



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

DESARROLLO DE VIDEOJUEGOS

T E S I S

**PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
INGENIERO MECÁNICO ELECTRICISTA**

**P R E S E N T A N
JULIO CESAR RAMÍREZ GURIDI
JUAN SERGIO MARTÍNEZ CRUZ
FERNANDO JACINTO ÁLVAREZ**

ASESOR

MAT. LUIS RAMÍREZ FLORES



FES Aragón



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Marzo 2014



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES

ARAGÓN

T E S I S DESARROLLO DE VIDEOJUEGOS

TESIS PARA OBTENER EL TITULO DE INGENIERO EN COMPUTACIÓN

TESIS PARA OBTENER EL TITULO DE INGENIERO MECÁNICO ELECTRICISTA

TESIS PARA OBTENER EL TITULO DE INGENIERO MECÁNICO ELECTRICISTA

**PRESENTAN: JULIO CÉSAR RAMÍREZ GURIDI
FERNANDO JACINTO ÁLVAREZ
JUAN SERGIO MARTÍNEZ CRUZ**

ASESOR: MAT. LUIS RAMÍREZ FLORES

SINODAL 1: MTRA. SILVIA VEGA MUYTOY

SINODAL 2: MTRA. MARÍA CONCEPCIÓN GARCÍA ESTRADA

SINODAL 3: M. EN I. FIDEL GUTIÉRREZ FLORES

SINODAL 4: M. EN C. FELIPE DE JESÚS GUTIÉRREZ LÓPEZ

San Juan de Aragón, Estado de México, abril de 2014

INTRODUCCIÓN

El objetivo general de este trabajo es crear una plantilla que integre las ideas principales y los pasos a seguir para desarrollar videojuegos desde algo sencillo a superproducciones. Todo se puede convertir en un videojuego, no importa su género, la idea es que no sea aburrido. El estado del arte de este trabajo es el equilibrio de los conocimientos y sus aplicaciones, iniciando con la parte fundamental que son las ideas, a estas dándoles su apropiada visión y objetividad se pueden convertir en una oportunidad de negocios.

Desarrollo de videojuegos comprende tres objetivos fundamentales que son: Proyectar la oportunidad de negocios a través de un plan de negocios y UN modelo de negocios, mostrar el desarrollo conceptual de la idea y la organización del sistema productivo, mostrar el funcionamiento de un videojuego y su organización como software. El enfoque sistémico que se pretende lograr es hacer fácil la aplicación del conocimiento en el desarrollo de videojuegos como una oportunidad de negocios rentable. Estamos en una era de la WIFI, la conectividad, la aldea global y la colectividad humana. Una era de las redes sociales, el legado de NAPSTER, así como el intercambio libre de contenidos multimedia, una era de libertad informática pero a la vez de paranoia por la seguridad. Hablar de videojuegos es porque la misma tecnología se ha prestado a que se abaraten los dispositivos donde pueden ejecutarse, hoy en día, cualquier persona por muy humilde que sea tiene un teléfono móvil y por ende tiene juegos ya sea en versión trial o demos, cualquiera tiene un rato de ocio por qué puede pasar ese esparcimiento jugando.

¿Por qué desarrollar videojuegos?, porque es entretenimiento, forman parte de nuestra cultura universal como humanos desde el inicio de la civilización siempre ha habido tiempo de esparcimiento y de ocio, entonces con la evolución de la tecnología, los videojuegos son solo una extensión de los juegos solamente proyectados en nuevos escenarios acordes a la evolución de la tecnología.

El octavo arte término acuñado durante la última década del siglo XX, ha transformado a los videojuegos en parte de nuestra cultura global, siendo parte básica de nuestra vida diaria. Un videojuego contemporáneo es como una película. Donde la diferencia es que el usuario forma parte de la narrativa que el videojuego proyecta, aquí se pueden transformar de forma virtual todas las ideas que la imaginación da.

Apreciemos las bondades de los videojuegos, exitosamente han formado parte de nuestra cultura en el desarrollo de la imaginación e inteligencia del hombre durante los últimos 60 años, han maximizado el desarrollando de habilidades como los reflejos, la intuición, pensamiento analítico, pensamiento lógico, pensamiento abstracto, pensamiento consciente e inconsciente, pensamiento estratégico, pensamiento individual y grupal. Y sin embargo, hay muchas mentiras y estudios mal intencionados en contra de los videojuegos, hasta donde sabemos el abuso de cualquier cosa perjudica la salud y no necesariamente tiene que ser un videojuego.

Ahora, como logramos crear un juego, pues bien fácil, teniendo una idea, se crean objetivos y reglas, pero hay que agregarle algo para volverlo interesante, cual es el trofeo o que se va a ganar para que nos motive a participar. Entonces, si esto lo tenemos presente, pasar esta idea a un medio digital ya no es tan

complicado como pareciera. Esto se logra sabiendo que hacer y como aterrizar la idea, dividiendo el proceso en dos etapas; la etapa conceptual y la etapa creativa, con su respectiva parte de negocios.

El capítulo uno, el objetivo es la organización y aplicación del conocimiento para la toma de decisiones de esta etapa experimental, y seleccionar una propuesta en función a las circunstancias existentes, así como fortalecer, modificar y argumentar sólidamente una propuesta para que sea producida, y con esto obtener la aprobación del proyecto.

Durante el desarrollo de concepto, el producto experimenta muchos cambios para darle forma y fuerza, el trabajo conjunto de escritores, artistas, diseñadores gráficos, organizadores, planeadores, directores, guionistas, correctores y analistas de marketing, los cuales estarán ajustando la idea hasta convertirla en un concepto sólido y listo para ser producido. La etapa conceptual su finalidad es aprobar o desechar propuestas en función a las condiciones del mercado. El diseño de concepto es parte estructural del modelo de negocios, es experimental porque aunque hay patrones y métricas definidas, está sujeta a la volubilidad del mercado por lo tanto no hay garantía de éxito.

El capítulo dos, el objetivo es planear, crear y asegurar que el producto funcionara bien. En la fase de preproducción el concepto pasara a ser un producto planeado, donde los recursos financieros, técnicos y humanos estarán coordinados en función del tiempo disponible para lograr un producto terminado en tiempo y forma.

Esta etapa es medible y se asegura sus resultados reales comparados con los resultados estimados en función al tiempo porque se conocen las mecánicas de cada grupo de trabajo. En esta parte se manejan criterios de conclusión de obra por parte de Planeación para tener un producto terminado por unidad de tiempo. El capítulo tres, es totalmente técnico, aplican algunas ideas para crear un videojuego.

El capítulo cuatro, plantea la oportunidad de negocios a través de un plan de negocios y un modelo de negocios que plantee ciertas ideas como la misión, visión, valores, así como la filosofía del corporativo.

El capítulo cinco comprende una aplicación de estas prácticas.

1 CONTENIDO

1	DISEÑO DE CONCEPTO	1
1.1	DISEÑO DE CONCEPTO	1
1.1.1	COMO SABER SI UNA IDEA ES BUENA.....	2
1.1.2	COMO ATERRIZAR UNA IDEA	3
1.1.3	CUANDO SE DEBE AJUSTAR UNA PROPUESTA	4
1.1.4	MERCADO META Y ANÁLISIS DE MERCADO.....	5
1.1.5	PRODUCTO FINAL REVISADO POR MERCADOTECNIA.....	6
1.2	ORGANIZACIÓN DEL PROYECTO.....	6
1.2.1	DISEÑAR UN PRODUCTO ECONÓMICAMENTE RENTABLE	8
1.2.2	DISEÑO DE CONTEXTO DEL PRODUCTO	8
1.2.3	DISEÑO DE CARACTERES DEL PRODUCTO	9
1.2.4	CUANDO SE DEBE RE ORIENTAR EL CONCEPTO	12
1.3	ARQUITECTURA DE CONCEPTO NARRATIVO.....	13
1.3.1	ESCRITORES	14
1.3.2	GUIÓN GRAFICO (STORY BOARD) Y ARGUMENTO.....	14
1.3.3	ARTE Y DISEÑO VISUAL.....	16
1.3.4	DISEÑO DE JUEGO Y DE PARTIDA	18
2	PREPRODUCCIÓN, PRODUCCIÓN, POSTPRODUCCIÓN	20
2.1	DISEÑO DE PRODUCCIÓN.....	20
2.1.1	MAPA DE TRABAJO POR DEPARTAMENTO	24
2.1.2	ESQUEMA DE ORGANIZACIÓN DEL GUIÓN GRAFICO.....	25
2.1.3	DIVISIÓN DEL TRABAJO POR ÁREA DE APLICACIÓN	28
2.1.4	DIVISIÓN DEL PROYECTO POR ETAPAS Y GRUPOS DE TRABAJO	29
2.1.5	MÉTRICAS DE PRODUCTIVIDAD	30
2.1.6	CRONOGRAMA DE ACTIVIDADES	30
2.1.7	GUIÓN GRAFICO	31
2.1.8	DISEÑO DE SECUENCIAS.....	32
2.1.9	DIAGRAMA DE ACCIÓN GENERAL	34
2.2	PRODUCCIÓN	35
2.2.1	ORGANIZACIÓN DEL VIDEOJUEGO	35
2.2.2	MAPA MAESTRO DEL VIDEOJUEGO	37
2.2.3	MAPA DE NIVEL MAESTRO	38
2.2.4	MAPA DINÁMICO DE SECUENCIA.....	38

2.2.5	SEGMENTACIÓN DEL VIDEO JUEGO	39
2.2.6	CARACTERES	40
2.2.7	ESCENARIOS	41
2.2.8	LUZ Y SOMBRAS.....	42
2.2.9	OBJETOS DINÁMICOS Y OBJETOS ESTÁTICOS	42
2.2.10	OBJETOS ESTÁTICOS.....	43
2.2.11	AUDIO	43
2.2.12	CINEMÁTICAS	44
2.2.13	ENSAMBLE DEL VIDEO JUEGO.....	45
2.3	POSTPRODUCCIÓN.....	45
2.3.1	ASEGURAMIENTO DE CALIDAD	46
2.3.2	ASEGURAMIENTO DE CALIDAD DE LOS BETA TESTERS	48
2.3.3	ASEGURAMIENTO DE CALIDAD DE COMPATIBILIDAD DE HARDWARE.....	49
2.3.4	CONTROL ESTADÍSTICO DEL PRODUCTO EN LÍNEA.....	50
2.3.5	SUPERVISIÓN DE CONTENIDO.....	51
2.3.6	RECURSOS ADICIONALES.....	53
2.3.7	SOPORTE EN LÍNEA.....	54
3	RECURSOS TECNOLÓGICOS.....	55
3.1	RECURSOS TECNOLÓGICOS.....	55
3.1.1	ESTRUCTURA BÁSICA DE UN VIDEOJUEGO	56
3.1.2	API (APPLICATION PROGRAMMING INTERFACE).....	58
3.2	¿QUÉ ES UN MOTOR DE JUEGO ?	61
3.2.1	PROPIEDADES DEL MOTOR DE JUEGO	63
3.3	LA FAMILIA DE MOTORES DE QUAKE.....	69
3.4	LA FAMILIA DE MOTORES UNREAL	70
3.5	ARQUITECTURA DEL MOTOR DE JUEGO	71
3.5.1	HARDWARE DESTINO	73
3.5.2	CONTROLADORES DE DISPOSITIVO.....	73
3.5.3	SISTEMA OPERATIVO.....	73
3.6	SDK DE TERCEROS Y RECURSOS INTERMEDIOS.....	74
3.6.1	ESTRUCTURAS DE DATOS Y ALGORITMOS	74
3.6.2	GRÁFICOS	75
3.6.3	COLISIONES Y FÍSICA.....	75
3.6.4	ANIMACIÓN DE CARÁCTER.....	75
3.6.5	INTELIGENCIA ARTIFICIAL.....	76

3.6.6	MODELOS DE CARÁCTER BIOMECÁNICOS	76
3.6.7	CAPA DE LA INDEPENDENCIA DE PLATAFORMA	76
3.6.8	SISTEMAS CENTRALES	77
3.6.9	ADMINISTRADOR DE RECURSOS	78
3.6.10	MOTOR DE RENDERIZADO	78
3.6.11	RENDERIZADO DE BAJO NIVEL	78
3.6.12	INTERFACE DE DISPOSITIVO GRAFICO.....	79
3.6.13	OTROS COMPONENTES DE RENDERIZACION	79
3.6.14	OPTIMIZACIÓN DE ESCENA DE GRÁFICOS/ELIMINACIÓN	79
3.6.15	EFECTOS VISUALES	80
3.6.16	FRONT END (INTERFACE FINAL USUARIO MAQUINA).....	81
3.6.17	HERRAMIENTAS DE ANÁLISIS Y DEPURACIÓN	82
3.6.18	COLISIONES Y FÍSICA.....	83
3.6.19	ANIMACIÓN	85
3.6.20	DISPOSITIVOS DE INTERFACE HUMANA.....	86
3.6.21	AUDIO	86
3.6.22	MULTIJUGADORES EN LÍNEA Y EN RED	87
3.6.23	BASES FUNDAMENTALES DEL GAMEPLAY	88
3.6.24	MUNDOS DE JUEGO Y MODELOS DE OBJETO	89
3.6.25	SISTEMA DE EVENTOS	90
3.6.26	SISTEMA DE ESCRITURADO	90
3.6.27	FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL.....	91
3.6.28	SUBSISTEMAS ESPECÍFICOS DEL JUEGO	92
3.6.29	HERRAMIENTAS Y ACTIVOS DE INFORMACIÓN	92
3.6.30	HERRAMIENTAS DE CREACIÓN DE CONTENIDO DIGITAL.....	92
3.6.31	CONDICIONAMIENTO DE ACTIVOS DE INFORMACIÓN	93
3.7	DATOS DE MODELOS/MALLAS EN 3D	94
3.7.1	GEOMETRÍA DE patrones (brush geometry)	94
3.7.2	MODELOS 3D (MALLAS)	95
3.7.3	DATOS DE ANIMACIÓN ESQUELÉTICA.....	95
3.7.4	DATOS DE AUDIO	96
3.7.5	DATOS PARA SISTEMAS DE PARTÍCULAS.....	96
3.7.6	DATOS DE AMBIENTE DE JUEGO Y EDITOR DE AMBIENTE.....	97
3.8	ALGUNOS ENFOQUES PARA LA ARQUITECTURA DE HERRAMIENTAS	97
3.9	KIT PARA DESARROLLO DE SOFTWARE (SOFTWARE DEVELOPMENT KIT)	99

3.9.1	UNREAL ENGINE 3	100
3.9.2	PLANIFICACIÓN Y PREPRODUCCIÓN	100
3.9.3	EL MOTOR Y CONFIGURACIÓN.....	100
3.9.4	UNREALSCRIPT Y BASE DE CODIGOS UE3.....	101
3.9.5	ELEMENTOS DE CREACIÓN DE GAMEPLAY	101
3.9.6	CREACIÓN DE NIVEL	102
3.9.7	CREACIÓN DE CONTENIDOS.....	102
3.9.8	COMPILACIÓN Y PRUEBAS	103
3.9.9	EMBALAJE Y DISTRIBUCIÓN.....	103
3.9.10	PROGRAMACIÓN DE GAMEPLAY.....	103
3.9.11	ANIMACIÓN.....	104
3.10	TÓPICOS DE DESARROLLO.....	104
3.10.1	INTELIGENCIA ARTIFICIAL Y NAVEGACIÓN	104
3.10.2	ANIMACIÓN.....	104
3.10.3	MARCO APEX.....	105
3.10.4	AUDIO.....	105
3.10.5	COMMANDLETS (COMANDOS)	105
3.10.6	FOLIAGE (FOLLAJE)	106
3.10.7	ANIMACIÓN DE MATERIALES Y VERTEX	106
3.10.8	ILUMINACIÓN	107
3.11	ELEMENTOS DEL GAMEPLAY.....	107
3.11.1	INPUT / OUTPUT (ENTRADAS / SALIDAS).....	108
3.11.2	KISMET VISUAL SCRIPTING (ESCRITURADO VISUAL KISMET).....	108
3.11.3	LANDSCAPE (PAISAJE)	108
3.11.4	LEVEL STREAMING (NIVEL DE TRANSMISIÓN DE INFORMACIÓN)	109
3.11.5	LIGHTING & SHADOWS (ILUMINACIÓN Y SOMBRAS)	109
3.11.6	MATERIALS & TEXTURES (MATERIALES Y TEXTURAS).....	109
3.11.7	MATERIAL.....	109
3.11.8	SPREAD (PROPAGACIÓN)	110
3.11.9	EMISSIVE (EMISIÓN).....	110
3.11.10	OPACITY (OPACIDAD)	110
3.11.11	NORMAL.....	110
3.11.12	MATINEE & CINEMATICS (MATINÉ Y CINEMÁTICAS).....	110
3.11.13	NETWORKING & REPLICATION (RED Y REPLICACIÓN).....	110
3.11.14	PARTICLES & EFFECTS (PARTÍCULAS Y EFECTOS)	111

3.11.15	PERFORMANCE, PROFILING, AND OPTIMIZATION (DESEMPEÑO, PERFIL Y OPTIMIZACIÓN)	111
3.11.16	GENERAL MAINTENANCE (MANTENIMIENTO GENERAL)	111
3.11.17	BASIC TOOLS AND TECHNIQUES (HERRAMIENTAS BÁSICAS Y TÉCNICAS)	111
3.11.18	MEMORY PROFILING (PERFIL DE MEMORIA)	112
3.11.19	PROFILING CONTENT AND PERFORMANCE (CONTENIDO DE PERFILES Y RENDIMIENTO)	112
3.11.20	CPU PERFORMANCE (RENDIMIENTO DEL CPU)	112
3.11.21	GAME PERFORMANCE (RENDIMIENTO DE JUEGO)	113
3.11.22	RENDER PERFORMANCE (RENDIMIENTO DE RENDERIZADO)	113
3.11.23	GPU PERFORMANCE (RENDIMIENTO DE GPU)	113
3.11.24	NETWORK PROFILES (PERFILES DE RED)	114
3.11.25	PERFILES MÓVILES	114
3.11.26	PHYSICS (FÍSICA)	115
3.11.27	POST PROCESS EFFECTS (EFECTOS POST PROCESOS)	115
3.11.28	RENDERING (RENDERIZADO)	115
3.11.29	SCALEFORM GFX	115
3.11.30	SKELETAL MESHES (MALLAS ESQUELÉTICAS)	116
3.11.31	STATIC MESHES (MALLAS ESTÁTICAS)	116
3.11.32	UNREAL EDITOR AND TOOLS (EDITOR UNREAL Y HERRAMIENTAS)	116
3.11.33	UNREALSCRIPT	117
3.11.34	USER INTERFACES & HUDS (INTERFACE DE USUARIO Y HUDS)	117
3.11.35	GRÁFICOS Y VIDEO	118
3.11.36	AUDIO	119
3.11.37	3D	120
3.11.38	NET, MMORG y HUDs	121
3.12	REVISIÓN DEL CÓDIGO FUENTE DE QUAKE 3	122
3.12.1	ARQUITECTURA DEL MOTOR DE JUEGO	123
3.12.2	EL CÓDIGO	125
3.12.3	ESTADÍSTICAS	126
3.12.4	ASIGNACIÓN DE MEMORIA	127
3.12.5	RENDERIZADO	127
3.12.6	ARQUITECTURA DEL RENDERIZADO	128
3.12.7	MODELO DE RED	133
3.12.8	ARQUITECTURA DE RED	134
3.12.9	SISTEMAS DE CAPTURAS	134

3.12.10	INTROSPECCIÓN DE MEMORIA CON C.....	138
3.12.11	MENSAJES FIABLES Y NO FIABLES	139
3.12.12	LA MAQUINA VIRTUAL	139
3.12.13	ARQUITECTURA DE LA MAQUINA VIRTUAL	139
3.12.14	QVM INTERNAS	140
3.12.15	Q3ASM.EXE INTERNOS	142
3.12.16	COMO FUNCIONA QVM	143
3.12.17	TEMA DE LA PRODUCTIVIDAD Y LA SOLUCIÓN	144
3.12.18	INTELIGENCIA ARTIFICIAL.....	145
3.12.19	ARQUITECTURA DE INTELIGENCIA ARTIFICIAL	145
3.13	HARDWARE Y PERIFERICOS.....	146
3.13.1	DISPOSITIVOS MÓVILES.....	148
4	MODELO DE NEGOCIOS	149
4.1	PLAN DE NEGOCIOS.....	149
4.1.1	RESUMEN EJECUTIVO.....	150
4.1.2	GESTIÓN DE LA CORPORACIÓN.....	151
4.1.3	PLAN DE MERCADOTECNIA	153
4.2	MODELO DE NEGOCIOS.....	154
4.2.1	RESUMEN EJECUTIVO.....	155
4.2.2	VISIÓN, MISIÓN Y ESTRATEGIAS DE GAMESKAPE	157
4.2.3	PROPUESTAS DE VALOR	159
4.2.4	CANALES PARA LLEGAR A ESOS CLIENTES.....	160
4.2.5	MODELO DE NEGOCIOS Y EL ANÁLISIS DE MERCADO.....	161
4.2.6	RESULTADOS DE UN TRABAJO DE EQUIPO	162
4.2.7	EL ARTE DE CONSTRUIR UNA MARCA	163
4.2.8	EL MAPA ESTRATÉGICO DE GAMESKAPE	164
4.2.9	BALANCE SCORE CARD PARA GAMESKAPE	166
4.2.10	FEEDBACK Y APRENDIZAJE ESTRATÉGICO	167
4.2.11	ASUNTOS DE NEGOCIO	168
5	EJEMPLO DE APLICACIÓN.....	169
5.1	EJEMPLO DE APLICACIÓN.....	169
5.1.1	LA PROPUESTA A PRODUCIR	170
5.1.2	OVERLANDS.....	170
5.1.3	TRANSFORMACIÓN DE UN CONCEPTO A UN VIDEOJUEGO.....	171
5.1.4	ARTE Y GUION GRAFICO DEL VIDEOJUEGO	172

5.1.5	BOCETOS Y PROTO ARTE DE OVERLANDS	173
5.1.6	ESQUEMA DE DISTRIBUCIÓN DEL PRODUCTO	179
5.1.7	ORGANIZACIÓN DEL VIDEOJUEGO	180
5.1.8	SEGMENTACIÓN Y ETAPAS DEL VIDEOJUEGO.....	180
5.1.9	ELEMENTOS DEL VIDEOJUEGO.....	183
5.1.10	DISEÑO DE PARTIDA.....	186
5.1.11	DISEÑO DE PRODUCCIÓN.....	191
5.1.12	ETAPAS DEL PROYECTO	193
5.1.13	ENSAMBLE DEL VIDEO JUEGO Y PRUEBAS BETA.....	195
5.2	RESUMEN EJECUTIVO PLAN DE NEGOCIOS.....	199
5.3	RESUMEN EJECUTIVO MODELO DE NEGOCIOS.....	200
6	CONCLUSIONES	204
7	GLOSARIO	205

1 DISEÑO DE CONCEPTO

1.1 DISEÑO DE CONCEPTO

La inteligencia es una entidad presente en todos los seres humanos y en los animales con cerebro, entonces constantemente estamos generando ideas que nos llevan a crear nuevas intrusiones en la realidad.

Nosotros vemos lo que queremos ver, estas ideas toman referencias de la observación y el cerebro les va dando forma en función a las necesidades que tenemos, nosotros como humanos moldeamos nuestro mundo en función a nuestra percepción.

La percepción es la habilidad de interpretar nuestro entorno y aprovecharlo y así crear conceptos que bien son los límites para comprender esta realidad.

El concepto es la integración de observaciones que enfocadas a cubrir una necesidad y que nuestra mente transforma en una entidad tangible y medible para ser cuantificada por sus cualidades.

Modelando el concepto viene la aplicación hacia la necesidad y realidad humana para ajustarla a la búsqueda de las oportunidades de negocio en función a las necesidades de la población.

El mercado meta es fácil delimitar estableciendo métricas y modelos del comportamiento de las necesidades de determinados grupos humanos.

Con modelos se obtienen parámetros para explorar la psique humana desarrollando así ideas más sutiles que llenaran esas necesidades.

Entonces la idea y concepto es la integración de las observaciones en función a las necesidades y así crear oportunidades de negocio.

En la práctica todo está enfocado al negocio pero las ideas más originales nacen del sentimiento, de las necesidades y de la observación. Para trasladar una idea a la realidad se tiene que observar el mundo como un todo y descubrir dónde están las carencias. De esta forma se pueden ver las oportunidades de negocio y así una idea puede convertirse en un concepto que cubre las necesidades de determinados grupos humanos.

Estas observaciones que se consideran están en el escenario mundial que involucran tópicos casi en su mayoría de aspecto humano y una minoría de aspecto natural.

Estos aspectos influyen decisivamente en la selección de una idea para enfocarla y ajustarla al requerimiento.

A partir de ahí es donde los accionistas con sus metodologías toman la decisión de producirla o no.

1.1.1 COMO SABER SI UNA IDEA ES BUENA

Saber si una idea es buena es como atreverse a decir que este negocio te hará rico, normalmente pensamos en que las buenas ideas permiten ganar mucho dinero, entonces si eso es cierto por qué no plantearlas bien y llevarlas a la realidad.

No hay formula que asegure el éxito de un proyecto pero si existen metodologías que respaldan con sus criterios evidentes la POSIBLE aceptación de la idea así como su éxito a nivel comercial.

Muchas veces las ideas que se transforman en conceptos llevan otras ideas de menor peso pero en alguna de estas pequeñas ideas se puede esconder el éxito de una gran oportunidad de negocios.

Realmente no hay forma de saber si una idea es buena, sin embargo se usa el sentido común, que está conformado de herramientas básicas de la lógica humana, no aplica variables cuantitativas y cualitativas, simplemente se hace uso de algo llamado INTUICIÓN, este criterio ayuda en la decisión de aceptar o desechar una propuesta, normalmente experiencias previas de prueba y error –corazonadas-, la mejor forma de saber si una idea tiene futuro es poniéndola en práctica, nadie ni nada puede asegurar el éxito de una idea.

Una idea brillante normalmente es 10 por ciento es inspiración y 90 por ciento trabajo.

Una idea bien organizada puede convertirse en un producto u oportunidad de negocios, cuando hay una propuesta que se desea concretar es porque hay los argumentos suficientemente fidedignos para ser sometida a discusión ante la junta de accionistas, si ellos encuentran que la propuesta es negocio entonces ellos financiaran el desarrollo de la propuesta.

A través de metodologías, heurísticas, creatividad, observación directa, observación indirecta, descubrimiento, PRUEBA Y ERROR, psicología inversa, psicología tradicional, inducción, deducción, asociación de esferas, entre otras técnicas una idea propuesta se puede fortalecer o desechar por sus cualidades originales que pueden adecuarse al contexto del momento.

La mejor idea es la que integra más opciones de éxito por su originalidad, adaptabilidad, expansión y aplicaciones en su momento clave.

1.1.2 COMO ATERRIZAR UNA IDEA

Una idea es una entidad con dos atributos que son el tiempo y su rentabilidad, hay proyectos que toman muchísimo tiempo, durante el cual suceden muchas cosas que debilitaran todas las buenas intenciones que tuvo al inicio así como su proyección en el futuro, una idea magnifica puede ser destruida por el tiempo y no por los recursos tecnológicos o económicos. El tiempo marca todo en la existencia de una idea, así que para aterrizarla no se debe perder la objetividad y racionalizarla en términos reales.

Para aterrizarla se debe cuantificar los resultados deseados en función a los recursos disponibles tantos humanos, tecnológicos y financieros.

En la línea del tiempo se establecen metas reales, estas mantendrán los criterios originales y podrán ser modificados al encontrar nuevas oportunidades de negocio, en otras palabras si la idea original se vuelve obsoleta pero ha dejado al descubierto nuevas vetas se explotaran estas nuevas oportunidades.

Este criterio es de algún modo fácil plantear.

Dependiendo de la naturaleza de la idea se hacen algunas rutinas cuya finalidad es estandarizar el escenario donde la idea se proyectara, como en una simulación, las ideas se transforman en proyectos los cuales tienen objetivos que se cumplirán satisfactoriamente.

El escenario consiste en un modelo de ejecución del proyecto.

Todavía en papel este se conceptualiza en función a los recursos económicos, la técnica y la tecnología del momento en que se propone y lo más importante es el capital humano, muchas personas de algún modo audaces para su momento histórico concretaron ideas grandiosas porque hubo con que hacerlas.

Hablamos de un trabajo en equipo de gente atrevida que no le tenía miedo al ridículo, que no le importaba perder dinero, sino la mayor satisfacción es descubrir lo que nadie ha tenido o donde nadie ha llegado porque ahí están las oportunidades de negocio.

No cabe duda que todas las grandes aventuras de la humanidad comenzaron con ideas que se pudieron aterrizar satisfactoriamente por los recursos y el capital humano que pudo llevarlas a buen término o al menos a descubrir algo nuevo.

1.1.3 CUANDO SE DEBE AJUSTAR UNA PROPUESTA

Durante el análisis de la idea, se llegan a encontrar muchas cosas favorecedoras o desfavorecedoras a la idea original.

Se suele recurrir a elementos adicionales para ajustar la idea original hacia nuevos horizontes que podrán fortalecerla y de algún modo garantizar una mejor rentabilidad a futuro.

Muchas veces se cae en el error de mantener la idea original y genera un producto que ya no causa impacto que el mercado meta que lo ve como basura, entonces, ¿cómo se orienta una idea?

Se usan herramientas estadísticas para ajustar criterios en la idea original, estos criterios pueden ser:

**FILOSÓFICOS
PERSONALES
EVENTOS DEPORTIVOS
EVENTOS CULTURALES
EVENTOS COMERCIALES
MODA
CAMBIOS EN LAS COSTUMBRES HUMANAS
TECNOLOGÍAS
DISPONIBILIDAD ECONÓMICA DEL MERCADO META
CONFLICTOS BÉLICOS EN EL MUNDO
PRECIO DEL CRUDO
FLOTABILIDAD DE DIVISAS COMO EL DÓLAR Y EL EURO.
VARIACIONES EN LA BOLSA DE VALORES**

Orientar la idea es ajustarla para “asegurar” el éxito lo más humanamente posible. Sin olvidar que los cambios que experimenta el mundo son muy rápidos y drásticos. Hacer un ajuste en la idea original es válido.

Esta variación en las tendencias puede darle un giro a una idea y convertirla en un éxito. Para hacer ajustes en la orientación de la idea es porque se tiene pleno dominio de la misma, entender el concepto y modificarlo, con el estilo apropiado se pueden capturar segmentos de mercado inimaginables con tan solo reajustar por decir el color de algún carácter involucrado, alguna melodía de fondo, alguna frase célebre, un esquema de colores de fondo, sonido etc.

Hay ideas que por su originalidad pueden ser demasiado revolucionarias para su época por lo que se someten a un proceso de adaptación.

Ideas que son tan innovadoras que tienen que ser disminuidas en alcances para hacerla comprensible al momento actual de su publicación.

Muchas veces al revisar la idea se encuentran detalles que deben ser corregidos por razones de naturaleza humana.

La adaptación de las ideas se hace para cambiar el contexto general de la idea original, muchas veces es por que ponen en evidencia detalles incómodos que pueden generar fricciones por parte de mercados ajenos al mercado meta.

Muchas veces estas adaptaciones llegan a ser tan radicales que la idea original se pierde pero da vida a ideas nuevas que no se tenían contempladas.

Las adaptaciones de una idea involucran elementos de naturaleza humana principalmente, tales como violencia, sexo, lenguaje obsceno, eventos humanos reales pero disfrazados, tópicos que pueden solo ser comprendidos por público con cierto criterio como la edad, por decirlo así, apto para públicos de 3 a 5 años, 6 a 12 años, 13 a 18 años, adultos, por mencionar algunos criterios restrictivos. Para deslindar estas circunstancias se usa la famosa frase célebre "cualquier parecido con la realidad es coincidencia".

Las filosofías de la idea original pueden ser poco comprendidas por el usuario y crear atmosferas erróneas de la proyección de la idea, lo cual debe corregirse para evitar malos entendidos y llegar a ser boicoteado con algunas instituciones reguladoras de comercio, ellos pueden argumentar cualquier cosa como afectar las buenas costumbres hasta persuadir al público de actos inmorales entre otras cosas. Por eso, las ideas deben ser adaptadas para crear dinero y no perdidas, ya que en un mundo tan competido y regulado por instituciones verificadoras, abogados voraces pueden afectar la conclusión de una gran idea.

1.1.4 MERCADO META Y ANÁLISIS DE MERCADO

La opinión de especialistas de diversas áreas propone los pros y los contras del prototipo mismo. Donde se han corregido esos detalles o incluso la misma idea ha tenido un giro para su producción. Muchas veces no se tienen certezas en los criterios empleados, sin embargo las herramientas estadísticas proporcionan algún respaldo y voto de confianza. En el prototipo mismo se explora la posibilidad de éxito de manera muy lejana, muchas veces estas expectativas son de carácter especulativo, sin embargo apoyándose en éxitos previos y análisis de mercados concienzudos es como se da el voto de aprobación de un producto.

Por mencionar un ejemplo para que una tienda tenga éxito debe de ofrecer productos a bajos precios, comodidad para llegar, facilidades de estacionamiento y entretenimiento, diversidad de opciones de pago, que donde esté ubicada el local sea de fácil acceso y de tráfico lento para poder entrar al estacionamiento en automóvil, que haya percepción de seguridad y no de hostilidad por parte de cuerpos de seguridad, amabilidad por parte de los empleados, atención al cliente, limpieza, sanitarios gratuitos, servicios financieros entre otras cosas. De una idea dada que fue sometida a evaluación por sus cualidades y aportaciones se llega a la decisión de producción.

Estos datos proporcionan estimaciones del impacto del producto. Estas referencias estadísticas sirven para modificar el perfil del producto mismo y ajustarlo al mercado meta cumpliendo con cierta normatividad que el mercado exige. Los accionistas consideran estos puntos como elementos para decidir el status de un proyecto de inversión. Todos los datos son estimaciones lo mas fidedignas posibles proporcionadas por las autoridades emisoras de las estadísticas.

Un video juego es un producto de cobertura mundial, estos proyectos son prorrateados para su lanzamiento mundial. Hay muchas economías que de manera legal no permitirían su promoción, sin embargo, así como se viven actualmente las cosas, la piratería es un instrumento para la difusión mundial de videojuegos. Claro no es una actividad lícita pero promueve de alguna manera a los videojuegos, además del internet que ha roto fronteras y aunque daña el modelo de negocios, es un mal necesario. Para obtener datos lo más confiables es a través de la psicología inversa aplicada a los negocios.

Este esquema basa su veracidad en este criterio "somos lo que hacemos y no lo que decimos" Infiriendo en los indicadores financieros mundiales se puede obtener cifras estimables y confiables sobre los hábitos de compra de la gente. Durante la observación de los indicadores financieros como el IPC, DOW JONES <http://www.dowjones.com/> , NASDAQ <http://www.nasdaq.com/>, BMV, entre otros se puede observar lo que la población está demandando a ver lo que está comprando.

Una forma coherente de encontrar los mercados metas es observando el comportamiento de la cartera vencida y lo que la gente compra, ya que si observamos el endeudamiento de la población es porque tiene forma de adquirir sus bienes sin importar como los obtuvo.

1.1.5 PRODUCTO FINAL REVISADO POR MERCADOTECNIA

Después de una revisión exhaustiva del producto se procede a llevarlo ante la junta de accionistas, ellos decidirán en función a sus criterios personales, el producto será o no producido en función a su rentabilidad.

Ellos emplean criterios y métodos simples y efectivos para decidir el futuro del producto. Estas fuentes de información que respaldan al producto provienen de motores de búsqueda confiables como son Google, Google Adwords.

Y se pueden dividir por regiones mundiales para ajustar el modelo de negocios. Los criterios que siguen son unificados por el modelo económico globalizado.

Producto mismo que fue diseñado para determinado segmento de mercado:

Edad
Educación
Ingreso per cápita
Caloría per cápita

Como comentario, la mejor época para lanzar un videojuego es cuando hay reajustes en las economías globales, la razón es porque la gente busca distraerse.

1.2 ORGANIZACIÓN DEL PROYECTO

La organización lo es todo para convertir una idea en realidad tangible y lucrativa.

Muchas veces la organización de la idea es fácil conociendo las técnicas apropiadas y los recursos técnicos.

El análisis de fortalezas y debilidades es básico por que plantea la división del trabajo y así asignar tareas a grupos de trabajo, este planteamiento siempre estará en función al tiempo. El tiempo es el recurso más preciado para construir cualquier proyecto.

No importa la magnitud del proyecto si se dispone de organización.

Partiendo de los recursos disponibles como el conocimiento y la técnica es como las cosas proyectadas tendrán un futuro prometedor.

Sin embargo, muchas veces el resultado se ve afectado por los recursos disponibles y la capacidad de la técnica en el momento que el proyecto se llevó a cabo. Sin embargo las cosas no quedan ahí, siempre habrá más de una opción para lograr las cosas en tiempo y forma.

Esforzarse por obtener una buena organización inicial así como un buen equipo de trabajo comprometido reeditara en ganancias y oportunidades de negocio futuras. Ahora una idea se puede ajustar al mejor perfil mercadotécnico. Esto significa que la organización de la idea estará en función a los requerimientos del mercado para pretender obtener la mejor ganancia posible.

Esto se refiere a que cada elemento de la idea fue diseñado en función a las circunstancias del momento en el escenario de lanzamiento y para capturar la atención de potenciales compradores.

La organización de la idea es el reflejo de cómo cada elemento tomara participación en el mercado y se reflejara en sus índices de ventas.

Vender lo es todo, pero como se le vende un producto a una persona, pues fácil, observando su perfil, sus hábitos y eso se concatena con los elementos del proyecto. El concepto tiene solo dos variables, las atemporales que mantendrán la idea original a lo largo de la línea del tiempo del concepto y las temporales que experimentaran una evolución a lo largo del desarrollo del concepto.

Empleando mapas de acción es como se puede sacar mejor provecho a la idea en función al momento histórico de su futuro lanzamiento.

Un estado presente en el concepto involucra un contexto inicial para el comienzo del desarrollo, no precisa que este sea el comienzo, sino únicamente es el arranque del concepto

Un estado futuro, planteado con expectativas bien definidas será el resultado final.

Durante la transición de un estado a otro es donde se desarrolla la evolución de los hechos y así comienza el manejo de la ubicación de los elementos del concepto para enfatizar momentos clave durante la línea del tiempo de la idea.

Empleado diferentes estilos de narrativa se obtiene capturar la atención del mercado meta. La organización adecuada de los elementos del concepto proporciona fuerza al objetivo del producto, en las diferentes etapas de desarrollo del concepto mismo, en la organización se busca diseñar una narrativa que maximice momentos clave de cada etapa además de involucrar elementos adicionales para enlazar eventos nuevos con eventos antiguos, segmentando la historia, concatenarla, incorporar nuevos elementos, eliminar elementos obsoletos o dejarlos en espera. Este esquema organizacional se aplica a las diferentes esferas del concepto como son ambientes, inmobiliario, mobiliario,

habilidades, semántica, ideas conceptuales, diálogos, tendencias, personajes, atuendos, acciones, etc....

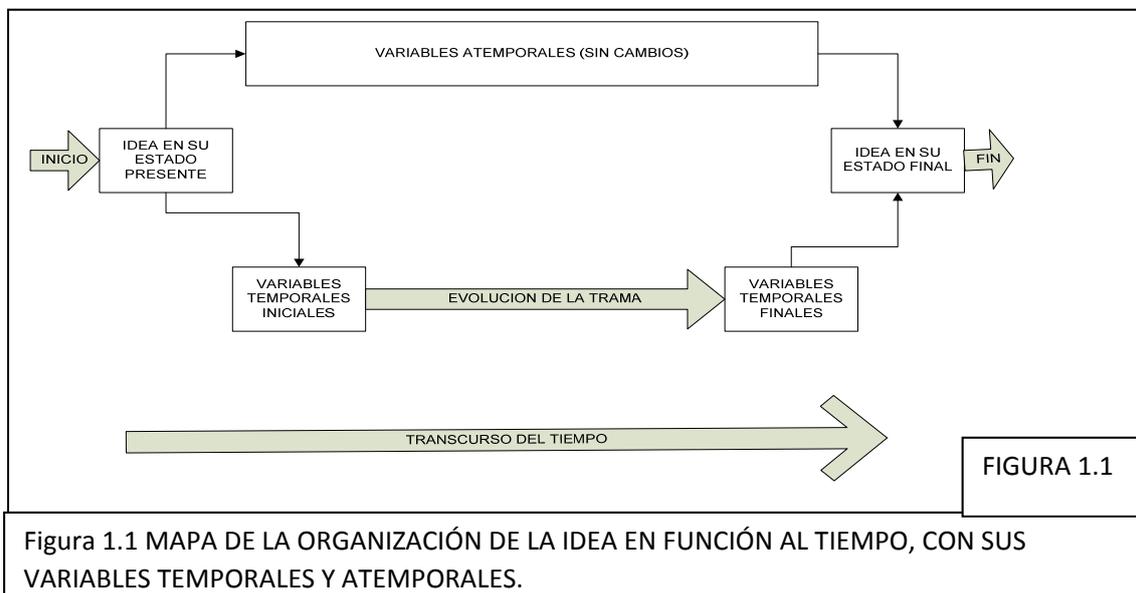


Figura 1.1 MAPA DE LA ORGANIZACIÓN DE LA IDEA EN FUNCIÓN AL TIEMPO, CON SUS VARIABLES TEMPORALES Y ATEMPORALES.

1.2.1 DISEÑAR UN PRODUCTO ECONÓMICAMENTE RENTABLE

Capitalizar el éxito de una idea lo hace un diseño original del manejo de ideas plenamente coherentes que se presten a una secuela o pre cuela, productos secundarios de mercadotecnia.

Durante el transcurso del desarrollo de una idea siempre se considera la rentabilidad, desde el inicio se puede “ver” el alcance de la misma a través de criterios básicos como la originalidad, la facilidad posicionarse en el mercado. La disposición de recursos financieros y tecnológicos para obtener el resultado deseado en relación al tiempo.

Aquí se integran los criterios de profesionales en proyectos de inversión, de equipos de producción, de directores, de analistas de mercado, de diseñadores de producción, equipos de producción, equipos de marketing. Si ellos en sus opiniones proponen un SI, entonces, se prosigue a un análisis más objetivo con cifras y manejos exactos de inicio a fin.

En la integración de opiniones objetivas de profesionales, se maneja las tendencias positivas y negativas del concepto en un escenario real que es el mercado mismo. Incluso situaciones de origen económico, conflictos en la junta de accionistas, sea de la naturaleza que fuese todo proyecto de inversión debe estar sustentado en la eficiencia del staff para su buen término.

Cuando no se tiene certeza del resultado final, se recurre a estrategias de cambio de objetivo o poner punto final a la idea.

1.2.2 DISEÑO DE CONTEXTO DEL PRODUCTO

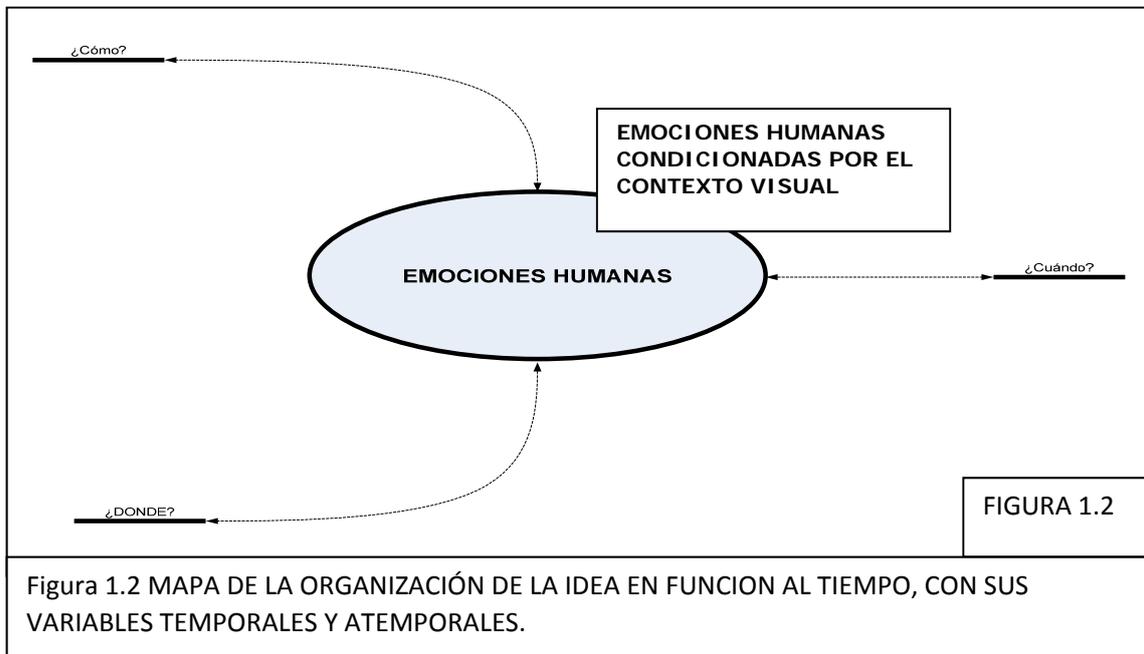
El diseño del contexto de la idea está en función a las esferas de emociones que crea la evolución del desarrollo de los hechos en el tiempo.

Se diseña con variables temporales que enfatizan emociones condicionadas por el contexto donde se desarrolla la idea.

El contexto se compone de un momento en el tiempo con sus respectivas variables de:

- ¿Cómo?
- ¿Cuándo?
- ¿Dónde?

Esto se emplea para darle fuerza a la idea y generar emociones que capturen al mercado meta. Empleando esferas de la psique humana como: angustia, desesperación, ira, furia, amargura, enojo, gula –por mencionar algunas-, se trasladan a otras esferas para traducirlas en colores, sonidos, escenarios, que hacen que la psique del usuario se desate y experimente emociones fabricadas por el contexto del diseño visual.



1.2.3 DISEÑO DE CARACTERES DEL PRODUCTO

El concepto se compone de elementos que narraran una historia, ellos transmitirán emociones y acción, entonces diseñar un carácter se compone de dos variables; físicas y psicológicas, basadas en propiedades cuantitativas y cualitativas, con estas métricas de diseño se puede crear el carácter deseado.

Cada carácter está diseñado en función a su participación dentro de la trama. El diseño de carácter está enfocado para pertenecer al contexto del concepto general.

Se propone una serie de esferas de cualidades cuantificadas en niveles de aplicación.

La idea principal es cuál es el rol del carácter en la idea, así se conceptualiza la esfera de personalidad y se le dan atributos físicos y psicológicos.

Estas esferas son:

Físicas; cabello, ojos, boca, estatura...

Psicológicas; inteligente, justo, obediente, creativo, moralista...

Habilidades; computación, correr, fuerza, inteligencia...

Indumentaria; playa, noche, coctel, gala, montaña, guerra...

Armamento; armas, cañones, granada, radares...

Se pueden usar métricas para enfatizar el perfil del carácter y darle fuerza a ciertos elementos. Así es como se conforma el diseño de carácter en función al contexto del concepto.

Los atributos del carácter se otorgan en base a una política definida dentro del mismo concepto, puede ser una evolución dentro del juego, subir nivel, acumular puntos, desarrollar habilidades, etc....

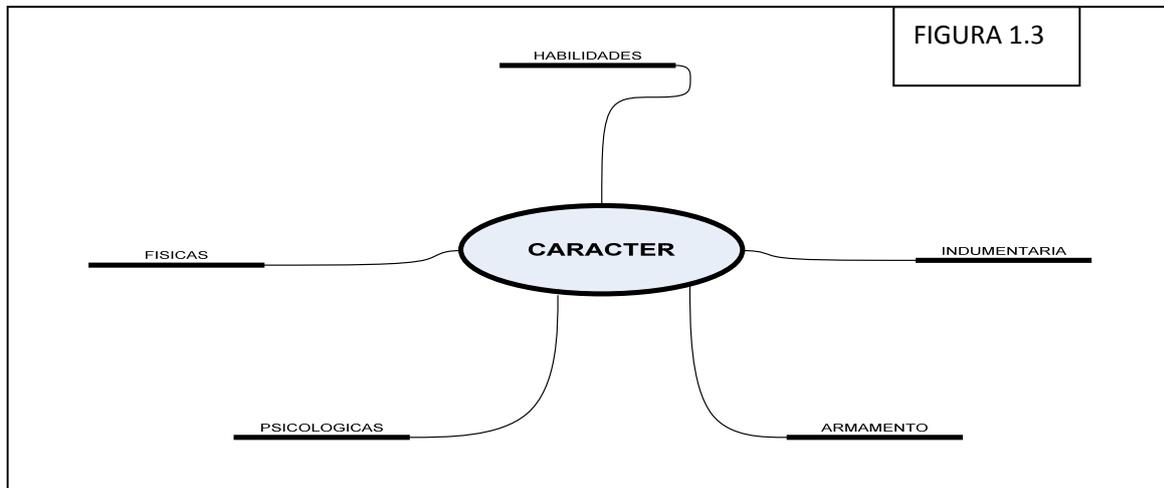


Figura 1.3 El carácter se compone de cinco capas de aplicación para darle forma a la personalidad. Que son cualidades son: físicas, habilidad, indumentaria, psicológicas, armamento. Estos privilegios se otorgan en función al perfil del usuario para identificarse el con el carácter para crear cierta afinidad e identidad.

DISEÑO DEL PERFIL PSICOLÓGICO DEL PRODUCTO

Una vez que se tiene una propuesta definida y clara se continúa dando más fuerza al diseño psicológico al concepto, esto es con la finalidad de capturar más mercados.

Un producto bien diseñado captura la atención de mercados de diferentes naturalezas, sin importar sexo, edad, status social, etc. Esto lo logra porque captura la imaginación, el cliente se identifica con el producto o con alguna parte del producto, es aquí en esta etapa donde se diseñan las esferas involucradas en el producto para lograr ese fin.

Para obtener esos criterios de diseño se tiene que observar al posible mercado meta y otros mercados satélites, las observaciones se basan estadísticamente en hábitos de compra, edades, sexo, género, preferencia sexual, alimentación, tiempo dedicado a ver tv, tiempo dedicado a escuchar radio, tiempo a navegar en internet, tiempo de hacer ejercicio físico, tiempo dedicado a actividades económicas.

Se puede modelar cada evento de hábitos de compra del cliente, por mencionar un ejemplo, cada vez que nos conectamos a internet, una cookie registra todos los movimientos, visitas, duración, clics dados, el orden de cada visita, la regularidad de la misma, sin importar los sitios visitados, únicamente registra y envía las estadísticas a una supercomputadora que procesa los datos asignándoles jerarquías de modo que se deduce estadísticamente los hábitos generales de un usuario de internet.

Partiendo de esto, se hacen observaciones estadísticas y el resultado proporciona información valiosa de la cual se presta atención en las singularidades para el diseño de productos.

La idea se creó a partir de una observación y se separó para darle vida a un concepto. Entonces el perfil del concepto se obtuvo de observar singularidades que se han trasladado a conceptos tangibles y medibles.

El perfil psicológico sirve para capturar mercados meta, donde el cliente se siente identificado con el producto o alguna característica del mismo.

En el estudio de mercado se diseñan modelos para capturar esas singularidades de los clientes y así obtener más ideas para reforzar al producto.

Utilizando técnicas de psicología inversa es como se diseña el perfil psicológico de un producto, que aunque como en todo se puede tener una percepción errónea y obtener basura como resultados, sin embargo el que no arriesga no gana.

Observar al mercado meta a través de medios indirectos revela más información que hacer encuesta, estas observaciones se basan en evidencias obvias, para medir esas observaciones se cuantifican con metodologías estadísticas como las frecuencias de ciertos hábitos.

Por decir un ejemplo, una persona con un ingreso de unos 1000 usd mensuales, con qué frecuencia va a la tienda de la esquina a comprar algo que le hace falta, cuanto es lo que gasta, si pago en efectivo o con otra forma de pago, a qué horas fue, como iba vestido, con quien iba, cuantas veces va a comprar al día o a la semana.

El ejemplo mostrado se puede cuantificar con las métricas apropiadas dando resultados tangibles en función del tiempo y del dinero que gasto esa persona pero que finalidad tiene esto, simple, es crear un perfil del comprador.

Siguiendo el análisis en otra etapa, ¿qué fue lo que compro, hora, productos?, estos hechos dicen más acerca del cliente potencial.

Este criterio empleado en el ejemplo es para modelar un perfil de hábitos de compra de productos disponibles en cualquier tienda de autoservicio.

De aquí se puede partir para desarrollar múltiples modelados de perfil de cliente y así desarrollar nuevos productos.

Sin embargo, con la evolución de la tecnología y el uso de medios de información masivos tendenciosos, a la gente se le está obligando a pagar con tarjetas de

crédito o débito para qué no carguen efectivo, pero la realidad es que cada vez que se hace una compra por estos medios se pierde el anonimato y se forma un expediente de nuestros hábitos de compra, creando un perfil de costumbres, estos datos de compra alimentan grandes computadoras que desarrollan los nuevos modelos de negocios de los años venideros, la información se ha convertido en el oro negro de este siglo.

Estos datos estadísticos y con herramientas apropiadas se puede deducir que productos busca la gente, así se configura un modelo dinámico de tendencias del mercado para modelar el perfil psicológico del producto.

Por mencionar otro ejemplo, si el concepto trata de personajes bélicos, estos se deben diseñar en función a la eficiencia disfrazada de paranoia, trastorno de personalidad o defectos de carácter. Y como son muy comunes estas anomalías será fácil que el comprador se identifique con el producto que se le vende.

Esta cualidad se toma para enfatizarla a través del diseño de carácter, por decirlo así, endurecer sus facciones, atribuirle un cuerpo atlético, fuerza, agilidad, inteligencia, estas cosas las hace el departamento de arte al leer el guion y se dan la idea del concepto entonces proponen varios bosquejos del carácter, en función del perfil psicológico del concepto

1.2.4 CUANDO SE DEBE RE ORIENTAR EL CONCEPTO

Si por alguna causa de fuerza mayor, se viera comprometida la idea original del concepto o el concepto mismo desarrollado se debe plantear una salida decorosa como una edición del proyecto o aplazar el lanzamiento del producto, o en última instancia cancelarlo para evitar más pérdidas, terminarlo y enlatarlo para promoverlo tiempo después y no comprometer todavía más la inversión.

A veces suceden eventos de fuerza mayor, que por su naturaleza logran afectar negativamente al producto y el resultado es catastrófico.

Es como promocionar una película de terremotos cuando se acaba de vivir una situación similar.

1.3 ARQUITECTURA DE CONCEPTO NARRATIVO

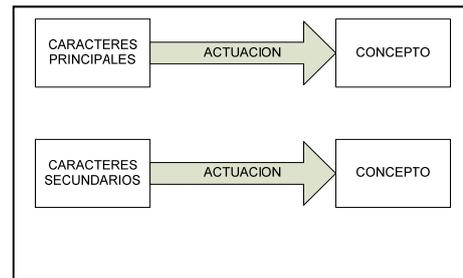
Es la forma de cómo la historia se organizara, existen dos formas para narrar una historia a través de caracteres o de eventos.

Pueden ser narradas por:

CARACTERES PRINCIPALES
CARACTERES SECUNDARIOS

FIGURA 1.4

Figura 1.4 ESTA GRAFICA ILUSTRAS LAS ARQUITECTURAS DEL ARGUMENTO PARA CARACTERES PRINCIPALES O SECUNDARIOS LOS RECTÁNGULOS INDICAN QUIENES SON LOS ACTORES O LOS EVENTOS Y EL RESULTADO FINAL TRANSFORMADO EN CONCEPTO



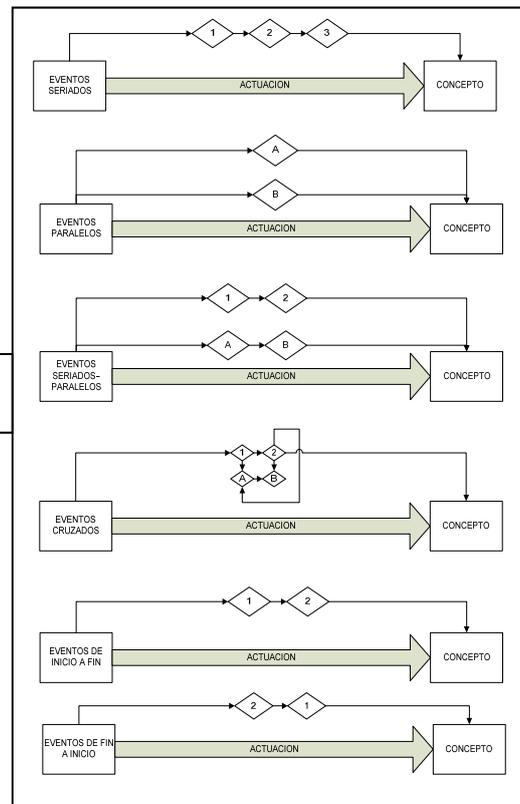
Y POR
EVENTOS SERIADOS
EVENTOS PARALELOS
EVENTOS SERIADOS Y PARALELOS
EVENTOS CRUZADOS
EVENTOS DE INICIO A FIN
EVENTOS DE FIN A INICIO

Figura 1.5 ESTA GRAFICA ILUSTRAS LAS ARQUITECTURAS DEL ARGUMENTO PARA EVENTOS SERIADOS, EVENTOS PARALELOS, EVENTOS CRUZADOS, EVENTOS DE INICIO A FIN Y DE FIN A INICIO.

LOS ROMBOS INDICAN LOS EVENTOS, LAS FLECHAS Y SU DIRECCIÓN INDICAN SU ORGANIZACIÓN CON RESPECTO AL TIEMPO.

LOS RECTÁNGULOS INDICAN QUIENES SON LOS ACTORES O LOS EVENTOS Y EL RESULTADO FINAL

FIGURA 1.5



1.3.1 ESCRITORES

Los escritores son la piedra angular de toda idea que se desea proyectar hablando de historias llevadas a la pantalla grande y en este caso de los videojuegos.

Ellos integran elementos de la realidad para transformarlos en una historia que el público desea conocer a través de novelas o publicaciones.

Su función como escritor es estructurar una historia partiendo de una idea, la organización, el estilo y la imagen a proyectar.

La idea principal se puede perder pero aun así se puede lograr mucho con las ideas secundarias que bien estructuradas podrán dar vida a conceptos más sólidos en filosofía.

La forma de narrarla debe ser entretenida, se logra estableciendo un mapa de eventos, un mapa de caracteres, un mapa de secuencias, un mapa de actos, un mapa de perfiles, un mapa de condiciones, todo esto en función al tiempo.

Se grafica la secuencia de eventos para mostrar la historia de forma entretenida e interesante.

Crear la personalidad de los caracteres refleja toques de realismo al juego, muestra ciertas esferas de perfiles que el jugador se identifica con ellos.

Para darle estilo a las ideas se requiere de alguien que su trabajo sea ese adaptar la idea a los requerimientos que el perfil del segmento de mercado demanda.

Los escritores son observadores de la realidad y obtienen sus ideas de la misma, entonces ellos pueden ajustar la narrativa para hacer cambios en el contexto original.

Dicho de otro modo ellos pueden narrar una historia donde un carácter heroico se vea como un siniestro personaje y conforme se desarrolla la trama aparecerán nuevas pistas que develaran sus verdaderas intenciones.

En el caso de historias narradas en videojuegos se puede cambiar el desenlace de la historia oficial y transformarla en un nuevo enfoque dado por las características que ofrece el videojuego.

1.3.2 GUIÓN GRÁFICO (STORY BOARD) Y ARGUMENTO

Teniendo la idea clara, darle vida a través de una narrativa descriptiva coherente con una ambientación adecuada se procede a revisiones, borradores, agregar o eliminar cabos sueltos, aclarar puntos, eliminar o agregar ideas, lo importante es la secuencia de los eventos involucrados sea coherente. Para esto se crean mapas de conglomerados de ideas estructuradas y así visualizarlas en el contexto general para no dejar elementos redundantes, confusos, absurdos etc. Se puede manejar como una edición, la cual al ser visualizada a través de mapas, se pueden notar fallas y eliminarlas. La línea de la historia se puede manipular según las circunstancias que el jugador vaya ejecutando al decidir el rol o perfil de personaje seleccionado o diseñado. Esto aplica según el tipo de videojuego.

Muchas veces entrar en historias complejas llega a ser tedioso y aburrido por lo que se trata de omitirlas, cuando el público comprador demanda más material

conceptual se recurren a estrategias de marketing como ofrecer videos, películas, libros, y muchos productos que cubren esas expectativas del mercado meta.

Algunas veces se da el caso de que un videojuego llega a convertirse en película pero la experiencia muestra que es tanta la expectativa del mercado que cuando llegan a ver este subproducto la decepción es tan grande que el producto se deja de vender.

Estas decisiones no dependen del escritor ni del guionista que hace la historia para productos de marketing secundario, depende de la junta de accionistas que están decidiendo sobre el futuro del modelo de negocios. Muchas veces ellos toman decisiones erróneas y convencerlos de tomar la decisión adecuada es totalmente un juego de azar ya que si se equivoca uno, lo pueden a uno despedir por incapacidad mental o lo que se les ocurra. Es la culminación de una idea, el proyectarla como un producto listo a ser considerado como una propuesta, con un diseño estructurado, con un inicio y un fin, con un estudio de mercado que respalda su rentabilidad por sus cualidades singulares. El guion escrito es la integración final de los elementos de un diseño conceptual. Es el producto final antes de ser nuevamente sometido a revisión, el guion grafico es el resultado de la parte conceptual del proyecto, a partir de aquí se pueden analizar nuevas propuestas para ser corregidas o adaptadas, una vez teniendo el guion grafico final empieza la preproducción que ya es el manejo de recursos y resultados en función al tiempo. Tomando como referencia un guion grafico es como se puede diseñar las escenas que integran al video juego. Partiendo del guion grafico el diseño del mapa de acción del video juego es relativamente fácil porque los recursos y el capital humano solamente se van asignando a tareas específicas en función del tiempo. Como se ha visto durante los títulos anteriores, un guion grafico es el conglomerado de ideas transformadas en conceptos visuales que reflejan diferentes esferas, las cuales al ser vistas crean emociones en el publico porque este se identifica con el concepto visual.

Estos conceptos visuales llevan técnicas para despertar el interés del mercado meta. Estos diseños visuales involucran:

- Dibujo**
- Tipo de arte**
- Expresiones faciales**
- Fondos**
- Mobiliario**
- Inmobiliario**
- Sombras**
- Iluminación**
- Vestuario**
- Sensualidad (si lo amerita)**
- Personaje**
- Poses**

Todos estos criterios suelen ser sujetos a cambios para mejorar la perspectiva de la idea misma.

1.3.3 ARTE Y DISEÑO VISUAL

El diseño visual conforma esferas de ambientación coherentes y en armonía que involucran caracteres, propiedades físicas y psicológicas, vestuarios, escenarios, detalles, lenguajes, estilos, artefactos, mobiliario, inmobiliario dentro de un contexto visual acorde a la idea conceptual en proyecto. Esto conforma la ambientación. Cuando hablamos de cualquier proyecto sin importar su naturaleza, hablamos de estética, del equilibrio de las cosas para que mantengan su armonía en el escenario. Es muy fácil notar cualquier elemento que este fuera de contexto y criticarlo por su notoriedad. En el diseño visual el objetivo es apropiarse de la mente del comprador a través de la armonía del escenario con todos sus elementos. La calidad de estos elementos se cuantifica por investigaciones documentales fidedignas o si se trata de diseños visuales ficticios por su apego a características técnicas y científicas del momento que se adecuan a al entorno visual del concepto incluso temas del argot técnico y científico sin embargo esto último suele ser un error frecuente por su falta de originalidad. Muchas veces el diseño visual es lo que captura al cliente por que evoca sus más profundos sentimientos.

En los videojuegos todo es arte, y el arte es la libre expresión de la psique humana. Mostrar ideas a través de conceptos visuales bien claros es la consecuencia de un arte enfocado a los negocios. El arte del guion grafico es el reflejo de enfatizar ciertas esferas de la psique humana para obtener una emoción deseada. Se busca generar emociones como alegría, tristeza, preocupación, ira, miedo. Existen diferentes filosofías para explicar el arte y generar emociones a través de ella. Por ejemplo el yin y yang refleja visualmente el equilibrio de estas emociones. Los cinco elementos son empleados para reflejar el equilibrio a través de un pentagrama, cuando hay un desequilibrio en estos se genera una emoción positiva o negativa. De este modo el arte buscara siempre capturar la imaginación del mercado meta. El diseño de guion escrito es la parte fundamental porque la pre producción de ahí toma las referencias básicas para presupuestar el diseño de producción y enviar la propuesta al ejecutivo.

Es la proyección visual de los elementos que componen al escenario, esta área investiga la ambientación en función a la época, la funcionalidad, la interacción de sus componentes para manejar esferas visuales coherentes en el contexto.

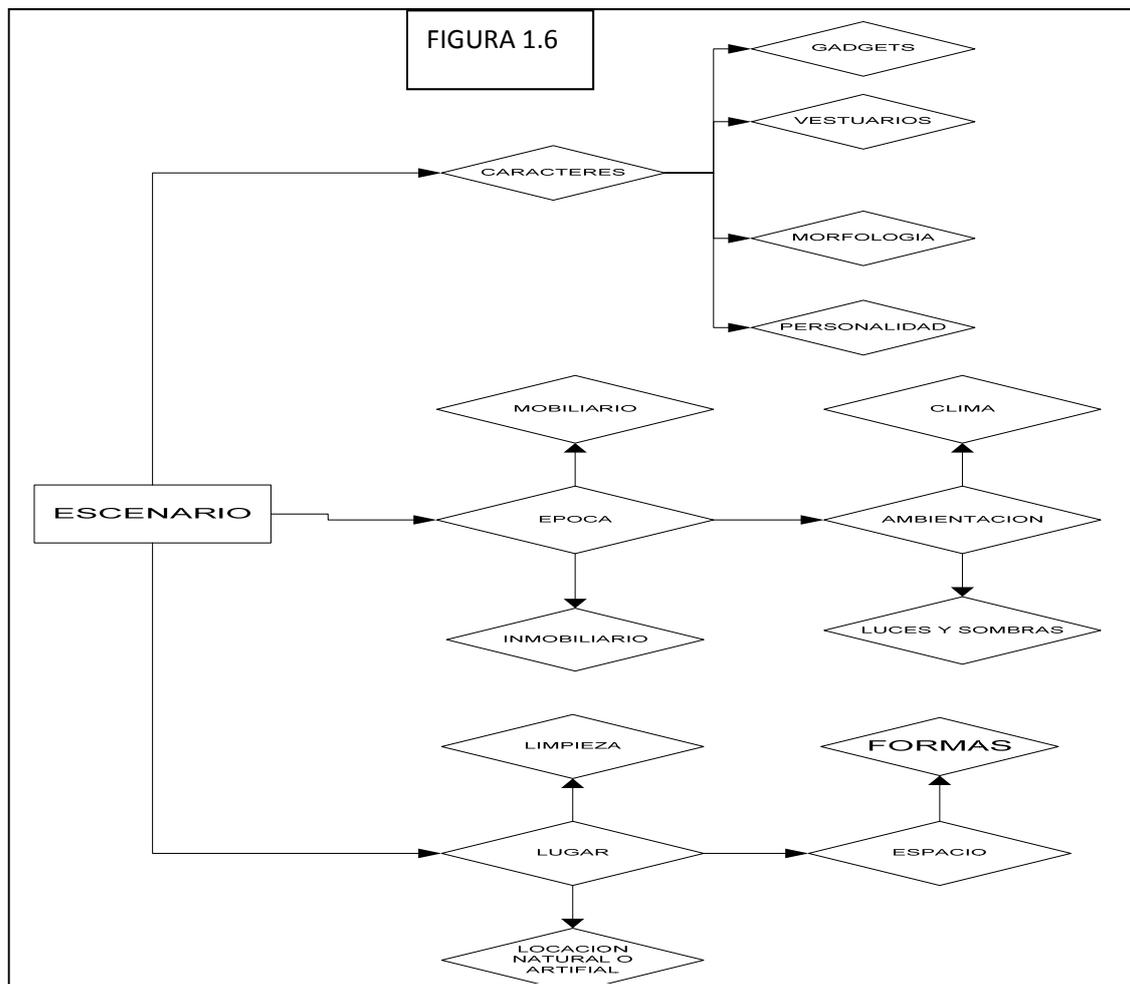


Figura 1.6 Este mapa muestra los elementos del arte técnico. Se refiere principalmente a darle la ambientación correcta al escenario y ganar la seriedad del comprador mostrando un producto de calidad donde se vea reflejada con los más mínimos detalles en su presentación. El arte técnico es una investigación concienzuda del guion escrito con la finalidad de proyectar las ideas y eventos en tiempo y forma.

En función al concepto, la creación de escenarios va ligada al perfil psicológico del concepto, el diseño de arte de fondo está motivado para generar emociones humanas a través de esferas de ambientación como son el clima, la luz, las formas, los colores, las sombras, elementos de mobiliario e inmobiliario, artefactos, personajes de reparto y personajes secundarios.

El sonido participa de manera estructural en la ambientación ya que el cerebro humano al escuchar interpreta esta esfera de diferentes formas emocionales que dependen de la musicalización y la forma en cómo esta paleta de sonidos es manipulada para crear suspenso, drama, pánico, tristeza, dolor, pesar, agonía, etc....

Los manejos de contexto de luz y sonido en un fondo generan estas emociones, muchas veces no se necesita que el fondo este completamente visible sino que el cerebro "vea" y llena esos espacios. La dinámica de estas dos variables motiva atmosferas de diseño visual muy variadas manteniendo la atención del usuario.

Esto se refiere al enfoque que se da al escenario, es decir si a través de ideas conceptuales se captura la atención del comprador.

Ya que la idea se ha convertido en concepto y tiene objetividad se comienza a darle el giro a su lugar en el tiempo y espacio para darle la ambientación correcta.

La ambientación consiste en que el grupo de especialistas en diseño de ambientación y arte investigaran y agregaran todos los elementos disponibles para darle al concepto la vida que requiere para proyectar las esferas necesarias dentro de la historia, como son vestuario, mobiliario, inmobiliario, artefactos, lugares, paisajes, luz, sombras, sonido, diseño de personaje, diseño de diálogos.

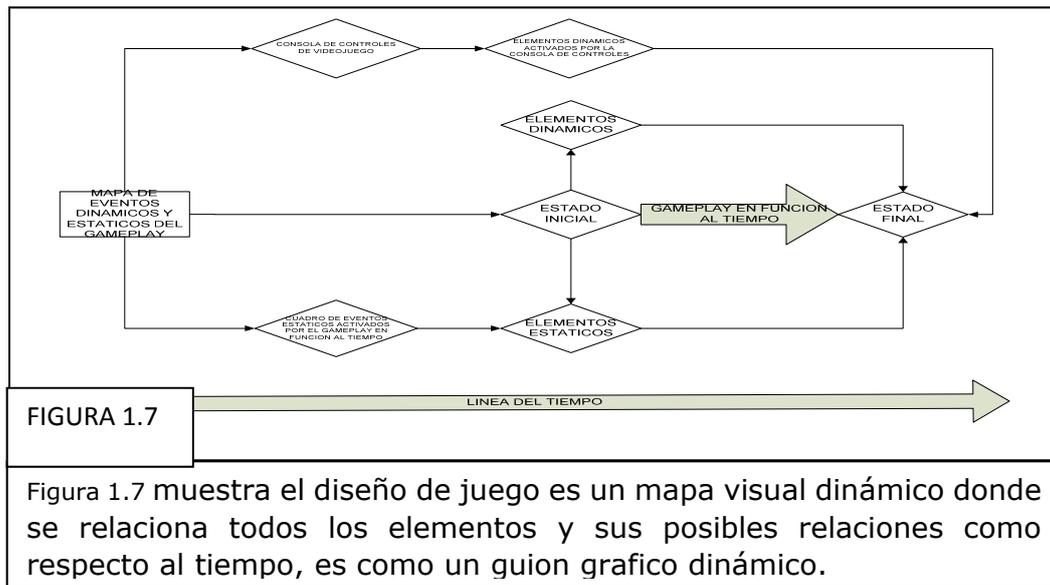
Una vez obtenida la parte conceptual que involucro desde la elección de la idea misma, su depuración, su perfeccionamiento, su enriquecimiento, su diseño, el diseño visual se forma con las ideas básicas del concepto y está enfocado a capturar al mercado meta con el propósito de que el cliente se identifique con lo que ve.

En la psique del cliente el diseño visual transporta al cliente a una atmosfera de sentimientos profundos desatados por la interacción del producto compuesto de escenarios, acciones, trama, historia, sonido, luces, sombras etc. ... con el cliente mismo al interactuar con el producto.

1.3.4 DISEÑO DE JUEGO Y DE PARTIDA

Basado en el guion grafico es como se proyectara el esquema del videojuego, de inicio a fin y con todos sus respectivos elementos, dividido en etapas.

El diseño del juego en si es la estructura de la sucesión de eventos con sus respectivas interacciones en el escenario, dicho de otra forma es un organigrama visual donde se proyecta la narrativa con sus respectivas secuencias y acciones de cada evento en función al guion gráfico. Cada segundo de gameplay el jugador interactuara con el escenario virtual el cual está compuesto por un número finito de elementos con los cuales se interactuaran tales como objetos físicos, rankings, indicadores, voces, otros jugadores, sonidos, efectos visuales, efectos de sonidos, escenarios dinámicos, escenarios estáticos. Por esta cantidad de eventos simultáneos el diseño de juego se organiza para reducir errores y si los hay corregirlos fácilmente. Esta es una estrategia para reducir tiempos en producción. Durante la concepción del producto que va a producirse, se han discutido diferentes opciones para plantear el proyecto principalmente hablando de la parte conceptual, la parte creativa es relativamente fácil porque se programa una producción con objetivos claros en función a tiempos determinados.



La partida es la parte más importante del juego, plantea el modo de juego, las reglas, haciendo notar que determinara la dinámica del juego.

Se plantean los objetivos del juego, el sistema de puntuación, se intenta no hacer controversial una partida, se plantea con reglas sencillas y breves para no crear espacios sin resolver.

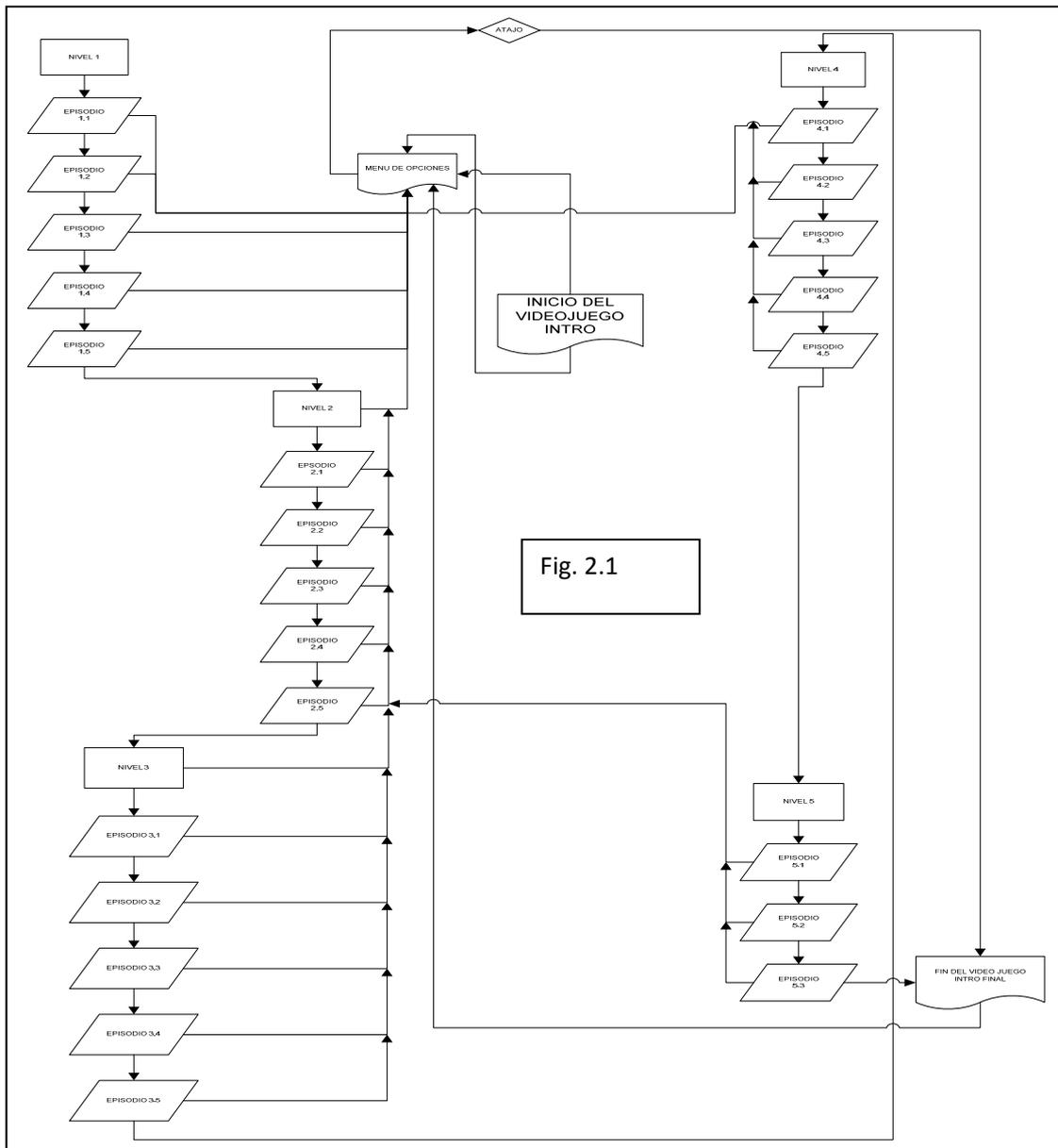
Como ejemplo la partida puede ser con el ajedrez, con unas cuantas reglas se pueden hacer infinidad de combinaciones en un escenario determinado por la cantidad de cuadros blancos y negro

2 PREPRODUCCIÓN, PRODUCCIÓN, POSTPRODUCCIÓN

2.1 DISEÑO DE PRODUCCIÓN

El diseño de producción es el organigrama en detalle de toda la producción. En función al tiempo todas las actividades estarán ejecutándose en tiempo y forma, con sus respectivos responsables,

MAPA MAESTRO DEL VIDEOJUEGO



esta distribución se observa en la fig. 2.1

Partiendo de este mapa maestro del videojuego se transforma en un producto tangible con métricas y resultados esperados.

Entre las distintas labores que requiere la realización de un videojuego, el diseño de producción es la especialidad de quien decide en líneas generales el aspecto que tendrán los escenarios en los que se desarrolla la acción, ya sean éstos fabricados (decorados) o previamente existentes (localizaciones).

Trabajando en estrecha relación con el director, toma todas las decisiones sobre la forma, el color y el estilo plástico del videojuego.

Habitualmente, esta figura sólo es necesaria en producciones de gran envergadura y en las que trabajan varios equipos dedicados a la escenografía, cada uno encabezado por su respectivo director artístico, ya que se trata de quien los coordina, garantizando que todos ellos sean fieles a la estética general previamente acordada.

Así pues, puede decirse que, en estos casos, la tarea fundamental del diseñador de producción es previa a la del director artístico y consiste en planear, por medio de bocetos ilustrativos, dibujos detallados o guion gráfico, lo que el segundo tendrá que realizar en la práctica durante el rodaje.

El diseño de producción es que determina el contenido visual de la película y de ella depende la que realizan los decoradores, diseñadores de vestuario y los efectos especiales con el fin de ayudar al director a conseguir el ambiente idóneo para la historia que narra.

En esta parte se tiene un producto bien definido, listo para ser producido por que están claramente vistos los alcances y límites. Y se sabe lo que se quiere proyectar al concluir el trabajo.

Para llevarlo a buen término se aplican criterios bien conocidos de eficiencia, eficacia, calidad, productividad en función a un capital y un periodo de tiempo finito. El aparato administrativo aplicado es el mismo que se emplea para cualquier proyecto de inversión razonable. Con el conocimiento necesario para concluirlo satisfactoriamente y la idea es convertirla en realidad con los recursos disponibles. Además de que el resultado obtenido sea el que se prometió. Entrando de lleno en materia la preproducción es la puesta en marcha del proyecto a nivel administrativo. Se aplicaran métodos, modelos, métricas, experiencias, decisiones, recursos humanos, técnicos, tecnológicos, financieros, todo esto estimado en base a presupuesto en función al tiempo.

Cada departamento con sus respectivas métricas de eficiencia entregara constantemente resultados a la administración, orquestándose así el buen término del producto. Se requiere un mapeo del video juego, de ahí parten todas las actividades hasta su conclusión.

MAPA DE RESPONSABLES

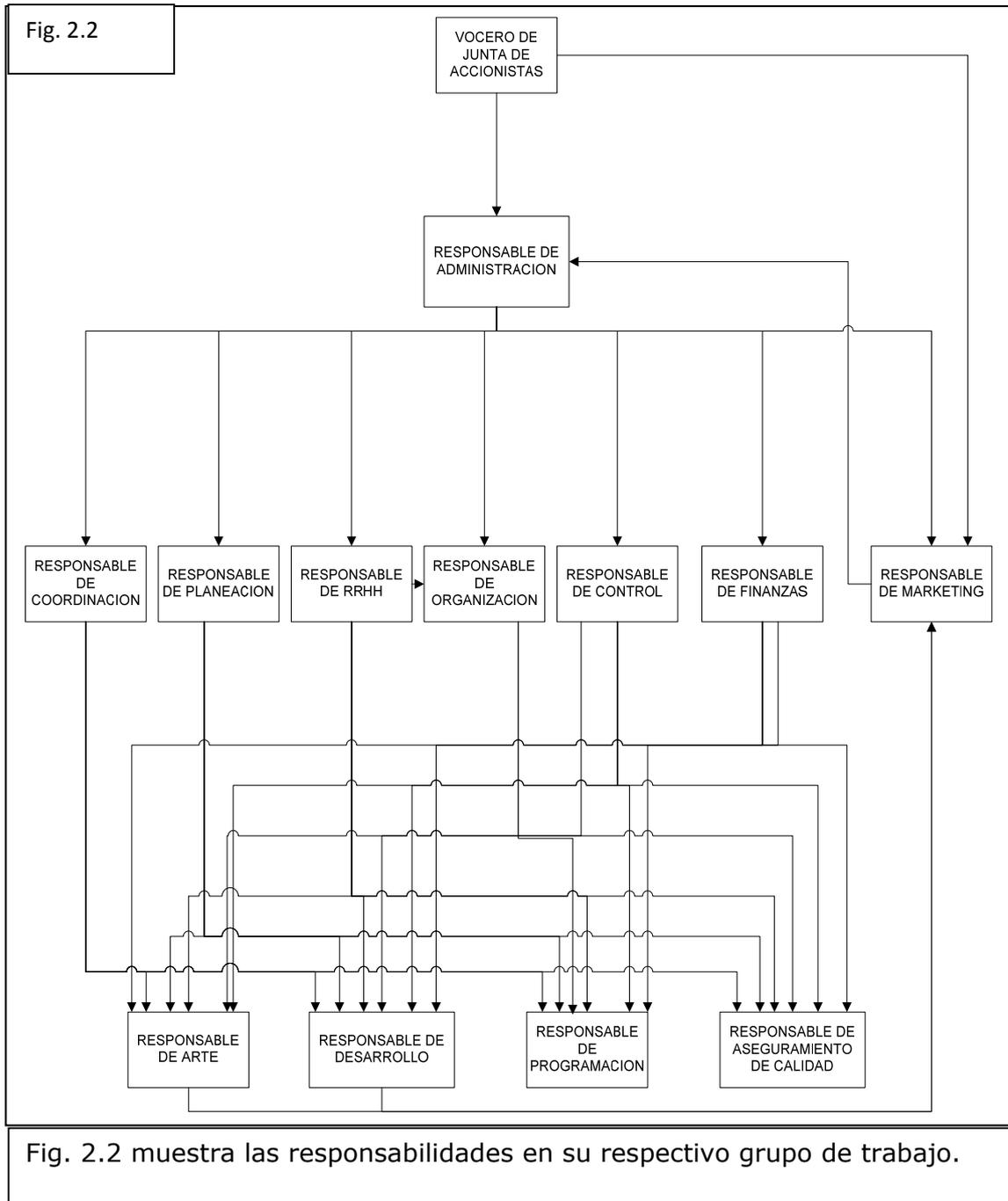


Fig. 2.2 muestra las responsabilidades en su respectivo grupo de trabajo.

En el mapa siguiente se pueden ver todos los grupos de trabajo involucrados en el proyecto, así como sus respectivas interrelaciones y gestiones entre un grupo y otro. Todos se monitorean por que las actividades han sido diseñadas secuencialmente poniendo a prueba constantemente el aseguramiento de la calidad en cada momento para evitar retrasos por operaciones incorrectas. Toda la información que fluye es bidireccional, para tener presente el control de todas las estadísticas de los procesos involucrados así como resultados tangibles.

MAPA GENERAL DE LA EMPRESA POR GRUPOS DE TRABAJO

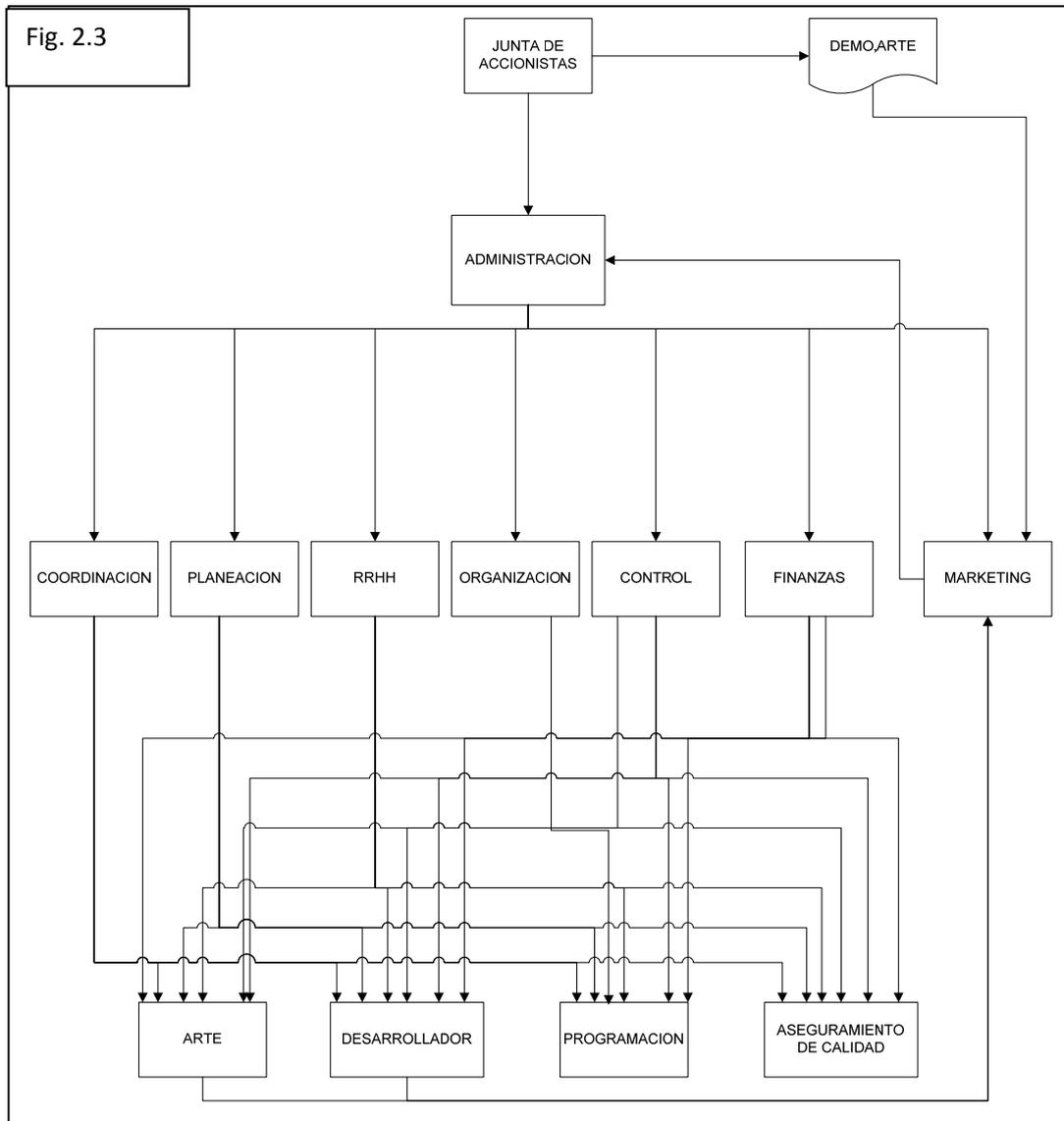


Fig. 2.3 muestra la empresa de videojuegos dividida en grupos de trabajo, las líneas y flechas señalan el flujo de las relaciones entre los grupos que son información, resultados en tiempo real.

2.1.1 MAPA DE TRABAJO POR DEPARTAMENTO

Explicando este mapa, observamos que está la etapa artística que entrega activos en video, audio y gráficos a la etapa desarrolladora esta proporciona interactividad de los elementos proporcionados por la etapa artística.

En la etapa de programación se están integrando todos los elementos de la etapa desarrolladora donde se integra la 3D, audio y la red. Por último en la etapa final de aseguramiento de calidad, el producto se reproduce para pruebas y tener un mapeo estadístico de fallas que se canalizan al grupo de trabajo correspondiente.

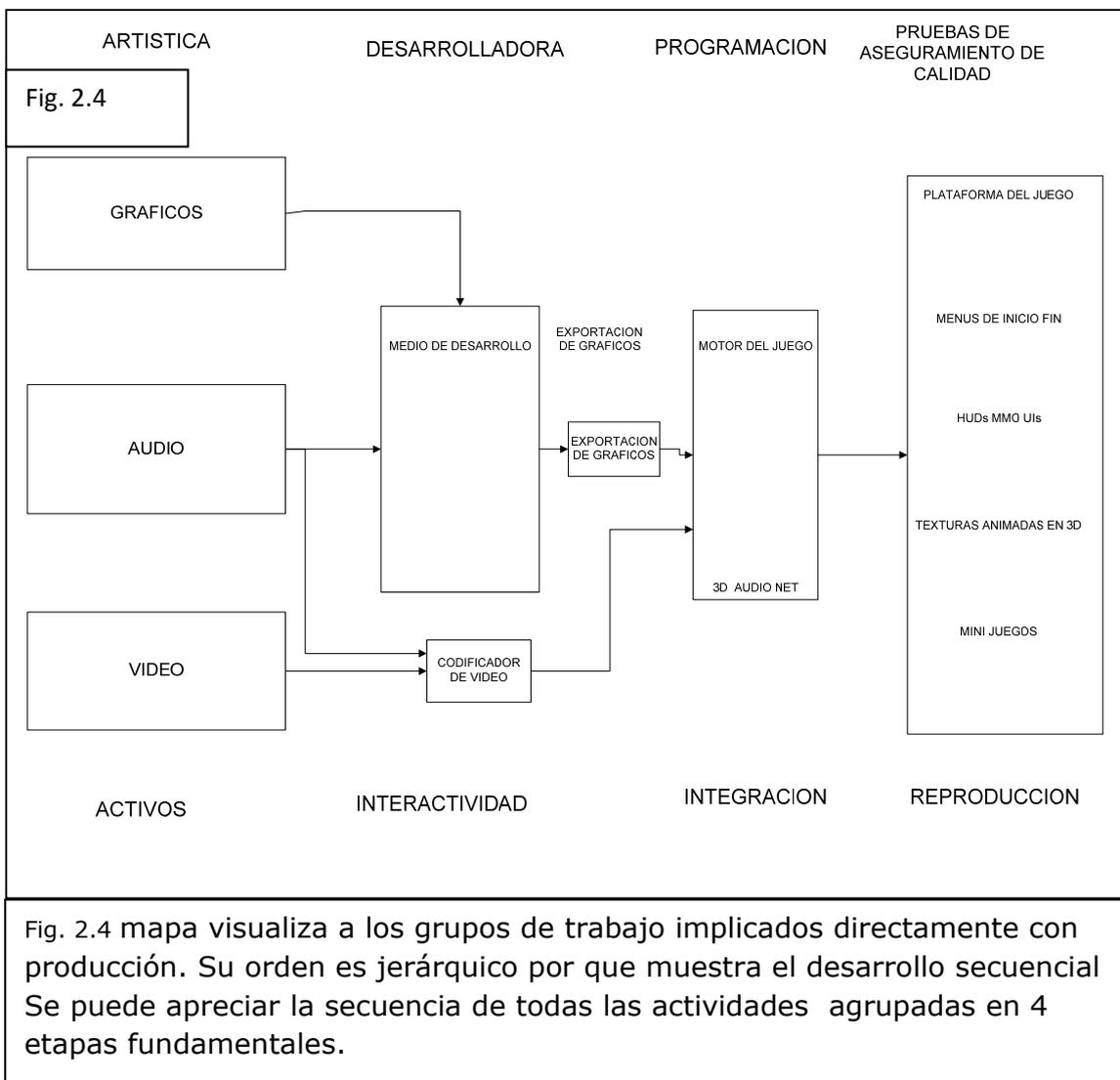


Fig. 2.4 mapa visualiza a los grupos de trabajo implicados directamente con producción. Su orden es jerárquico por que muestra el desarrollo secuencial Se puede apreciar la secuencia de todas las actividades agrupadas en 4 etapas fundamentales.

2.1.2 ESQUEMA DE ORGANIZACIÓN DEL GUIÓN GRAFICO

Guion grafico segmentado por niveles

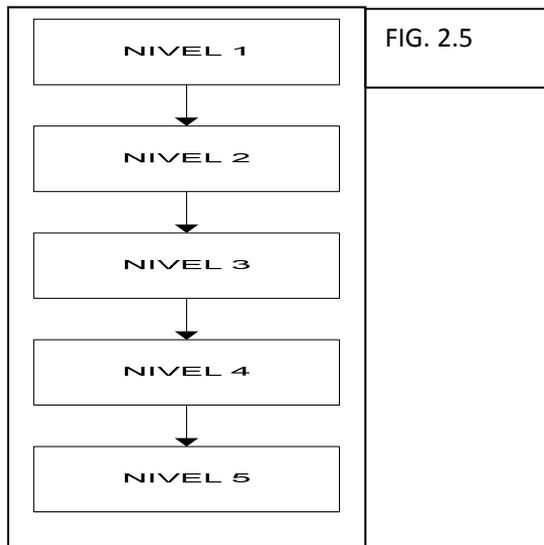


Fig. 2.5 muestran la forma más común de ordenar un esquema secuencial de niveles y de escenas, la idea es tener el control del proyecto por medio de este tipo de herramientas gráficas y los datos del responsable a cargo así como anotaciones de fechas, observaciones, entre otros detalles.

Cada nivel es segmentado por escenas

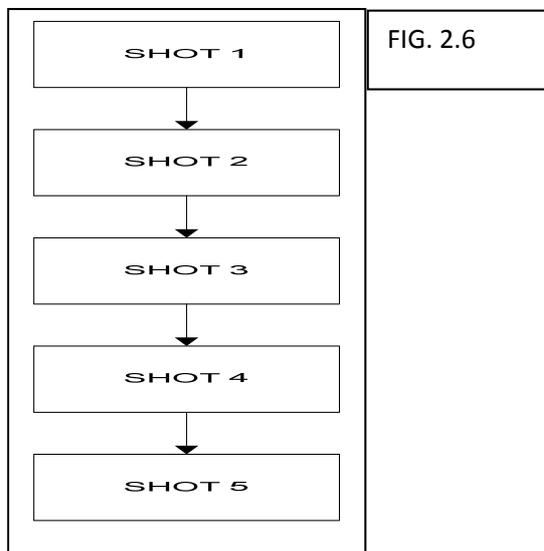
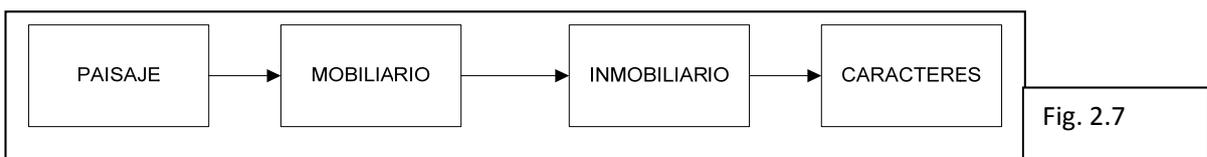


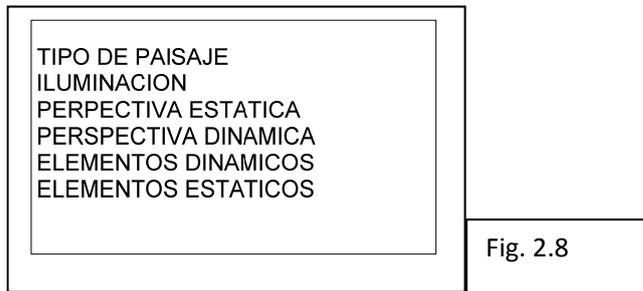
Fig. 2.6 y 2.7 muestran la forma más común de ordenar un esquema secuencial de niveles y de escenas, la idea es tener el control del proyecto por medio de este tipo de herramientas gráficas y los datos del responsable a cargo así como anotaciones de fechas, observaciones, entre otros detalles.

Cada escena se compone de caracteres, paisajes, mobiliario, inmobiliario

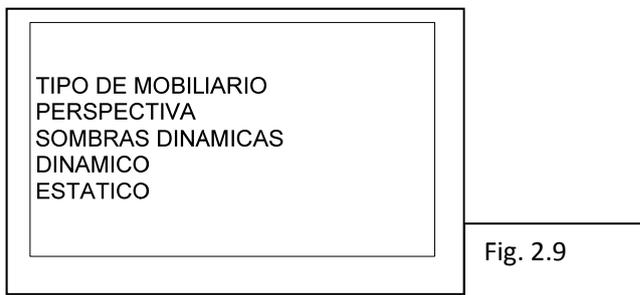


En cada escena se compone de cuatro capas básicas

PAISAJE



MOBILIARIO



CARÁCTER

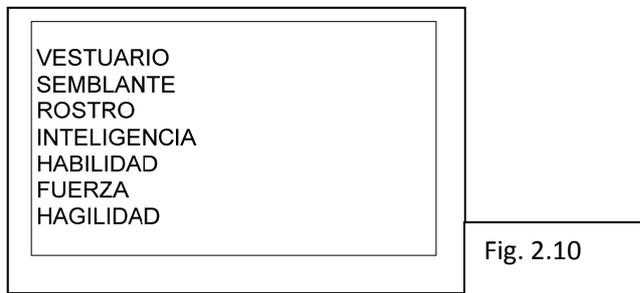


Figura 2.8, 2.9, 2.10 y 2.11 muestran los elementos integradores de una escena del videojuego.

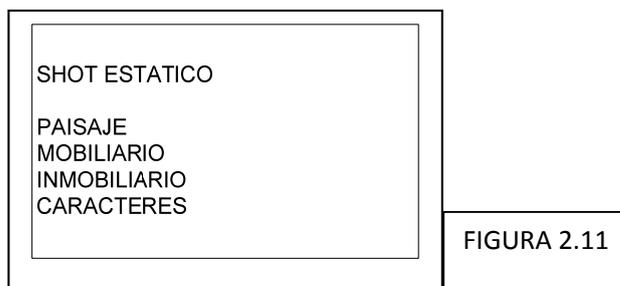
Administrados por capas de aplicación, los elementos del paisaje, del mobiliario y del carácter mismo (o caracteres según el caso) interactuarán en función a los eventos que el jugador active conforme el gameplay se desarrolle,

Hay elementos dentro del paisaje y del mobiliario que aunque no se efectúen acciones por parte del jugador estas actuarán de forma estática, por decirlo de algún modo, al estar en la escena.

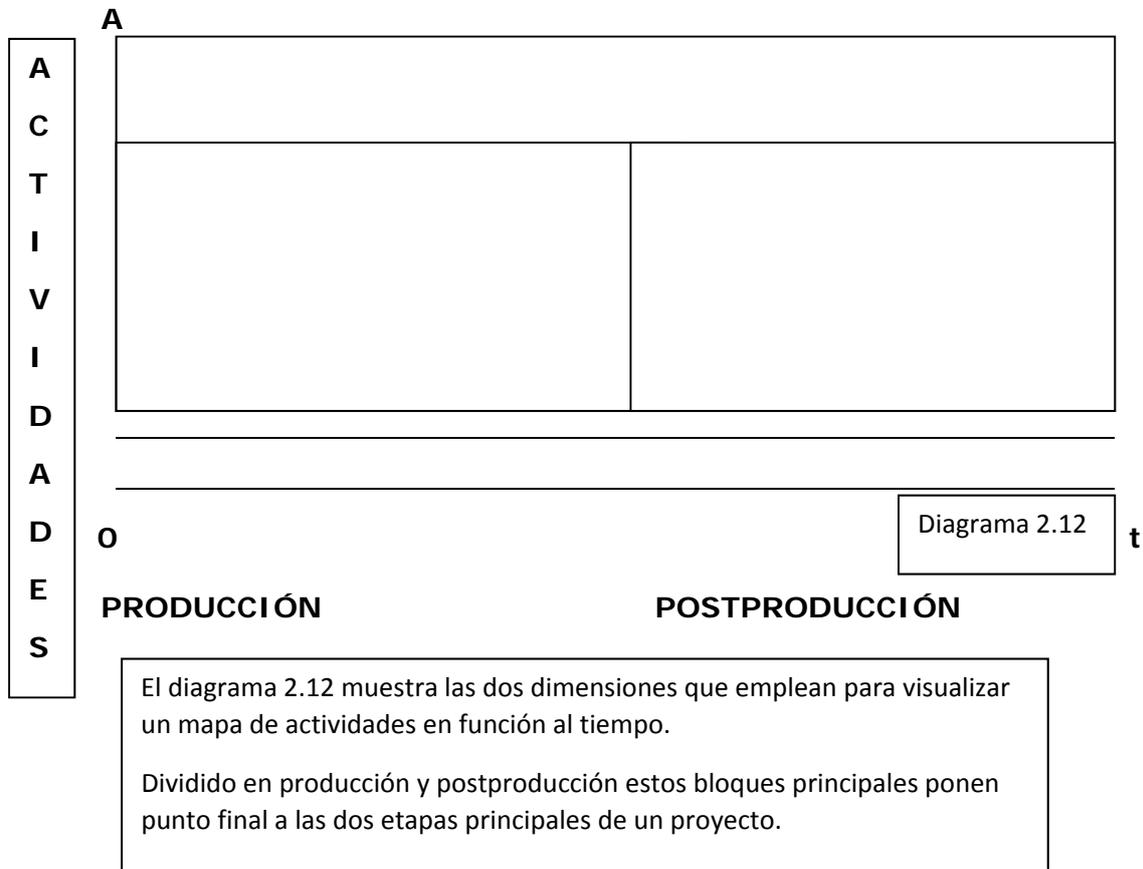
Una vez que estos elementos están listos se procede a armar la escena por capas para asignar eventos al conjunto.

El sonido se va agregando conforme las escenas van siendo ensambladas.

En lo que corresponde a la escena estática se emplea como base para la escena dinámica.



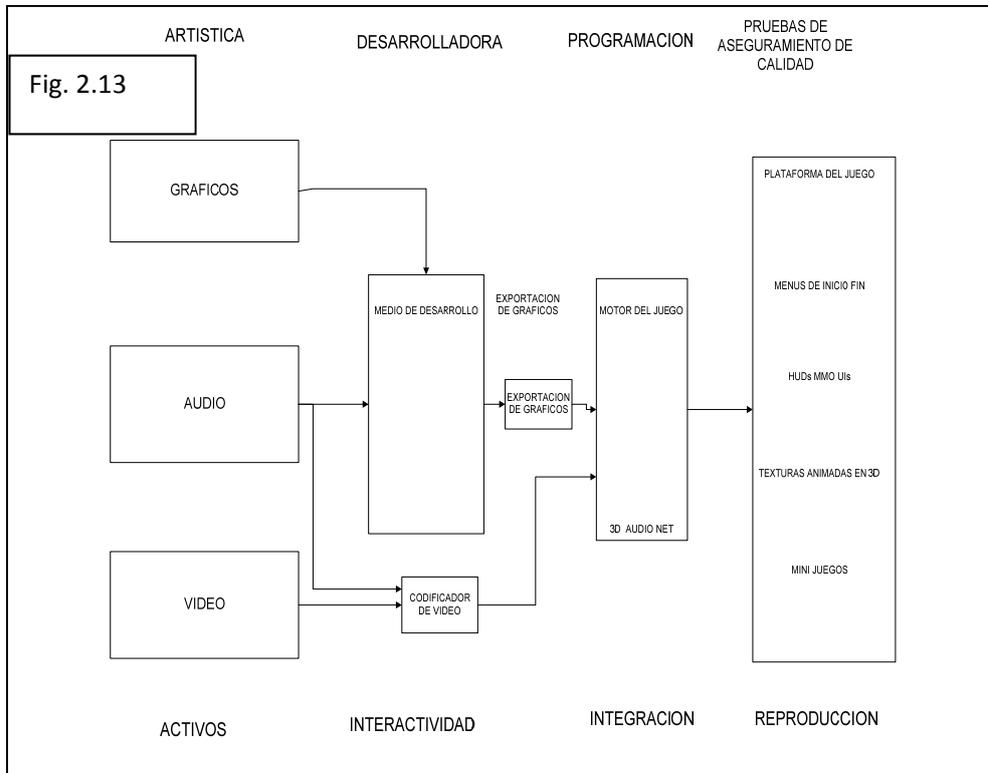
Organigrama de actividades para producción y postproducción



En este diagrama se mapea la línea del tiempo con respecto a las actividades planeadas. Para su ejecución se proyectan esquemas temporales de actividades.

Con este mapeo de actividades segmentas se asignan a sus respectivos departamentos.

2.1.3 DIVISIÓN DEL TRABAJO POR ÁREA DE APLICACIÓN



La fig. 2.13 muestra la división del proyecto por áreas de aplicación.

Dividido el proyecto en cuatro áreas fundamentales que son:

- Artística
- Desarrolladora
- Programación
- Aseguramiento de calidad

En cada respectiva área de aplicación el proyecto contempla los siguientes elementos.

Artística: es el desarrollo de los patrones visuales y acústicos que son los activos del videojuego.

Desarrolladora: se maneja la codificación del video y el medio de desarrollo del proyecto, es decir la interoperabilidad de elementos.

Programación: se integran al motor del juego los gráficos y el video además del sonido, la 3D y la red.

Aseguramiento de calidad: es la reproducción del videojuego, asegurando su calidad en su operación de los elementos integradores como son las texturas, los gráficos, sonido, plataformas, menús.

2.1.4 DIVISIÓN DEL PROYECTO POR ETAPAS Y GRUPOS DE TRABAJO

En producción el trabajo está dividido en 3 departamentos los cuales trabajan secuencialmente:

**Artística
Desarrolladora
Programación**

En postproducción el departamento de aseguramiento de calidad hace requerimientos de revisión de productos que no cumplen la norma mínima de calidad, por lo que los cuatro departamentos esta en constaste trabajo

Aseguramiento de calidad

**Artística
Desarrollador
Programación**

Esta división de responsabilidad se compone de los siguientes departamentos.

Artística: gráficos, audio, video

Desarrolladora: medio de desarrollo y codificación del video.

Programación: motor del videojuego, 3D (tercera dimensión), audio, NET (red). Aseguramiento de calidad: plataforma de juego, menús, HUD (head-up display, información que llega en todo momento a la pantalla), 3D.

En cada departamento hay un responsable que entregara resultados en función a la planeación.

Nivel 1	JUNTA DE ACCIONISTAS
Nivel 2	RESPONSABLE DE ADMINISTRACIÓN
Nivel 3	RESPONSABLE DE DIRECCIÓN RESPONSABLE DE COORDINACIÓN RESPONSABLE DE PLANEACIÓN RESPONSABLE DE RR.HH. RESPONSABLE DE ORGANIZACIÓN RESPONSABLE DE CONTROL RESPONSABLE DE FINANZAS RESPONSABLE DE MARKETING
Nivel 4	RESPONSABLE DE ARTE RESPONSABLE DE DESARROLLO RESPONSABLE DE PROGRAMACIÓN RESPONSABLE DE ASEGURAMIENTO DE CALIDAD

2.1.5 MÉTRICAS DE PRODUCTIVIDAD

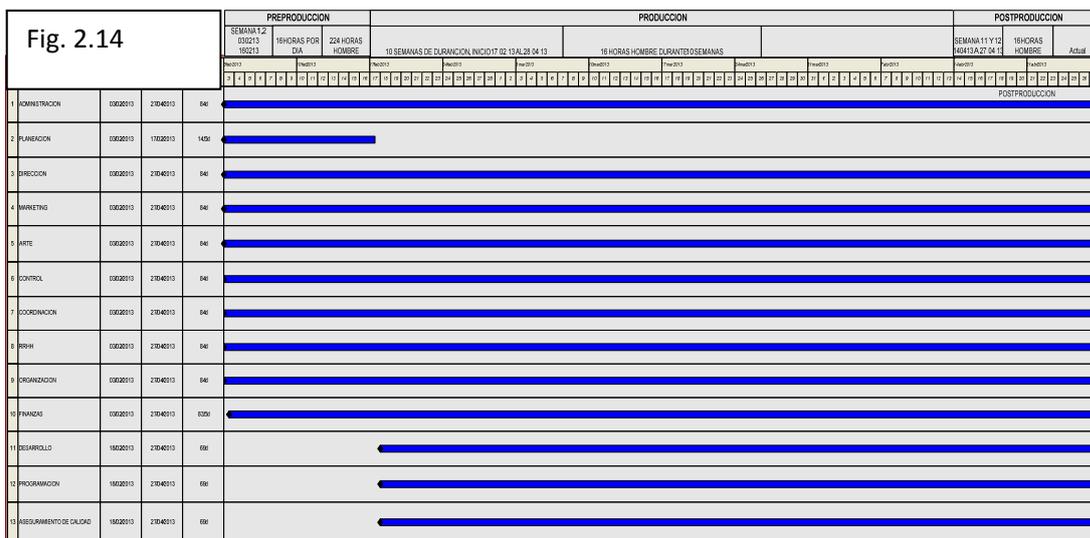
Como en todo proyecto, una vez establecida la planeación por objetivos, se cuantifica en periodos de tiempo cada actividad para otorgar esquemas secuenciales en función a la duración de cada actividad, el tiempo que se asignara para cada actividad estará prorrateado, considerando ciertas holguras para la terminación de cada actividad. Importante es el cumplimiento de estas actividades, el retraso se paga caro y la productividad se compromete. Esta metodología de tiempos y movimientos se basa en la capacidad de entender una operación y dominarla. Entonces se procede a cuantificarla en términos de calidad en función al tiempo. En una gráfica de 3D se puede observar como un evento esquematiza por orden de acción, se cuantifica su duración, y se califica la calidad del trabajo entregado por evento completo. Sin embargo se analiza en que parte del proceso de generan más fallas y se mejora la aplicación del procedimiento. Entonces el resultado es una mejora del proceso por unidad de tiempo determinado. Para cada actividad del proyecto se cuantifica en unidad de tiempo todas las actividades las cuales ya tienen un número de control. En términos prácticos y de conocimiento, estas actividades se asocian por parecido y se logra la optimización de los recursos. Dicho de otro modo todas las actividades se jerarquizan por repetibles y semejantes, entonces se hacen bloques de trabajo para su ejecución.

2.1.6 CRONOGRAMA DE ACTIVIDADES

Esta es una racionalización del tiempo del tiempo disponible y las actividades involucradas.

Se estandariza de forma secuencial.

Para estos eventos se crea un calendario basado en horas hombre (fig. 2.14)

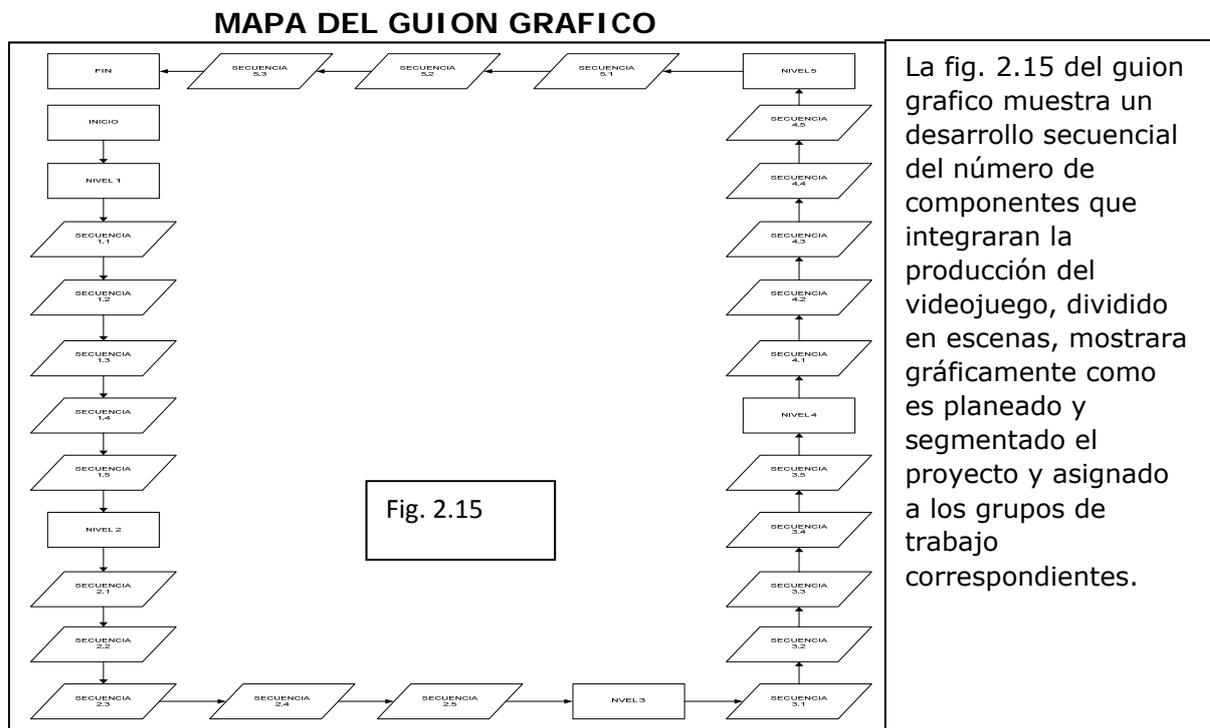


Se ha establecido la duración del proyecto en 12 semanas, con 16 horas hombre por persona y laborando 7 días por semana. Se ha dividido en tres etapas, 2 semanas para preproducción, 10 semanas para producción, y dos semanas para postproducción. Los grupos de trabajo de administración,

mercadotecnia, coordinación, RRHH (recursos humanos), organización, control, finanzas y arte laboraran las 12 semanas. Planeación solo laborara 2 semanas. Desarrollo, programación y aseguramiento de calidad trabajaran a partir de la tercera semana.

2.1.7 GUION GRAFICO

El guion grafico es el punto de partida para todo el proyecto. Se toma como el parte aguas del diseño de producción, es donde todos los recursos tanto humanos como técnicos tendrán su plena participación en la obtención de resultados satisfactorios. A partir de aquí la producción ya está en víspera de iniciar actividades una vez que el diseño de producción esté listo. Usando un ejemplo de un guion gráfico, se mapea y se distribuye la historia en 5 niveles de 5 secuencias cada uno, en el nivel 5 son 3 niveles. Se construye un mapa de historia el cual relatara la trama en dentro de sus límites para no perder la fuerza de la idea y no divagar. Se plantean los elementos psicológicos necesarios en cada nivel tales como luces y sonido. Le plantean los elementos estáticos y dinámicos conforme a la historia, se interpolan los elementos dinámicos de su estado inicial y su estado final, esto le dará vida al gameplay. Se tiene que señalar claramente en el guion grafico en el momento de su programación. La interpolación del guion grafico en su punto inicial y final se conocerá más adelante como el gameplay. Se hacen las correcciones adecuadas.



La fig. 2.15 del guion grafico muestra un desarrollo secuencial del número de componentes que integraran la producción del videojuego, dividido en escenas,

mostrara gráficamente como es planeado y segmentado el proyecto y asignado a los grupos de trabajo correspondientes

2.1.8 DISEÑO DE SECUENCIAS

Es la asignación de tareas específicas hacia cada grupo de trabajo involucrado en la producción general del proyecto, todo es en base al guion gráfico. Cada secuencia tiene un conjunto de elementos que son video y sonido, con sus respectivas relaciones, así como puntos de interacción, animaciones, (dependiendo del tiempo de contexto), la idea es tener un guion grafico técnico para dividirlo en función a capas de aplicación.

DIAGRAMA MAESTRO DE NIVEL

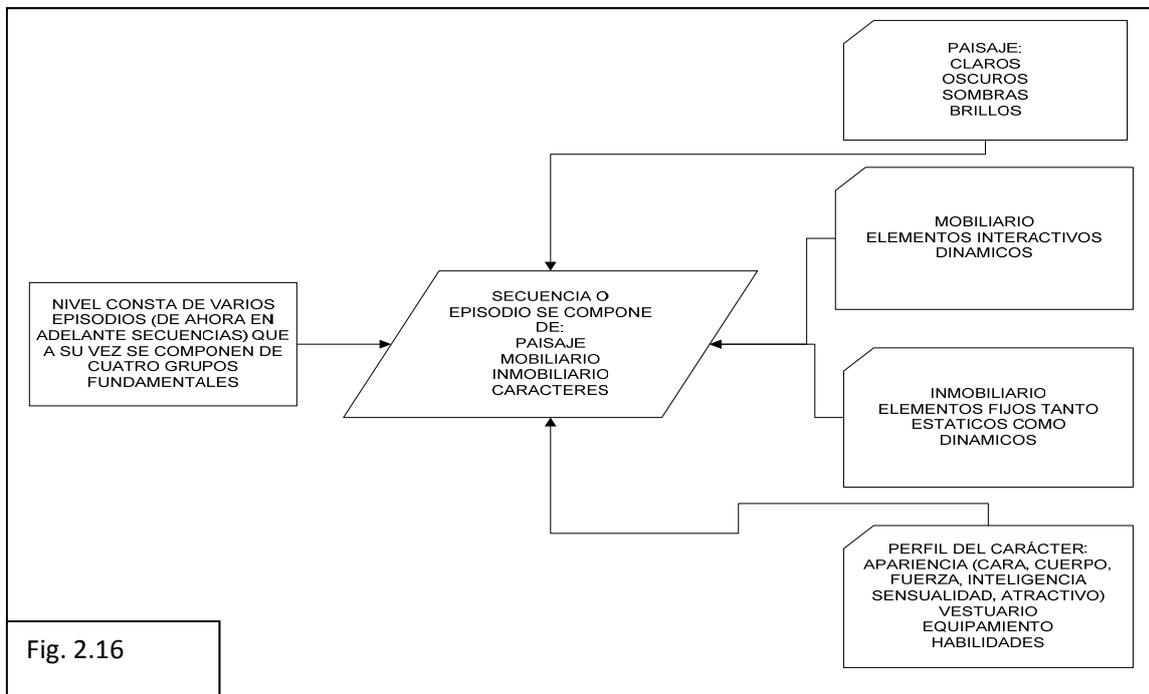


Fig. 2.16

La fig. 2.16 muestra que utilizar las capas para darle sencillez y manejo a cada estructura de aplicación y así corregir fácilmente cualquier incoherencia. Partiendo de esta filosofía es como el trabajo se divide y se asigna a su área correspondiente donde entregaran productos satisfactorios que se irán ensamblando en la construcción maestra del videojuego.

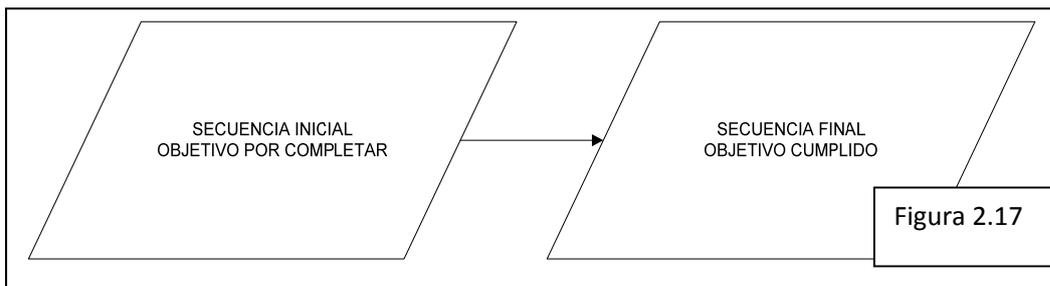
Estas capas son;

Elementos tangibles:

Fondos
Elementos activos
Elementos pasivos
Sonidos dinámicos
Sonidos de fondo
Carácter
Consola de control

Elementos intangibles:

Bases de datos dinámicas
Registros
Eventos
Objetos



La figura 2.17 muestra que partiendo de este punto todas las actividades contempladas en cada secuencia tienen un inicio y un fin. El rombo izquierdo muestra un estado inicial. La flecha muestra el transcurso del tiempo, y el rombo derecho muestra el estado final.

2.1.9 DIAGRAMA DE ACCIÓN GENERAL

Es un guion grafico dinámico, donde se contabilizan todas las acciones de los elementos de cada nivel así como escena del videojuego.

Juego consta de un determinado número de:

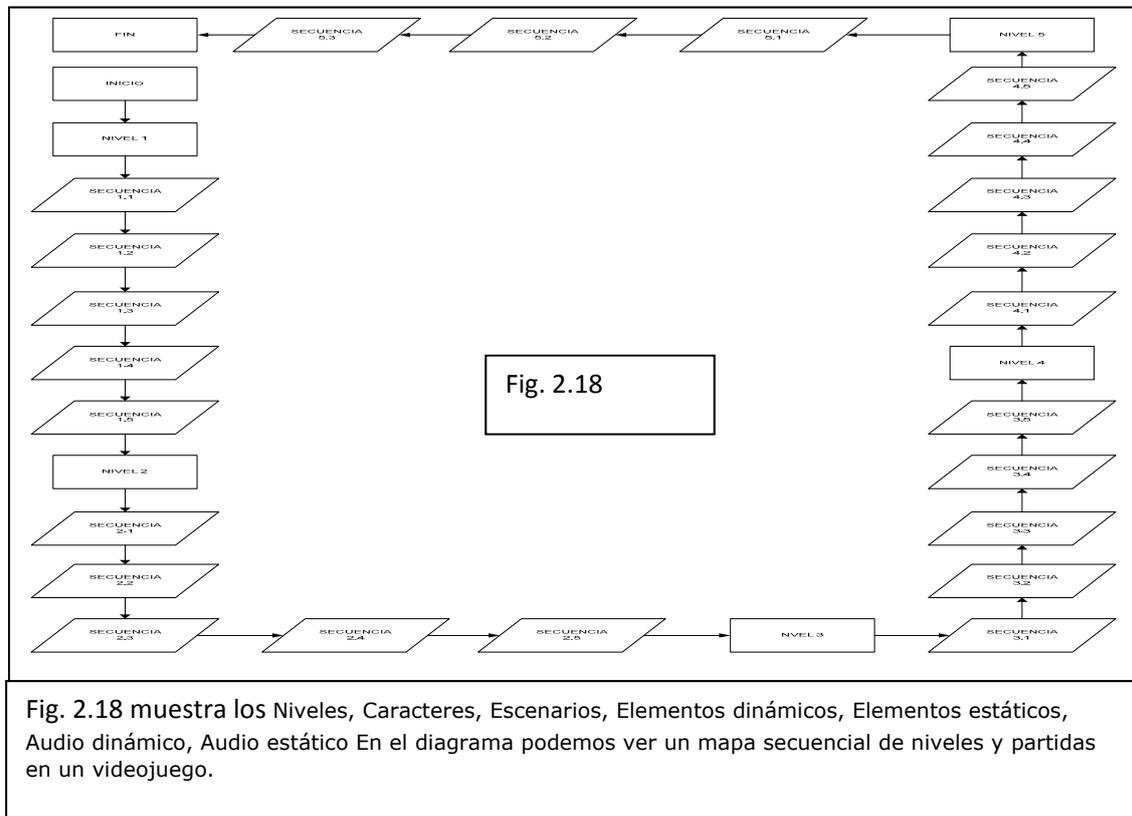


Fig. 2.18 muestra los Niveles, Caracteres, Escenarios, Elementos dinámicos, Elementos estáticos, Audio dinámico, Audio estático En el diagrama podemos ver un mapa secuencial de niveles y partidas en un videojuego.

Tomando de referencia el guion gráfico, es detallar dinámicamente cada escena del videojuego.

Partiendo del proyecto y continuamente solicitando información a ARTE, el diseño de partida es una hoja de especificaciones que debe de ser cubierta por producción.

Involucra los siguientes niveles de realización.

Consiste en la relación entre todos los elementos dinámicos y estáticos partiendo de un punto inicial a un punto final.

Cubriendo todas las relaciones operativas que el videojuego contempla.

El diseño de partida está estructurado por los siguientes niveles de ejecución y representados por su funcionalidad durante la reproducción del escena.

- Nivel A** Animación
- Nivel B** Integración
- Nivel C** Reproducción
- Nivel D** aseguramiento de calidad

2.2 PRODUCCIÓN

Producción es la ejecución del proyecto. Basado en una planeación, cada actividad se ejecutara conforme a lo estimado en tiempo y forma. El desarrollo esta monitoreado, enviando resultados en función al tiempo y todos los grupos de trabajo están interconectados por lo que constantemente están enviando datos y resultados para dar continuación a las actividades siguientes.

Durante esta etapa se subcontratan algunas compañías especializadas en su respectiva área. Se muestran los organigramas con sus respectivos, métodos, procesos, métricas, responsables, capital humano, que trabajaran para lograr el buen término del proyecto. Y como en todo se busca la optimización de los recursos. Siendo la planeación la piedra angular del proyecto a nivel producción se aplica la correcta interpretación del diseño de secuencias que viene dada por el guion gráfico. La segmentación del videojuego proporciona la facilidad con que el videojuego se creara. Muchos elementos en el gameplay del videojuego son estructuras cíclicas que solamente se van repitiendo conforme se desarrolla la trama, normalmente lo que va cambiando es el escenario. Y pues la producción inicia conociendo al todo los grupos de trabajo y el proyecto dado por planeación.

2.2.1 ORGANIZACIÓN DEL VIDEOJUEGO

Teniendo la planeación y la distribución del trabajo que el producto requerirá para su realización se procede a producirlo. Para tal efecto se han hecho mapas de trabajo y se siguen filosofías de trabajo.

La planeación se segmenta de la siguiente manera

Esquema de distribución de los elementos (LAYOUT)

Es acomodar todos los elementos del videojuego para capturar el mejor concepto visual posible.

Diagrama de acción general

Es un guion grafico dinámico, donde se estandarizan todas las acciones de los elementos de cada nivel así como escena del videojuego.

Diseño de acción

Tomando de referencia el guion gráfico, es detallar dinámicamente cada escena del videojuego.

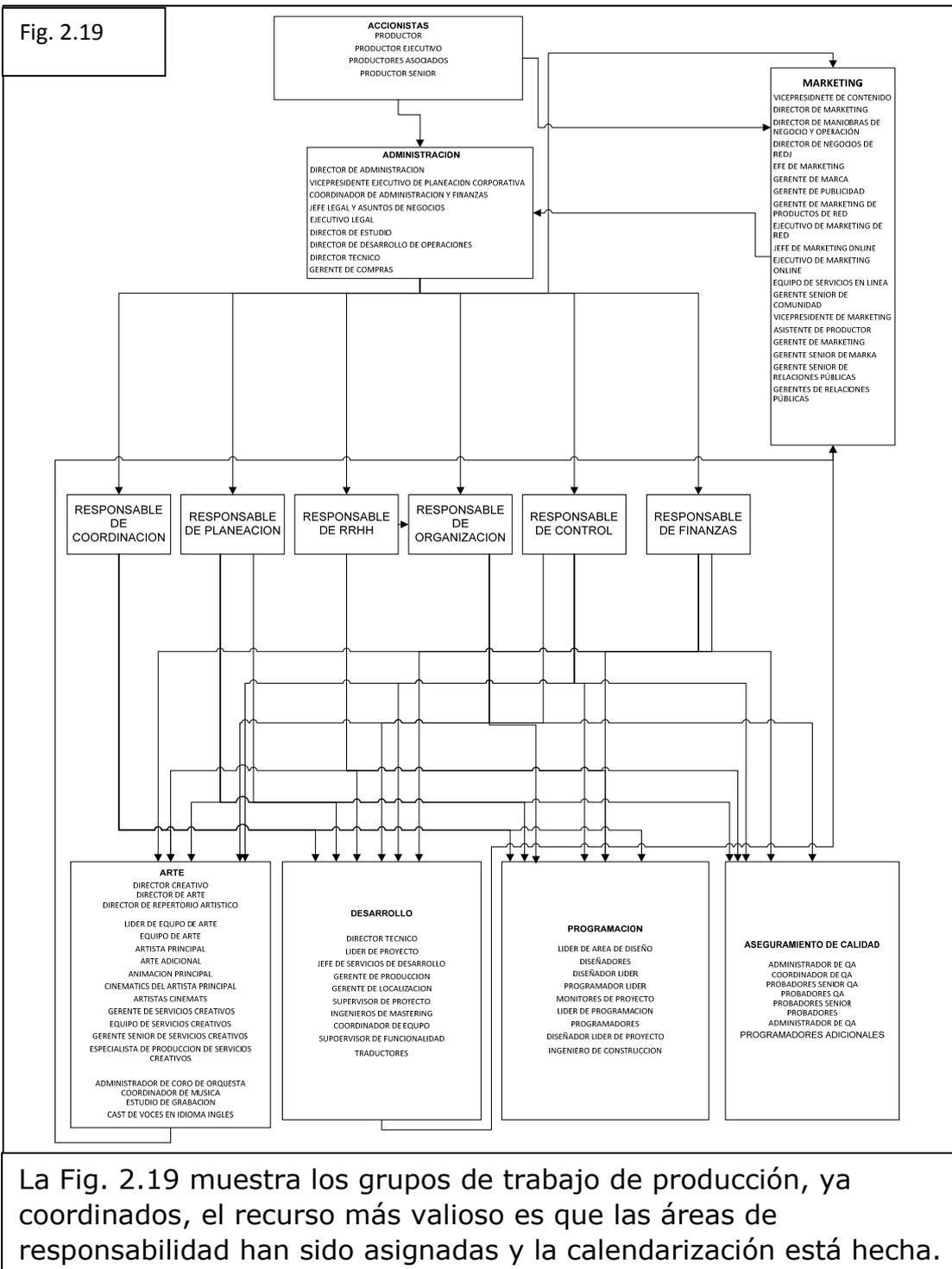
Diseño de partida

Consiste en la relación entre el menú y el diseño de acción.

Nivel A	Caracteres
	Fondos
	Objetos fijos
	Objetos interactivos
	Audio
	Menús
Nivel B	Animación
Nivel C	Integración
Nivel D	Reproducción

Estos cinco esquemas son los que producción realizara en base al proyecto dado por planeación.

MAPA DE GRUPOS DE TRABAJO DE PRODUCCIÓN



2.2.2 MAPA MAESTRO DEL VIDEOJUEGO

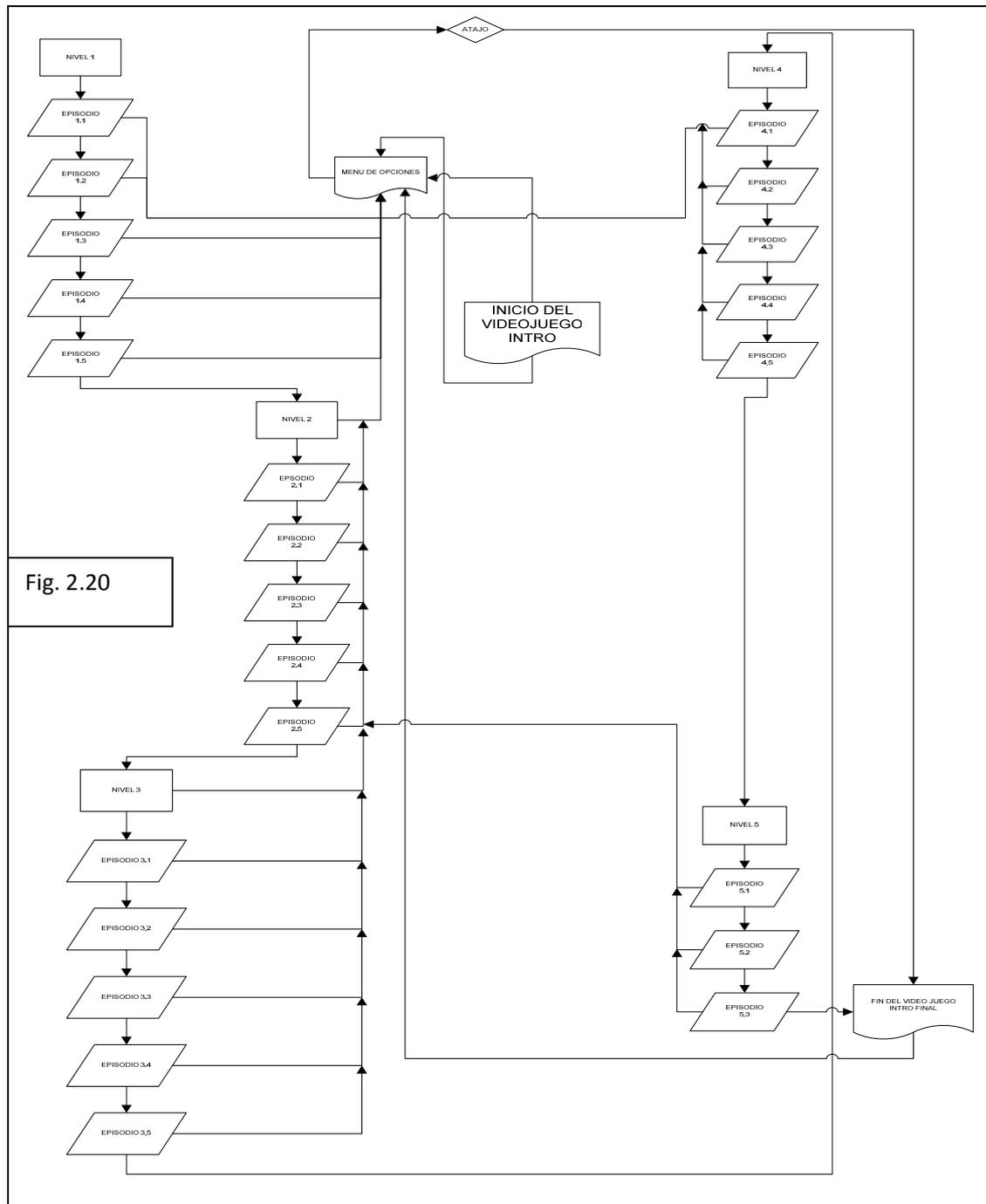
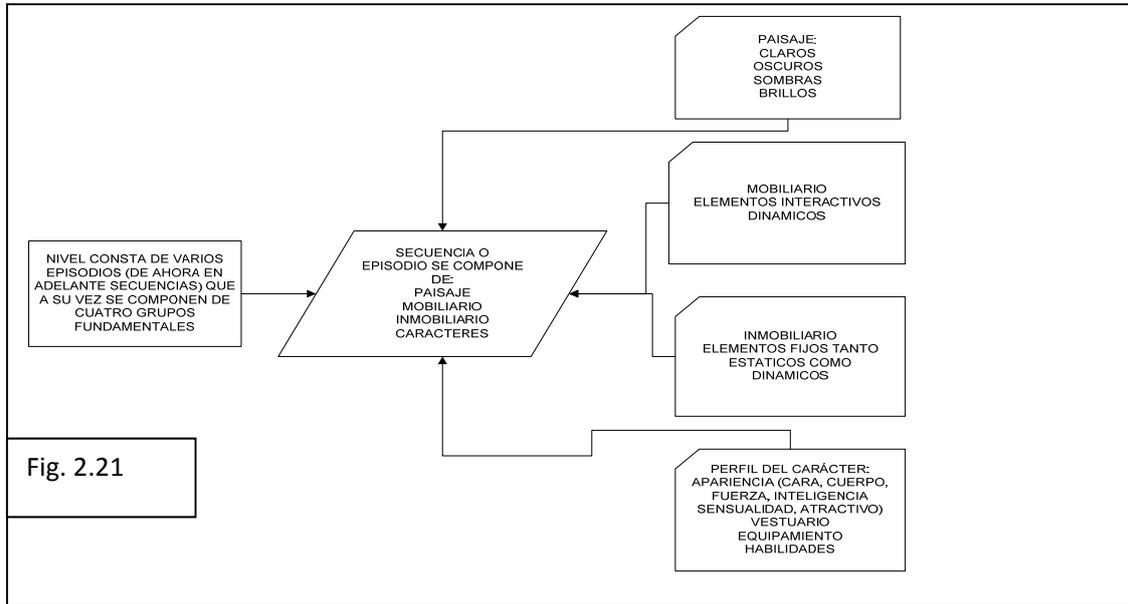


Fig. 2.20

Fig. 2.20 es el punto de referencia para el grupo de trabajo de programación, partiendo de aquí es como el juego estará conformado.

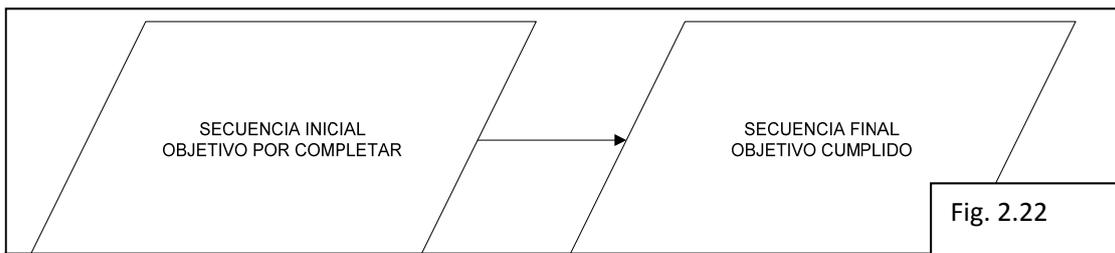
2.2.3 MAPA DE NIVEL MAESTRO



La fig. 2.21 ilustra cómo se está ensamblando el videojuego de forma tangible, previamente se había visto que iniciaba como una planeación, ahora es una tarea medible en eficiencia con respecto al tiempo.

En estos mapas trabajaran arte, desarrollo y programación para ir armando el videojuego por niveles.

2.2.4 MAPA DINÁMICO DE SECUENCIA



La fig. 2.22 ilustra la elaboración de escenas y sus respectivos resultados en función al tiempo.

Este diagrama muestra como arte, desarrollo y programación crearan las secuencias que ensamblaran al producto. Consiste en que la secuencia contiene elementos en un estado inicial, el mapa dinámico consiste en señalar que harán esos elementos contenidos en ese estado inicial dentro de un periodo de tiempo determinado.

2.2.5 SEGMENTACIÓN DEL VIDEO JUEGO

Para una optimización lo más eficiente posible del video juego, este se segmenta con una filosofía de capas de aplicación.

Empieza con la segmentación del proyecto por:

Numero de caracteres

Vestuarios
Habilidades
Inteligencia
Semblante
Genero
Interacciones

Numero de niveles

Numero de escenarios
Numero de objetos estáticos
Numero de objetos dinámicos
Numero de sombras
Numero de focos
Numero de luces
Numero de colores
Numero de texturas
Numero de interacciones
Numero de audio

Estático
Dinámico

Numero de menús

Número de opciones de sistema
Sonido
Gráficos
Texturas
Guardar

Con esta metodología se obtiene un número determinado de eventos de fácil desarrollo para la etapa creadora.

Esta segmentación es necesaria para el control de las actividades y sean monitoreadas. Para que entreguen en tiempo y forma los resultados esperados. Muchos de los procesos involucrados en el video juego son cíclicos Por lo que cada evento puede repetirse y lo que va cambiando es el escenario donde se desarrolla la acción.

2.2.6 CARACTERES

Los caracteres se diseñan también por capas de aplicación, como si se vistiera un muñeco. Se comienza por el diseño solicitado por planeación y se revisan las concepciones por parte de arte. Bajo esos esquemas es como se procede a darle forma a los caracteres.

Se considera en su fabricación:

Físicas

Estatura
Cabello
Tez
Nariz
Ojos
Raza
Semblante
Fuerza

Psicológicas

Habilidad
Inteligencia

Vestuario y atavíos

Tipo de vestimenta

Menú de vestuarios

Armamento

Menú de armamento

Con un patrón de opciones se diseña el carácter de acuerdo al contexto del proyecto

Este patrón de opciones para la adaptación visual y dinámica del carácter es un menú interactivo que el jugador puede adaptar a sus necesidades.

Entonces se obtiene un número finito de opciones que se desarrollan y se programan para que estén disponibles en el menú principal.

2.2.7 ESCENARIOS

Los escenarios se diseñan también por capas de aplicación, como si se pintara un cuadro. Se comienza por el diseño solicitado por planeación y se revisan las concepciones por parte de ARTE.

Bajo esos esquemas es como se procede a darle forma a los escenarios. Se considera en su fabricación:

Físicas

- Lugar**
- Horario**
- Humedad**
- Temperatura**
- Interior**
- Exterior**
- Temporada**
- Vegetación**
- Contaminación**
- Populoso**
- Entre otros**

Dinámicas

- Lluvias**
- Vientos**
- Luz**
- Sombra**
- Transcurso del tiempo**
- Iluminación**
- Oscurecimiento**
- Plantas**
- Polvo**
- Piedras**
- Caracteres extras.**

Estáticas

- Piedras**
- Construcciones**
- Arboles**
- Autos,**
- Entre otros**

El diseño de los escenarios se desarrolla por capas de aplicación y según lo que pidan los requerimientos de planeación es como se proyectan los efectos de perspectiva para dar efectos del transcurso de los eventos.

Con un patrón de opciones se diseñan los escenarios de acuerdo al proyecto. Este patrón de opciones para la adaptación visual y dinámica del escenario

Entonces se obtiene un número finito de opciones que se desarrollan y se programan para que estén presentes en el gameplay.

2.2.8 LUZ Y SOMBRAS

Las luces y sombras se diseñan también por capas de aplicación. Se comienza por el diseño solicitado por planeación y se revisan las concepciones por parte de ARTE. Bajo esos esquemas es como se procede a darle forma a las luces y sombras involucradas en los escenarios así como todos los elementos que las requieren.

Se considera en su fabricación:

Formas	Objetos dinámicos Objetos estáticos
Focos	Directos Indirectos

Con un número finito de eventos que requieren luces y sombras se procede a ajustar todos los focos para que automáticamente las sombras se formen. Muchas veces estas sombras se pueden eliminar para mejorar el rendimiento del videojuego en función al hardware disponible.

2.2.9 OBJETOS DINÁMICOS Y OBJETOS ESTÁTICOS

Los objetos dinámicos se diseñan también por capas de aplicación además de que durante el gameplay se puede interactuar con ellos. Se comienza por el diseño solicitado por planeación y se revisan las concepciones por parte de ARTE. Bajo esos esquemas es como se procede a darle forma a los objetos dinámicos

Se considera en su fabricación:

Interactivos	Armas Municiones Gadgets Comunicaciones Recargas Estimulantes Recuperadores Objetos para arrojar Secretos
Interactivos pasivos	Hojas Basura Banderas Entre otros

Estos objetos dinámicos su función es formar parte de la acción
Con un menú interactivo de opciones es como se puede tener manejo de estos objetos para formar parte del gameplay del videojuego.
Entonces se obtiene un numero finito de opciones que se desarrollan y se programan para que estén disponibles en el menú principal.

2.2.10 OBJETOS ESTÁTICOS

Los objetos estáticos se diseñan también por capas de aplicación e interactúan de forma indirecta con el jugador durante el gameplay. Se comienza por el diseño solicitado por planeación y se revisan las concepciones por parte de ARTE. Bajo esos esquemas es como se procede a darle forma a los objetos estáticos.

Se considera en su creación el número de elementos:

Estáticos activos

- Casas**
- Piedras**
- Paredes**
- Agujeros**
- Mobiliario**
- Arboles**
- Automóviles**
- Tinacos**
- Entre otros**

Estáticos pasivos

- Montañas**
- Colinas**
- Ríos**
- Calles**
- Entradas**
- Pavimentos**
- Entre otros**

Estos objetos estáticos su función es formar parte de la acción
Con un menú interactivo de opciones es como se puede tener manejo de estos objetos para formar parte del gameplay del videojuego estas funciones son como las texturas y detalles del escenario.
Con un número finito de elementos estáticos es como se van colocando por capas en el escenario.

2.2.11 AUDIO

El audio se divide en dos grupos, el audio dinámico y el audio estático.
Partiendo de referencia del guion gráfico y de proyecto proporcionado por Planeación. En el juego Se crea una librería finita de efectos de sonido para cada evento condicionado que lo requiera. Con la biblioteca de sonidos se integran a

cada elemento según la condición se presente durante el gameplay. El audio es muy importante y normalmente se estandariza para emplear un número finito de efectos acústicos para igual número de eventos que requieren apoyo acústico. El audio dinámico está en función a los eventos que se van desarrollando durante el gameplay.

Audio dinámico

- Disparos**
- Explosiones**
- Diálogos**
- Activación de algún recurso**
- Pisadas de 1 persona**
- Ataques de otros elementos**

El audio estático es el sonido de fondo que le da atmosfera al gameplay durante su reproducción.

Audio estático

- Sonido de fondo**
 - Viento**
 - Truenos**
 - Relámpagos**
 - Cantar de pájaros**
 - Aleteos de plumas**
- Pisadas**
- Explosiones lejanas**
- Trafico**
- Entre otros.**

El audio es la parte fundamental de toda producción audio visual interactiva como es el videojuego.

Un buen audio es vida, es atmosfera, es sentirse parte de la acción. Su aplicación es por eventos.

2.2.12 CINEMÁTICAS

Las cinemáticas son elementos de refuerzo visual para complementar al videojuego en belleza estructural. Una cinemática es un elemento de animación estática donde el jugador no participa de ningún modo, solo ve como es reproducida la película y ya. Es solamente un espectador de lo que sucede en la acción del cinemática. Considerados como elementos finitos dentro del videojuego, el cinemática forma parte del concepto visual del videojuego.

Algunas veces el proyecto involucra ciertas escenas que requieren participaciones con características humanas, para darle vida al videojuego y que no se vea tan "artificial". Estas escenas se les llaman CINEMÁTICAS, la idea es dar refuerzo al concepto. Para solucionar este detalle se subcontrata una compañía actoral que represente la idea visual que se desea agregar. Todos ellos con su propia producción. Únicamente requieren saber que quiere la producción principal y ellos entregan su producto terminado.

Una cinemática es una representación digital de la realidad, es como decir una pequeña película. De la planeación del guion grafico se determina el número de cinemáticas requerido.

2.2.13 ENSAMBLE DEL VIDEO JUEGO

Construcción maestra es la integración de los 4 niveles de ejecución proporcionados por producción. Aquí se ensambla el videojuego por jerarquías de aplicación.

Siendo:

**Animación
Integración
Reproducción
Aseguramiento de calidad.**

Se maneja esta filosofía de capas de aplicación por qué un videojuego consta de muchos elementos estáticos y dinámicos. Cuando se trata de resolver problemas es fácil encontrarlos en función a la capa de aplicación donde se originó el problema. Los problemas son cosas comunes que requieren ser tratados de forma objetiva. Cuando una falla se presenta en determinada circunstancia del videojuego inmediatamente se procede a su análisis y consta de esta forma:

**Nivel
Secuencia
Partida
Lugar en el espacio del gameplay
Interacción con elementos estáticos
Interacción con elementos dinámicos
Efectos de audio involucrados**

Esta metodología funciona y permite revisar fácilmente todas las rutinas y eliminar líneas de código redundantes. El ensamble del videojuego es una prueba de aseguramiento de calidad donde se verifican los objetivos de planeación.

2.3 POSTPRODUCCIÓN

Esta parte del proceso creativo de un proyecto, en este caso el videojuego, forma una de las piedras angulares de la estructura de negocios.

La razón es que un software de calidad proporciona prestigio para la marca y confianza para el usuario. No importa que sean videojuegos o aplicaciones de cualquier naturaleza, el común denominador es la mínima presentación de fallas. Cada producto es susceptible de tener fallas sin embargo con una filosofía de calidad se logran resultados sorprendentes por que la calidad no cuesta, la calidad es un modo de ver la vida, de hacer bien las cosas desde el principio. En términos generales la calidad en los procesos, en el material, en el capital humano es un valor que no cuesta dinero, sino es una actitud.

Para efectos en el proceso creativo de videojuegos la calidad comienza desde la etapa de producción. En las múltiples pruebas de procesos y generación de

resultados se van descubriendo los errores de aplicación que el software va experimentando durante su reproducción. Las metodologías de software de calidad son las mismas para cualquier proceso. Para efectos de este proyecto, la aplicación de métodos estadísticos resultara efectiva porque hablamos de billones de procesos iterativos que ocurren por segundo durante la ejecución del gameplay. Para controlar estos resultados obtenidos se manejan modelos probabilísticos y estadísticos de control de calidad y una adecuada interpretación de los datos que arrojan.

En el esquema informático continuamente se monitorean los resultados obtenidos por cada grupo de trabajo. La generación de estos valores y comparados con los modelos ideales de ejecución proporcionan los criterios para decidir si el proceso está controlado o fuera de control. El muestreo de iteraciones correctas y resultados positivos se contrasta con resultados negativos. Como se obtiene esto, la generación de estos resultados proviene de las terminales donde se están desarrollando y programando los objetos, eventos, módulos, capas, servicios, estructuras. Y planteando una metodología de trabajo que permita fácilmente encontrar errores en las capas de aplicación se estandariza los procesos para alejarnos de la programación artística y tener programación bajo un mismo criterio de aplicación.

2.3.1 ASEGURAMIENTO DE CALIDAD

Independientemente si la empresa está certificada o no, la calidad es una filosofía que genera buenos resultados porque no cuesta. Aplicarla cuesta trabajo más no dinero. Para efectos de certificación primero se estandarizan los procesos más comunes. Para el caso del software la calidad va muy de la mano con finanzas por qué se ve reflejado en los ahorros a la empresa así como la eliminación de redundancias innecesarias en los ciclos de trabajo de todo el corporativo. Para aplicar un modelo de gestión de calidad es fácil, siempre que se tengan presentes el conocimiento adecuado de los procesos. Gestionar calidad significa aplicar métodos estadísticos de procesos con valores máximos y mínimos donde el proceso estará monitoreado y controlado. Estas holguras presentan las tolerancias permitidas para el funcionamiento correcto del sistema. Para establecer el sistema de gestión de calidad se plantean los siguientes estándares.

Estrategias:

Definir políticas, objetivos y lineamientos para el logro de la calidad y satisfacción del cliente. Estas políticas y objetivos deben de estar alineados a los resultados que la organización desee obtener.

Procesos:

Se deben de determinar, analizar e implementar los procesos, actividades y procedimientos requeridos para la realización del producto o servicio, y a su vez, que se encuentren alineados al logro de los objetivos planteados. También se deben definir las

actividades de seguimiento y control para la operación eficaz de los procesos.

Recursos:

Definir asignaciones claras del personal, Equipo y/o maquinarias necesarias para la producción o prestación del servicio, el ambiente de trabajo y el recurso financiero necesario para apoyar las actividades de la calidad.

Estructura Organizacional:

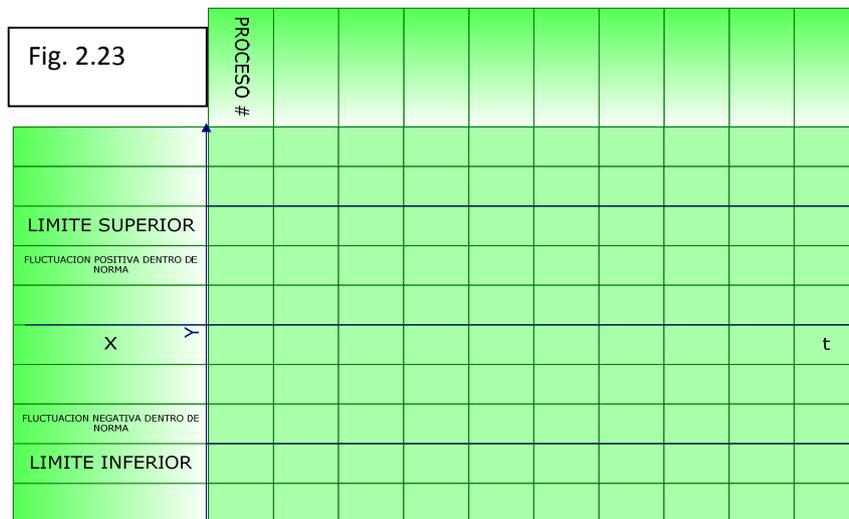
Definir y establecer una estructura de responsabilidades, autoridades y de flujo de la comunicación dentro de la organización. Definir y establecer una estructura de responsabilidades, autoridades y de flujo de la comunicación dentro de la organización.

Documentos:

Establecer los procedimientos documentos, formularios, registros y cualquier otra documentación para la operación eficaz y eficiente de los procesos y por ende de la organización

También existen varias normativas estandarizadas que establecen requisitos para la implementación de un Sistema de Gestión de la Calidad, y que son emitidas por organismos normalizadores como la ISO, DIS, entre otros.

GRAFICO BASICO DE CONTROL DE PROCESOS CON LIMITES EN FUNCION AL TIEMPO



Estas métricas de control de procesos están relacionadas directamente con el proceso mismo que se está controlando.

Para explicar esto es de la siguiente manera:

- Todos los procesos que el proyecto involucra se contabilizan.**
- Estos procesos se agrupan en áreas funcionales.**
- Se eliminan todas las redundancias.**

El proyecto mismo tiene en su planeación los objetivos claros y sus alcances. La estandarización de los procesos se realiza conociendo bien el proceso en su resultado obtenido que sea satisfactorio y en la optimización de la ejecución del mismo. Se ha logrado una estandarización de todos los procesos del proyecto se generan los modelos de resultados satisfactorios en función al tiempo. Estos modelos de cómputo envían datos a un servidor principal que maneja estadísticamente toda la información y genera gráficos y proyecta los focos rojos donde se está generando ciertas eventualidades producidas por procesos fuera de estándar o fuera de control. La fig. 2.23 muestra la gráfica de aseguramiento de calidad de producto en función a los procesos.

El pleno conocimiento de metodologías de flujo de datos que generen información confiable se ve reflejada en la eficiencia del proyecto en general.

Para visualizar esta optimización en términos financieros se refleja en forma de ahorros para el proyecto tanto en tiempo como en dinero. Hay muchos software que sirven para tal propósito de administrar la calidad de un proyecto. La facilidad de implantar un control de procesos y su manejo apropiado de la información obtenida genera buenos criterios para la toma de decisiones.

2.3.2 ASEGURAMIENTO DE CALIDAD DE LOS BETA TESTERS

Ellos se encargan de probar el software bajo las condiciones más extremas de aplicación con la finalidad de encontrar errores. Los probadores BETA son personas distribuidas alrededor del mundo, son un grupo de trabajo coordinado y administrado por CALIDAD y su función es PROBAR el producto bajo condiciones extremas de aplicación. Lo pueden hacer en todo tipo de situaciones posibles. Su función es generar errores de ejecución del producto para que sean registradas y enviadas al servidor de calidad.

De ahí se envían a los responsables de la capa donde se generó el problema. Estas condiciones de operación extremas son para software.

Windows® en todas sus versiones

(Se puede hablar de Linux, Android, macos, pero su aceptación en videojuegos es limitada y sujeta a condiciones de operación básicas por los emuladores, la única opinión es que la mayoría de los videojuegos están diseñados para ambiente Windows y consolas).

Las condiciones son ilimitadas bajo parámetros normales de operación.

Estas condiciones son por:

Versiones de Windows®
Aplicaciones en operación
Tareas en ejecución
Periféricos

Estas serían las condiciones de reproducción más comunes y extremas para un software. Muchas de estas condiciones experimentarían cambios debido al hardware disponible además de características propias del hardware. Para cada

esquema de pruebas BETA, se diseña un modelo estadístico de comportamiento con límites inferiores y límites superiores de aceptación.

Si estos límites son rebasados forman un expediente de datos que servirán para diseñar el manual de operación del producto.

Sin olvidar los sistemas operativos alternos como:

MACOS® ,
GNU/LINUX ®
UNIX®
SOLARIS®
GOOGLE CHROME OS®
DEBIAN®
UBUNTU®
MANDRIVA®
SABAYON®
FEDORA®
REACTOS®

PARA DISPOSITIVOS MÓVILES

SYMBIAN OS®
ANDROID®
IOS®
WINDOWS PHONE®
BLACKBERRY OS®
HP webOS®
BADA®

Las pruebas de aseguramiento de calidad para consolas y dispositivos móviles son las mismas, porque son criterios estadísticos de repetitividad de eventos bajo determinadas condiciones que cubren el modelo estadístico. Estos criterios se aplican para las consolas disponibles en el mercado como son NINTENDO®, PLAYSTATION®, XBOX 360®.

2.3.3 ASEGURAMIENTO DE CALIDAD DE COMPATIBILIDAD DE HARDWARE

Este criterio resuelve a través de condiciones severas de uso la compatibilidad que existe entre el videojuego y las plataformas existentes como son PC, Mac, ps, xbox, nintendo, entre las más comunes, la idea es la misma buscar los errores que solo durante la marcha se pueden encontrar, señalarlos y que el área correspondiente los resuelva. De igual forma para el software, se plantea un modelo estadístico de condiciones y repetitividad. Este modelo se diseña para cada condición en particular de hardware, esto es porque existen infinitas de modelos y configuraciones de hardware. Para computadoras personales, consolas y dispositivos móviles. El diseño del modelo de aplicación estadístico comenzaría por configuraciones básicas de hardware como son:

HARDWARE:

PROCESADORES
MEMORIA RAM
BIOS
MOBOS
PROCESADORES DE GRÁFICOS
PROCESADORES DE SONIDO
DISPLAYS
JOYSTICKS
GAMEPADS

Cabe señalar que para los dispositivos móviles como teléfonos y tablets los videojuegos están en versiones más ligeras, que no comprometen el desempeño del hardware.

2.3.4 CONTROL ESTADÍSTICO DEL PRODUCTO EN LÍNEA

Una vez puesta en marcha la venta del producto es conveniente el seguimiento del producto, esta información obtenida cada vez que el usuario hace uso del él en cada conexión a internet genera datos valiosos para el diseño de nuevos productos y crear nuevas necesidades al usuario. En términos especulativos, todas las conexiones a internet generan información que tratada de forma especial generara las oportunidades de negocio de los próximos años. Aunque el usuario diga mentiras en el registro de su perfil, sus hábitos lo delatan

Los criterios estadísticos así como su frecuencia:

Horarios,
Días,
Días del mes,
Días del año,
Duración de la sesión
Tipo de dispositivo donde la aplicación está instalada
Errores generados
Perfil del jugador
Número de participantes en línea
Edades
Chat de voz

Por mencionar algunos parámetros válidos, para modelar un sistema estadístico de seguimiento del producto en línea parte de la misma aplicación. Cuando esta se reproduce en la consola o la PC, está ya está conectada a internet y está enviando información sobre la partida así como los elementos externos involucrados en el runtime.

2.3.5 SUPERVISIÓN DE CONTENIDO

Consiste en supervisar todo el producto terminado. Estas supervisiones consisten en observar el comportamiento de gameplay en sus diferentes capas de aplicación. Es como una revisión de calidad pero de contenido visual.

Aquí se revisan los errores visuales producidos por algún fallo en la comunicación con ARTE, DESARROLLO, PROGRAMACIÓN.

Estas revisiones normalmente no arrojan resultados negativos pero de haberlo, se turna la observación al grupo correspondiente.

Esta supervisión visual del gameplay es un cotejo entre el guion escrito del proyecto y el producto final.

Todos los contenidos del videojuego así como sus características extras.

Hablando de censura y la clasificación correcta de un videojuego se estandariza a través de este criterio.

G – “Público en general”:

Aplica cuando un videojuego no contiene desnudos, contenido sexual, uso de drogas o lenguaje fuerte. La violencia es mínima y el tema del videojuego es bastante apropiado para niños. Según la MPAA, el calificativo “G” no indica necesariamente que sea un videojuego para niños.

PG – “Control de Padres Sugerido. Algún contenido puede no ser apropiado:

Esta calificación de aplica cuando los miembros creen que el videojuego contiene temática o contenido que los padres pueden encontrar inapropiado para niños. El videojuego puede contener algún tipo de palabras soeces, violencia o breves desnudos pero solo en pequeña intensidad. Un videojuego calificado como “PG” no puede incluir el uso de drogas.

PG-13 – “Advertencia severa a los padres. Algún contenido puede ser inapropiado para niños menores de 13 años”:

La MPAA añadió esta calificación en 1984 para denotar un videojuego en el cual la violencia, uso de lenguaje o contenido sexual es lo suficientemente intenso como para que algunos padres no quisieran exponer a sus hijos tal producto, pero no tan intenso para una calificación “R”. Cualquier videojuego que incluya el uso de drogas recibirá al menos el calificativo de “PG-13”. Una cinta de este tipo puede incluir el uso simple de lo que el comité llama una “fuerte palabra de contenido sexual” mientras sea usada como exclamación, más no en un contexto meramente sexual.

R - “Restringido. Personas menores de 17 años requieren la compañía del padre o un adulto”:

El Comité aplica esta calificación a videojuegos en las cuales se consideran que incluyen un alto nivel de contenido para adultos, como uso excesivo de palabras soeces, violencia intensa, contenido explícitamente sexual y uso extensivo de drogas. En algunos estados, la edad mínima para ver una película de tipo "R" sin acompañamiento es 18.

NC-17:

"Ninguna persona menor de 17 debe ser admitida": Originalmente llamada "X", esta calificación es aplicada a filmes en los cuales el Comité considera que el contenido adulto es más intenso que en una película "R", pero no implica algún tipo de obscenidad. Como en los videojuegos calificados como "R", la edad mínima para ver un videojuego "NC-17" es 18 en algunos estados.

Si por alguna causa el videojuego llega a tener contenidos visuales y acústicos extremos puede ser sujeto a edición o incluso ser prohibido.

Una edición es el conjunto de opiniones generadas por un grupo de expertos, ellos proporcionaran su punto de vista para la mejora del producto.

Durante las etapas de revisión, siempre estará un supervisor en jefe de todos los departamentos, el basando en métodos estadísticos dará el visto bueno al cierre del producto y pasarlo a fabricación y distribución.

En base a las siguientes observaciones.

Es un producto diseñado para público de todas las edades con todos sus sentidos totalmente funcionales.

Estos productos también están destinados a públicos con ciertas características pero para simplificarlas únicamente utilizaremos la edad como criterio de ventas.

Para mayor información acerca de la ESRB consulte el siguiente sitio

<http://www.esrb.org/index-js.jsp>

Estos son los logos empleados para certificar el contenido de un videojuego y hacia cual público está destinado.

Este criterio es muy útil para deslindar responsabilidades y evitar demandas del público hacia la compañía emisora del videojuego.



2.3.6 RECURSOS ADICIONALES

Durante el desarrollo del producto, se presentan muchos detalles no contemplados en el diseño original del concepto, algunos pueden ser extremadamente visibles que nadie los vio y otros pueden estar totalmente obvios. Y como el producto ya está terminado, lo que se hace es modificar parte de los alcances esperados del producto para no generar garantías incumplidas. Esta ingeniería adicional se contempla dentro del proyecto como una holgura calculada, la idea es que el producto pueda ser corregido sin que cueste tanto. Estos resultados se obtienen de las estadísticas de aseguramiento de la calidad. Así pues el esquema de trabajo se focaliza en donde están esos errores que ya se tienen localizados. Como resultado de la opinión de los probadores beta, artistas, concedores, profesionales del marketing, ellos pueden tener peso al opinar acerca del producto como tal, algunas veces hay tópicos que llegan verse fuera de contexto lo cual altera la percepción de la idea original.

Es normal que todos los participantes en la creación del producto estén tan ocupados en sus respectivas responsabilidades que esos errores de percepción pasan desapercibidos, por lo que, gente fuera del medio es capaz de notarlos. Las correcciones de arte se dan por razones muy diversas. Igual que en el arte adicional, se requiere de un grupo de profesionales ajenos a la producción para opinar sobre el producto terminado, escuchar con detenimiento cada parte del gameplay proporcionara las opiniones generales sobre la decisión de alguna corrección en la voz, que normalmente seria cambiar voces.

Algunas veces la musicalización es la "correcta" pero al recibir opiniones repetidas de un tema en particular se puede pensar que no es lo correcto y se pide una segunda opinión sobre la propuesta, entonces se corrige. Durante la ejecución del videojuego puede haber detalles que no agraden al contexto general, entonces se procede a corregirlo, estos detalles se toman a partir de opiniones generadas por los probadores beta. Se modela en base a cada secuencia que el videojuego contiene para tener un mejor control del producto. Compatibilidad adicional se refiere a errores que se generan durante la reproducción y que pueden originarse por muchas razones lógicas y algunas ilógicas. Pueden ser de software o de hardware, tanto en modo single como multiplayer así como por hardware inadecuado o carencia de alguna actualización del firmware o de algún driver o un actualización del sistema operativo originen funciones erróneas. Se soluciona sometiendo a pruebas extremas la aplicación, generar el error, manejarlo estadísticamente y resolverlo en la capa y circunstancias que lo originaron. Los modelos de calidad de software existentes, tanto como su aplicación de los modelos más importantes.

Cuando se aplican modelos de calidad, finalmente lo que se logra es que mejoren los procesos de software, así como la calidad del software que se desarrolla.

Modelos de gestión de calidad
Modelos Gestión de Calidad del Software
Modelos de Calidad del Software existentes
Estructura de los modelos de calidad de software
Concepto
Niveles
CMM (Capability Maturity Model)
Estructura del modelo CMM
Niveles CMMI
Modelos de calidad de software
Modelo McCall
Modelo de Boehm
ISO/IEC 9126

2.3.7 SOPORTE EN LÍNEA

Cuando el producto sale al mercado y es comprado, es muy normal que tenga errores, sin embargo, se proporcionara actualizaciones, parches, ayuda al usuario en la solución de posibles fallas técnicas que el producto haya tenido durante tu ejecución. Esta ayuda se proporciona en línea por que actualmente es un común denominador el uso de internet para muchas aplicaciones tales como la del entretenimiento. El videojuego en su estructura contiene los recursos para conectarse a internet y descargar contenidos extras, actualizaciones, certificaciones, cinemáticas entre otras cosas, así como actualizaciones de software para reproducirse correctamente con los controladores disponibles.

De forma automática o manual el videojuego puede actualizarse.

3 RECURSOS TECNOLÓGICOS

3.1 RECURSOS TECNOLÓGICOS

Cuando se habla de desarrollo de juegos, se suele pensar que existe un programador, que debe ser capaz de hacer todo el juego, pero pensar esto es un gran error, ya que realizar un videojuego involucra aparte de programadores, otros especialistas como diseñadores, músicos, escritores, etc.

Esto se centrará en los programadores, pero hay que tener en cuenta que si se quiere producir un buen juego, además de la originalidad, debe tener buenos gráficos, buena música, jugabilidad (gameplay), etc. Aunque todo lo que involucra este proceso puede ser construido por un solo programador, es más adecuado tener a un especialista en cada área.

Para comenzar en el desarrollo de videojuegos es preferible hacerlo con proyectos pequeños, con el fin de ser capaces de cumplir las metas que nos fijamos, nunca pensar en proyectos extremadamente grandes que se nos escapen de las manos, ya que lo más probable es que nunca los terminemos, lo cual nos llevará a la frustración.

Es común que las personas que se dedican a esta área del desarrollo, comiencen con re ediciones de juegos como viejos como: Pong®, Tetris®, Arkanoid®, Pacman®, Space Invaders®, Sokoban®, Snake®, Mario Bros®, etc.

La selección de juegos nombrados anteriormente son algunos de los primeros que aparecieron en la historia de los videojuegos, como el famoso Pong®. Todos estos juegos tienen algo en común, son juegos en 2D. Hoy en día la gran mayoría de los títulos que salen al mercado son juegos 3D como Doom®, Quake®, Call of Duty®, Unreal Tournament®, Need for Speed® entre otros.

Muchos pretenden comenzar a desarrollar un juego del tipo Doom, tienen que pensar que realizar uno de estos juegos de forma independiente (solo un programador) es una tarea titánica, además todos estos videojuegos fueron realizados por grandes equipos de personas (no solo programadores) y con presupuestos grandes. Esto no quiere decir que no podamos crear uno de estos, pero hay que ir paso a paso. Cuando ya se tenga dominado los juegos 2D, se pasa a crear algún proyecto 3D de un tamaño moderado o una buena idea es adaptar un juego 2D a 3D.

3.1.1 ESTRUCTURA BÁSICA DE UN VIDEOJUEGO

Básicamente como funciona por dentro un videojuego. Primero hay que aclarar que un videojuego es un programa diferente a los programas convencionales. Un videojuego debe funcionar en tiempo real, en todo momento mientras se está ejecutando, el juego debe estar realizando alguna tarea como: dibujar los objetos, actualizar coordenadas, calcular colisiones, etc., independiente de si el usuario hace algo. Obviamente debe también estar esperando que ocurra algún evento, ver si el usuario presiona alguna tecla, si mueve el mouse, o presiona algún botón de este y luego actuar en consecuencia. Todo esto y más ocurren en un ciclo o loop, apreciándose mejor en la fig. 3.1.

Básicamente la estructura de un videojuego consta de las siguientes partes:

Inicialización
Ciclo del videojuego
Entrada
Procesamiento
Salida
Finalización

Explicación en más detalle de cada una de ellas:

1. Inicialización: aquí inicializaremos todo lo que será usado luego en el ciclo del videojuego. Por ejemplo aquí inicializaremos la librería gráfica, un modo gráfico, el sistema de sonido/música, de texto y cualquier otro tipo de sistema necesario. Además reservaremos memoria para los objetos que intervienen en el juego, creación de estructuras de datos, etc. Carga de sonidos, de imágenes y de recursos en general. También en este proceso se inicializarán las posiciones iniciales de los personajes, carga de puntajes desde un archivo, etc.

2. Ciclo del videojuego: el ciclo del videojuego es un loop que se estará repitiendo una y otra vez. Aquí es donde ocurre toda la acción del juego, y la única forma para poder salir de este ciclo es cuando el jugador pierde, llega al final del juego o sale del videojuego con alguna combinación de teclas o presionando algún botón del mouse, etc. El ciclo del juego consta básicamente de tres partes:

2.1. Entrada: en esta parte se obtiene desde algún dispositivo de entrada (teclado, mouse, joystick, etc.) todo lo que realiza el jugador, por ejemplo que tecla presionó/soltó del teclado, que botón del mouse presionó/soltó, si movió el mouse en alguna dirección, etc.

2.2. Procesamiento: aquí se procesa toda la información que se recibió en el punto anterior y se toman decisiones a partir de los datos de entrada. Es decir aquí está toda la lógica del juego. Se procesa la física, inteligencia artificial, comunicación de datos en red, etc.

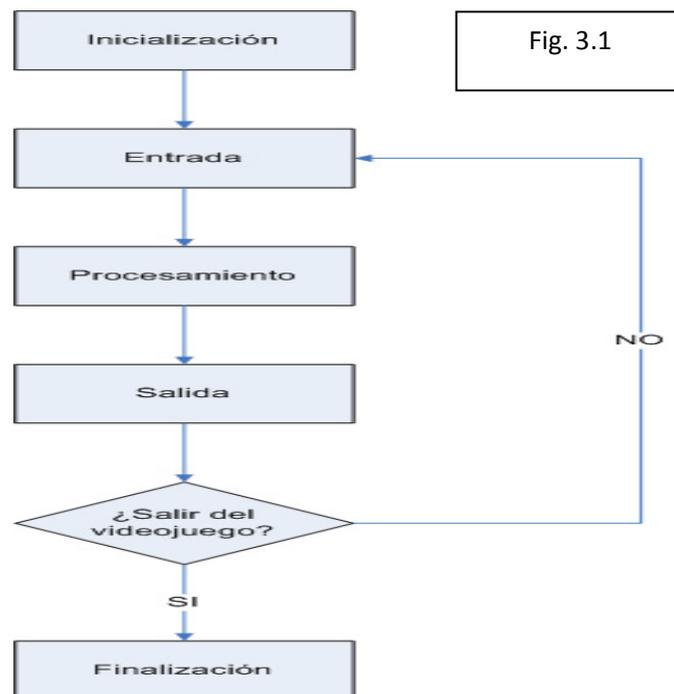
2.3. Salida: en este punto se muestra toda la información ya procesada en el punto anterior, aquí es donde mostramos los gráficos en pantalla, reproducimos sonidos, etc.

3. Finalización: por último en esta parte se hace básicamente lo opuesto a lo que hicimos en la inicialización, es decir, eliminar de la memoria todos los recursos almacenados, ya sea imágenes, sonidos, música, etc. Cerrar todos los sistemas que se abrieron en la inicialización. Guardar datos de puntajes en un archivo, etc.

Resumiendo vimos que los conocimientos básicos para desarrollar juegos son:

Lenguaje de programación.
Compilador/IDE.
Matemáticas.
Física.
Networking.
Scripting.
Inteligencia Artificial.
API Gráfica.
API para Sonido y Música.

Y la estructura básica de un videojuego la podemos ver en un simple diagrama de flujo:



La parte técnica del video juego comprende los elementos básicos que se ven en el diagrama. Todo videojuego se compone de 4 etapas básicas siendo la artística, la desarrolladora, la programación y las pruebas de aseguramiento de calidad. Cada etapa se integra de métodos, técnicas, esquemas, modelos de trabajo que dependen del software que se esté empleando para producirlo.

Esta etapa realmente es laboriosa. Se llevan a cabo millones de líneas de código, sin embargo con los métodos, procesos, técnicas apropiadas estos modelos se vuelven iterativos los cuales se automatizan logrando que cada línea de código redundante se elimine y solamente se empleen ciclos de trabajo según se vayan presentando los eventos requeridos. El video juego se mapea con la finalidad de facilitar el trabajo. Se asignan bloques de trabajo. La finalidad es hacerlo fácil, eficiente, cumplir en tiempo y ahorrar costos.

Con una filosofía eficiente de trabajo estas metas se cumplen.

Se procede a un mapeo de:

Videojuego.
Caracteres
Escenarios
Acciones
Objetos
Audio de fondo
Audio de eventos

3.1.2 API (APPLICATION PROGRAMMING INTERFACE)

Una API (Application Programming Interface) o Interfaz de Programación de Aplicaciones, es un conjunto de funciones que realizan tareas específicas facilitando la vida al programador. Al hablar de API grafica nos referimos a un grupo de funciones para inicializar por ejemplo modos gráficos, realizar copiado de datos de la memoria del computador a la tarjeta de video (blitting), etc. Existen API's específicas para cada tipo de tarea que se desee realizar. Actualmente las API's gráficas más usadas son OPENGL Y DIRECTX.

OpenGL (Open Graphics Library) es una API multiplataforma creada por Silicon Graphics® en 1992, que maneja solo el aspecto gráfico de un sistema, dejando afuera el sonido, música, control de teclado, mouse, joysticks, gamepads, etc., los que deben ser controlados con otras API's especializadas. Existe una biblioteca multiplataforma para el manejo de audio tridimensional llamada OpenAL (Open Audio Library).

DirectX es una API multimedia creada por Microsoft® en 1995, que consta básicamente de Direct3D para la parte gráfica, DirectSound y DirectMusic para la parte de audio, y DirecInput para el control de teclados, joysticks, etc. También incluye DirectPlay para comunicación de datos en redes.

El uso de estas dos API's en un comienzo puede ser un poco complicado y engorroso, especialmente DirectX. Existen algunas alternativas como Allegro o SDL, las que son multiplataforma.

SDL (Simple DirectMedia Layer) es una API gráfica para realizar operaciones de dibujo en 2D, gestionar efectos de sonido y música, y cargar imágenes.

Existen varias librerías que complementan SDL, algunas de ellas son:

SDL_image: Carga de imágenes en diferentes formatos: png, jpg, etc.

SDL_mixer: Carga de formatos de sonido como wav, mp3, ogg, etc.

SDL_net: Comunicación de datos en redes.

SDL_ttf: Uso de fuentes TrueType.

SDL_gfx: Dibujo de primitivas gráficas, escalar/rotar imágenes, control de framerate, Filtros de imagen MMX.

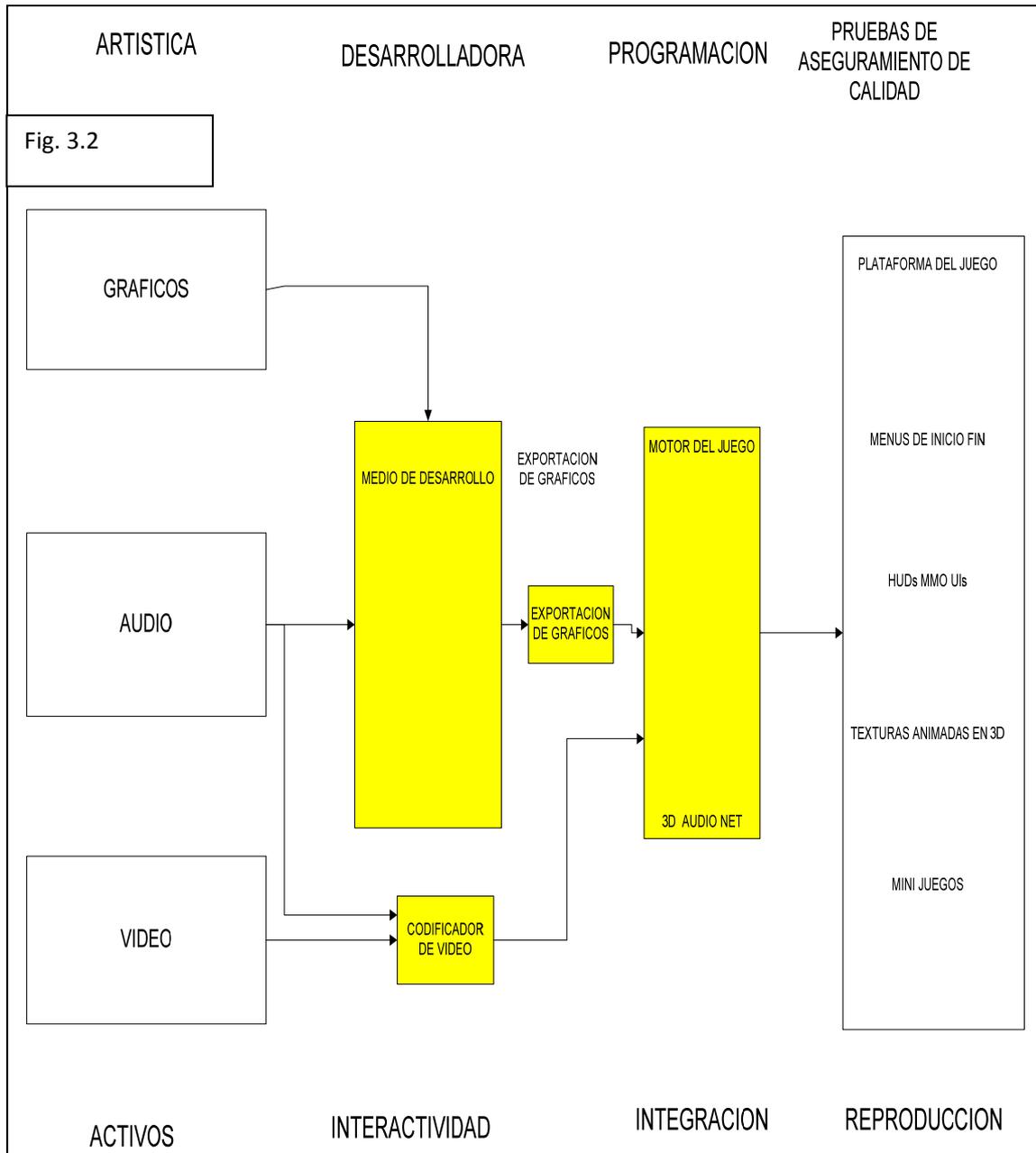
SDL permite básicamente crear aplicaciones en 2D, pero si queremos extender estas capacidades podemos usarla en conjunto con OpenGL, ya sea para crear nuevamente aplicaciones 2D que aprovechen las características de aceleración por hardware, que todas las tarjetas de video poseen hoy en día, como para crear aplicaciones completamente en 3D.

Ya vimos las herramientas y conocimientos necesarios para comenzar en el desarrollo de videojuegos, pero no podemos dejar de mencionar que también existen los Engines o Motores de juego, que están basados en algunas de las API's ya vistas, OpenGL o DirectX principalmente, que proveen al programador todas las funcionalidades necesarias para el desarrollo de un juego.

Básicamente un Engine está formado por varios sistemas y subsistemas, por ejemplo un sistema gráfico para manejar objetos 2D o 3D, un sistema de control de entrada (teclado, mouse, etc.), sistema de texto, sistema de red, sistema de scripts, sistema de audio, etc. Todos los conocimientos necesarios que vimos antes ahora están aplicados en un Engine. API's Gráficas. Existen Engines gratuitos (algunos bastante buenos y complejos) y otros comerciales, por ejemplo podemos nombrar los siguientes: Irrlicht, Ogre 3D, Crystal Space 3D, PopCap, Torque, etc.

Para comenzar a desarrollar conviene utilizar algo más sencillo.

Por ejemplo podríamos usar la dupla SDL/OpenGL junto a otras librerías externas para construir nuestro propio Engine. No solo es necesario disponer de las herramientas y conocimientos que ya vimos para el desarrollo de juegos, aun hay más, pero esto ya no es tarea del programador (aunque conviene también tener estos conocimientos en forma general por lo menos) estamos hablando del Diseño Gráfico de un juego. Alguien debe ser capaz de realizar las imágenes que aparecerán, ya sea en menús, pantallas de configuración y en el juego mismo.



Las API's (Application Programming Interface) o Interfaz de Programación de Aplicaciones, es un conjunto de funciones que realizan tareas específicas en la fig 3.2 muestra en donde operan las API's en el desarrollo del videojuego.

3.2 ¿QUÉ ES UN MOTOR DE JUEGO ?

El término " motor de juego " surgió a mediados de la década de 1990 en referencia a FIRST PERSON SHOOTING (FPS), en juegos como Doom increíblemente popular por id Software. con arquitectura con una separación razonablemente bien definida (como sistema de representación de gráficos en tres dimensiones, sistema de detección de colisiones, sistema de audio) y arte en el ambiente de juego, y un excelente gameplay es el legado de Doom.

El valor de esta separación se hizo evidente en que los desarrolladores comenzaron la concesión de licencias de juegos y reequipamiento en nuevos productos mediante la creación de nuevos diseños de arte, ambientes , armas, personajes , vehículos, y las reglas de juego con sólo cambios mínimos en el "motor" de juego.

Esto marcó el nacimiento de la "comunidad mod " -a grupo de jugadores individuales y pequeños estudios independientes que construyeron nuevos juegos modificando los juegos existentes, utilizando herramientas gratuitas proporcionadas por los desarrolladores originales.

Hacia el final de la década de 1990 , algunos juegos como Quake III Arena y Unreal fueron diseñados con la reutilización y el "modding" en mente. Los motores eran hecho altamente personalizable a través de lenguaje de scripting como C de Quake de id, y la concesión de licencias del motor comenzó a ser una fuente de ingresos secundaria viable para los desarrolladores que los creó . Hoy en día, los desarrolladores de juegos pueden licenciar el motor del juego y reutilizar partes significativas de sus componentes clave de software para construir juegos. Si bien esta práctica todavía implica una inversión considerable en la costumbre ingeniería de software, puede ser mucho más económico que el desarrollo de todos los componentes principales del motor.

La línea entre un juego y su motor es a menudo borrosa. Algunos motores hacen una distinción bastante clara , mientras que otros no. En otro juego, el motor de renderizado puede proporcionar facilidades materiales de uso general y de sombra, y el poder se define en su totalidad en los datos.

Podría decirse que una arquitectura basada en datos es lo que diferencia a un motor de juego a partir de una pieza de software que es un juego, pero no un motor . Cuando un juego contiene reglas lógicas o de juego codificadas de forma rígida, o emplea - caso especial código para hacer que tipos específicos de objetos del juego, se hace difícil o imposible para volver a usar el software para hacer un juego diferente. Probablemente deberíamos reservar el término "motor de juego" para el software que es extensible y se puede utilizar como la base para muchos juegos diferentes sin mayor modificación.

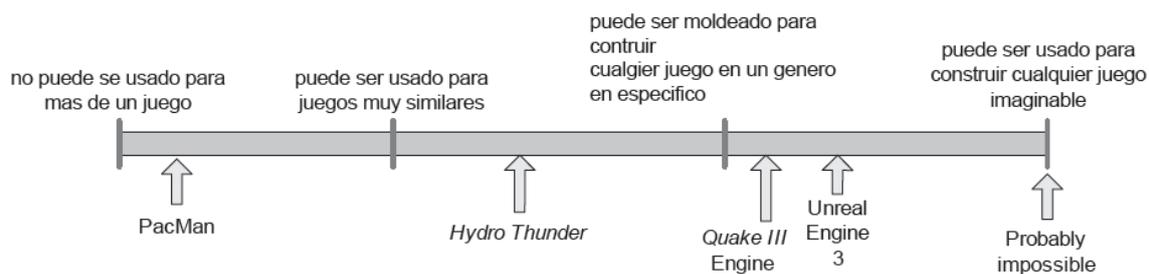


fig. 3.1. gama de reutilización de motore de juego

Claramente esto no es una distinción blanco y negro . Podemos pensar en una gama de reutilización en el que cada motor cae . Figura 1.1 toma una parte en las ubicaciones de algunos conocidos juegos/motores a lo largo de esta gama. Uno podría pensar que un motor de juego podría ser algo parecido a Apple QuickTime o Microsoft Windows Media Player, una pieza de uso general de software capaz de reproducir prácticamente cualquier contenido de juego imaginable. sin embargo este ideal todavía no se ha alcanzado (y puede nunca ser). La mayoría de los motores de juego son una cuidadosa obra de arte fino, ajustado para ejecutar un juego en particular en una determinado plataforma de hardware. E incluso los motores multiplataforma de mayoría de uso general son realmente adecuados para la construcción de juegos en un género en particular, como como tiradores de primera persona o juegos de carreras . Es seguro decir que cuanto más sea de propósito general sea un motor o su software intermedio componente de juego, menos óptima es la ejecución del juego en particular en una plataforma concreta.

Este fenómeno se debe a que el diseño de piezas eficiente de software, Por ejemplo, un motor de renderizado que fue diseñado para manejar los ambientes interiores íntimos probablemente no va a ser muy bueno en representación de vastos entornos al aire libre. El motor en espacios interiores podría utilizar un árbol BSP o sistema de portal para garantizar que no se dibuja la geometría que esta tapada por paredes u objetos que se encuentran más cerca de la cámara. El motor al aire libre, por otra parte, podría utilizar un mecanismo de oclusión menos exacta, o ninguno en absoluto, pero es probable que tenga un uso agresivo de técnicas del grado de detalle para asegurarse de que los objetos distantes se representan con un número mínimo de triángulos , durante el uso de alta resolución mallas de triángulos de la geometría que está cerca de la cámara. La llegada de equipos de computación cada vez más rápidos y tarjetas de gráficos especializadas, junto con algoritmos de renderizado y estructuras de datos cada vez más eficientes, está empezando a diferenciar entre los motores gráficos de géneros dedicados. Ahora es posible utilizar un tirador de primera persona para construir un juego de estrategia en tiempo real, por ejemplo. Sin embargo, el equilibrio entre la generalidad y la optimización persiste. Un juego siempre se puede hacer más impresionante en función a las necesidades específicas y las limitaciones en particular y/o plataforma de hardware.

3.2.1 PROPIEDADES DEL MOTOR DE JUEGO

Un motor de videojuego es un término que hace referencia a una serie de rutinas de programación que permiten el diseño, la creación y la representación de un videojuego. Del mismo modo existen motores de juegos que operan tanto en consolas de videojuegos como en sistemas operativos.

La funcionalidad básica de un motor es proveer al videojuego de un motor de renderizado para los gráficos 2D y 3D, motor físico o detector de colisiones, sonidos, scripting, animación, inteligencia artificial, redes, streaming, administración de memoria y un escenario gráfico.

El proceso de desarrollo de un videojuego puede variar notablemente por reutilizar o adaptar un mismo motor de videojuego para crear diferentes juegos. Un ejemplo de motor de juego sería el motor gráfico Doom engine, Euphoria® o el Rockstar Advanced Game Engine®.

DirectX® consta de las siguientes API :

Direct3D: utilizado para el procesamiento y la programación de gráficos en tres dimensiones (una de las características más usadas de DirectX).

Direct Graphics: para dibujar imágenes en dos dimensiones (planas), y para representar imágenes en tres dimensiones.

DirectInput: para procesar datos del teclado, mouse, joystick y otros controles para juegos.

DirectPlay: para comunicaciones en red.

DirectSound: para la reproducción y grabación de sonidos de ondas.

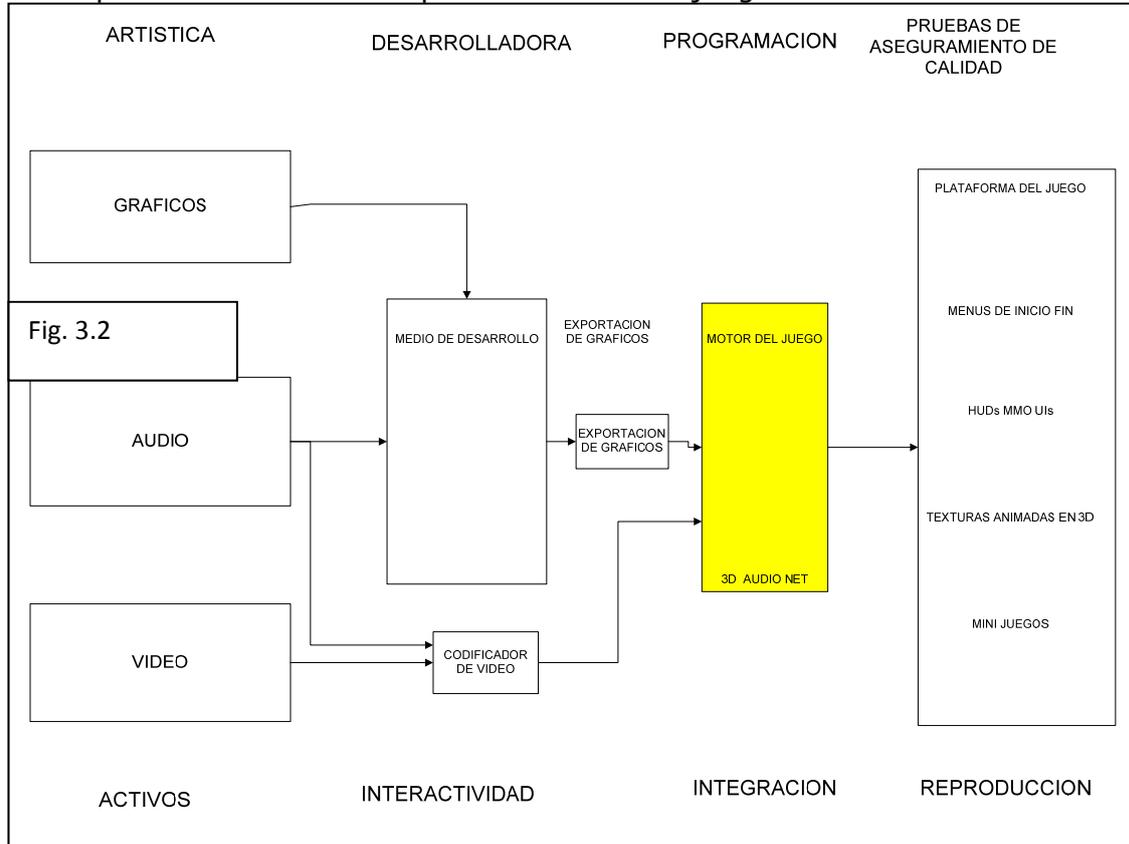
DirectMusic: para la reproducción de pistas musicales compuestas con DirectMusic Producer.

DirectShow: para reproducir audio y vídeo con transparencia de red.

DirectSetup: para la instalación de componentes DirectX.

DirectCompute: lenguaje e instrucciones especiales para el manejo de cientos o miles de hilos de procesamiento, especial para procesadores de núcleos masivos.

La fig. 3.2 muestra en qué etapa se encuentra el motor de juego y su participación en el desarrollo del producto. El motor de juego esta dividido en jerarquías básicas que ofrecen toda la interoperabilidad de los componentes del videojuego.



El motor gráfico o de renderización es el que proporciona funciones de renderizado 2D, 3D y de sprites. Se suele apoyar en librerías/APIs gráficas como OpenGL o DirectX. Define una capa de abstracción entre el HW/SO y el juego en sí. Además, se encarga de la visibilidad (BSP y buffers), el mapeado, el antialiasing, la gestión de mallas 3D, entre otras cosas.

El grafo de escena es una parte muy importante de un motor de juegos. Ésta es una estructura de datos muy utilizada en editores gráficos basados en vectores y en videojuegos. Se encarga de ordenar la representación lógica y, también a veces, espacial de una escena gráfica. Cada nodo puede tener muchos hijos, pero, un nodo sólo suele tener un padre. El efecto de algo sobre un padre es visible por todos sus hijos. Así, una operación sobre un grupo se propaga hacia todos sus nodos hijo. Esto permite manejar (escalar/mover/rotar/etc.) grupos de objetos como si fueran uno solo. En los juegos, cada nodo suele representar un objeto en la escena.

El motor de físicas es software que simula modelos de física utilizando variables del tipo velocidad, masa, etc. Su objetivo es simular y predecir los efectos bajo diversas situaciones de lo que ocurre en la vida real o en un mundo de fantasía. En los juegos se utilizan cada vez más, sobre todo en los de carreras donde

factores como la fuerza centrífuga o derrapes necesitan de ecuaciones físicas para aportar realismo.

El detector de colisiones es un algoritmo empleado para detectar cuándo dos o más objetos colisionan. Se suelen realizar mediante el cálculo de la intersección de volúmenes simples (Bounding Volumes). Se pueden utilizar distintos volúmenes para envolver el objeto y así detectar mejor las colisiones, como cubos o esferas. Se puede tomar como un componente independiente, aunque tiene mucho que ver con el grafo de escena y el motor de físicas.

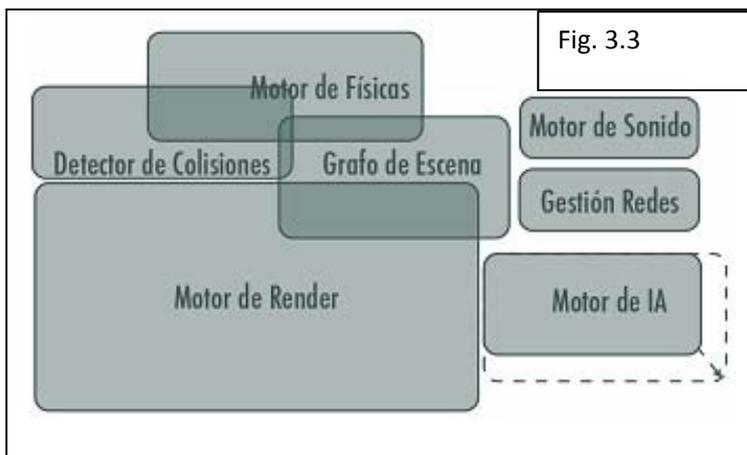
El motor de inteligencia artificial es el encargado de dotar a ciertos elementos del juego un comportamiento pseudointeligente. Por regla general, lo que se aplica en los juegos son técnicas muy simples de inteligencia artificial como máquinas de estados o algoritmos de búsquedas. Sin embargo, cada vez más se utilizan nuevas técnicas para conseguir una inteligencia más "realista" como redes neuronales o algoritmos genéticos.

El motor de sonido es el encargado de reproducir la banda sonora del juego y los efectos de sonido, como disparos o explosiones, en sincronía con la acción del juego.

En la gestión de envío/recepción de datos como coordenadas y voz, la parte de gestión de redes es la ventana de comunicación del videojuego a través de esquemas de cliente servidor.

Plantea un hecho radical la evolución de los juegos solitarios hacia los juegos online.

La fig. 3.3 muestra las relaciones que gestiona el motor de juego como parte fundamental del software de aplicación.



DirectX es una colección de API desarrolladas para facilitar las complejas tareas relacionadas con multimedia, especialmente programación de juegos y vídeo, en la plataforma Microsoft Windows. La fig. 3.4 muestra las características de los motores de juego disponibles.

Nombre	Desarrollador	Juegos Notables	Licencia	Lenguaje	Multiplataforma	Orientado a 2D/3D	SDL	Script	Notas y referencias
3D Rad	3impact		Freeware	C++	No	3D	Sí	AngelScript	
Adventure Game Studio	Chris Jones	Monkey Island Blackwell	Artistic License 2.0	C++	Sí	solo 2D	No	AGSScript	
CryEngine									
CryEngine 2									
Crystal Space									
Doom engine									
Euphoria	NaturalMotion	Grand Theft Auto The Force Unleashed	Propietario						
Id Tech 2									
Id Tech 3									
Id Tech 4									
Id Tech 5									
Irrlicht									
Game Maker									
Havok									
Infinity Engine									
Microsoft XNA									
Panda3D									
Pygame									
Rockstar Advanced									

Fig. 3.4

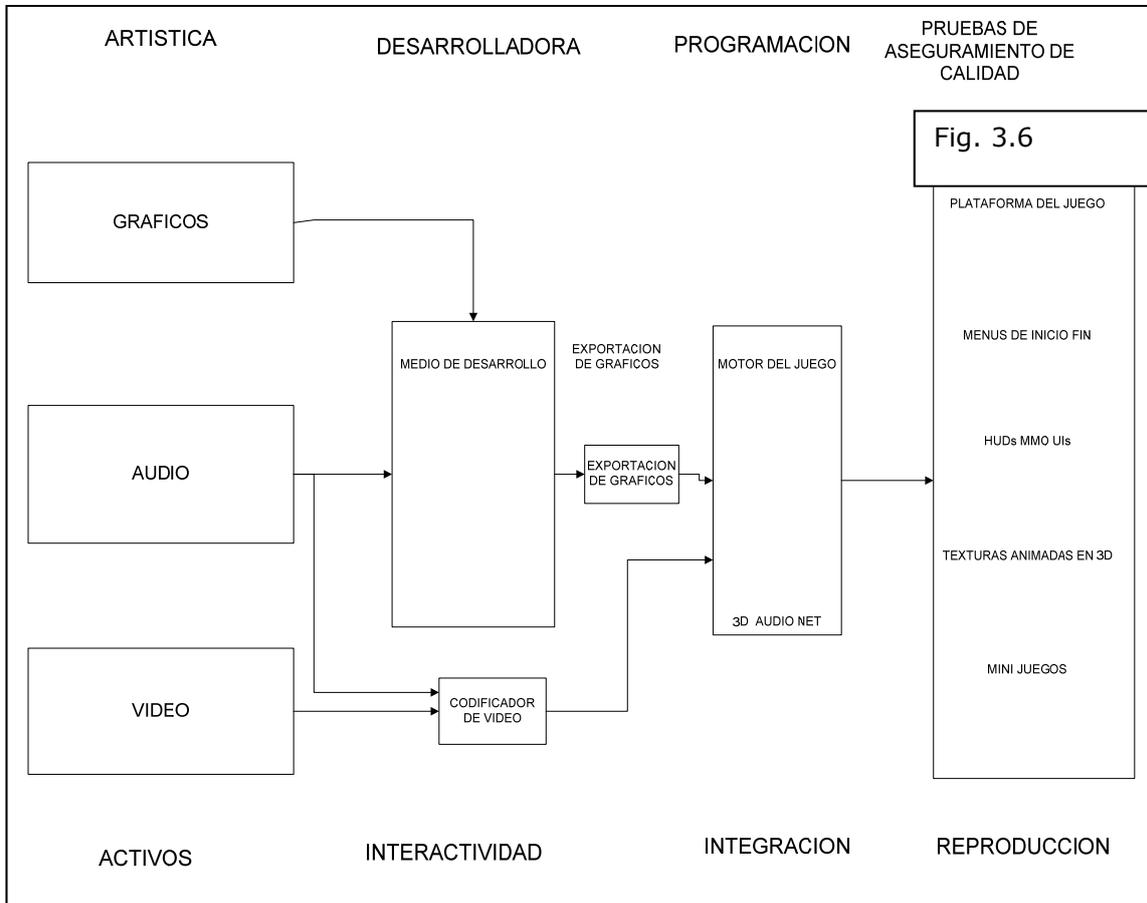
A pesar de ser desarrollado exclusivamente para la plataforma Windows, se está desarrollando una implementación de código abierto de su API para sistemas Unix (en particular Linux) y X Windows System por el proyecto WineHQ, del cual existe fork propietario, Cedega, desarrollada por la empresa de software Transgaming® y orientada a la ejecución de juegos desarrollados

para Windows® bajo sistemas Unix®. La fig. 3.5 muestra la evolución de las cualidades de directx, (no esta incluido directx12)

Versión de DirectX	Número de versión	Sistema Operativo	Fecha de Lanzamiento
DirectX 5.2	4.05.01.1998	Windows 98	5 de mayo de 1998
DirectX 6.0	4.06.00.0318	Windows CE para la Dreamcast	7 de agosto de 1998
DirectX 8.0a	4.08.00.0400 (RC14) + parches de instalación	Última versión para Windows 95	7 de noviembre de 2000
DirectX 8.1	4.08.01.0810 4.08.01.0881 (RC7)	Windows XP	12 de noviembre de 2001
DirectX 9.0	4.09.00.0900	Windows Server 2003	19 de diciembre de 2002
DirectX 9.0a	4.09.00.0901		26 de marzo de 2003
DirectX 9.0b	4.09.00.0902 (RC2)		13 de agosto de 2003
DirectX 9.0c	4.09.0000.0904 (RC0)	Windows XP SP2, Windows Server 2003 SP1, y Xbox 360 . Última versión exclusiva para 32-bit.	13 de diciembre de 2004
DirectX 9.0c	4.09.0000.0904	Compatible con todos los Sistemas Windows que soporten 9.0c (RC0) Primera versión en incluir D3DX DLLs	9 de diciembre de 2005
DirectX 9.0c - actualizaciones bimensuales	4.09.29.1974		Desde agosto del 2005 hasta el presente. Última versión: 18 de Abril de 2011
DirectX 10	6.00.6000.16386	Windows Vista y Windows 7	30 de noviembre de 2006
	6.00.6001.18000	Service Pack 1 para Windows Vista , Windows Server 2008 y Windows 7 incluye Direct3D 10.1	4 de febrero de 2008

Fig. 3.5

La etapa técnica del videojuego es un conglomerado de aplicaciones ya definidas por algunos fabricantes de software, facilitando el esquema operativo con sus respectivos ahorros de tiempo.



El fig. 3.6 muestra la división estructural y jerárquica donde se aprecia la estructura de la creación del software del videojuego en sus distintas etapas. Durante el desarrollo de esta tesis y también con el sentido común que involucra el conocimiento de las cosas llegamos a la conclusión de que hablar de tecnología puede parecer irrelevante por su constante cambio, lo cual nos deja atrás en esta cuestión. Sin embargo, para propósitos didácticos y aplicados solo mencionaremos las ideas principales (difícilmente cambiarán por la evolución de la tecnología). Podemos ver que un videojuego es efímero, su programación es de algún modo artesanal (para evitar ser pirateado fácilmente), también su codificación se hace en lenguajes comunes y se presta a ser modificado por terceros. Un videojuego es como cualquier aplicación informática, se compone de dos elementos básicos, que son el software y el hardware. Actualmente se divide el software en aplicaciones de 32 y 64 bits el hardware también, y las filosofías de programación siguen vigentes, entonces pasamos a explicarlas de forma sencilla

3.3 LA FAMILIA DE MOTORES DE QUAKE

El primer FIRST PERSON SHOOTER en 3D (FPS) fue y generalmente aceptado es *Castle Wolfenstein 3D* (1992). Escrito por id Software de Texas para la plataforma PC, este juego llevó a la industria de los juegos en una dirección nueva y emocionante. Id Software llegó a crear Doom, Quake, Quake II y Quake III. Todos estos motores son muy similares en la arquitectura, y refiriéndose a ellos como la familia de motores de Quake. La Tecnología de Quake se ha utilizado para crear muchos otros juegos e incluso de otros motores. Por ejemplo, el linaje de Medal of Honor para la plataforma PC es algo como esto:

Quake II (id)

Sin (Ritual)

F.A.K.K.2 (Ritual)

Medal of Honor: Allied Assault (2005 & Dreamworks Interactive)

Medal of Honor: Pacific Assault (Electronics Arts, Los Angeles)

Muchos otros juegos basados en la tecnología de Quake siguen caminos tortuosos a través de juegos y estudios . De hecho, el Valve's Source engine (utilizado para crear los juegos de Half-Life) también tiene raíces lejanas en la tecnología de Quake.

El código fuente de Quake y Quake II es de libre acceso, y los motores Quake originales está razonablemente buena estructurados y "limpios" (aunque, por supuesto, un poco anticuado, y escritos en su totalidad en C). Estas bases de código sirven como grandes ejemplos de cómo se construyen los motores de juego de potencia industrial. El código fuente completo de Quake y Quake II está disponible en la página web de la identificación en <http://www.idsoftware.com/negocio/techdownloads>.

Teniendo los juegos Quake/Quake II, se puede construir el código utilizando Microsoft Visual Studio y ejecutar el juego en el debugger utilizando los activos del juego verdadero. Esto es increíblemente instructivo. Puede establecer puntos de interrupción, ejecutar el juego, y luego analizar cómo el motor realmente funciona paso a paso por el código. Es recomendable descargar uno o ambos de estos motores y analizar el código fuente en esta manera.

3.4 LA FAMILIA DE MOTORES UNREAL

Epic Games Inc. irrumpió en la escena FPS en 1998, con su legendario juego de Unreal. Desde entonces, Unreal Engine se ha convertido en un importante competidor para la tecnología de Quake en el espacio de FPS. Unreal Engine 2 (UE2) es la base para Unreal Tournament 2004 (UT2004) y se ha utilizado para un sinnúmero de "mods", proyectos universitarios, y los juegos comerciales. Unreal Engine 3 (UE3) es el siguiente paso evolutivo, con algunas de las mejores herramientas y más ricos conjuntos de características del motor de la industria, incluyendo una interfaz gráfica de usuario conveniente y eficaz para crear shaders y una interfaz gráfica de usuario para la programación lógica juego llamado Kismet. Muchos juegos se están desarrollando con el UE3 últimamente, incluyendo, por supuesto, el popular GEARS OF WAR.

El Unreal Engine ha hecho conocido por su amplio conjunto de características y herramientas coherentes y fáciles de usar. Unreal Engine no es perfecto, y la mayoría de los desarrolladores modificarla de varias formas de ejecutar su juego de manera óptima en una plataforma de hardware en particular. Sin embargo, Unreal es una herramienta de prototipo increíblemente poderosa y la plataforma de desarrollo de juegos comerciales se puede utilizar para construir prácticamente cualquier juego 3D de primera persona o en tercera persona (por no hablar de los juegos de otros géneros también).

El Unreal Developer Network (UDN) proporciona un amplio conjunto de documentación e información sobre las diferentes versiones de Unreal Engine (ver <http://udn.epicgames.com>). Parte de la documentación de Unreal Engine 2 es de libre acceso, y "mods" pueden ser construidos por cualquier persona que posea una copia de UT2004. Sin embargo, el acceso al resto de la documentación UE2 y todos los documentos UE3 se limitan a licenciarios del motor. Por desgracia, las licencias son muy caras, y por lo tanto fuera del alcance de los desarrolladores de juegos independientes y pequeños estudios también. Pero hay bastantes sitios web y wikis útiles sobre Unreal. Uno muy popular es <http://www.beyondunreal.com>.

3.5 ARQUITECTURA DEL MOTOR DE JUEGO

Un motor de juego en general, consiste en un conjunto de herramientas y un componente de ejecución. la arquitectura es la primera pieza y después de entrar en las herramientas de arquitectura en la siguientes secciones.

Figura 3.7 muestra todos los principales componentes de ejecución que constituyen un motor de juego 3D típico. Sí, es grande! Y este diagrama ni siquiera cuenta para todas las herramientas. Motores de juegos son grandes sistemas software.

Al igual que todos los sistemas de software, los motores de juego se construyen en capas. Normalmente las capas superiores dependen de capas inferiores, pero no viceversa. Cuando una capa inferior depende de una capa superior, la llamamos una dependencia circular. dependencias de ciclos se deben evitar en cualquier sistema de software, porque conducen al indeseable acoplamiento entre sistemas, hacen que el software inestable, e inhiben la reutilización de código. Esto es especialmente cierto para un sistema a gran escala como un motor de juego.

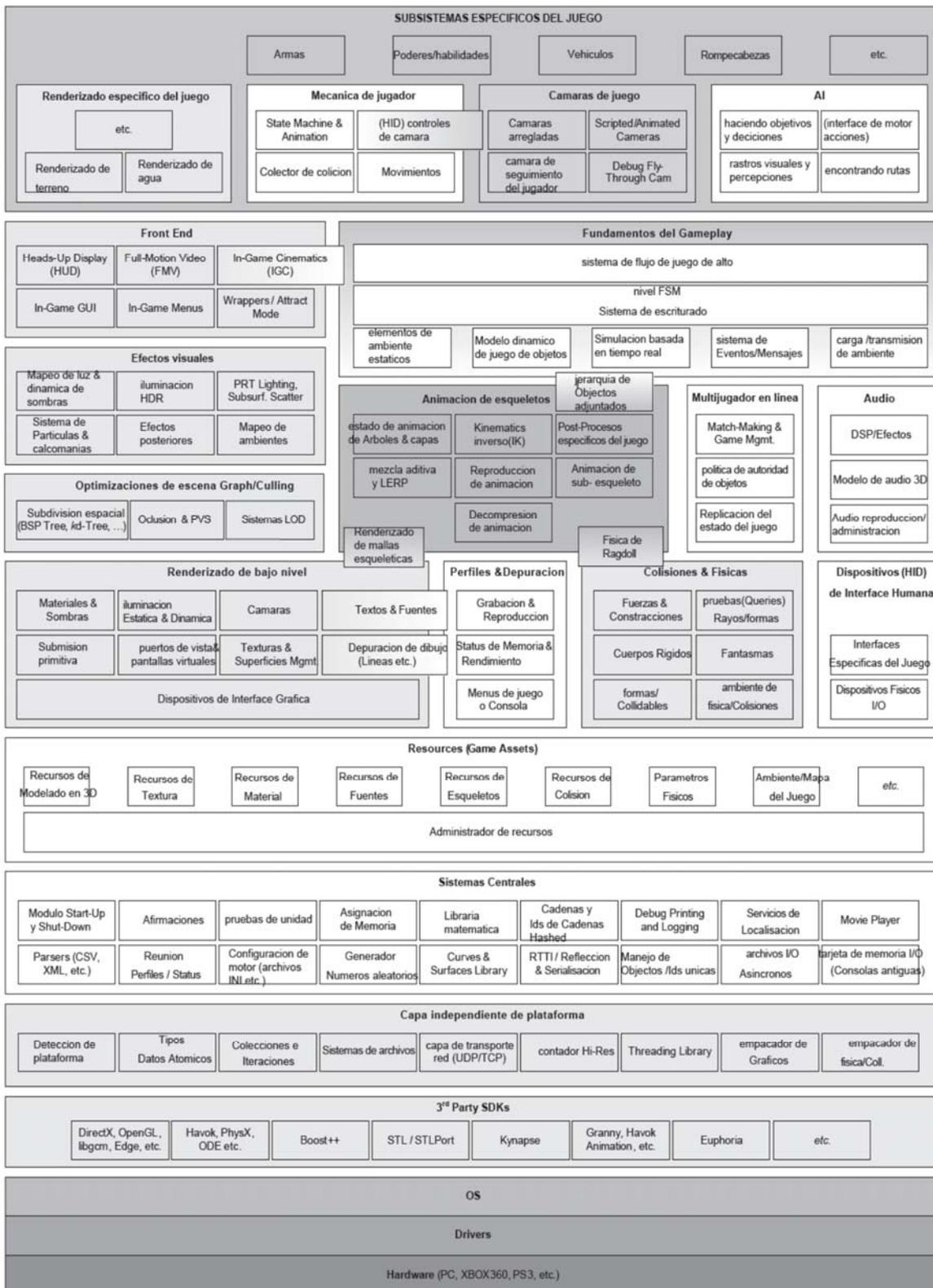


Fig. 3.7 Arquitectura de la ejecución del motor de juego.

3.5.1 HARDWARE DESTINO

La capa de hardware de destino, que se muestra de forma aislada en la Fig.3.8, representa el sistema de ordenador o consola en la que se ejecutará el juego. Plataformas típicas incluyen Microsoft Windows y Linux basados en PCs, el iPhone de Apple y Macintosh, Microsoft 's Xbox y Xbox 360, Sony PlayStation, PlayStation 2, PlayStation Portable (PSP) y PLAYSTATION 3, y DS de Nintendo, Game-Cube, y Wii. La mayoría de los temas son independientes de la plataforma, pero también vamos a tocar en algunas de las consideraciones de diseño propios de PC o consola de desarrollo, donde las distinciones son relevantes.



Fig. 3.8 Capa de hardware

3.5.2 CONTROLADORES DE DISPOSITIVO

Como muestra en la Figura 3.9, controladores de dispositivos son componentes de software de bajo nivel que proporciona el sistema operativo o el proveedor de hardware. Los controladores manejan los recursos de hardware y protegen el sistema operativo y las capas superiores de los motores de los detalles de la comunicación con las múltiples variantes de dispositivos de hardware disponibles.



Fig. 3.9 Capa de controlador de dispositivo

3.5.3 SISTEMA OPERATIVO

En un PC, el sistema operativo (OS) está funcionando todo el tiempo. Se organiza la ejecución de varios programas en un solo equipo, una de las cuales es el juego. La capa del sistema operativo se muestra en la Figura 3.10 Sistemas operativos como Microsoft Windows utilizan un enfoque de tiempo en secciones para compartir el hardware con los programas en ejecución múltiple, conocidos como multitarea preventiva. Esto significa que un juego de PC nunca puede asumir que tiene el control total del hardware debe "jugar limpio" con otros programas en el sistema.



Fig. 3.10 Capa del sistema operativo.

En una consola, el sistema operativo es a menudo es sólo una capa delgada de la biblioteca que se compila directamente en el ejecutable del juego. En una consola, el juego normalmente "dueño" de toda la máquina. Sin embargo, con la introducción de la Xbox 360 y PLAYSTATION 3, esto ya no es estrictamente el

caso. El funcionamiento sistema de estas consolas puede interrumpir la ejecución de su juego, o hacerse cargo de ciertos recursos del sistema, con el fin de mostrar los mensajes en línea, o para permitir al jugador para pausar el juego y traer a un Xross Media Bar de la PS3 o el tablero de instrumentos del Xbox 360, por ejemplo. Así que la brecha entre la consola y laPC se está cerrando poco a poco (para mejorar o empeorar).

3.6 SDK DE TERCEROS Y RECURSOS INTERMEDIOS

La mayoría de los motores de juegos aprovechan una serie de kits de desarrollo de software de terceros (SDK) y el middleware, como se muestra en la Fig. 3.11. La interfaz funcional o basado en la clase proporcionada por un SDK es a menudo es llamado interfaz de programación de aplicaciones (API). algunos ejemplos.



Fig. 3.11 Capa del kit de desarrollo de software por un tercero.

3.6.1 ESTRUCTURAS DE DATOS Y ALGORITMOS

Al igual que cualquier sistema de software, juegos dependen en gran medida de las estructuras de datos y algoritmos para manipularlos. Aquí hay algunos ejemplos de bibliotecas de terceros que proporcionan este tipo de servicios.

□ STL. Las plantillas de la biblioteca del estándar C++ proporciona una gran cantidad de código y algoritmos para el manejo de estructuras de datos, cadenas y basada en secuencias I/O.

□ STLport. Esta es una implementación portátil optimizada de STL.

Boost. Boost es un poderosa biblioteca de estructura de datos y algoritmos, diseñado en el estilo de STL.

Loki. Loki es una poderosa biblioteca de plantillas de programación genérica.

Los desarrolladores de juegos están divididos sobre la cuestión de si se debe usar bibliotecas de plantillas como STL en sus motores de juego. Algunos creen que los patrones de asignación de memoria de STL, no son propicias para alto rendimiento de la programación y que tienden a conducir a la fragmentación de la memoria, haga STL inutilizable en un juego. Otros creen que el poder y la conveniencia de STL superan sus problemas, y que la mayoría de los problemas pueden de hecho ser trabajados de todos modos.

3.6.2 GRÁFICOS

La mayoría de los motores de renderizado de juego se construyen en la parte superior de una biblioteca de interfaz de hardware, tales como los siguientes:

- Glide es el SDK de gráficos 3D para las antiguas tarjetas gráficas Voodoo. Este SDK era popular antes de la era de la transformación de hardware e iluminación (hardware T&L), que se inició con DirectX 8.
- OpenGL es un SDK gráficos 3D portátil utilizado.
- DirectX SDK es gráficos 3D Microsoft 's y su rival principal para OpenGL.
- libgcm es una interfaz directa de bajo nivel para el hardware de gráficos de RSX de PLAYSTATION 3, que fue facilitada por Sony como una más eficiente alternativa a OpenGL.
- Edge es un potente y altamente eficiente motor de renderizado y animación. producido por Naughty Dog y Sony para la PlayStation 3 y usados por una serie de estudios de juegos de primer y tercer participante.

3.6.3 COLISIONES Y FÍSICA

La detección de colisiones y la dinámica de cuerpos rígidos (conocida simplemente como "la física" en la comunidad de desarrollo de juegos) son proporcionados por los siguientes bien conocidos SDK.

- Havok es un popular motor de física y colisión de potencia industrial.
- PhysX es otro popular motor de física y colisión de potencia industrial, disponible para descarga gratuita desde NVIDIA.
- Open Dynamics Engine (ODE) es un bien conocido paquete de física/colisión de código abierto.

3.6.4 ANIMACIÓN DE CARÁCTER

Existe una serie de paquetes de animación comercial, pero sin duda no limitado a lo siguiente.

Granny. El Popular toolkit Granny de Rad Game Tools incluye robustos exportadores modelo y animación 3D para todas las principales modelado y animación 3D como Maya, 3D Studio MAX, etc, una biblioteca de tiempo de ejecución para la lectura y la manipulación de los datos del modelo y de animación exportados, y un sistema de animación de tiempo de ejecución de gran alcance. En mi opinión, el SDK Granny tiene el mejor diseñado y más lógica animación API de todos los que he visto, comercial o de propiedad, sobre todo su excelente manejo del tiempo.

Havok Animation. La línea entre la física y la animación es cada vez más borrosas como personajes se vuelven más y más realista. la empresa que fabrica los populares de física Havok SDK decidió crear un SDK gratuito de animación, lo

que hace que la reducción de la brecha entre la física y la animación mucho más fácil de lo que nunca lo ha sido.

Edge. La biblioteca de punta producida para PS3 por el equipo de ICE en Naughty Dog, las Herramientas y el grupo de Tecnología de Sony Computer Entertainment América, y Advanced Technology Group de Sony en Europa incluyen un motor poderoso de animación de gran alcance y eficiente procesamiento de geometría

3.6.5 INTELIGENCIA ARTIFICIAL

Kynapse. Hasta hace poco, la inteligencia artificial (IA) fue manejado de una forma acostumbrada para cada juego. Sin embargo, una compañía llamada Kynogon tiene producido un SDK middleware llamado Kynapse. Este SDK proporciona construcción de Bloques de bajo nivel de cómo encontrar un camino, evitar objetos estáticos y dinámicos, identificación de vulnerabilidades dentro de un espacio y una buena interfaz entre la IA y la animación.

3.6.6 MODELOS DE CARÁCTER BIOMECÁNICOS

Endorphin y Euphoria. son paquetes de animación que producen movimiento de carácter utilizando modelos humanos biomecánicos avanzados de movimiento realista.

Como mencionamos anteriormente, la línea entre la animación de los personajes y la física comienza a desdibujarse. Los paquetes como Havok Animation tratan de fusionarse con la física y la animación de una manera tradicional, con un animador humano que proporcionen la mayoría del movimiento a través de una herramienta como Maya y con la física de aumento que el movimiento en tiempo de ejecución. Pero recientemente un firma llamada Natural Motion Ltd. ha producido un producto que adelanta redefiniendo la forma de movimiento de caracteres que se maneja en los juegos y otras formas de medios digitales.

Su primer producto, endorfina, es un plug-in que permite a los animadores Maya para ejecutar simulaciones biomecánicas completos sobre personajes y exportar el archivo resultante en animaciones como si hubieran sido animados a mano. Las cuentas del modelo biomecánico para el centro de gravedad, distribución del peso del personaje, y detallada conocimiento de cómo un verdadero humanos se balancea y se mueve bajo la influencia de la gravedad y otras fuerzas.

Su segundo producto, Euphoria, es una versión en tiempo real de endorphin previsto para producir físicamente el movimiento carácter biomecánicamente preciso en tiempo de ejecución bajo la influencia de las fuerzas impredecibles.

3.6.7 CAPA DE LA INDEPENDENCIA DE PLATAFORMA

La mayoría de los motores de juego están obligados a ser capaces de funcionar en más de una plataforma de hardware. Las compañías como Electronic Arts y Activision/Blizzard, Por ejemplo, siempre dirigen sus juegos en una amplia variedad de plataformas, porque expone sus juegos para el mayor mercado

posible. Por lo general, los únicos estudios de juego que no están orientadas al menos dos plataformas por juego son los estudios de Naughty Dog de Sony y Insomniac studios. Por lo tanto, la mayoría de los motores de juego tienen una arquitectura con una capa de independencia de la plataforma, al igual que la que se muestra en la Fig. 3.12 Esta capa se encuentra sobre el hardware, los controladores de funcionamiento del sistema, y otros software de terceros, el resto del motor pertenece a la plataforma subyacente.

Envolviendo o reemplazando la biblioteca más utilizada de las funciones del estándar C, las llamadas al sistema operativo, y otra interfaces de programación de aplicaciones fundacional (APIs), la capa de la independencia de plataforma asegura consistente comportamiento en todas las plataformas de hardware. Esto es necesario porque hay una buena parte de la variación en todas las plataformas, incluso entre bibliotecas "estandarizados" como la biblioteca de C estándar.

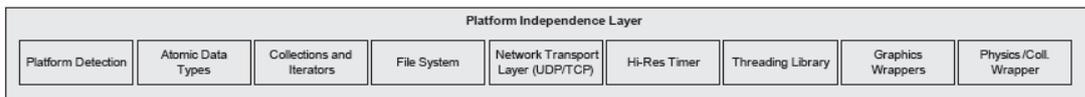


Fig. 3.12 Capa independiente de plataforma

3.6.8 SISTEMAS CENTRALES

Cada motor de juego, y todas las aplicaciones grandes y complejas en C++, requiere de utilidades software. categorizando estas bajo la etiqueta de "sistemas centrales." Una capa típica de sistemas básicos se muestra en la Fig. 3.13 Éstos son algunos ejemplos de las instalaciones de la capa de núcleo que generalmente ofrece.

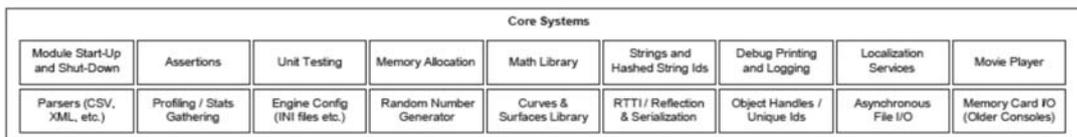


Fig. 3.13 Sistemas de motor nuclear

□ Las afirmaciones son líneas de código de comprobación de errores que se insertan para atrapar lógica errores y violaciones de los supuestos originales del programador . Chequeos de afirmación están generalmente despojados de la final de construcción de producción del juego .

□ Gestión de la memoria . Virtualmente cada motor de juego implementa su propio sistema(s) de asignación de memoria a medida para asegurar las asignaciones de alta velocidad y cancelaciones de asignación y para limitar los efectos negativos de la fragmentación de la memoria.

□ Math Library. Los juegos son por su naturaleza altamente matemáticos. Como tal, todos los motores de juego tiene al menos una, sino muchas bibliotecas

matemáticas. Estas bibliotecas proporcionan facilidades para el vectores, matrices, cuaterniones , trigonometría, operaciones geométricas con líneas , rayos , esferas, etc , manipulación de ranuras, integración numérica , solución de sistemas de ecuaciones , y todo lo demás que los programadores de juegos requieren .

□ estructuras de datos y algoritmos personalizados . A menos que los diseñadores de un motor decidieron a depender por completo de un paquete de terceros, tales como STL , un conjunto de herramientas para la gestión de las estructuras de datos fundamentales (listas enlazadas , dinámica matrices, árboles binarios , mapas de hash, etc) y algoritmos (buscar, ordenar , etc) por lo general se requiere . Estos a menudo son codificada a mano para minimizar o eliminar asignación dinámica de memoria y para garantizar un rendimiento óptimo de ejecución en la plataforma destino(s).

3.6.9 ADMINISTRADOR DE RECURSOS

Presente en todos los motores de juego de alguna forma, el administrador de recursos proporciona una interfaz de Unificada o conjunto de interfaces) para tener acceso a cualquier y todos los tipos de activos de juego y otros datos de entrada del motor. Algunos motores hacen esto de forma altamente centralizada y coherente (por ejemplo los paquetes de Unreal, OGRE 3D ResourceManagers class). Otros motores tienen un enfoque apropiado, a menudo se deja en manos del programador de juegos para acceder directamente a archivos crudos en el disco o dentro archivos comprimidos, como Quake 's PAK files. Un típico administrador de recursos de capa se representa en la Fig. 3.14



Fig. 3.14 Administrador de recursos

3.6.10 MOTOR DE RENDERIZADO

El motor de renderizado es uno de los componentes más grandes y complejos de cualquier motor de juego. Los Renderizadores pueden con ser organizados de muchas maneras. No hay manera de hacerlo, la representación moderna de motores comparten algunas filosofías de diseño fundamentales, impulsadas en gran parte por el diseño del hardware de gráficos 3D de la que dependen.

Un enfoque efectivo y común cercano al diseño de motor de renderizado es emplear una arquitectura en capas como sigue.

3.6.11 RENDERIZADO DE BAJO NIVEL

El renderizado de bajo nivel, que se muestra en la Fig. 3.15, cubre la totalidad en bruto de las instalaciones del motor. En este nivel, el diseño se centra en la prestación de una colección de primitivas geométricas tanto rápido y abundante

sea posible, sin mucho sentido para el que porciones de una escena pueden ser visibles. Este componente es dividido en varios sub-componentes, que se describen a continuación.

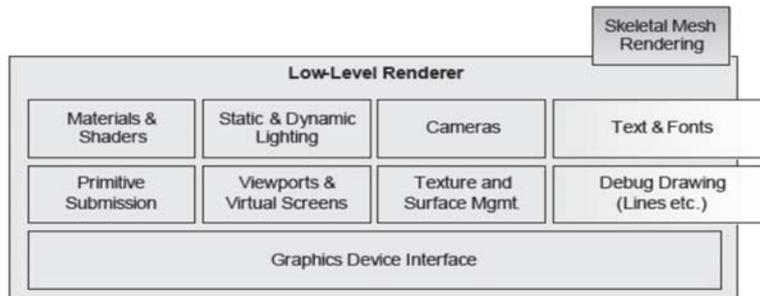


Fig. 3.15 Motor de renderizado de bajo nivel.

3.6.12 INTERFAZ DE DISPOSITIVO GRAFICO

Los SDK para Gráficos como DirectX y OpenGL, requieren una cantidad razonable de código para ser escrito es sólo para enumerar los dispositivos de gráficos disponibles, inicializarlos, establecer ajustes de reproducción (back-buffer, stencil-buffer etc), y así sucesivamente. Este suele ser manejado por un componente que se llama la interfaz de dispositivo gráfico (aunque cada motor utiliza su propia terminología).

Para un motor de juego de PC, también es necesario código para integrar su procesador con el bucle de mensajes de Windows. Por lo general, escribir un "mensaje emergente" de los servicios de Mensajes de Windows cuando están pendientes y de lo contrario corre el bucle de renderizado una y otra vez lo más rápido posible.

3.6.13 OTROS COMPONENTES DE RENDERIZACION

Los otros componentes de renderizado de bajo nivel que cooperan con el fin de recoger renderizados geométricos primitivos (a veces llamada renderizado de paquetes) , tales como mallas , listas de líneas, listas de puntos, partículas, parches de terreno, cadenas de texto, y lo que sea otra cosa que quiera dibujar, y hacerlos lo más rápido posible.

El procesador de bajo nivel por lo general proporciona una ventana gráfica de abstracción con un matriz asociada y parámetros de proyección 3D. El procesador de bajo nivel También gestiona el estado del hardware de gráficos y los shaders del juego a través su sistema de material y su sistema de iluminación dinámica.

3.6.14 OPTIMIZACIÓN DE ESCENA DE GRÁFICOS/ELIMINACIÓN

El procesador de bajo nivel se basa toda la geometría enviada a la misma, sin mucho respecto a si o no que la geometría es realmente visible. Un nivel más alto suele ser necesario con el fin de limitar el número de envíos primitivas para

la representación, sobre la base de algún tipo de determinación de la visibilidad. este capa se muestra en la Fig. 3.16.

Por muy pequeños mundos del juego, un simple corte de tronco (es decir, la eliminación de objetos que el cámara no puede "ver") es probablemente todo lo que se requiere. Para la caza mayor mundos, una estructura de datos de subdivisión espacial más avanzado puede ser utilizado para mejorar la prestación de eficiencia, al permitir que el conjunto potencialmente visible (PVS) de los objetos que se determine muy rápidamente.

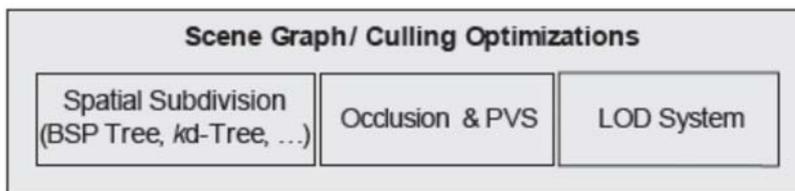


Fig. 3.16 Una típica escena de subdivisión de capas graph/spatial, para optimización del recortado.

Subdivisiones espaciales pueden tomar muchas formas , incluyendo un espacio particionado binario (BSP) de árbol, un árbol de cuatro ramas , una octree , un kd- árbol, o una jerarquía esfera . Una subdivisión espacial es a veces llamado un gráfico de la escena , aunque técnicamente la er latt es un tipo particular de estructura de datos y no subsumir la primera. Portales o métodos de sacrificio occlusión También podría aplicarse en esta capa del motor de renderizado.

idealmente el renderizado de bajo nivel debe estar completamente agnóstico al tipo de la subdivisión territorial o la escena gráfica que se utiliza. El diseño del Motor de renderizado de código abierto OGRE 3D ([http:// www.ogre3d.org](http://www.ogre3d.org)) es un gran ejemplo de este principio en acción. OGRE ofrece arquitectura de gráficos plug- and-play. Los desarrolladores de juegos pueden seleccionar entre una serie de diseños de gráficos pre implementado, o pueden proporcionar una aplicación de escenario gráfico personalizado.

3.6.15 EFECTOS VISUALES

Motores de juegos modernos soportan una amplia gama de efectos visuales, como se muestra en Fig. 3.17, incluyendo:

- sistema de partículas (por humo, salpicaduras de agua, etc);
- sistemas calcomanía (para agujeros de bala, huellas, etc);
- mapeo luz y mapeado de entorno;
- sombras dinámicas;
- efectos posteriores a pantalla completa aplicados después de una escena en 3D que ha sido renderizada para un buffer fuera de pantalla.

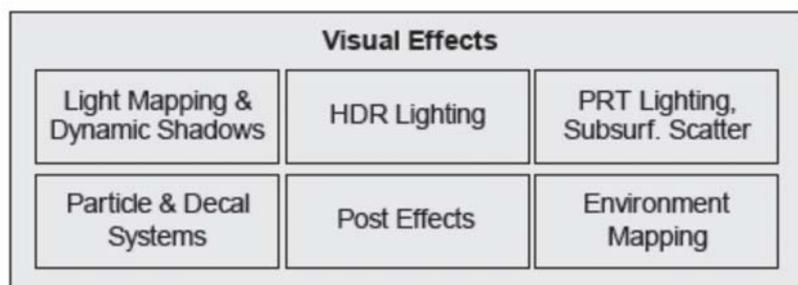


Fig. 3.17 Efectos visuales.

Algunos ejemplos de pantalla completa de efectos posteriores incluyen

- alto rango dinámico (HDR) iluminación y resplandor;
- pantalla completa anti-aliasing (FSAA);
- corrección de color y efectos de cambio de color, incluyendo la saturación bleach bypass, y efectos de desaturación, etc

Es común que un motor de juego tener un componente del sistema efectos que gestiona las necesidades de representación especializados de partículas, calcomanías y otros efectos visuales. Los sistemas de partículas y la calcomanía suelen ser componentes distintos del motor de renderizado y actúan como entradas al procesador de bajo nivel. Por

Por otro lado, la cartografía de la luz ambiente y las sombras son por lo general manejado internamente en el motor de renderizado adecuado. Efectos posteriores a pantalla completa se ha implementado ya sea como una parte integral del procesador o como una separada componente que opera en buffers de salida del procesador.

3.6.16 FRONT END (INTERFACE FINAL USUARIO MAQUINA)

La mayoría de los juegos emplean algún tipo de gráficos en 2D superpuestos en la escena 3D para diversos fines. Estos incluyen

- heads-up display del juego (HUD);
- menus en el juego, herramientas de desarrollo en una u otra consola, la cual puede ser enviada o no con el producto final.
- posiblemente, una interfaz gráfica de usuario (GUI) en el juego, permitiendo que el jugador manipular el inventario de su carácter, configurar las unidades de batalla, o realizar otras tareas complejas en el juego.

Esta capa se muestra en la Fig. 3.18 Gráficos bidimensionales como estos son usualmente implementado dibujando quads texturadas (pares de triángulos) con un proyección ortográfica. O pueden ser renderizados en 3D, con los quads bien arreglados pueden estar de frente a la cámara.

También hemos incluido el sistema de movimiento completo de video (FMV) en esta capa. este sistema es responsable de la reproducción de películas a pantalla completa que se han registrado anteriormente (ya sea prestado con motor de renderizado del juego o el uso de otro paquete de la RENDERIZACION).

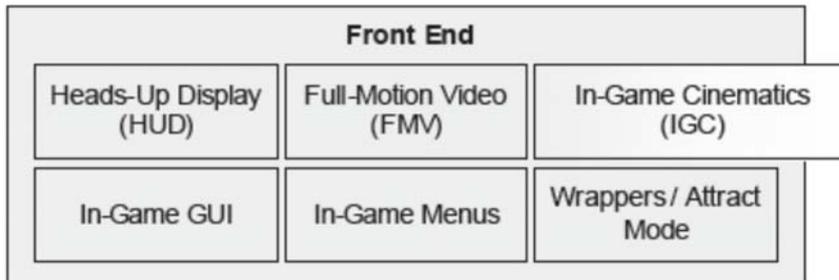


Fig. 3.18 Front y gráficos

Un sistema relacionado es las cinemáticas en el juego del sistema (IGC). Este componente normalmente permite secuencias cinemáticas que pueden ser coreografiados en el juego en sí, en 3D. Por ejemplo, ya que el jugador camina por una ciudad, una conversación entre dos personajes clave podrían ser implementados como un cine en el juego. IGC pueden o no pueden incluir el carácter (s) del jugador. Se pueden llevar a cabo como un corte deliberado en el que el jugador no tiene ningún control, o pueden ser sutilmente integrados en el juego sin que el jugador humano ni siquiera darse cuenta de que una IGC está teniendo lugar.

3.6.17 HERRAMIENTAS DE ANÁLISIS Y DEPURACIÓN

Los juegos son los sistemas en tiempo real y , como tal, los ingenieros de juego tienen la necesidad de optimizar el rendimiento. Además , recursos de memoria suelen ser escasos, por lo que los desarrolladores hacen un uso intensivo de la memoria. La capa de análisis y capa de depuración , que se muestra en la Fig. 3.19 , abarca estas herramientas y también incluye instalaciones de depuración en el juego , como el dibujo de depuración, un sistema de menú en el juego o la consola , y la capacidad de grabar y reproducir el juego con fines de prueba y depuración .

Hay suficientes herramientas de propósito general para este fin propósito:

- VTune de Intel,
- IBM`s Quantify and Purify.
- Compuware`s Bounds Checker .

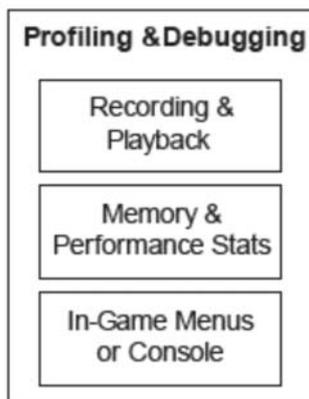


Fig. 3.19 Herramientas de análisis y depuración.

Sin embargo, la mayoría de los motores de juego también incorporan un conjunto de personalizada de herramientas de depuración y análisis. Por ejemplo, se pueden incluir uno o más de los siguientes:

- un mecanismo para instrumentar manualmente el código, por lo que las secciones específicos de código se puede programar;
- una instalación para la visualización de las estadísticas de análisis en pantalla mientras el juego se está ejecutando;
- una instalación para el vertido de las estadísticas de rendimiento a un fichero de texto o una hoja de cálculo Excel;
- una instalación para la determinación de la cantidad de memoria que está siendo utilizada por el motor, y por cada subsistema, incluyendo varias exhibiciones de pantalla;
- la capacidad de eliminar el uso de memoria , las estadísticas de fuga cuando el juego termina y / o durante el juego;
- herramientas que permiten imprimir las declaraciones, junto con la capacidad de activar o desactivar las categorías de depuración y controlar el nivel de excedentes de la salida;
- la posibilidad de grabar los eventos del juego y luego reproducirlas. Esto es duro para hacerlo bien, pero cuando se hace correctamente, puede ser una herramienta muy valiosa para rastrear errores.

3.6.18 COLISIONES Y FÍSICA

La detección de colisiones es importante para todos los juegos. Sin ella , los objetos se inter penetran , y sería imposible para interactuar con el mundo virtual en cualquier forma razonable . Algunos juegos también incluyen una dinámica realista o semi-realista de la simulación. A esto le llamamos el "sistema de la física" en la industria del juego, aunque la dinámica de cuerpos rígidos

realmente más apropiado, ya que son por lo general sólo se ocupa del movimiento (cinemática) de cuerpos rígidos y las fuerzas y pares de torsión (dinámica) que hacen que este movimiento que se produzca . Esta capa se representa en la Fig. 3.20

Colisión y física son por lo general bastante fuertemente acoplados . Esto es porque cuando se detectan colisiones , que casi siempre se resuelven como parte de la integración física y lógica satisfacen restricciones . Hoy en día , muy pocos compañías de juegos que escriban su propio motor de colisión / física. En cambio, un SDK está integrado típicamente en el motor.

□ Havok es el estándar de oro en la industria hoy en día. Es rico en características y desempeña bien sobre las tablas .

□ PhysX de NVIDIA es otra excelente motor de colisiones y dinámica. Se integra en el Unreal Engine 3 y también está disponible de forma gratuita como un producto independiente para el desarrollo de juegos de PC. PhysX fue originalmente diseñado como la interfaz para el nuevo chip acelerador de física de Ageia. el SDK pertenece y es distribuido por NVIDIA, y PhysX es adaptando para funcionar en sus últimas GPU.

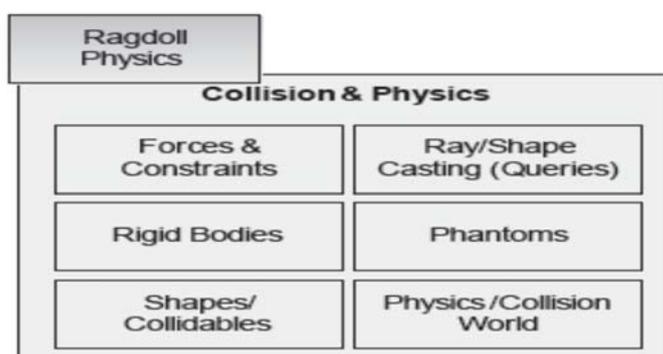


Fig. 3.20 Subsistemas de física y colisión.

La física de código abierto y los motores de colisión también están disponibles. tal vez el más conocido de ellos es el motor de dinámica abierta (ODE). Para obtener más información, ver <http://www.ode.org>.

I-Collide, V-Collide, y RAPID son otros populares motores de detección de colisión no comerciales. Los tres fueron desarrollados en el Universidad de Carolina del Norte (UNC). Para obtener más información, consulte http://www.cs.unc.edu/~geom/I_COLLIDE/index.html, http://www.cs.unc.edu/~geom/V_COLLIDE/index.html, y <http://www.cs.unc.edu/~geom/OBB/OBBT.html>

3.6.19 ANIMACIÓN

Cualquier juego que tiene caracteres orgánicos o semi-orgánicos (humanos, animales, dibujos animados personajes, o incluso robots) necesita un sistema de animación. Hay cinco tipos básicos de animación utilizada en los juegos:

- sprites/texturean de aproximación
- jerarquía de animación de cuerpo rígido
- animación de esqueleto
- animación de vértices
- objetivos de forma.

Animación esquelética permite una detallada malla de personajes 3D que se plantea un animador utilizando un sistema relativamente simple de los huesos. A medida que los huesos se mueven, la vértices de la malla 3D se mueven con ellos. Aunque objetivos morph y animación de vértice se usan en algunos motores, animación del esqueleto es el más frecuente método de animación en los juegos de hoy en día. Un sistema típico de animación del esqueleto se muestra en la Fig. 3.21

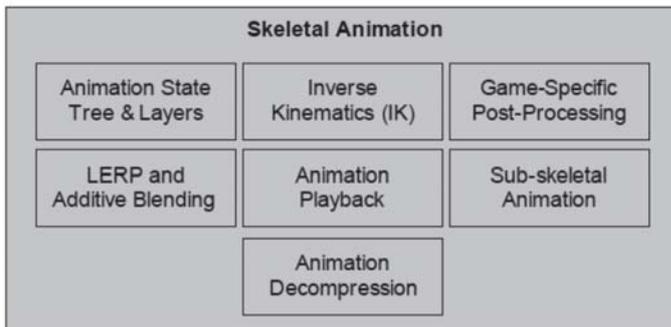


Fig. 3.21 Subsistema de animación esquelética.

El renderizado de esqueletos de mallas es un componente que cierra la brecha entre el renderizado y el sistema de animación. Es una cooperación estrecha, pero la interfaz está muy bien definida. El sistema de animación produce una pose para cada hueso en el esqueleto, y a continuación, estas posturas se pasan al motor de renderizado como una paleta de matrices. El procesador transforma cada vértice por la matriz o matrices de la paleta, con el fin de generar un final mezclado de posición de vértice.

Este proceso se conoce como desollado. También hay un estrecho acoplamiento entre los sistemas de animación y la física, cuando se emplean muñecos de trapo. Una muñeca de trapo es un personaje animado (a menudo en muerte), cuyo movimiento corporal es simulado por el sistema de la física. La física del sistema determina las posiciones y orientaciones de las diversas partes de la

cuerpo al tratarlos como un sistema limitado de cuerpos rígidos. El sistema de animación calcula la paleta de matrices requeridas por el motor de renderizado en Para dibujar el personaje en la pantalla.

3.6.20 DISPOSITIVOS DE INTERFACE HUMANA

Cada juego tiene que procesar las entradas del jugador, obtenida de diversos dispositivo de interfaz humana (HID s), incluyendo

- el teclado y el ratón,
- un joypad
- otros controles para juegos especializados, como volantes, joysticks, cojines de la danza, gamepads, etc

A veces llamamos este componente del jugador I/O, ya que También puede proporcionar una salida al jugador a través de la HID , tales como la fuerza de respuesta/ retumbar en un joypad o el audio producido por el WiiMote. La típica capa de HID se muestra en la fig. 3.22

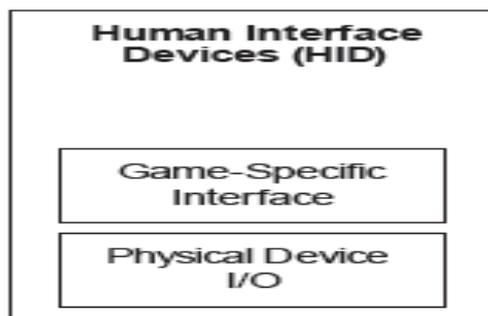


fig. 3.22 El sistema de input/output, también conocido como la capa de dispositivos de interface humana.

El componente de motor de HID es la separación entre los detalles de bajo nivel de los controladores de juego en una plataforma de hardware de controles de juego de alto nivel. Maneja datos brutos de la entrada del hardware, introduciendo un punto central alrededor de una zona muerta del joystick. Esto a menudo proporciona un mecanismo lo que permite al jugador personalizar la asignación entre los controles físicos y funciones de juego lógicas. A veces también incluye un sistema para detectar acordes, secuencias y gestos.

3.6.21 AUDIO

El audio es igual de importante que los gráficos en cualquier motor de juego. Desafortunadamente , audio menudo recibe menos atención que la renderizacion, la física, la animación, la IA, y el gameplay. La capa de audio se representa en Fig. 3.23

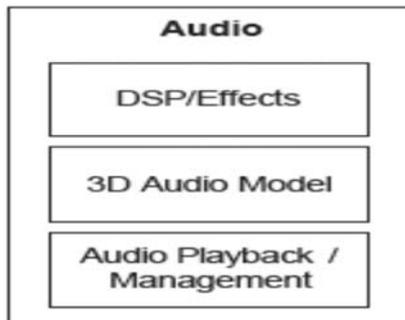


Fig. 3.23 Subsistema de audio.

Motores de audio varían mucho en complejidad. Los motores de audio de Quake y Unreal son muy básicos, y los equipos de juego por lo general los aumentan con funcionalidad personalizada o reemplazarlos con una solución interna. Para plataformas DirectX (PC y Xbox 360), Microsoft proporciona un excelente conjunto de herramientas de audio llamado XACT. Electronic Arts ha desarrollado un avanzado motor de audio de alta potencia llamado internamente SoundR!OT. Junto como fabricantes como estudios Naughty Dog, Sony Computer Entertainment America (SCEA) proporciona un potente motor de audio 3D llamado Scream, que se ha utilizado en una serie de títulos de PS3 incluyendo de Naughty Dog Uncharted. Sin embargo, incluso si un equipo de juego utiliza un motor de audio pre-existente, todos los partidos requieren una gran cantidad de desarrollo a medida personalizada suave, el trabajo de integración y atención a los detalles con el fin de producir audio de alta calidad en el producto final.

3.6.22 MULTIJUGADORES EN LÍNEA Y EN RED

Muchos juegos permiten múltiples jugadores jugar dentro de un solo mundo virtual. Juegos multijugador vienen en al menos cuatro esquemas básicos.

- multijugador de pantalla individual. Dos o más dispositivos de interfaz humana (joypads, teclados, ratones, etc) están conectados a una única máquina arcade, PC, o consola. Múltiples personajes jugadores habitan un mismo mundo virtual, y una sola cámara mantiene todos los personajes jugadores en marco al mismo tiempo.
- multijugador de pantalla dividida. Múltiples personajes habitan un solo jugador virtual mundo, con múltiples en una única máquina de juego, pero cada uno con su propia cámara, y la pantalla se divide en secciones para que cada jugador puede ver su carácter.
- multijugador en red. Varios equipos o consolas están conectados en red juntos, con cada máquina que aloja uno de los jugadores.

□ juegos multijugador masivos online (MMOG). Literalmente, cientos demiles de usuarios pueden reproducir simultáneamente dentro de un gigante mundo virtual en línea organizada por un poderoso centro de servidores.

La capa de red multijugador se muestra en la Fig. 3.24 Juegos multijugador son muy similares en muchos aspectos a sus contrapartes de un solo jugador. Sin embargo , el soporte para múltiples jugadores pueden tener un profundo impacto en el diseño de ciertos componentes del motor de juego. El mundo del modelo juego de objetos, sistema de dispositivo de entrada humana , el sistema de control de renderizado, y sistemas de animación son todos afectados por las características multijugador en un motor de un solo jugador preexistente ciertamente no es imposible, aunque puede ser una tarea desalentadora. Aun así, muchos equipos de juego han hecho con éxito . que Dicho esto, es mejor diseñar características multijugador desde el primer día.

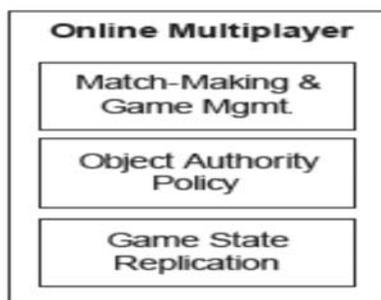


Fig. 3.24 Subsistema de multijugador en línea.

Es interesante observar que en la otra dirección - la conversión de un juego multijugador en un juego de un solo jugador - suele ser trivial. De hecho, muchos de motores de juego tratan el modo para un solo jugador como un caso especial de un juego multijugador, en que no pasa a ser sólo un jugador. El motor de Quake es bien conocida por su modo cliente- on-top -of- servidor, en el que un único ejecutable , se ejecuta en un solo PC , actúa como el cliente y el servidor en las campañas para un solo jugador.

3.6.23 BASES FUNDAMENTALES DEL GAMEPLAY

El término jugabilidad se refiere a la acción que tiene lugar en el juego, las reglas que rigen el mundo virtual en el que el juego se lleva a cabo, las habilidades del personaje del jugador(es) (conocida como mecánica del jugador) y de los otros personajes y los objetos del mundo, y las metas y objetivos del jugador(es). gameplay se implementa típicamente en la lengua nativa en la que el resto del motor está escrita, o en una secuencia de comandos de lenguaje de alto nivel, a veces ambos. Para cerrar la brecha entre el código del juego y el motor de bajo nivel los sistemas que hemos discutido hasta ahora, la mayoría de los motores de juego introducen una capa que se llama la capa de bases de juego (a falta de un nombre normalizado).

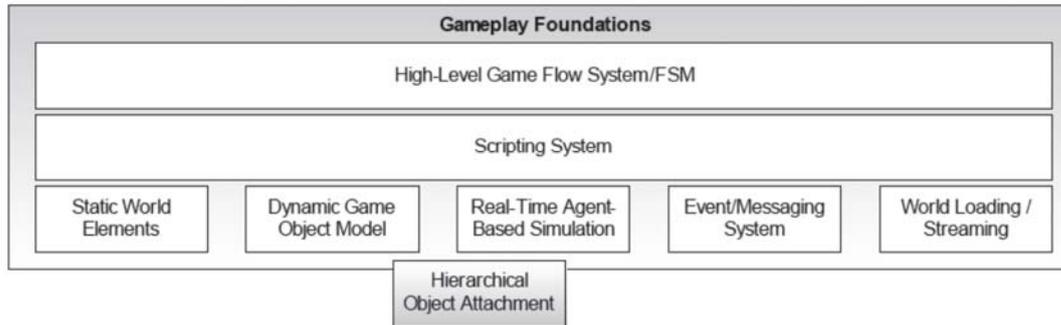


Fig. 3.25 Sistemas fundamentales del gameplay.

Se muestra en la Fig. 3.25, esta capa proporciona un conjunto de servicios básicos, en los que la lógica del juego de la especificación puede ser implementado convenientemente.

3.6.24 MUNDOS DE JUEGO Y MODELOS DE OBJETO

La capa de fundamental de juego introduce la noción de un mundo de juego, que contiene elementos estáticos y dinámicos. El contenido del mundo son usualmente modelada de una manera orientada a objetos (a menudo es, pero no siempre, el uso de un lenguaje de programación orientado a objetos). La colección de los tipos de objetos que componen un juego se llama el modelo de objetos de juego. El modelo de objeto de juego proporciona una simulación en tiempo real de una colección heterogénea de objetos en el mundo del juego virtual.

Los tipos típicos de los objetos del juego incluyen

- geometría estática de fondo , como edificios , caminos , terrenos a menudo en un especial caso), etc;
- cuerpos rígidos dinámicos, piedras, latas de refrescos, sillas, etc
- personajes jugadores (CP);
- Los personajes no jugables (NPC);
- Armas;
- proyectiles;
- vehículos;
- luces (que puede estar presente en la escena dinámica en tiempo de ejecución, o sólo utilizado para la iluminación estática offline);
- cámaras;
- y la lista sigue.

El modelo de mundo del juego está íntimamente ligada a un modelo de software de objetos, y este modelo puede acabar impregnando todo el motor. El software de objetos se refiere al conjunto de características de lenguaje, las políticas y las convenciones que se utilizan para implementar una pieza de software orientado a objetos. En el contexto de los motores de juego, el modelo de software de objetos responde a preguntas, tales como:

- ¿Es su motor de juego diseñado de una manera orientada a objetos?
- ¿Qué lenguaje utilizarán? C? C++? Java?
- ¿Cómo se organizará la jerarquía de clase estática ? Una jerarquía monolítica gigante? Muchos de los componentes de acoplamiento flexible?
- ¿Va a utilizar las plantillas y el diseño basado en políticas , o polimorfismo tradicional?
- Cómo se hace referencia a los objetos? Punteros?, punteros inteligentes?
- ¿Cómo van a ser identificados los objetos con una id única? Por dirección en la memoria? Por nombre? Por identificación única global (GUID)?
- ¿Cómo son los manejados tiempos de vida de los objetos del juego?
- Cómo están los estados de los objetos del juego simulan el paso del tiempo?

3.6.25 SISTEMA DE EVENTOS

Los objetos del juego, invariablemente, tienen que comunicarse entre sí. Esto puede ser logrado en todo tipo de formas. Por ejemplo, el objeto de enviar el mensaje podría simplemente llamar a una función miembro del objeto receptor. Un evento impulsado por arquitectura, al igual que lo que uno encuentra en una interfaz gráfica de típica, es también un enfoque común para la comunicación entre objetos. En un sistema controlador de eventos, el remitente crea una estructura de datos que llama un evento o mensaje, conteniendo el tipo de mensaje y cualquier argumento de que se van a enviar.

El evento se pasa al objeto receptor llamando a su función de control de eventos. Los eventos también se pueden almacenar en una cola para la manipulación en algún momento futuro.

3.6.26 SISTEMA DE ESCRITURADO

Muchos motores de juego emplean un lenguaje de programación con el fin de hacer que el desarrollo de las reglas del juego en especificación y contenido sean más fáciles y más rápidas. Sin un lenguaje de script, debe volver a compilar y volver a enlazar el ejecutable del juego cada vez que se realiza un cambio en las estructuras lógicas o de datos utilizados en el motor. Pero cuando un lenguaje de programación está integrado en el motor, los cambios a lógica juego y los datos se puede hacer mediante la modificación y volver a cargar el código del script. Algunos motores permiten script para volver a cargar mientras el juego

continúa ejecutándose. Pero de cualquier manera, el tiempo de vuelta es todavía mucho más rápido de lo que sería si tuviera que volver a compilar y vincular el ejecutable del juego.

3.6.27 FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL

Muchos motores de juego emplean un lenguaje de programación con el fin de hacer que el desarrollo de las reglas de juego específicas y el contenido más fácil y más rápido. Sin un lenguaje de script, debe volver a compilar y volver a enlazar el ejecutable del juego cada vez que se realiza un cambio en las estructuras lógicas o de datos utilizados en el motor. Pero cuando un lenguaje de programación integrado en el motor, los cambios al juego de la lógica y los datos se puede hacer mediante la modificación y volver a cargar el código script. Algunos motores permiten script para volver a cargar mientras el juego continúa ejecutándose. Otros motores requieren el juego que sea cerrado antes de la recopilación. Pero de cualquier manera, el tiempo de vuelta es todavía mucho más rápido de lo que sería si tuviera que volver a compilar y vincular el ejecutable del juego.

una red de nodos de ruta o los volúmenes de itinerancia, que definen áreas o caminos donde personajes de inteligencia artificial son libres de moverse sin temor a chocar con la geometría estática;

información de colisiones simplificada alrededor de los bordes de cada área libre de itinerancia;

conocimiento de las entradas y salidas de una región, y de donde en cada región un enemigo podría ser capaz de ver y/o emboscar ;

un motor de búsqueda de camino en el bien conocido algoritmo A;

ganchos en el modelo de sistema de colisiones y el mundo, para la línea de visión directa(LOS) y otras percepciones ;

un modelo mundial adaptado que dice al sistema de inteligencia artificial, donde todas las entidades del interés (amigos, enemigos, obstáculos) son, permite evitar dinámica de los objetos en movimiento, y así sucesivamente .

Kynapse también proporciona una arquitectura para la capa de decisión IA, incluyendo el concepto de cerebros (uno por carácter), agentes (cada uno de los cuales es responsable para la ejecución de una tarea específica, tales como la adopción de un punto a punto, anillo fuego en un enemigo, busca de enemigos, etc.), y las acciones (responsables de permitir que al carácter realizar un movimiento fundamental, que a menudo es resultado en juego de animaciones del esqueleto del personaje) .

3.6.28 SUBSISTEMAS ESPECÍFICOS DEL JUEGO

En la parte superior de la capa base del juego y los otros componentes del motor de bajo nivel, programadores y diseñadores de juego cooperan para implementar las características del juego en sí. Los sistemas de juego suelen ser numerosos, muy variados, y específicos para el juego que está siendo desarrollado. Como se muestra en la Fig. 3.26, estos sistemas incluyen, pero no son ciertamente limitados a la mecánica del personaje del jugador, varios sistemas de cámara en el juego, la inteligencia artificial para el control de los personajes no jugables (NPC), sistemas de armas, vehículos y la lista sigue. Si una línea clara se puede extraer entre el motor y el juego. En términos prácticos, esta línea no es perfectamente clara. Por lo menos el conocimiento del juego específicamente invariablemente se filtra a través de la capa fundamental del juego y a veces incluso se extiende en el núcleo de la propio motor.

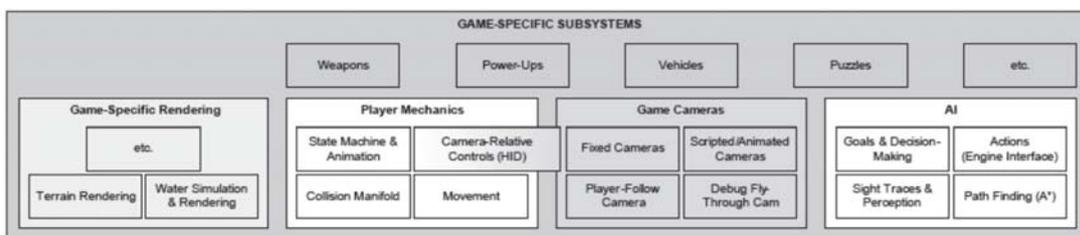


Fig. 3.26 Subsistemas específicos del juego

3.6.29 HERRAMIENTAS Y ACTIVOS DE INFORMACIÓN

Cualquier motor de juego debe ser alimentado con una gran cantidad de datos, en forma de activos de juego, archivos de configuración, scripts, etc. La figura 21 representa algunos de los tipos de activos de juego se encuentran típicamente en los motores de juego modernos. Las flechas color gris oscuro muestran cómo los flujos datos de las herramientas utilizadas para crear la fuente original de la ruta de activos a través del propio motor del juego. Las flechas de color gris claro delgadas muestran cómo los diferentes tipos de activos se refieren o utilizan otros activos.

3.6.30 HERRAMIENTAS DE CREACIÓN DE CONTENIDO DIGITAL

Los juegos son aplicaciones multimedia por naturaleza. los datos de entrada de un motor de juego vienen en una amplia variedad de formas, desde datos de malla en 3D a los mapas de bits de textura de datos de la animación a archivos de audio. Todos estos datos de origen deben ser creados y manipulados por los artistas. Las herramientas que los artistas utilizan se denominan aplicaciones de creación de contenido digital (DCC).

Una aplicación de DCC es generalmente dirigido a la creación de un particular, tipo de datos - aunque algunas herramientas pueden producir múltiples tipos de datos. Por ejemplo, Autodesk Maya y 3ds Max son frecuentes en la creación de animación en 3D. Photoshop de Adobe y sus productos están dirigidos a creación

y edición de mapas de bits (texturas). SoundForge es una herramienta popular para la creación de clips de audio. Algunos tipos de datos del juego no pueden ser creados utilizando una plataforma de aplicaciones DCC. Por ejemplo, observe la fig 3.27 la mayoría de los motores de juego proporcionan un editor personalizado para el tendido de los mundos del juego. Aun así, algunos motores no hacen uso de herramientas preexistente para el diseño del ambiente del juego. algunos equipos de juegos utilizan 3ds Max o Maya como un herramienta de diseño de ambiente, con o sin plug- ins personalizados para ayudar al usuario. Dicho esto, herramientas deben ser relativamente fácil de usar, y que absolutamente deben ser fiables , si un equipo de juego va a ser capaz de desarrollar un producto muy pulido de manera oportuna.

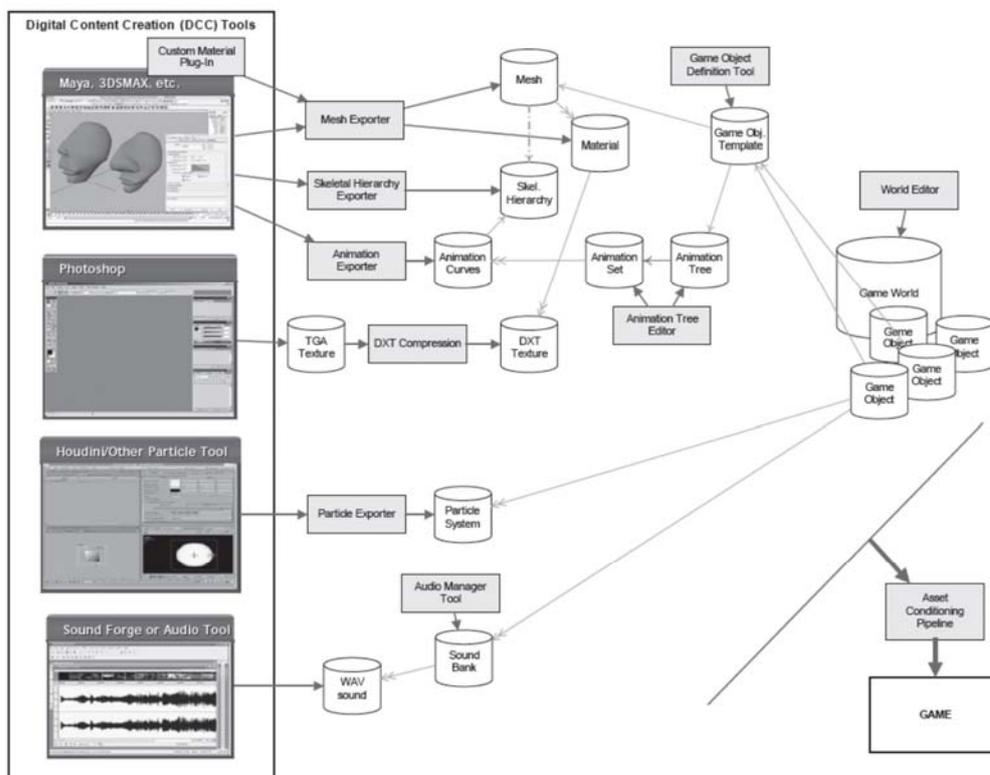


Figure 1.31. Tools and the asset pipeline.

Fig. 3.27 Herramientas y activos de información

3.6.31 CONDICIONAMIENTO DE ACTIVOS DE INFORMACIÓN

Los formatos de datos utilizados por aplicaciones de creación de contenido digital (DCC) son raramente adecuados para su uso directo en el juego. Hay dos razones principales para esto.

1. El modelo de aplicación DCC en memoria de datos es por lo general mucho más compleja de lo que requiere el motor del juego. Por ejemplo, Maya almacena un gráfico acíclico dirigido (DAG) un nodos de escena, con una compleja red de

interconexiones. Almacena un historial de todos los cambios que se han realizado en el fichero. Representa la posición, la orientación y la escala de cada objeto en la escena como una jerarquía llena de transformaciones 3D, descompuesto en componentes de traducción, rotación, escala, y de corte. un motor de juego normalmente sólo necesita una pequeña fracción de esta información en Para hacer el modelo en el juego.

2 . los Formatos de archivos de la aplicación DCC son a menudo demasiado lento para leer en tiempo de ejecución , y en algunos casos se trata de un formato propietario cerrado.

Por lo tanto , los datos producidos por una aplicación de DCC suele exportar a un formato estándar más accesible o un formato personalizado, para su uso en el juego .

Una vez que los datos se han exportado desde la aplicación DCC, que a menudo es preciso seguir procesados antes de ser enviado al motor de juego . Y si un estudio de juegos envía su juego a más de una plataforma , los archivos intermedios pueden ser procesados diferentemente en cada plataforma. Por ejemplo , los datos en 3D podrían ser exportados a un formato intermedio , tales como XML o un simple formato binario. Entonces podría ser procesado para combinar mallas que utilizan el mismo material , o se separan mallas que son demasiado grandes para el motor de digerir . La malla de datos pueden ser organizadas y se empaquetan en una imagen de memoria adecuada para en carga en la plataforma de hardware específico . los activos de información de la aplicación DCC para el motor de juego es a veces llamado el activo de información acondicionado. Cada motor de juego tiene esto de alguna forma.

3.7 DATOS DE MODELOS/MALLAS EN 3D

La geometría visible que se ve en un juego se compone generalmente de dos tipos de datos.

3.7.1 GEOMETRÍA DE PATRONES (BRUSH GEOMETRY)

Geometría de patrones se define como una colección de templates, cada uno de los cuales es definido por múltiples planos. Las brochas suelen crearse y editarse directamente en el editor del ambiente de juego. Esto es lo que algunos llaman enfoque "de la vieja escuela" a la creación de la geometría renderizable, pero todavía se utiliza.

Pros:

- rápida y fácil de crear;
- accesible para los diseñadores del juego, a menudo es usado para "bloquear" un nivel de juego para fines de prototipos;
- puede servir tanto de los volúmenes de colisión y como geometría renderizable.

Contras:

- baja resolución - dificultades para crear formas complejas;
- no puede apoyar objetos articulados o personajes animados.

3.7.2 MODELOS 3D (MALLAS)

Para los elementos detallados de escena, modelos 3D (también conocidos como mallas) son geometría de brocha superior. Una malla es una forma compleja compuesta de triángulos y vértices (Una malla también puede ser construido a partir de quads o de subdivisión de superficies de orden superior. Pero en el hardware de gráficos de hoy en día, que es casi exclusivamente orientada a la prestación de triángulos rasterizadas , todas las formas deben eventualmente ser traducido en triángulos antes de su representación). Una malla suele tener uno o más materiales aplicados a la misma, con el fin de definir propiedades de la superficie visual (color, reflectividad, baches , texturas difusas, etc.) El término " Malla " para referirse a una sola forma renderizable , y "modelo" para referirse a un material compuesto del objeto que puede contener varias mallas, además de datos de animación y otros metadatos para su uso por el juego.

Las mallas se crean normalmente en un paquete de modelado 3D como 3ds Max, Maya o Imagen Soft. Relativamente una nueva herramienta llamada ZBrush de alta resolución permite ultra-mallas para ser construido de una manera muy intuitiva y luego se degradan a un modelo de baja resolución con mapas normales para aproximar la alta frecuencia del detalle .

Los exportadores deben ser escritos para extraer los datos de las herramientas de contenidos digitales (DCC) (Maya , Max, etc) y guarda en el disco en una forma que es aceptada por el motor. Las aplicaciones DCC proporcionan una gran cantidad de estándar o formatos de exportación semi - estándar , aunque ninguno se adapta perfectamente para el desarrollo de juego (con la posible excepción de COLLADA). Por lo tanto, los equipos de juego a menudo crean costumbre en formatos de archivo y exportadores para ellos.

3.7.3 DATOS DE ANIMACIÓN ESQUELÉTICA

Una malla esquelética es un tipo especial de malla que se une a una jerarquía esquelética para fines de articulación de animación. Una malla de este tipo es a veces llamado piel, debido a que forma la piel que rodea el esqueleto subyacente invisible. Cada vértice de una malla del esqueleto contiene una lista de índices que indica conjunta(s) en el esqueleto. Un vértice generalmente también incluye un juego de pesas conjuntas, especificando la cantidad de influencia en cada articulación que tiene el vértice.

A fin de hacer una malla esquelética, el motor del juego requiere de tres distintos tipos de datos.

1. la malla en sí,

2. la jerarquía esqueléticas (nombre enlazados, relaciones entre padres e hijos y la base de pose del esqueleto se encontraba original a la malla), y

3. uno o más clips de animación, que especifican cómo las articulaciones deben moverse con el tiempo.

La malla y el esqueleto son exportados desde la aplicación DCC como un solo fichero de datos. Sin embargo, si varias mallas están obligados a un solo esqueleto, entonces es mejor exportar el esqueleto a un archivo diferente. Las animaciones generalmente se exportan individualmente, permitiendo sólo aquellas animaciones que se utilizan para ser cargadas en la memoria en cualquier momento dado. Sin embargo, algunos motores de juego permiten un banco de animaciones para ser exportado como un fichero, y algunos incluso comprimir malla, esqueleto, y animaciones en un solo fichero.

Una animación del esqueleto sin optimizar está definida por una corriente de matriz 4×3 , tomadas a una frecuencia de al menos 30 fotogramas por segundo, para cada uno de los articulaciones en un esqueleto (de los cuales hay a menudo es 100 o más) . Así, los datos de animación es de por sí mucha memoria. Por esta razón, los datos de animación son casi siempre almacenados en un formato muy comprimido. Esquemas de compresión varían de un motor a otro, y algunos son de propiedad. No existe un formato normalizado para los datos de animación del juego terminado.

3.7.4 DATOS DE AUDIO

Clips de audio por lo general se exportan desde Sound Forge o alguna otra herramienta de producción de audio en una variedad de formatos y en un número de muestreo de datos a diferentes tasas. Archivos de Audio puede haber en mono, estéreo, 5.1, 7.1, o de otro tipo multicanal configuraciones. Archivos Wave(.Wav) son comunes, pero otros formatos como PlayStation ADPCM (vag. Y. Xvag) también son comunes. clips de audio están a menudo organizados en bancos paraia los fines de la organización, facilitar la carga en el motor y la transmisión.

3.7.5 DATOS PARA SISTEMAS DE PARTÍCULAS

Los juegos modernos hacen uso de complejas efectos de partículas. Estos están escritos por artistas que se especializan en la creación de efectos visuales. Herramientas de terceros, tales como Houdini, permiten efectos de calidad, sin embargo, para la mayoría de juegos Los motores no son capaces de hacer que toda la gama de efectos que se pueden crear con Houdini. Por esta razón, muchas compañías de juegos acostumbra herramientas de edición de efectos de partículas, lo que expone sólo los efectos que el motor realmente apoya. Una herramienta personalizada también puede dejar que el artista vea los efectos exactamente como aparecerá en el juego.

3.7.6 DATOS DE AMBIENTE DE JUEGO Y EDITOR DE AMBIENTE

El ambiente de juego es donde todo en un motor de juego viene junto. No hay editores de ambiente de juegos disponibles en el mercado (es decir, el ambiente de juego equivalente de Maya o Max). Sin embargo, un número de motores de juego comercialmente disponibles proporcionan buenos editores ambiente.

- Algunos variante del editor del juego RADIANT es utilizado por la mayoría de los motores de juego basado en la tecnología de Quake;
- El motor fuente de Half-Life 2 proporciona un editor de ambiente llamado Hammer;
- UnrealEd es editor de ambiente del motor Unreal Engine. Esta potente herramienta también sirve como gestor de activos para todos los tipos de datos que el motor puede consumir.

Escribir un buen editor de ambiente es difícil, pero es una parte muy importante de cualquier buen motor de juego.

3.8 ALGUNOS ENFOQUES PARA LA ARQUITECTURA DE HERRAMIENTAS

Una Suite de herramientas de un motor de juego puede ser organizada en cualquier forma. algunas herramientas pueden ser piezas independientes de software, como se muestra en la Fig. 3.28 algunos herramientas pueden ser construidas en la parte superior de algunas de las capas más bajas utilizadas por el motor de ejecución, como ilustra la Fig. 3.29 Algunas herramientas pueden ser incorporados en el juego en sí. Por ejemplo, Quake - y los juegos basados en Unreal ambos cuentan con una consola en el juego que permite a los desarrolladores y modificadores para depurar y configurar comandos mientras se ejecuta el juego.

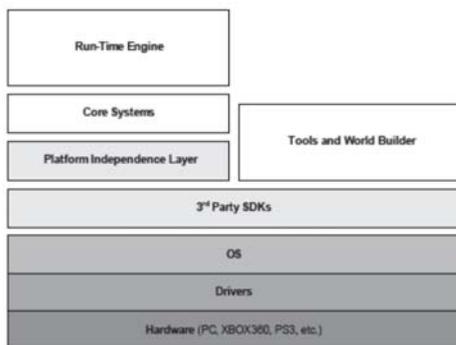


Fig. 3.28 Arquitectura de herramientas independiente.

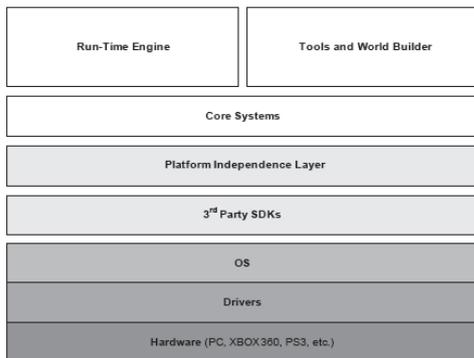


Fig. 3.29 Herramientas integradas en un marco de trabajo compartido con el juego.

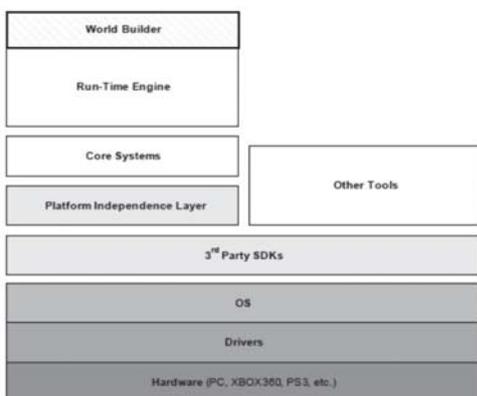


Fig. 3.30 Arquitectura de herramientas del motor Unreal.

Como un ejemplo interesante y único, el de ambiente Unreal y el administrador de activos, UnrealEd, está integrada en el motor del juego en ejecución. Para ejecutar el editor, se ejecuta el juego con un argumento de línea de comandos de "editor". Este estilo arquitectónico único se representa en la fig. 3.30 Permite las herramientas para tener acceso total a toda la gama de estructuras de datos utilizadas por el motor y evita un problema común de tener que tener dos representaciones de todas las estructuras de datos - uno para el motor ejecución y una para las herramientas.

También significa que se ejecuta el juego desde el editor es muy rápido (porque el juego en realidad ya se está ejecutando). La edición de un juego en vivo, es una característica que normalmente es muy difícil de poner en práctica, se pueden desarrollar con relativa facilidad cuando el editor que es una parte del juego. Sin embargo, un diseño de editor en el motor como éste tiene su cuota de problemas. Por ejemplo, cuando el motor se bloquea, las herramientas se convierten en inutilizable también. Por lo tanto un estrecho acoplamiento entre el motor y la creación de herramientas pueden ocasionar la reducción de la velocidad de producción.

3.9 KIT PARA DESARROLLO DE SOFTWARE (SOFTWARE DEVELOPMENT KIT)

Estos productos son conjuntos de herramientas diseñadas para crear software.

Estas suites de herramientas se llaman kit para desarrollo de software, existen para el desarrollo de muchas aplicaciones, en este caso de videojuegos, las marcas disponibles en el mercado son: Unreal development Kit, Cryware development kit, adobe gaming, gamasutra, gamestreamer, Korona sdk, autodesk gameware, autodesk scaleforms, y la lista sigue.

Continuamente nuevas marcas se incorporan al mercado y otras desaparecen, cabe mencionar que estas aplicaciones son plantillas de desarrollo con las cuales está organizado el videojuego.

La mayoría de las veces estos kits son licenciados tanto por usuario, por proyecto. Antes de elegir la marca correcta hay que tomar en cuenta detalles de planificación y preproducción esto con la finalidad de tener el mejor producto para las expectativas que deseamos obtener del SDK elegido.

Estos criterios mencionados son los más importantes para elegir el sdk correcto, muchas marcas argumentan que su software es el mejor y que nadie los iguala pero ya durante la marcha es donde se encuentran los tropiezos dados por la carencia de información previa a la elección de compra de licencias. Los gastos que genera un error son pérdidas de tiempo que se verán reflejadas en retrasos o productos defectuosos que continuamente requerirán actualizaciones o parches correctores.

La ingeniería del software para videojuegos comprende la ejecución en tiempo real de los productos de desarrollo para que el resultado sea el deseado.

Los productos de terceros como los middleware son para crear subproductos que el sdk para juegos no puede ofrecer, estos middleware son como MAYA, 3DMAX, CINEMA, etc.

Y en épocas donde la evolución de la tecnología como hoy la vivimos donde un producto se hace obsoleto incluso en horas, lo que le da valor es saber usar esas herramientas llamadas SDKs, (durante los últimos ajustes a este trabajo de tesis, se lanzó al mercado UNREAL ENGINE 4).

3.9.1 UNREAL ENGINE 3

Este documento es una visión general del proceso de planificación, producción y publicación de un juego usando Unreal Engine 3. Comenzando en un nuevo proyecto utilizando Unreal Engine 3 puede ser una tarea desalentadora. El motor es complejo y saber cómo empezar, obviamente así como la forma de pasar de una fase a otra no siempre es inmediata. Este resumen está aquí para ayudarle a lo largo del camino, usted apunta en la dirección correcta en cuanto a lo que hay que hacer para empezar, donde la documentación se puede encontrar para los distintos aspectos del ciclo de producción, y dónde ir una vez que la producción de su juego arranca.

Las secciones se presentan en orden cronológico a partir de la planificación y terminando con la publicación. En el camino, cada sección tendrá un breve resumen de lo que está involucrado en esa tarea o aspecto del desarrollo , junto con enlaces a documentos más detallados que cubren áreas importantes del aspecto en cuestión.

Esto es de ninguna manera una lista completa o exhaustiva de todas y todos los documentos en la red de desarrolladores de Unreal , y usted puede encontrar que su juego en particular no sigue los pasos o aspectos del desarrollo que figuran aquí , precisamente . Esta es la intención de darle una idea de lo que está involucrado y proporcionar una base para la realización de las tareas importantes que se requieren para desarrollar un juego con Unreal Engine 3.

3.9.2 PLANIFICACIÓN Y PREPRODUCCIÓN

Antes de que comience la producción, de hecho incluso antes de decidir Unreal Engine es el sistema más adecuado para su juego, hay algunos aspectos importantes y resolverlos con el fin de evitar desaires.

¿Qué características requeridas por son soportadas por Unreal Engine 3?

¿Los requisitos mínimos del Unreal Engine 3 se ajustan a su mercado objetivo?

¿Tiene trabajadores suficientes y suficientemente calificados para satisfacer su calendario de lanzamientos?

¿Tiene información de contenidos para Unreal Engine 3 se adapte a sus necesidades?

¿Qué pautas legales tendrán que seguir?

¿Qué puntos de apoyo proporciona Epic para Unreal Engine 3?

3.9.3 EL MOTOR Y CONFIGURACIÓN

Ya sea para fines de prueba o cuando la producción real está lista para comenzar, el primer paso es adquirir la versión adecuada del motor: Unreal Engine 3 o Unreal Development Kit. Para la obtención de Unreal Engine 3, el proceso puede

parecer difícil para los usuarios de primera vez , pero EPIC proporciona una guía detallada para cada paso en el camino. Si usted tiene un conocimiento práctico de los depósitos de fuerza, esto debe ser un proceso relativamente indoloro. Además , Epic ofrece una herramienta personalizada para actualizar de forma rápida a la nueva QA. La adquisición de Unreal Development Kit es un proceso rápido y sencillo que sólo requiere que descargue el instalador e instalar la aplicación.

Una vez que haya adquirido la versión apropiada del motor , hay alguna configuración adicional necesaria antes de que pueda empezar a trabajar en su propio proyecto de juego personalizado. Los concesionarios tienen un proceso de instalación más complicada , ya que deben crear un nuevo proyecto en la solución de motor y configurar Visual Studio. Los usuarios UDK sólo tienen que crear un nuevo proyecto de guión y el entorno de codificación de su elección. Una vez más , se proporcionan guías detalladas para el proceso de configuración para ayudarle a lo largo del camino.

3.9.4 UNREALSCRIPT Y BASE DE CODIGOS UE3

Conocer y comprender el lenguaje UnrealScript así como el código base del motor, tanto nativos como el script, es extremadamente importante. Esto va a mejorar la capacidad de aprovechar toda la potencia del motor y hacer que la implementación de nuevas características mucho más eficiente, ya que no constantemente busca a través de la documentación y perdido por código tratando de encontrar qué clase de extender o cuando se implementa una función.

Por supuesto, la comprensión y el conocimiento de un código base del tamaño de Unreal Engine 3 no es realmente una expectativa práctica. Sin embargo, usted debe ser capaz de captar una visión general de los principales aspectos y las clases junto con una sólida comprensión de los topicos a su área de especialización en un período razonable de tiempo.

La Introducción a: la página de programación contiene información y enlaces a documentos adicionales sobre estos aspectos para que pueda empezar aprender a programar con Unreal Engine3.

3.9.5 ELEMENTOS DE CREACIÓN DE GAMEPLAY

Creación de elementos de juego personalizados es una parte integral de conseguir el juego. Elementos de juego en este sentido se refieren a elementos como el tipo de juego (o reglas de juego), el jugador (incluyendo la cámara y de entrada), y el HUD / UI (interfaz de Heads Up Display/usuario). Por supuesto, estos no son más que una fracción de los elementos de juego a medida de su equipo técnico pondrá en práctica durante la creación de un nuevo juego, pero éstos constituyen la base del marco de juego y son comunes a todos los juegos y, como tal, requieren especial atención.

3.9.6 CREACIÓN DE NIVEL

Los niveles o mapas, son colecciones de artículos que conforman el mundo o ambiente en el que el juego se desarrolla. La construcción de ambientes y mundos para su juego es, obviamente, un aspecto muy importante del ciclo de producción. El proceso de creación de niveles implica una amplia colección de temas, pero en su base que consiste en colocar diversos tipos de actores en un mapa dentro de Editor Unreal, la suite de edición incluida con Unreal Engine 3.

Como el diseño de niveles implica un amplio abanico de disciplinas, puede haber varias personas que trabajan en un solo nivel en un momento dado. Puede haber una persona a cargo del nivel en su conjunto, mientras que una o más personas están a cargo de eventos de secuencias de comandos que utilizan el sistema de scripting visual Kismet y otros se pueden controlar los eventos de animación y secuencias cinemáticas utilizando Matinee. El sistema de transmisión en el nivel de Unreal Engine 3 hace posible todo esto en un solo nivel persistente puede contener múltiples sub-niveles, llamado niveles de streaming. Estos sub-niveles pueden contener el scripting Kismet y secuencias Matinee mantener las cosas organizadas y permite que varias personas estarán trabajando en el mismo nivel general, al mismo tiempo.

3.9.7 CREACIÓN DE CONTENIDOS

Unreal Engine 3 ofrece dos herramientas personalizadas dentro UnrealEd así como activos de información para traer contenido desde aplicaciones externas. Estas características dan a los artistas la libertad para crear contenido que traerá el entorno del juego a la vida. Desde mallas estáticas para crear el medio ambiente, edificios, estructuras, y decoraciones a las mallas del esqueleto utilizados para personajes, armas y vehículos. De los materiales complejos, dinámicos a efectos de partículas increíbles, Unreal Engine 3 hace que la creación de contenido simple y eficiente.

Las animaciones se crean en aplicaciones externas, como los programas de animación en 3D, como Autodesk Maya y 3dMax. O bien, algunos prefieren utilizar herramientas de animación dedicados, tales como Autodesk MotionBuilder. Todos estos programas ofrecen la posibilidad de manipular fotogramas clave de la rotación y la posición de los huesos o las articulaciones de un esqueleto con el tiempo. La selección de fotogramas que componen la animación puede ser exportado desde la aplicación externa y se importa a Unreal Editor como secuencias de animación y se guarda en juegos de animación. Cuando se aplica a las mallas del esqueleto con esqueletos subyacentes a juego, que también han sido importados desde aplicaciones

FBX Animación Pipeline - Transferencia animaciones esqueléticas de las aplicaciones 3D para UnrealEd.

FBX Morph Target Pipeline - Transferencia de objetivos morph de las aplicaciones 3D para Unreal Editor.

Skeletal Mesh, Importación de Nivel de detalle - Cómo importar y utilizar malla esqueléticas LD.

Morph Objetivos - Cómo crear estados de animación de vértices de mallas esqueléticas.

Clasificación de pelo, Translucent - Clasificación de triángulos para resolver los problemas en la prestación del cabello translúcido.

ActorX - ActorX es una herramienta para la exportación al formato PSK, que Unreal Editor puede importar.

Tutorial de Importación de Textura - Cómo texturizar en Unreal Engine 3.

Creación Maps - Consejos prácticos y los recursos para la creación de mapas.

Creación de Shade Maps - Guía sobre la creación de mapas de sombra utilizando 3dsMax

3.9.8 COMPILACIÓN Y PRUEBAS

En numerosas ocasiones a lo largo del proceso de desarrollo, tendrán que ser construido y compilado con el fin de ejecutar el juego para probar el motor y scripts. Este proceso se ha hecho tan eficiente como sea posible a través de la adición de varias herramientas para Unreal Engine 3. Construcción y compilación es manejado por el sistema de generación de Unreal y Unreal Frontend. Estos automatizan gran parte de lo que solía requerir una intervención manual.

Las pruebas y la depuración puede ser un proceso tedioso, pero hay herramientas incluidas que puede ayudar a localizar los errores, los problemas de rendimiento, y ajustar el juego de una manera muy eficiente.

3.9.9 EMBALAJE Y DISTRIBUCIÓN

La preparación y la creación de una versión final de juego para la distribución es el paso final en el proceso de creación de un juego con Unreal Engine 3 o Unreal Development Kit. En los casos en que el espacio de almacenamiento es una preocupación, es importante asegurarse de que sólo se está incluyendo que el contenido que se utiliza realmente en el juego. Usted también querrá asegurarse de que usted cumpla con cualquier acuerdo de licencia que existen con cualquier software que se incluye con el juego o dependencias de software de terceros a su juego puede tener.

3.9.10 PROGRAMACIÓN DE GAMEPLAY

Programación de elementos de juego se realiza principalmente utilizando el lenguaje de programación UnrealScript. Nuevas clases personalizadas se crean que pueden extenderse desde una de las clases existentes en el motor que proporcionan una funcionalidad básica, genérica que se hace a mano, las clases personalizadas pueden adaptarse a las necesidades del juego. Unreal Engine 3 contiene sistemas genéricos para la creación de jugadores, NPCs, reglas de

juego, perspectivas de cámara, artículos de colección y armas, vehículos, etc, que se pueden fabricar para adaptarse a cualquier tipo de juego. Por supuesto, puede encontrar que necesita para poner en práctica los elementos para su juego que simplemente no existen en ninguna forma en el motor existente. UnrealScript ofrece una gran flexibilidad en estos casos por lo que es posible crear prácticamente cualquier cosa que sea necesaria para hacer su trabajo de campo.

3.9.11 ANIMACIÓN

El trabajo del animador es extremadamente importante en el desarrollo del juego moderno. Los personajes hacen una gran parte de cualquier juego de vídeo y , a menudo, los personajes son seres humanos u otras criaturas conocidas. cómo se mueven y se ven en la pantalla puede hacer o deshacer la experiencia de inmersión del jugador. El ojo humano tiene una extraña habilidad para detectar fallas en el movimiento de los seres vivos, y en especial el movimiento de los seres humanos. Esto hace que el trabajo del animador aún más difícil, ya que no sólo tienen que trabajar dentro de las limitaciones del motor en términos de consumo de memoria y los límites técnicos de lo que es posible , sino que también tienen que hacerlo mientras hace movimientos que pueden engañar al ojo humano en la creencia de que son naturales y reales.

3.10 TÓPICOS DE DESARROLLO

3.10.1 INTELIGENCIA ARTIFICIAL Y NAVEGACIÓN

Creación interesante y útil AI se basa en esas entidades la posibilidad de navegar de forma inteligente y realista en el ambiente Tradicionalmente, Unreal Engine se ha apoyado en un sistema basado en el punto de referencia que se utiliza para crear una red de ruta. Con la adición del sistema basado en malla de navegación, el mundo puede ser representado de una manera mucho más eficiente y precisa. Esto significa que la navegación se ha convertido en mucho más eficiente en términos de memoria necesaria para almacenar las estructuras de datos y es capaz de producir mucho más realista el movimiento. Esta sección contiene documentación sobre movimiento entidades controladas por la IA de un lugar a otro utilizando uno de los sistemas de navegación, waypoints o navegación de malla, siempre en el Unreal Engine 3.

3.10.2 ANIMACIÓN

Las animaciones ayudan a crear la ilusión de movimiento complejo realizado por mallas esqueléticas. Las animaciones ayudan a sentirse vivo o tener algún sentido de la personalidad mediante la definición de la forma en que los personajes en pie, personajes hablan y actúan en el camino. A partir de estos comportamientos, los jugadores sienten que el avatar virtual es un poco más real que sólo gráficos en una pantalla.

Unreal Engine 3 emplea una amplia gama de tecnologías y herramientas para hacer más fácil la exportación de animaciones de su paquete de creación de

contenido 3D favorito, de mezclar animaciones raws para crear nuevas animaciones dentro de Unreal Editor.

3.10.3 MARCO APEX

APEX es un framework desarrollado por NVIDIA que proporciona ambas herramientas para la creación de contenidos y bibliotecas de tiempo de ejecución que permiten a los artistas y desarrolladores crear simulaciones físicas complejas con más facilidad. La integración Unreal Engine 3 de APEX incluye realismo que se puede utilizar para la simulación de banderas, pancartas, cortinas, ropa para los personajes. Además, la integración de APEX incluye la capacidad de simular la destrucción de los objetos físicos para la creación de ambientes destructibles.

3.10.4 AUDIO

El sonido es un aspecto que a veces se pasa por alto en el desarrollo del juego, pero es sumamente importante para la creación de entornos inmersivos y creíbles. Desde sonidos ambientales en el nivel de los sonidos interactivos de vehículos o armas a hablado de diálogo de los personajes, el audio en el juego puede hacer o deshacer la experiencia del usuario. Hacer que el audio en el juego en realidad suena como debe sonar puede ser una tarea difícil. El Sistema de audio del Unreal 3 Engine proporciona herramientas y características para moldear los sonidos en el juego para darles la sensación deseada. Esto es importante porque significa que una versión limpia del sonido se puede producir una vez en una aplicación externa, importada, y luego hace a mano dentro del motor para crear el resultado adecuado.

3.10.5 COMMANDLETS (COMANDOS)

Commandlets son programas de línea de comandos que se ejecutan dentro del entorno de Unreal Engine. Ellos son los más utilizados para hacer cambios masivos en el contenido, iterar sobre contenido para obtener información sobre el mismo, o como un mecanismo de la unidad de pruebas. El motor contiene muchos commandlets por defecto y nuevos commandlets se puede añadir para realizar funciones adicionales.

Commandlets se ejecutan pasando el nombre del commandlet junto con los parámetros necesarios en el juego en la línea de comandos, o bien ejecutando el juego desde un símbolo del sistema en Windows o mediante el destino de un acceso directo al ejecutable principal del juego (es decir UDK. exe).

FACEFX FACIAL ANIMATION (ANIMACIÓN FACIAL FACEFX)

FaceFX es una solución de animación facial desarrollado por OC3 Entertainment y se integra con el Unreal Engine 3.0. Todos los usuarios de Unreal Engine 3.0 licenciatarios y UDK tienen una licencia para FaceFX. Esta herramienta le permite crear secuencias de animación complejas, animaciones específicamente faciales, que están sincronizados con el audio. Estos pueden ser usados para escenas

detalladas y cinemáticas o simplemente secuencias jugado durante el juego para proporcionar realismo aumentado dramáticamente durante el habla de diálogo y otros.

3.10.6 FOLIAGE (FOLLAJE)

En el pasado, el Unreal Engine 3 ha tenido varios sistemas heredados, incluidos los volúmenes de follaje otoñal y decoraciones del terreno, pero eran bastante inflexibles, no realizar especialmente bien y no interactuar bien con Unreal Lightmass. El nuevo sistema de Foliage permite pintar rápidamente o editar y borrar conjuntos de mallas sobre Paisaje, StaticMeshes y terreno legado. Estas mallas se agrupan automáticamente en lotes que se representan mediante la creación de instancias de hardware, lo que significa muchos casos se pueden representar con sólo una única llamada de sorteo.

3.10.7 ANIMACIÓN DE MATERIALES Y VERTEX

Count instance Muestra el número de instancias de esta malla actualmente colocada en el nivel de transmisión de corriente

Cluster Count Muestra el número de grupos utilizan para representar las instancias de esta malla en el nivel de transmisión actual. Grupos de casos se prestan juntos en una sola llamada empate, llamado un clúster, para mejorar la eficiencia de procesamiento. La asignación de grupos se determina automáticamente en función del número de instancias que ya están en el clúster y el radio de la agrupación en el espacio universal . Los dos parámetros siguientes se pueden ajustar esto.

Instancias Por Cluster, establece el número máximo de instancias por clúster. todas las instancias en un solo clúster siempre se prestan en una sola llamada empate con el número total de triángulos iguales a los polígonos por malla multiplicado por las instancias por racimo. El número ideal depende de la cantidad de polígonos de la malla.

Cluster Radio Este es el tamaño máximo de un grupo puede crecer ante las instancias se asignan a un nuevo grupo. La reducción de este número se incrementa el número de racimos pero mejora la oclusión como el clúster tiene un cuadro delimitador más pequeño.

Cull Distance Start Esta es la distancia en unidades del mundo en el que los casos comenzarán a desaparecer. Tenga en cuenta que esto debe ser establecido en el material.

Cull Distance End, Esta es la distancia en unidades del ambiente en el que los casos serán completamente vaciados. Si el material no está configurado para desaparecer instancias individuales, grupos enteros desaparecerán y reaparecen juntos.

3.10.8 ILUMINACIÓN

Cada instancia de malla individual tiene su propia sombra y/o mapa luz generada por Lightmass , y estos son de azulejos juntos para cada lote lote precomputed puede apoyar precomputed . Hay varias opciones en la StaticMesh que deben revisarse para iluminación precomputed para operar bien con follaje de instancia. Lightmass es menos indulgente cuando se generan mapas de sombras para mallas de instancia , y una configuración incorrecta podría llevar a las mallas negras después de la reconstrucción de la iluminación.

Índice de coordenadas de mapas de luz - se debe establecer en un canal UV válido que tenga una desvenolver UV único. Función "Generar Unique UVs " La estática del acoplamiento del editor puede generar rápidamente un desembalaje único.

Resolución lightmap - esto debe ser un número reducido lo suficiente para que todo el mapa de sombras para las instancias en un solo grupo (por defecto 100) puede ser de azulejos , sin exceder el máximo de resolución de la textura (4094x4096) .

Hubo algunos errores en el shadowmap y asignación lightmap para el follaje en las versiones del motor antes de abril de 2012, que causaron algunos casos totalmente en sombra - para ser encendido y algunos casos sin mácula a quedar sombreados incorrectamente. Estos errores han sido corregidos para la versión de abril 2012, y también una advertencia se ha añadido para alertar al usuario si se supera la resolución máxima lightmap.

3.11 ELEMENTOS DEL GAMEPLAY

Los Elementos de juego constituyen la mayor parte del juego real desde la perspectiva del jugador, la mecánica y las reglas del juego , la forma en que el jugador controla el personaje en el juego, los elementos que pueden ser utilizados por o en contra del jugador. Estos son todos los aspectos del juego. El Juego cubre un amplio espectro de temas y es muy dependiente del tipo de juego que se está creando.

Para un simple juego de puzzle , juego podría consistir en :

Los controles usados por el jugador para interactuar y manipular las piezas del rompecabezas

Las piezas del rompecabezas y cómo interactúan entre sí después de un movimiento (por ejemplo, ciertos movimientos pueden dar lugar a "combos" que premian al jugador)

Cualquier "power-ups" el jugador puede ser capaz de obtener.

Las reglas del juego y cómo se determina si el jugador gana o pierde.

Un juego aparentemente más complejo, como un juego de rol, puede tener elementos de juego que consiste en:

El personaje del jugador, controles y estadísticas (nivel, la salud, etc)

Los productos que el jugador puede recoger y utilizar mejorar las estadísticas o realizar acciones (pociones , hechizos , armas , etc)

NPCs que el jugador puede interactuar con una manera u otra (enemigos, gente del pueblo, jefes , etc) Las reglas del juego y la forma en que se determina el juego . Unreal Engine 3 contiene sistemas genéricos para la creación de jugadores, NPCs, reglas de juego, perspectivas de cámara, artículos de colección y armas, vehículos, etc, que se pueden fabricar para adaptarse a cualquier tipo de juego.

3.11.1 INPUT / OUTPUT (ENTRADAS / SALIDAS)

Entrada y salida en Unreal Engine 3 se consigue de tres maneras diferentes. El primer método es mediante el uso de archivos de configuración; que se extienden al sistema de unión clave. El segundo método es mediante el uso de paquetes TCP/IP para leer y escribir información. El tercer método es a través del uso de la serialización y deserialización de objetos de Unreal. Con estos tres sistemas, se pueden crear partidas guardadas, leer noticias de un servidor web, enviar datos de perfil a un servidor web o guardar datos arbitrarios en alguna parte.

3.11.2 KISMET VISUAL SCRIPTING (ESCRITURADO VISUAL KISMET)

Eventos de Scripting dentro de los niveles del juego es un aspecto muy importante de la creación de entornos interesantes e interactivos para el jugador. No sería tan eficiente si se exige a los programadores para implementar las ideas de diseño de niveles. El sistema de scripting visual Kismet pone el poder en manos del diseñador de niveles para crear eventos que van de lo simple a extremadamente complejo. El concepto es simple: las acciones de diseño de Enlaces a ejecutar cuando se producen determinados eventos en el juego. Los resultados son un sistema flexible, de gran alcance que se puede utilizar para cualquier cosa de juego de prototipos o modos de juego, incluso toda la creación de simples actores móviles, tales como puertas.

3.11.3 LANDSCAPE (PAISAJE)

Landscape en el nuevo sistema de terreno de Unreal Engine 3, actualmente en desarrollo. Se trata de un sistema basado en el mapa de alturas que hace uso de las características de la transmisión de contenido de Unreal Engine 3 para permitir un uso más eficiente de los recursos, mientras que al mismo tiempo que proporciona una mayor funcionalidad y características sobre el terreno del sistema anterior disponible en el motor. Paisaje tiene por objeto sustituir el

sistema Age Terrain, que quedará obsoleto cuando Landscape haya alcanzado la paridad de características.

3.11.4 LEVEL STREAMING (NIVEL DE TRANSMISIÓN DE INFORMACIÓN)

La funcionalidad de transmisión de nivel de Unreal Engine 3 que permite cargar y descargar archivos de mapas en la memoria, así como alternar su visibilidad durante el juego. Esto hace que sea posible disponer de ambientes fragmentados en pedazos más pequeños para que sólo las partes relevantes del mundo están tomando los recursos y que se queden en ningún punto. Si se hace correctamente, esto permite la creación de grandes niveles, sin fisuras que pueden hacer que el jugador se sienta como si estuvieran jugando en un ambiente natural.

3.11.5 LIGHTING & SHADOWS (ILUMINACIÓN Y SOMBRAS)

Cuando se trata de la creación de entornos creíbles, la forma de la geometría en el nivel iluminado juega un papel muy importante. El ojo y el cerebro humanos esperan que la luz interactúe con las superficies, llenar una habitación o sombras proyectadas de una manera particular. Cualquier cosa que se desvíe de esto se puede romper con la experiencia de inmersión para el usuario. Sistema de iluminación del Unreal Engine 3 es muy flexible, lo que permite diferentes tipos de luces y sombras a todo ser utilizados en armonía para crear la apariencia adecuada para cualquier juego. El sistema de iluminación estática Lightmass facilita la configuración de ambientes enteros extremadamente fácil ya que la mayoría del trabajo es realizado por el proceso de generación de iluminación y luces dominantes que prevén sombras detalladas y sin sacrificar el rendimiento.

3.11.6 MATERIALS & TEXTURES (MATERIALES Y TEXTURAS)

Materiales y las texturas utilizadas dentro determinan la apariencia de las superficies dentro del ambiente. Ellos controlan no sólo el color de las superficies, sino también cómo cada superficie interactúa con la luz mediante la aplicación de un modelo de iluminación elegido y el modo de fusión. El modelo de iluminación determina cómo la luz interactúa con la superficie y el modo de fusión controla la forma en la superficie que se representa mezclas con el resto de la escena.

3.11.7 MATERIAL

Un material es un activo que se puede aplicar a una malla para controlar el aspecto visual de la escena. En general, cuando la luz de la escena incide sobre la superficie, el modelo de iluminación del material se utiliza para calcular la forma en que la luz interactúa con la superficie. Hay varios modelos diferentes de iluminación disponibles dentro del sistema de materiales de Unreal Engine 3 , pero el modelo de iluminación de Phong es podría llamarse la iluminación "regular". Cuando se realiza el cálculo de iluminación, varias propiedades de los materiales se utilizan para subir con el color final de la superficie. Estas

propiedades del material también se llaman entradas de material en el editor de materiales y son cosas como diffuseColor , emissiveColor , specularColor, SpecularPower, etc.

3.11.8 SPREAD (PROPAGACIÓN)

El color difuso de un material representa la cantidad de luz entrante reflejada por igual en todas las direcciones. Un valor de (1,1,1) significa que el 100 % de la luz entrante se refleja en todas las direcciones. Difusa no es dependiente por lo que tendrá el mismo aspecto desde cualquier ángulo, pero no depende de la normal en cada píxel. Color difuso sólo aparecerá cuando se ven afectados por una cierta iluminación no sombreada, por las escalas de la iluminación entrante.

3.11.9 EMISSIVE (EMISIÓN)

El color emisivo de un material representa la cantidad de luz que emiten los materiales, como si se tratara de una fuente de luz, aunque en realidad no se ilumina otras superficies en la escena. Desde emisiva está actuando como una fuente de luz, no se ve afectado por las luces o sombras. Emisivo a veces se utiliza para representar el término iluminación ambiental.

3.11.10 OPACITY (OPACIDAD)

Opacidad controla la cantidad de luz que pasa a través de la superficie de los materiales translúcidos. Esto significa superficies translúcidas como un filtro de escena que se ve a través de él. Un material con una opacidad de 1 de opacidad total, mientras que una opacidad de 0 significa que deja que pase toda la luz a través de él.

3.11.11 NORMAL

La propiedad normales representa la orientación de la superficie. Puede ser especificado por píxel con el uso de un mapa normal, y tiene un efecto tanto en la iluminación difusa y especular.

3.11.12 MATINEE & CINEMATICS (MATINÉ Y CINEMÁTICAS)

La herramienta de animación Matinee proporciona la capacidad para animar las propiedades de los actores a través del tiempo, ya sea para crear un juego dinámico o cinematográfico en el juego secuencias. El sistema se basa en el uso de pistas de animación especializados que pueden han fotogramas clave colocados para establecer los valores de ciertas propiedades de los actores en el nivel. El editor Matinee es similar a los editores no lineales utilizadas para la edición de vídeo, por lo que es familiar para los profesionales del vídeo.

3.11.13 NETWORKING & REPLICATION (RED Y REPLICACIÓN)

Replicación en Unreal Engine 3 permite que varios jugadores experimentar eventos juntos. Al utilizar el modelo cliente-servidor generalizada junto con UnrealScript es posible elaborar la experiencia multijugador.

Replicación garantiza que los datos en el cliente es suficiente para estimar lo que está sucediendo en el juego preciso, por lo que el jugador puede experimentar el juego y sus cambios en tiempo real. La replicación también se asegura de que los datos sean generados por el cliente y recibidos en el servidor de forma que el cliente está cambiando activamente la experiencia.

3.11.14 PARTICLES & EFFECTS (PARTÍCULAS Y EFECTOS)

Efectos dentro de un juego son importantes para crear la apariencia de un entorno de vida. Añaden la interacción, el movimiento y la variación que es esencial para la creación de mundos creíbles para el jugador. Sin ellos, es más difícil para el jugador a participar su imaginación y sumergirse en el juego en conjunto.

3.11.15 PERFORMANCE, PROFILING, AND OPTIMIZATION (DESEMPEÑO, PERFIL Y OPTIMIZACIÓN)

Hacer un juego que se ejecuta bien y se adapte con restricciones de memoria es un problema interesante que toca muchos sistemas y muchas disciplinas. Esta página es su puerta de entrada a un gran número de herramientas y técnicas en las cuales le ayudarán a ganar la batalla. Incluso con las mejores herramientas en el mundo, usted todavía tiene que inculcar la mentalidad de asegurarse que la gente está utilizando la mínima cantidad de ciclos de datos/cpu para crear lo que quieren. Las herramientas y las técnicas que se enumeran aquí le ayudará a hacer exactamente eso.

Para ello partimos del gran problema en una serie de problemas secundarios que nos permitirá centrarnos en cada área de la mayor parte posible de manera ortogonal posible. Esto permite a varias personas para trabajar en escala de forma independiente.

3.11.16 GENERAL MAINTENANCE (MANTENIMIENTO GENERAL)

Mantener un juego es un acto de equilibrio entre añadir constantemente nuevas características vs presupuestos. El juego tiene que funcionar dentro de las limitaciones de las plataformas de destino. Hacer un seguimiento de todos estos aspectos y asegurarse de que el juego sigue siendo funcional y reproducible requiere una vigilancia constante.

3.11.17 BASIC TOOLS AND TECHNIQUES (HERRAMIENTAS BÁSICAS Y TÉCNICAS)

Ser capaz de encontrar rápida y fácilmente las partes de su juego que están causando problemas es muy importante durante la producción. Esto puede ahorrar tiempo y dinero. Pero lo que puede ser un ahorro de tiempo aún mayor es la identificación de áreas con problemas potenciales, incluso antes de que se empiecen a causar problemas en situaciones de prueba del juego estándar. La optimización del juego para ver la página de juego Programadores describe un flujo de trabajo de ejemplo para optimizar el juego y la representación que se utiliza durante el desarrollo de Infinity Blade Dungeons .

3.11.18 MEMORY PROFILING (PERFIL DE MEMORIA)

El uso de memoria es siempre una preocupación para los videojuegos, especialmente los destinados a las consolas o dispositivos móviles, donde el espacio de memoria es limitada. Esto se extiende a partir de la cantidad de espacio que ocupan los activos de contenido en el disco para el uso de memoria de los diferentes sistemas en tiempo de ejecución a la cantidad de asignaciones de memoria y cancelaciones de asignación . Estas son todas las piezas de información extremadamente importantes que hacen que sea posible para limitar el uso de memoria de los límites aceptables .

El uso de la memoria y la página de perfiles muestra varias herramientas y técnicas que se pueden utilizar para conocer y controlar el uso de la memoria por su juego.

3.11.19 PROFILING CONTENT AND PERFORMANCE (CONTENIDO DE PERFILES Y RENDIMIENTO)

Dada la cantidad de contenido utilizado en los juegos de hoy en día, es extremadamente importante para limitar el impacto de ese contenido en el desempeño del juego siempre que sea posible. Optimizar el contenido tiene el potencial de proporcionar el mayor retorno de la inversión por lo que a menudo es la primera área que se perfila y optimiza. Unreal Engine 3 ofrece varias herramientas para ayudar en este proceso. Estas herramientas , junto con muchos consejos sobre cómo limitar el ipacet de contenidos se detallan en el perfil y la optimización de la página de contenido.

3.11.20 CPU PERFORMANCE (RENDIMIENTO DEL CPU)

El trabajo realizado por el motor de la CPU se divide en las discusiones . Si bien puede haber varios subprocesos de trabajo que realizan varias tareas a la vez , los dos temas principales son:

GameThread - El GameThread maneja jugabilidad actualización. Esto incluye Actores, componentes de recolección, etc.

RenderThread - El RenderThread encarga la realización de cálculos de iluminación, cálculos de sombreado, configuración de transluminancia, actualización de capturas de escena, oclusión , etc

Después de aislar el cuello de botella en el ambiente del juego como un problema de rendimiento de la CPU, uno de los próximos pasos debe ser para determinar en qué parte de su código C++ se está gastando el tiempo. Además, es posible que desee buscar los costos ocultos de rendimiento como de carga - Hit Tiendas- o fallos de caché .

La mayoría de las plataformas tienen sólidas herramientas de muestreo de CPU disponibles que pueden proporcionar métricas de código detalladas e incluso llamar a los gráficos para C + + usando datos capturados de una sesión en vivo . Por ejemplo , en la plataforma de Windows podría utilizar Intel VTune para

capturar los datos de muestreo en un período corto para aislar los puntos calientes. Plataformas de consola tienen herramientas similares disponibles, a menudo se incluyen con la cadena de herramientas de desarrollo. Usted querrá familiarizarse con estas aplicaciones y utilizarlas ampliamente en todo el desarrollo. tenga en cuenta que en algunas consolas, Unreal tiene una función de utilidad para ayudar a realizar capturas de muestreo simple en el CPU. Puede escribir GAME TRACE para iniciar un rastreo en el CPU. La mayoría del profiling de CPU soportan la captura de datos de muestreo y / o datos del gráfico de llamadas durante un corto período. Esto es muy bueno para la depuración de problemas de rendimiento que implican una velocidad de fotogramas pobres. Capturas de muestreo suelen dar a funcionar llame puntos alerta en el código C++. Esto puede ser útil a veces, pero más de las veces usted encontrará que una captura de Call Graph es mucho más útil. Gráficos de llamadas a menudo tienen más sobrecarga en tiempo de captura, pero proporcionan información detallada sobre las llamadas de funciones de punto alerta, que a menudo pueden conducir directamente a la fuente de un problema. Incluso en los casos en que muchos objetos/actores están llamando una única función que se muestra como lento, si tiene suerte, el gráfico de llamadas le llevará de nuevo a un nombre de clase de actor en particular u otro patrón que hace alusión a la causa.

3.11.21 GAME PERFORMANCE (RENDIMIENTO DE JUEGO)

Todos los objetos de juego que se añade a la escena tiene algo de recursos, y por lo general los objetos más interesantes toman la mayor cantidad de recursos. Con el fin de tener suficiente tiempo de CPU GameThread para aquellos objetos que necesita para asegurarse de que otros objetos no están utilizando los recursos de manera injusta y que todos los objetos hacen la mínima cantidad de trabajo necesario para lograr su objetivo. Muchas de las herramientas, consejos y técnicas para perfilar y optimizar el trabajo GameThread se detallan en la generación de perfiles del juego y en la página de optimización.

3.11.22 RENDER PERFORMANCE (RENDIMIENTO DE RENDERIZADO)

La complejidad de la escena en términos de iluminación y sombreado, así como varios otros aspectos visuales intensivos del procesador, puede tener un enorme impacto en el tiempo de permanencia en el RenderThread. Averiguar cuál de estas áreas se alentan, RenderThread ayudará a centrar los esfuerzos en disminuir el impacto de elementos que no son tan importantes, al tiempo que más recursos a disposición de los rasgos cruciales y obras maestras visuales. El Procesamiento Profiling y la página de optimización demuestra varias herramientas y métodos de elaboración de perfiles y la optimización de los elementos visuales se manejan en el RenderThread.

3.11.23 GPU PERFORMANCE (RENDIMIENTO DE GPU)

Mientras que el CPU se encarga de los cálculos y la actualización del juego de cada cuadro, la GPU es el encargado de mostrar el juego. Obviamente, esto significa que los polígonos de dibujo, sino que también implica hacer frente a la

complejidad de sombreado y sobregirar, mallas -GPU piel esqueléticos, efectos de procesamiento de correos, las superficies de fluidos, etc Si el GPU está gastando una gran cantidad de tiempo de trabajo, podría significar que su juego se llene de otros elementos de renderizado causando problemas de rendimiento. El GPU recibe órdenes del subproceso de representación y lo hace : Transformación de Vértices y el exhibición de mallas esqueléticas. Rasterization (averiguar qué píxeles caiga sobre los que los triángulos) Sombreado (ejecución del sombreado de píxeles en todos los píxeles de un triángulo dado. El sombreado de píxeles es controlado por el material dado a ese triángulo) Con perfiles GPU hay una gran cantidad de características complicadas. La regla de oro en cuanto a rendimiento de la GPU es siempre experimentar. La operación que consume tiempo principal desarrollada por la GPU es el sombreado de píxeles. La complejidad de los shaders usados, la cantidad de espacio de pantalla que los shaders ocupan, la cantidad de sobregirar causada por shaders translúcidos, etc , son factores muy importantes a considerar al perfilar la GPU. Otra área que la GPU es responsable de es la transformación de mallas de piel. La cantidad de vértices de las mallas del esqueleto que se utilizan es un factor grande como vértices sin piel son mucho más costosos que los vértices de la malla estáticas. Obviamente , la cantidad de polígonos que se dibuja es también un área del problema potencial de rendimiento del GPU, aunque las tarjetas actuales tienen la capacidad de dibujar grandes cantidades de polígonos. Aún así, uno siempre quiere estar seguro de que los polígonos que no necesitan ser dibujado no se están elaborando, y que engrana sólo tiene tantos polígonos que sean necesarios para lograr el resultado visual deseado. Para ayudar en la captura de información acerca del GPU, un GPU de un solo cuadro de captura de muestreo se puede realizar en algunas consolas utilizando el comando TRACE PERFORM. Hay varias herramientas disponibles para ayudar a perfilar del GPU tanto en el PC como en las distintas consolas. La página de rendimiento del GPU y de perfiles explica cómo utilizar las herramientas proporcionadas con Unreal Engine 3, así como otras herramientas externas para determinar dónde los problemas de rendimiento pueden estar ocurriendo en el GPU.

3.11.24 NETWORK PROFILES (PERFILES DE RED)

Mantener atención en la cantidad de datos que se envían a través de la red del juego en línea es importante. no quiere el envío de un montón de datos innecesarios o tener objetos específicos spamming en la red. Mantener el tráfico de red simplificado hará juegos online funcionen mejor para todos los jugadores, pero especialmente los que tienen un ancho de banda limitado.

3.11.25 PERFILES MÓVILES

Perfiles de los juegos en los dispositivos móviles es bastante similar al perfilado para PC. Hay algunas herramientas especiales para las distintas plataformas y algunas consideraciones a tener en cuenta.

3.11.26 PHYSICS (FÍSICA)

Unreal Engine 3 utiliza el motor de física PhysX para conducir sus cálculos de simulación física. PhysX proporciona la capacidad para llevar a cabo la detección de colisiones precisa, así como simular interacciones físicas entre los objetos dentro del mundo. Tener la física en su juego va a ayudar a mejorar el valor de inmersión de cada escena, ya que ayuda a los jugadores creen que están interactuando con la escena y que la escena está respondiendo de nuevo en una forma u otra. PhysX permite simular capas, El uso de cuerpos suaves de PhysX, puede simular cuerpos flexibles que se doblan y tuercen como el jugador que Prods. También es posible crear niebla físicamente interactiva, u otras simulaciones de partículas físicas complejas. Y todo esto, en tiempo real.

3.11.27 POST PROCESS EFFECTS (EFECTOS POST PROCESOS)

Procesamiento posterior permite diferentes efectos que se deben realizar en el escenario después de haber sido dictada, pero antes de dar salida a la vista. El marco de proceso posterior en el Unreal Engine 3 permite los efectos del proceso post-artista configurable, a través de las agrupaciones de los elementos del proceso de post formando un efecto general. Ejemplos de elementos y efectos incluyen la floración (HDR efecto blooming en objetos brillantes), profundidad de campo, desenfoque de movimiento, oclusión del ambiente y efectos materiales, que son materiales personalizados se ejecutan en la imagen de la escena final.

3.11.28 RENDERING (RENDERIZADO)

Desde un punto de vista visual, lo que el jugador ve en la pantalla, la escena representada, es el aspecto más importante. Lo que el jugador por lo general no se da cuenta es todo el trabajo que se dedica a la producción de ese escena renderizada, el trabajo hecho por el sistema de representación. El subsistema de rendering en Unreal Engine 3 no sólo se encarga de la representación de la geometría y otros elementos a la pantalla, sino que también abarca una gama de otros sistemas, como la iluminación y las sombras, los materiales y el procesamiento posterior. Todos ellos trabajan en conjunto para producir la escena final prestados y dar al juego su apariencia.

3.11.29 SCALEFORM GFX

Scaleform GFX proporciona un medio para el uso de gráficos en movimiento creados en Adobe Flash Professional como interfaces de usuario y las pantallas de heads-up en el Unreal Engine 3. Estos pueden ser prestados directamente a la pantalla o representan como texturas que se pueden utilizar en materiales y aplicadas a la geometría en el ambiente. Scaleform GFX pone el poder para diseñar interfaces de usuario en las manos del artista, ofreciendo una colección de widgets prefabricados diseñados específicamente para su uso en juegos y soporte integrado para efectos como la animación. Con su integración en el Unreal Engine 3, Scaleform permite UnrealScript, así como secuencias de comandos visual del Kismet para trabajar en conjunción con el lenguaje de

scripting de Flash ActionScript para crear interfaces atractivas e interactivas que se aprovechan de movimiento capacidades gráficas de Flash y la familiaridad de los desarrolladores con las herramientas de Unreal Engine 3.

3.11.30 SKELETAL MESHES (MALLAS ESQUELÉTICAS)

Las mallas esqueléticas están compuestas de dos partes, un conjunto de polígonos compuestos para compensar la superficie de la malla esquelético y un conjunto jerárquico de huesos interconectados que pueden ser utilizados para animar los polígonos. Mallas esqueléticas se utilizan a menudo en Unreal Engine 3 para representar los caracteres u otros objetos animados. Los modelos 3D, los aparejos y las animaciones se crean en un modelado y animación aplicaciones externas (3dsMax, Maya, Softimage, etc), que luego se importan a Unreal Engine 3 utilizando el Navegador de contenido de Unreal Editor y guardan en paquetes.

3.11.31 STATIC MESHES (MALLAS ESTÁTICAS)

Una malla estática es una pieza de geometría que consiste en un conjunto de polígonos que pueden ser almacenados en la caché de memoria de vídeo y los servicios proporcionados por la tarjeta gráfica. Esto les permite ser prestados de manera eficiente, lo que significa que pueden ser mucho más complejos que otros tipos de geometría tales como cepillos BSP. Puesto que están en la memoria de vídeo, sin embargo, mallas estáticas se pueden traducir, rotar y escalar, pero no pueden tener sus vértices animados de ninguna manera.

Las mallas estáticas son la unidad básica que se utiliza para crear la geometría ambiente de niveles creados en Unreal Engine 3. Estos son los modelos 3D creados en aplicaciones de modelado externos (como 3dMax, Maya, Softimage, etc.) que se importan a Unreal Editor a través del Navegador de contenido, guardándose en paquetes, y luego utiliza en diversas formas de crear elementos renderizable. La gran mayoría de cualquier mapa de un juego hecho con Unreal Engine 3 constará de mallas estáticas , generalmente en forma de StaticMeshActors . Otros usos de las mallas estáticas son para la creación de motores, tales como puertas o ascensores, los objetos de la física de cuerpos rígidos, decoraciones follaje y del terreno, los edificios, los objetivos del juego, y muchos más elementos visuales.

3.11.32 UNREAL EDITOR AND TOOLS (EDITOR UNREAL Y HERRAMIENTAS)

Unreal Engine 3 contiene un conjunto de herramientas utilizadas para crear y editar los niveles, importar y organizar los activos de arte, crear activos de contenido especializado, y mucho más. La piedra angular de estas herramientas es el Unreal Editor, o UnrealEd. Unreal Editor se compone de un editor de niveles, una colección de los navegadores, y un conjunto de herramientas especializadas. Prácticamente todos los desarrolladores que trabajan en un proyecto pasarán por lo menos parte de su tiempo de trabajo en el interior de Unreal Editor. Diseñadores de niveles, obviamente, utilizarán el editor de niveles para crear

ambientes para que el juego tenga lugar. Los creadores de contenido importarán sus activos y el uso de las herramientas especializadas para ponerlos en marcha para su uso en el juego. Los programadores utilizarán los niveles de prueba para asegurar su código de juego funciona como se espera y puede ser la tarea de ampliar la funcionalidad del editor o la adición de nuevas herramientas. Estar familiarizado con el Unreal Editor y sus herramientas es esencial para cualquier desarrollador que trabaja con el Unreal Engine 3.

3.11.33 UNREALSCRIPT

UnrealScript es el lenguaje de programación para el Unreal Engine 3. Debido a que fue diseñado para juegos, contiene ciertas características integradas, como los estados y los temporizadores, que hacen el juego mucho más fácil de implementar. UnrealScript se compila, pero no es parte del ejecutable principal. Esto hace que sea un perfecto medio de implementación y la iteración de nuevos elementos de juego sin la necesidad de volver a compilar todo el motor cuando se realizan cambios. El lenguaje en que se construye es orientado a objetos y se asemeja a Java o C++ en su sintaxis. Esto hace que sea familiar para los programadores de juegos con experiencia, y muy fácil de aprender. Al mismo tiempo, ya que se utiliza en el contexto de un juego, algunos de las metodologías y principios difieren de la programación tradicional. Por ejemplo, no hay constructores explícitos estándar o destructores para objetos que pueden ser confusos a los utilizados para los idiomas más tradicionales de programación. Una vez que se familiarice con Unreal, sin embargo, las cosas empiezan a encajar en su sitio y la adición de nuevos elementos a los juegos será un proceso rápido.

3.11.34 USER INTERFACES & HUDS (INTERFACE DE USUARIO Y HUDS)

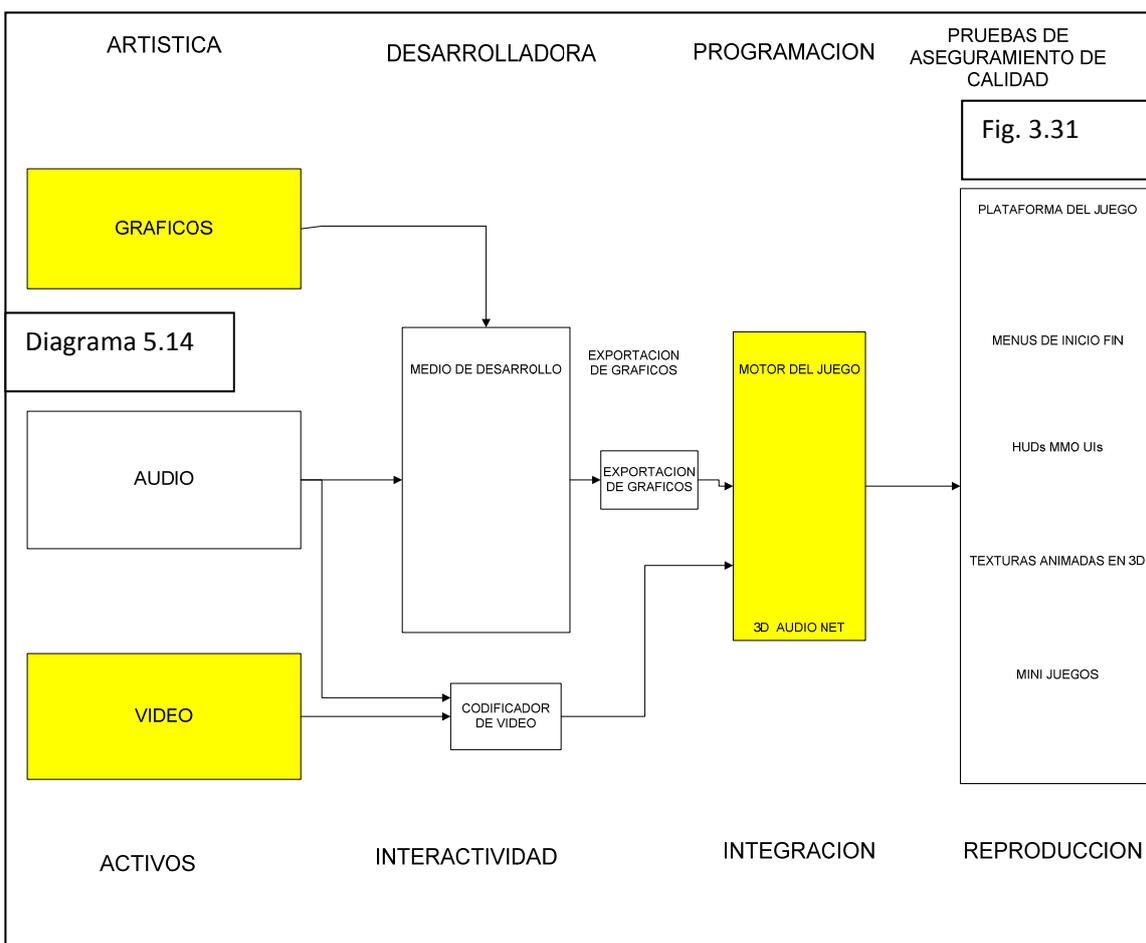
La forma en que el juego se comunica e interactúa con el jugador es extremadamente importante. Las interfaces de usuario (UI) y heads-up displays (HUD) son la manera de los juegos de proporcionar información sobre el juego al jugador y en algunos casos permitiendo al jugador interactuar con el juego. Unreal Engine 3 ofrece múltiples medios de creación de interfaces de usuario y HUD. La clase Canvas se puede utilizar para dibujar directamente en la pantalla a un bajo nivel, superpuesto sobre el mundo. Scaleform GFX también está integrado en el motor y proporciona un menú completo y solución de HUD utilizando las herramientas profesionales tales como Adobe Flash Professional. Para los juegos móviles, también hay un sistema de menús móvil construido en la cima de la funcionalidad de la lona que ofrece la posibilidad de crear menús usando la entrada táctil.

3.11.35 GRÁFICOS Y VIDEO

La fig. 3.31 muestra la relación de gráficos y video en la etapa artística como elementos separados y en la etapa desarrolladora se asocian y se codifican para agregarles atributos en la etapa de programación.

Esta lógica de procesos facilita el funcionamiento de los objetos en los procesos por ende la detección de errores y su fácil corrección.

Durante las pruebas simultáneas de calidad hechas por los beta testers aparecen problemas relacionados a todo tipo de procesos y objetos.



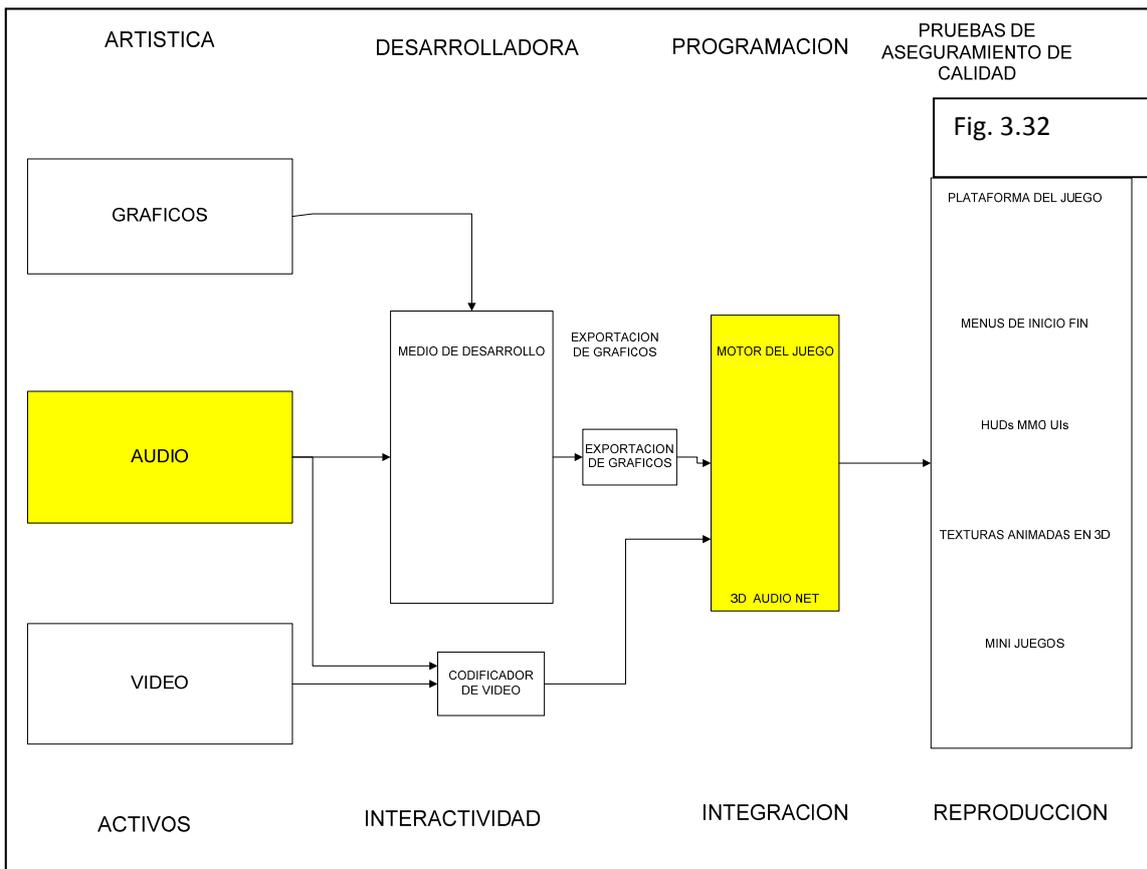
En apariencia parecer lo mismo gráficos y video pero existen diferencias substanciales, los gráficos se refieren a la forma en que están creados y conformados los objetos así como su actuación e interoperabilidad en el medio de desarrollo.

Y el video es la forma de proyectar estos elementos multimedia en la interface visual del hardware, esta parte está vinculada a los recursos existentes en el dispositivo donde la aplicación corre.

3.11.36 AUDIO

La fig. 3.32 muestra la jerarquía del audio y su respectiva evolución en el desarrollo del videojuego, esta evolución está normalizada por API's dedicadas a manejo de hardware y a efectos sonoros

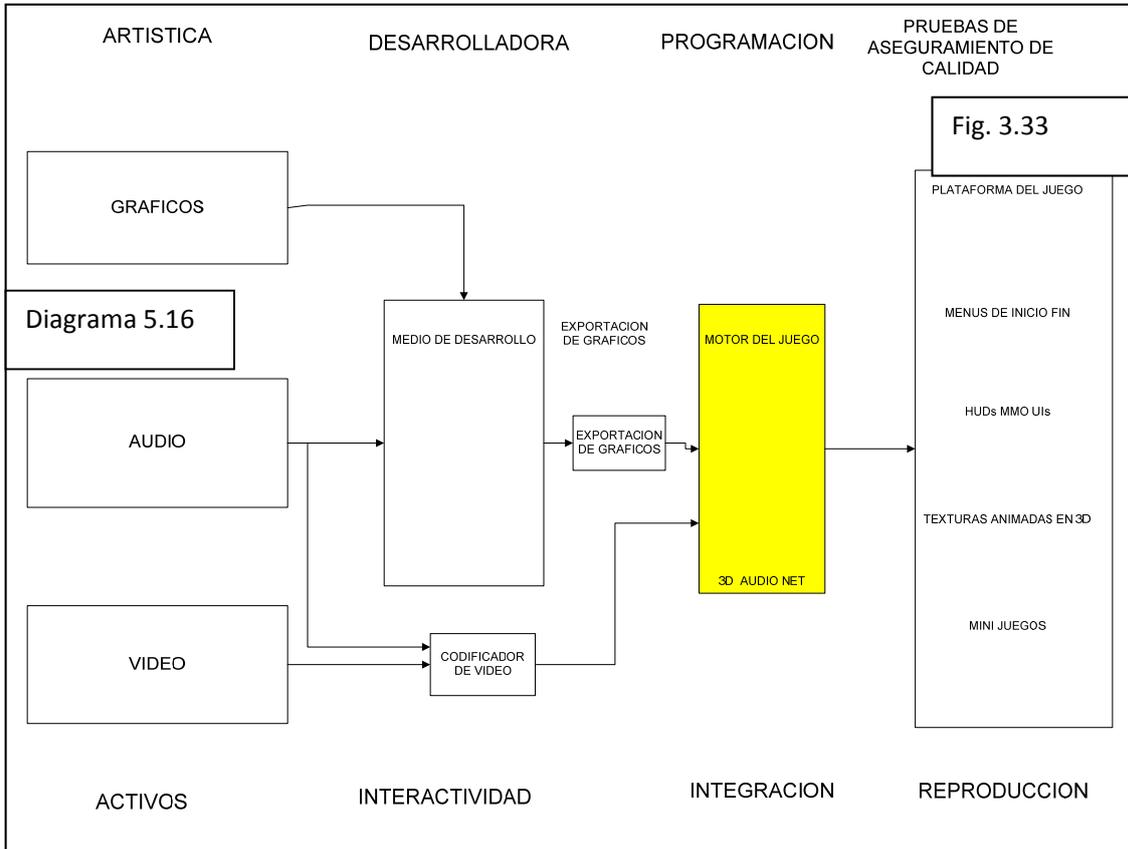
Estas rutinas de programación son las que conforman al motor de juego, conforme avanza el desarrollo del producto, el motor de juego adquiere mayor complejidad en cuanto al manejo de librerías y su asociación a eventos estáticos y dinámicos.



En la etapa artística las librerías de audio se crean y en la etapa de programación se integran asociándose a eventos.

3.11.37 3D

La fig. 3.33 muestra como la 3D es parte de la etapa de integración durante la programación del producto, con el uso del motor de juego los gráficos obtenidos adquieren propiedades visuales de profundidad y sombras ofreciendo un efecto tridimensional en el plano de la interface gráfica.

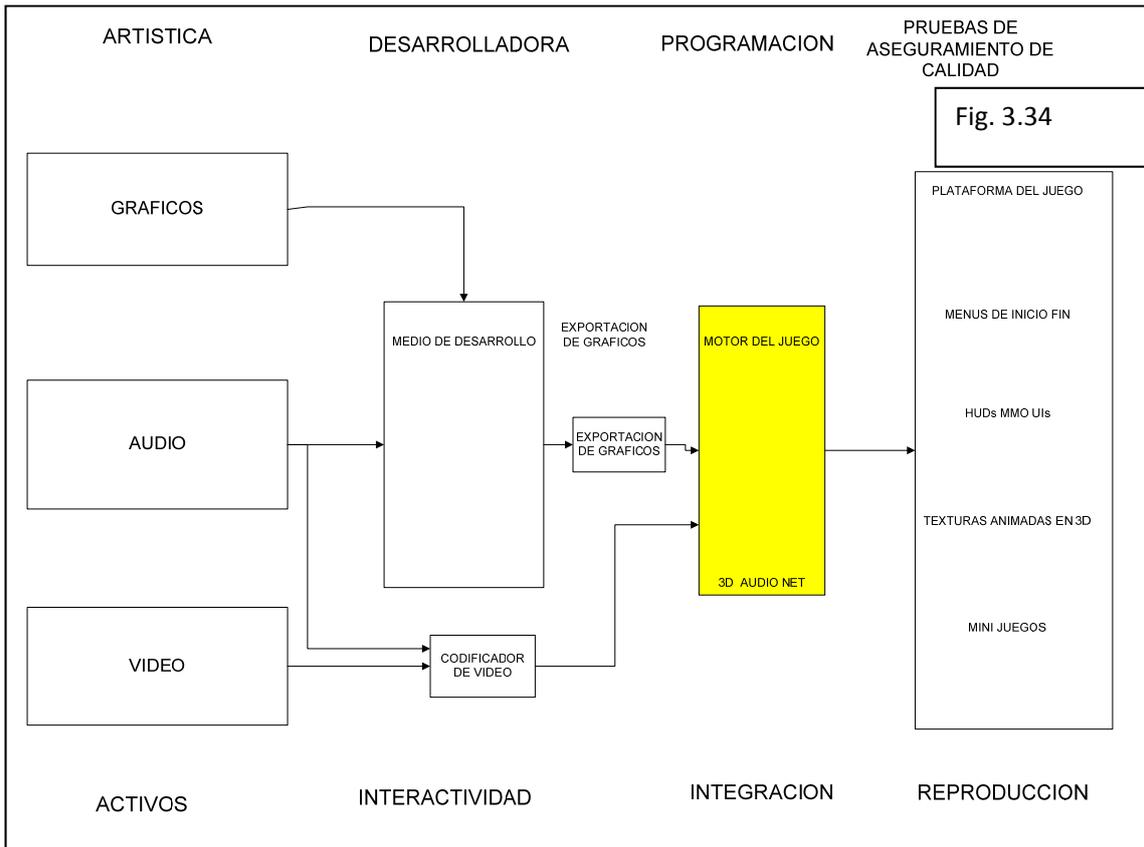


Esta etapa consiste en la asociación de atributos a los fondos como son luz, sombra, foco, cámara, en el espacio de 3 dimensiones, al mover coordenadas se experimentan efectos de profundidad, lo que hace percibir al usuario volumen en los objetos en movimiento.

Estos atributos ya vienen dados por el software de desarrollo empleado en los gráficos, el motor de juego proporciona las relaciones existentes entre estos objetos y su participación motivada por el usuario del producto.

3.11.38 NET, MMORG Y HUDS

La conectividad del videojuego es un estándar actual, son muy escasos los videojuegos que no tengan presente el modo online (puede estar deshabilitado por carecer de una licencia original), esta conectividad se logra bajo el esquema de cliente – servidor, con este esquema se pueden enviar y recibir datos en tiempo real pudiendo visualizar en pantalla a otros jugadores, tener comunicación con ellos y poder ver mensajes emergentes de la partida en línea.



La fig. 3.34 representa el lugar donde se efectúan los procesos de NET, MMORG Y HUD.

3.12 REVISIÓN DEL CÓDIGO FUENTE DE QUAKE 3

Después de doom, doom iphone, quake1, quake2, iphone wolfenstein y doom 3, idtech3 es el motor 3d que impulsa quake 3 y quake live. El motor es sobre todo una evolución del idtech2 pero hay algunas novedades interesantes. Vea la fig. 3.35

los puntos clave se pueden resumir de la siguiente manera:

Nuevo procesador DualCore con shaders a base de materiales (construidas sobre OpenGL con activos de información fijo).

Nuevo modelo de red basado en las capturas.

nuevas máquinas virtuales que juegan un papel esencial en el motor, combinando Quake1 portabilidad/seguridad con la velocidad Quake2.

Nueva Inteligencia Artificial de los bots.

Haciéndose notar que:

- El sistema de máquinas virtuales y la cadena de herramientas asociadas que en conjunto representan el 30% del código liberado. Bajo esta perspectiva idTech3 es un mini sistema operativo que proporciona las llamadas al sistema de tres procesos.
- El sistema de red basado en las instantáneas y la introspección de la memoria.

PRIMER CONTACTO Y COMPILACION



fig 3.35 vista de QUAKE a través de Windows.

el código se puede encontrar en cuenta GitHub id Software: Cuando se trata de comprender una gran base de códigos es preferible utilizar XCode: Velocidad de Spotlight, Comando y clic para buscar definiciones en cadenas de relieve que la herramienta más poderosa que Visual Studio. Pero la apertura del proyecto Quake III mostró que el código en descomposición no es siempre sobre el código, sino también acerca de las herramientas: XCode 4.0 no puede abrir los Quake III proyectos XCode 2.0. Al final utilizado Visual Studio 2010 Professional en Windows 8: Tras la instalación de las herramientas de productividad de potencia de Visual Studio 2010, el conjunto de herramientas de realidad fue agradable.

Lo primero que llama la atención es que el espacio de trabajo visual Studio no está hecha de un proyecto, sino ocho. No todos ellos son utilizados, dependiendo si la construcción es DEBUG o RELEASE (especialmente game, cgame y q3_ui: los proyectos de máquinas virtuales). Algunos de los proyectos no se usan nunca (splines y ui).

La fig. 3.36 ilustra los módulos de QUAKE:

Projects	Type	DEBUG Builds	RELEASE Builds	Comments
botlib	Static Library	botlib.lib	botlib.lib	A.I
cgame	Dynamic Library/Bytecode	cgamex86.dll	-	
game	Dynamic Library/Bytecode	qagamex86.dll	-	
q3_ui	Dynamic Library/Bytecode	uix86.dll	-	
quake3	Executable	quake3.exe	quake3.exe	
renderer	Static Library	renderer.lib	renderer.lib	OpenGL based
Splines	Static Library	Splines.lib	Splines.lib	Used NOWHERE !
ui	Dynamic Library/Bytecode	uix86_new.dll	-	Used for Quake III Arena.

FIG. 3.36

3.12.1 ARQUITECTURA DEL MOTOR DE JUEGO

Una conveniente forma para entender una arquitectura es mirar primero al software como una caja negra recibiendo entradas (flechas superiores izquierdas) y generando salidas (flechas inferiores) fig. 3.37

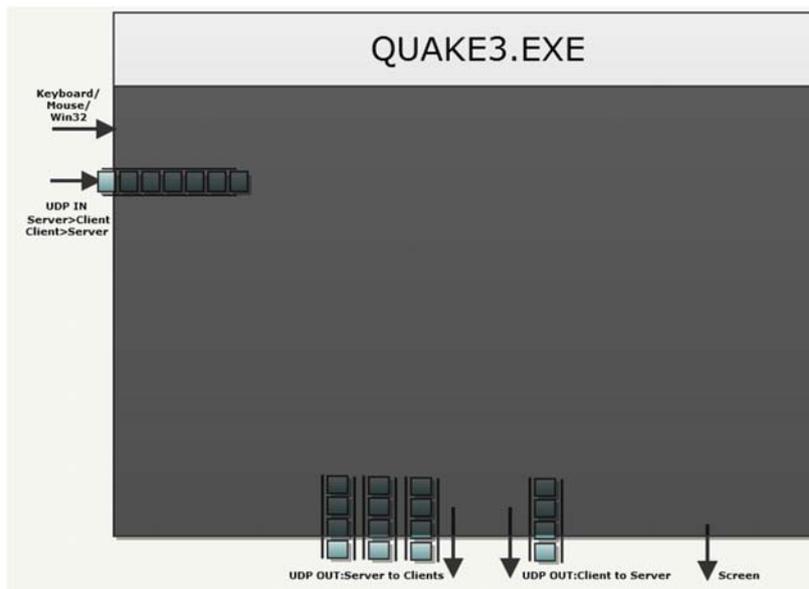


fig. 3.37

Entonces se ven cómo fluyen las entradas hacia las salidas de una manera (cuadro blanco) con los 6 módulos (quake3.exe, renderer.lib, bot.lib, game, cgame y q3_ui) que interactúan de la siguiente manera fig. 3.38:

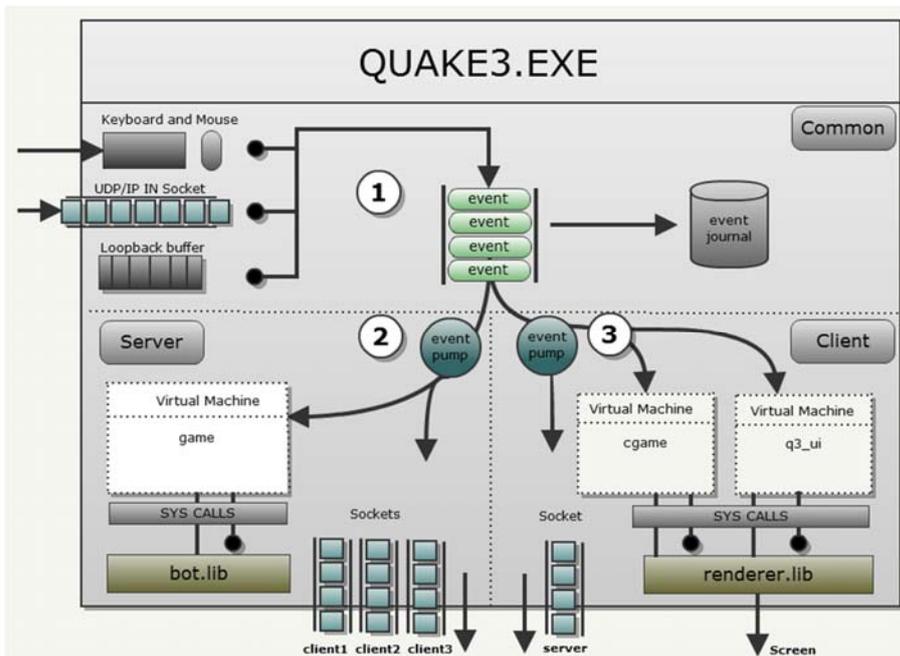


fig. 3.38

Dos cosas importantes para entender el diseño:

1. cada entrada (teclado, message win32, mouse, socket UDP) es convertida en una event_t y se coloca en una cola de eventos centralizado (sysEvent_t eventQue [256]). Esto permite, entre otras cosas grabar (Contabilizar) cada entrada con el fin de recrear errores.

2. La División explícita del cliente y servidor.

El servidor es responsable de mantener el estado del juego, determinar lo que se necesita por los clientes y difundirlo por la red. Está enlazado estáticamente contra bot.lib que es un proyecto independiente.

El lado del cliente es responsable para predecir las entidades son (compensación de latencia) reproducibles a la vista. Está enlazado estáticamente contra rendererproject: Un proyecto independiente que habría permitido un Direct3D o incluso software renderizable para estar conectados fácilmente.

3.12.2 EL CÓDIGO

Desde un punto de vista el código aquí es un bucle parcialmente desenrollado que ilustra la producción de eventos y el consumo por el cliente y el servidor:

```
int WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow)
{
    Com_Init
    NET_Init

    while( 1 )
    {
        // Common code
        IN_Frame() // Add Win32 joystick and mouse inputs as event_t to unified event queue.
        {
            IN_JoyMove
            IN_ActivateMouse
            IN_MouseMove
        }

        Com_Frame
        {
            // Common code
            Com_EventLoop // Pump win32 message, UDP socket and console commands to the queue
            (sysEvent_t eventQue[256])
            Cbuf_Execute

            // Server code
            SV_Frame
            {
                SV_BotFrame // Jump in bot.lib
                VM_Call( gvm, GAME_RUN_FRAME, sv.time ) // saltar a la maquina virtual donde la logica
del juego es realizada

                SV_CheckTimeouts
                SV_SendClientMessages // Send snapshot or delta snapshot to connected
clients
            }

            // Common code
            Com_EventLoop
            Cbuf_Execute

            // Client code
            CL_Frame
            {
                CL_SendCmd // Pump the event queue and send commands to server.

                SCR_UpdateScreen
                VM_Call( cgvm, CG_DRAW_ACTIVE_FRAME); // Send message to the Client Virtual
Machine (do Predictions).
                or
                VM_Call( uivm, UI_DRAW_CONNECT_SCREEN); // If a menu is visible a message is sent
to the UI Virtual Machine.

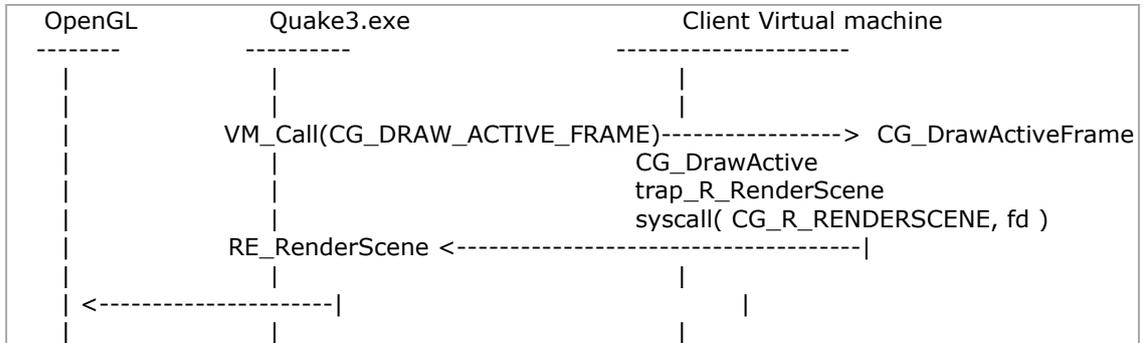
                S_Update // Update sound buffers
            }
        }
    }
}
```

Una cosa interesante a notar aquí que ilustra perfectamente la forma primordial de las máquinas virtuales son:

En ninguna parte vemos una llamada a RE_RenderScene: la función que realiza la suprecion y emision de comandos de OpenGL. En su lugar lo que sucederá es:

Quake3.exe envía un mensaje a la máquina virtual del cliente: CG_DRAW_ACTIVE_FRAME que la señal de que se necesita una actualización.

La máquina virtual realiza alguna suprecion de entidad y la predicción a continuación requieren la interpretación de OpenGL a través de una llamada al sistema Quake3 (CG_R_RENDERSCENE). Quake3.exe recibe la llamada al sistema y, de hecho llama RE_RenderScene.



3.12.3 ESTADÍSTICAS

Aquí están algunos estadísticas del documento.

	files	blank	comment	code
cloc-1.56.exe code common	559	48630	73501	233952
cloc-1.56.exe lcc	116	2270	1513	28067
cloc-1.56.exe q3asm q3map	44	4987	5565	22877
cloc-1.56.exe q3radiant	206	11870	13113	54922
TOTAL	919	68293	95509	341994

En un gráfico circular, podemos ver claramente cómo la inusual proporción del 30% del código base está dedicada a herramientas vea la fig. 3.39

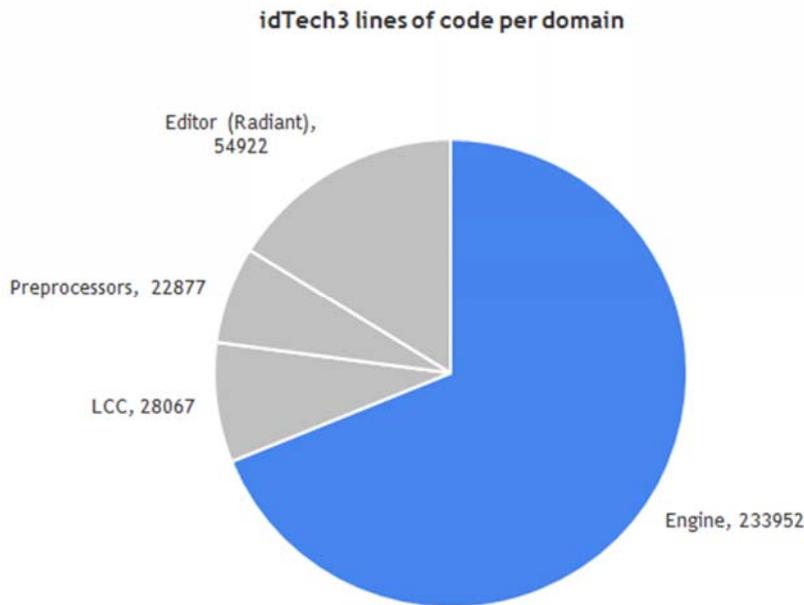


fig. 3.39

Esto explica en parte porque idtech3 cuenta con un compilador C ANSI: El código abierto de Little C Compiler (LCC) se utiliza para generar bytecode para las máquinas virtuales

3.12.4 ASIGNACIÓN DE MEMORIA

Dos asignadores personalizados trabajan aquí:

- Asignación de zona: Responsable para runtime.small y las asignaciones de memoria de corto plazo.
- Asignación de sección: Es responsable de la carga de nivel, y asignaciones grandes a largo plazo a partir de los archivos pak (geometry.map, textures, animations).

3.12.5 RENDERIZADO

El RENDERIZADO de Quake III es una evolución del renderizado por aceleración de hardware de Quake II: La parte clásica es que está construido sobre una arquitectura "Binary Partition"/"Potential Visible Set" con dos nuevos aspectos clave notables:

- Un sistema de Sombreado construido en parte superior de la información arreglada de OpenGL 1.X. Esto fue un gran logro en 1999. Se proporcionó una gran cantidad de espacio para la innovación en una era anterior a los ahora omnipresentes vértices, geometría y fragmentos de sombra.
- Soporte para arquitecturas multinúcleo: El modelo cliente/servidor de OpenGL está bloqueando en algunos métodos y un sistema de cadenas elimina parcialmente este problema.

3.12.6 ARQUITECTURA DEL RENDERIZADO

El renderizado está contenido enteramente en `renderer.lib` y estadísticamente enlazado a `quake3.exe`, vea fig. 3.40

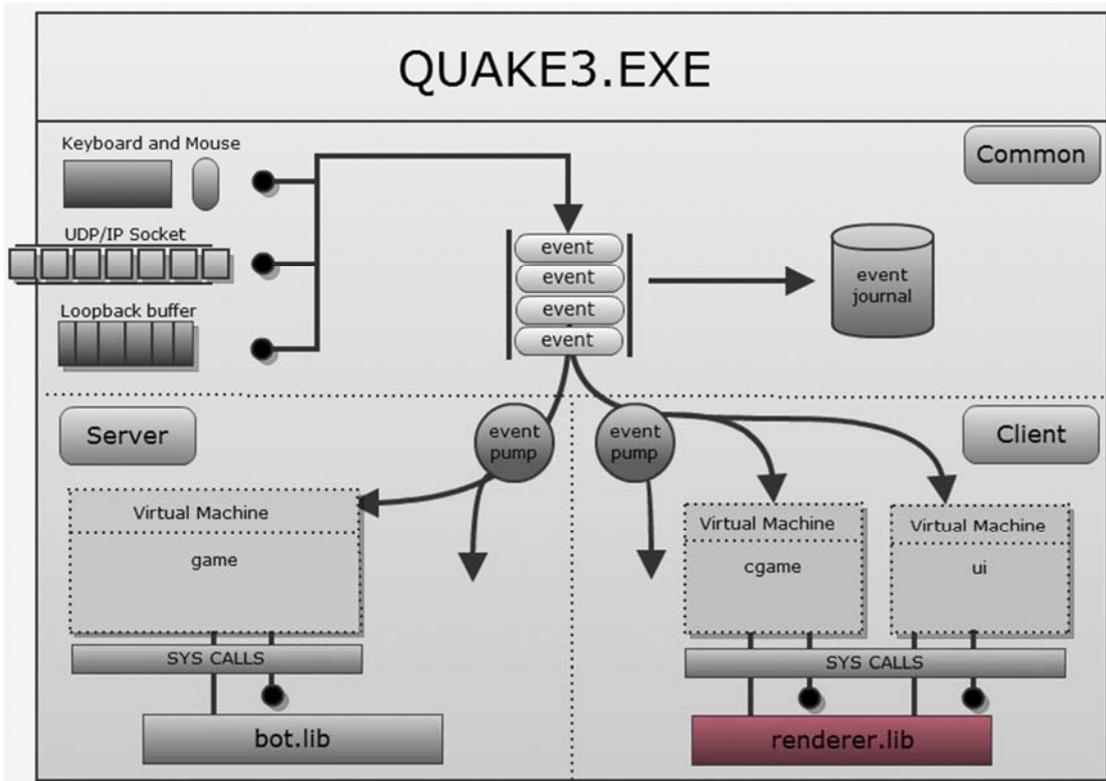


fig. 3.40

La arquitectura vanguardista de renderizado es la de QUAKE CLASICO: se basa en el famoso combo BSP/PVS/Lightmap:

- procesamiento previo:

- 1 . El diseñador de juegos crea y guarda un mapa. Utilizando QRadiant .
- 2 . q3bsp.exe secciona el mapa a través de particionamiento de espacio binario.
- 3 . Fuera del BSP un portal es generado.
- 4 . q3vis.exe utiliza el portal y genera un PVS (Potentially Visible Set) para cada entrega. Cada PVS se comprime y se almacena en un archivo bsp
- 5 . El portal es descartado.
- 6 . q3light.exe calcula todas las iluminaciones para cada polígono en el mapa y guarda el resultado como lightmap de texturas en el archivo bsp .
- 7 . En este punto, todo el procesamiento previo (PVSS y Lightmaps) se almacenan en el archivo de BSP .

- ejecución :

- 1 . El motor carga el mapa y el BSP .

- 2 . Cuando se solicita la entrega :

- 3 . El motor descomprime el PSV para la licencia actual y determina que es visible actualmente.

- 4 . Para cada polígono que combina ed lightmap con el color a través de multitexturas.

El paso de multitexturas y lightmap es claramente visible cuando el motor se modifica para mostrar sólo uno u otro fig. 3.41:

El dibujado de textura por los diseñadores/artistas de nivel :



fig. 3.41

El lightmap generado por q3light.exe fig. 3.42:



fig. 3.42

El resultado final cuando el tiempo de ejecución a través de multitexturizado fig. 3.43.



fig. 3.43

RENDERIZADO MULTINUCLEO Y SMP (symmetric multiprocessing)

Desconocido para mucha gente: Quake III Arena envió con SMP el soporte a través de `r_smp` variable. El frontend y el backend se comunican a través de un diseño estándar de productor/consumidor. Cuando `r_smp` se establece en 1 superficie de dibujo se almacenan alternativamente en un búfer doble situado en la memoria RAM. El frontend (que es llamado Main thread en este ejemplo) se escribe alternativamente a un buffer mientras que el otro es leído por el backend (llamado cadena de renderizado en este ejemplo, fig 3.44).

Un ejemplo para ilustrar cómo funcionan las cosas:

De t_0 a t_1

- La línea Principal decide qué dibujar y escribir superficies al `surfacebuffer1`.
- La línea de Render se inhabilita y por lo tanto bloqueada.
- La línea GPU también está haciendo nada.

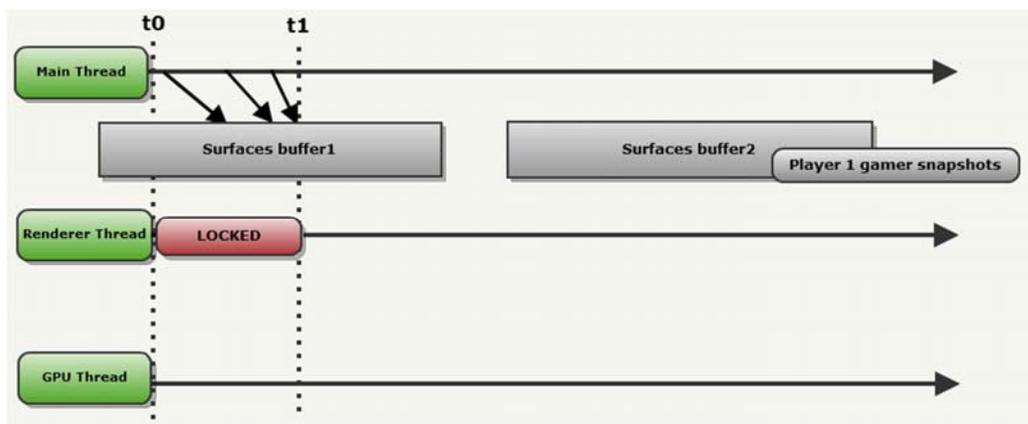


fig. 3.44

De t_1 a t_2 Los elementos empiezan a moverse por todo el lugar:

- la cadena principal está decidiendo lo que es visible para el siguiente fotograma. escribe sobre la superficie para `surfacebuffer2`: Este es un ejemplo típico de doble búfer.
- Mientras tanto, la cadena de Renderizado hace llamadas OpenGL y espera pacientemente hasta que la rosca GPU ha copiado todo en un espacio seguro.
- la cadena GPU lee la superficie desde donde está señalada la cadena de Renderizado.

notar que en t2, vea fig. 3.45:

- la cadena de renderizado todavía está transfiriendo datos a la GPU: SurfaceBuffer1 está en uso.
- la cadena principal ha escrito en SurfaceBuffer2 pero no puede empezar a escribir a SurfaceBuffer1: Está bloqueado

Este caso (en el que el cadena de Renderizado está bloqueando cadena principal es en realidad el más común durante la reproducción de Quake III: Se ilustra la limitación de bloqueo de algunos de métodos de las API de OpenGL.

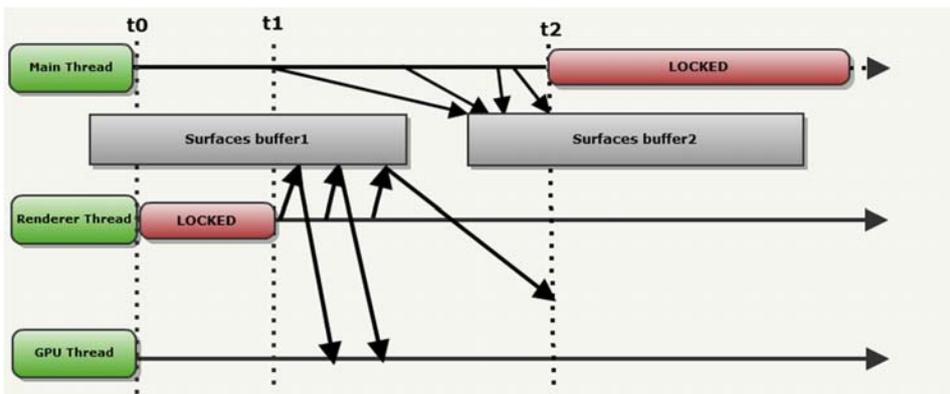


fig. 3.45

- Tan pronto como la cadena de renderizado se hace con el SurfaceBuffer1 (t3) se inicia el bombeo de las superficies de SurfaceBuffer2, fig 3.46.
- Tan pronto como se abrió (en t3), la cadena principal empieza a trabajar en la siguiente trama, escribiendo a SurfaceBuffer1.
- La GPU casi nunca es ociosa con esta configuración.

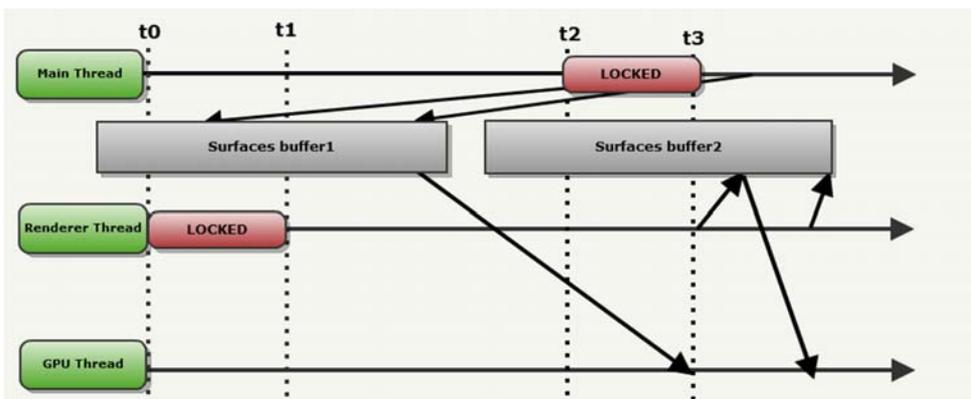


fig. 3.46

La sincronización esta realizada a traves de [Windows Event Objects](#) in [winglmp.c](#) (SMP seccion de aceleracion al final).

3.12.7 MODELO DE RED

El modelo de red de Quake3 es sin duda la parte más elegante del motor. En el nivel inferior de comunicaciones de Quake III todavía abstractos con el módulo NetChannel que apareció por primera vez en el ambiente de Quake. Lo más importante a entender es, vea fig. 3.47:

En un entorno de ritmo rápido a cualquier información que no se recibe en primera transmisión no vale la pena enviar de nuevo porque va a ser demasiado viejo de todos modos.

Como resultado, el motor se basa esencialmente en UDP/IP: No hay rastro de TCP/IP en cualquier lugar ya que el aspecto "transmisiónj real" introduce latencia intolerable. La pila de red se ha ampliado con dos capas que se excluyen mutuamente:

- El cifrado con la clave compartida previamente.
- La compresión con clave huffman pre-computada.

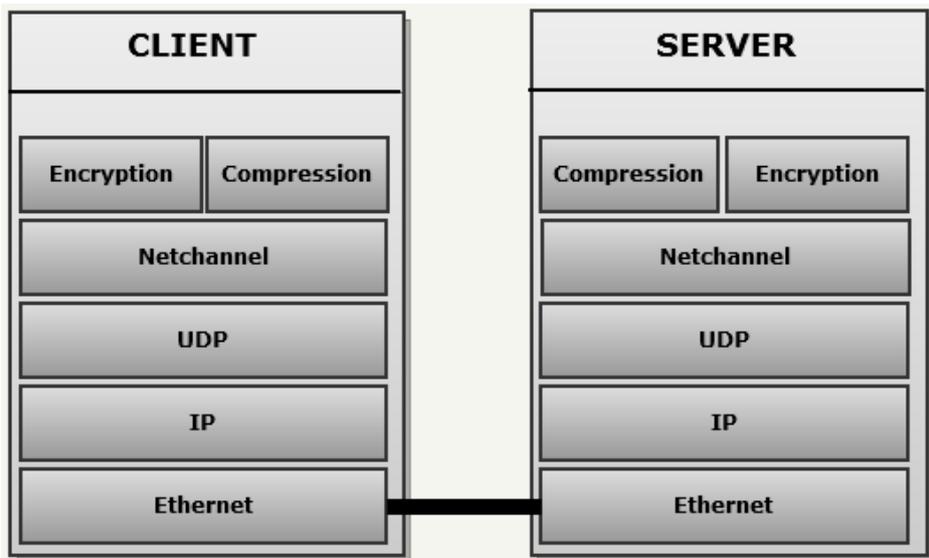


fig. 3.47

Pero donde el diseño realmente brilla está en el lado de servidor en el que un sistema elegante a minimizar el tamaño de cada datagrama UDP, mientras que la compensación de la falta de fiabilidad de UDP: Una historia de instantáneas genera paquetes deltas a través de la introspección de memoria.

3.12.8 ARQUITECTURA DE RED

El lado del cliente del modelo de red es bastante simple: El cliente envía comandos al servidor en cada fotograma y recibe la actualización para el gamestate. El lado del servidor es como poco más complejo ya que tiene que propagar el gamestate Maestro a cada cliente mientras se contabilizan paquetes UDP perdidos. Este mecanismo cuenta con tres elementos clave, vea fig. 3.48:

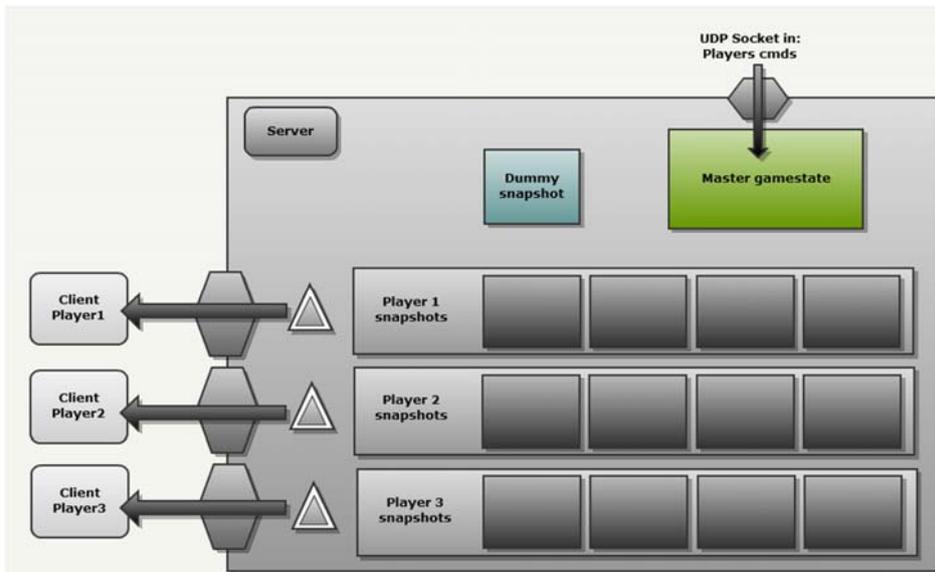


fig. 3.48

- Un gamestate maestro es el verdadero estado universal de las cosas. Los clientes envían sus comandos en el Netchannel. Se transforman en event_t que modificara el estado del juego cuando llegan al servidor.
- Para cada cliente el servidor mantiene los últimos 32 gamestate enviados a través de la red en un arreglo ciclico: Se llaman instantáneas. El arreglo ciclico con el famoso truco máscara binaria mencionado en Quake World Network.
- El servidor también cuenta con un gamestate "ficticio" con cada conjunto de campos tiende a cero. Esto se utiliza en capturas delta cuando no hay un "estado anterior" disponible.

Cuando el servidor decide enviar una actualización a un cliente que usatres elementos en orden para generar un mensaje que es enviado sobre el NetChannel.

3.12.9 SISTEMAS DE CAPTURAS

Con el fin de comprender el sistema capturas, aquí hay un ejemplo con las siguientes condiciones:

- El servidor envía la actualización a un Cliente1.
- El servidor está intentando propagar el estado de Client2 que tiene cuatro 4 campos (posición 3 ints [X], posición [Y], la posición [Z] y uno int aceptable).

- La comunicación se realiza a través de UDP / IP: Esos mensajes se pierden muy a menudo en el Internet.

El servidor ha recibido algunas actualizaciones de cada cliente. Ellos han impactado el gamestate Maestro (en verde). Ahora es el momento para propagar el estado de Client1, vea fig. 3.49:

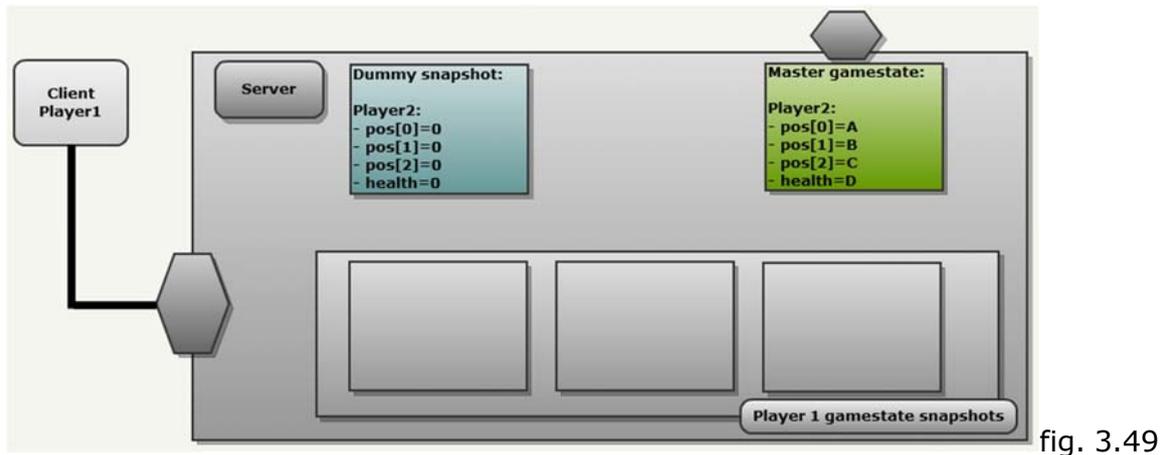


fig. 3.49

Con el fin de generar un mensaje al módulo de red siempre hará lo siguiente:

1. Copie el gamestate Master en la siguiente ranura de historia del cliente.
2. Compararlo con otra captura.

Esto es lo que podemos ver en el siguiente gráfico:

1. el Gamestate Maestro se copia en el índice 0 en la historia Cliente1, vea fig.3.50: Ahora se llama "Snapshot 1.2 Dado que esta es la primera actualización, hay ninguna captura válida en la historia Cliente1 lo que el motor va a utilizar en el archivo." Captura falsa", donde todos los campos son siempre cero. Esto da lugar a una actualización completa desde todos los campos solo se envía al NetChannel.

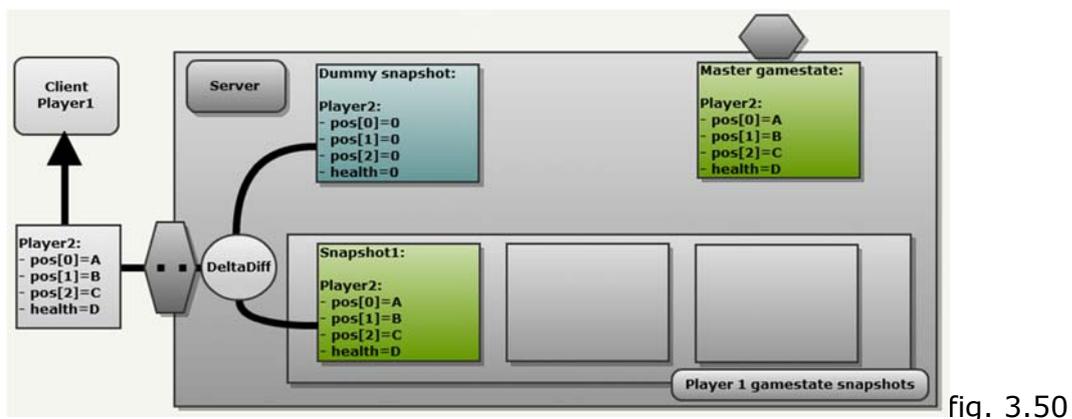


fig. 3.50

El punto clave para entender aquí es que si no hay capturas válidas están disponibles en el historial del cliente, el motor recoge "capturas falsas" para generar un mensaje delta. Esto dará lugar a un UPDATE completo enviado al cliente mediante 132 bits (por cada campo es preceded by a bit marker): [1 A_on32bits 1 B_on32bits 1 B_on32bits 1 C_on32bits].
Server Frame 2:

Ahora vamos a pasar adelante en el tiempo: se trata de la segunda trama del Servidor. Como podemos ver cada cliente han enviado órdenes y han impactado el gamestate Maestro: el Client2 se ha movido en el eje Y de manera pos [1] ahora es igual a E (en azul). Cliente1 también ha enviado comandos pero lo más importante, también ha reconocido que recibe el UPDATE anterior para que captura se ha marcada como "ACK", vea fig. 3.51:

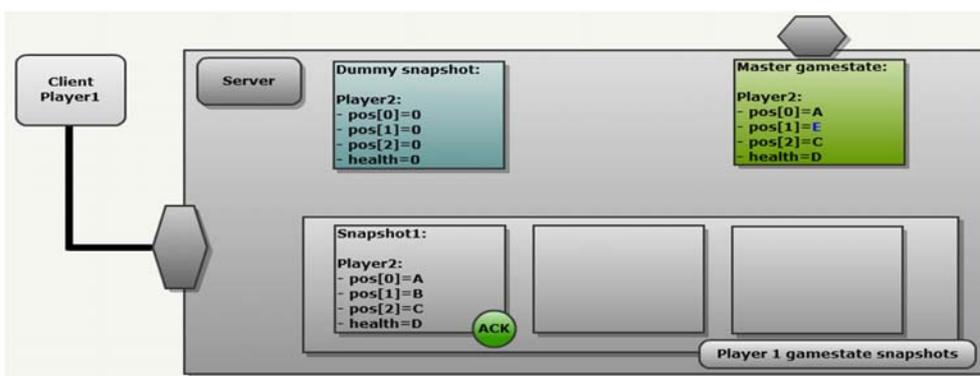


fig. 3.52

El proceso es el mismo, vea fig. 3.53:

1. Copie el gamestate Master en la siguiente ranura de historia del cliente: (índice 1): Se trata de instantáneas 2
2. Esta vez tenemos una instantánea válida en la historia del cliente (instantánea1). Comparar estas dos instantáneas

Como resultado se envía sólo una actualización parcial (POS [1] = E) sobre la red. Esta es la belleza del diseño: El proceso es siempre el mismo.

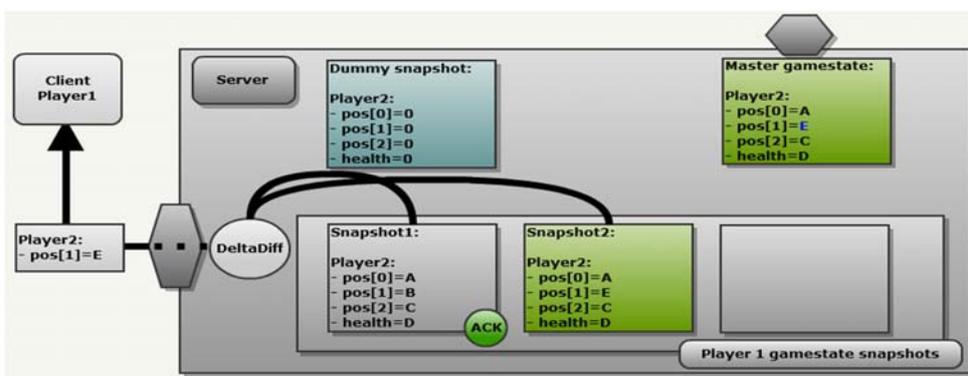


fig. 3.53

Nota: Debido a que cada campo es precedido por un bit de marcador (1= cambiado, 0 = sin cambios) la actualización parcial anterior sería usa 36 bits [0 1 32bitsNewValue 0 0]

Server Frame 3:

cómo el sistema trata los paquetes perdidos. Este es ahora el fotograma 3. Los clientes se han mantenido en el envío de comandos al servidor. El cliente ha perdido la vida y la salud es ahora igual a H. Pero Client no ha reconocido la última actualización. Quizás el UDP del servidor se perdió, tal vez el ACK del cliente se perdió, pero la línea de fondo es que no se puede utilizar, fig. 3.54.

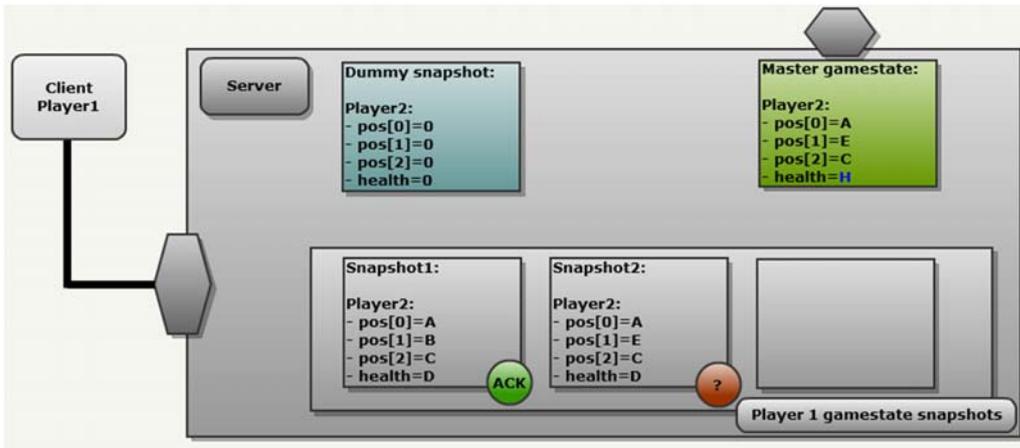


fig. 3.54

Independientemente del proceso sigue siendo el mismo, fig. 3.55:

1. Copie el gamestate Master en la siguiente ranura de historia del cliente: (índice 2): Se trata de Snapshot3
2. Comparar con la última instantánea reconocida válido (snapshot1).

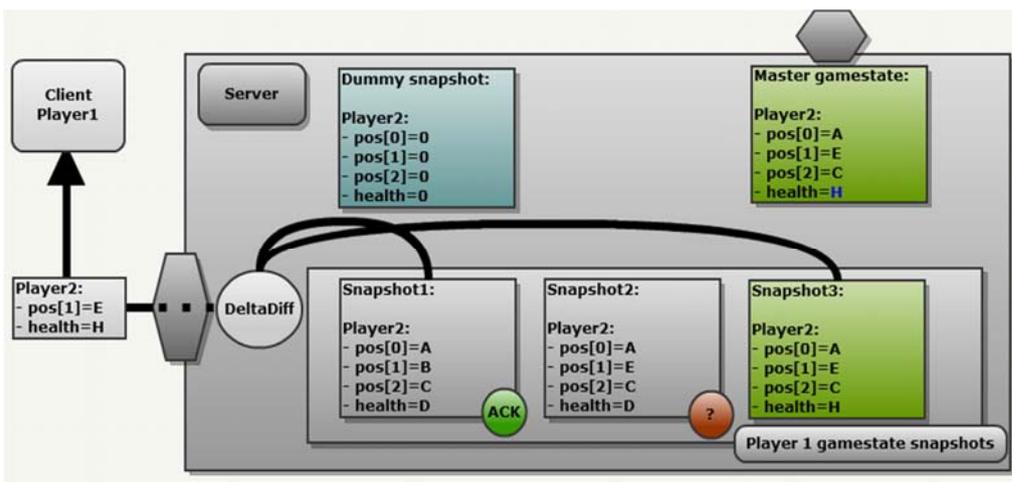


fig. 3.55

Como resultado, el mensaje enviado es parcial y contiene una combinación de cambios de edad y los nuevos cambios: (pos [1] = E y salud = H). Tenga en

cuenta que snapshot1 podría haber sido demasiado viejo para ser utilizado. En este caso el motor habría utilizado la "instantánea falsa" de nuevo, lo que resulta en una actualización completa.

La elegancia del sistema reside en su simplicidad. El mismo algoritmo automáticamente

- Generar actualización parcial o completa.
- reenvió de información antigua que no fueron recibidas y nueva información en un solo mensaje.

3.12.10 INTROSPECCIÓN DE MEMORIA CON C

Usted puede preguntarse cómo Quake3 está comparando capturas con la introspección ... ya que C no tiene introspección.

La respuesta es que cada uno de lugares sobre el terreno para un netField_t se prefabricado a través de una matriz y algunas directivas de preprocesamiento inteligente:

```
typedef struct {
    char *name;
    int offset;
    int bits;
} netField_t;

// using the stringizing operator to save typing...
#define NETF(x) #x,(int)&((entityState_t*)0)->x
netField_t entityStateFields[] =
{
    { NETF(pos.trTime), 32 },
    { NETF(pos.trBase[0]), 0 },
    { NETF(pos.trBase[1]), 0 },
    ...
}
```

El código completo de esta parte se puede encontrar en MSG_WriteDeltaEntity snapshot.c's. Quake3 ni siquiera sabe lo que es la comparación: Sólo sigue ciegamente el índice de entityStateFields, de compensacion y tamaño y envía la diferencia en la red.

3.12.11 MENSAJES FIABLES Y NO FIABLES

Si el sistema de capturas compensa datagramas perdidos en la red para UDP, algunos de los mensajes y comandos deben ser garantizados para ser entregados (cuando un jugador abandona o cuando el servidor necesita el cliente para cargar un nuevo nivel).

Esta garantía es abstraída por el NetChannel

3.12.12 LA MAQUINA VIRTUAL

Si los motores anteriores delegarán solo al gameplay para la máquina virtual, idtech3 en gran medida dependen de ellos para las tareas esenciales. Entre otras cosas:

- la interpretación es activada desde la maquina virtual del cliente.
- El mecanismo de compensación de retraso es en su totalidad en la máquina virtual del cliente.

3.12.13 ARQUITECTURA DE LA MAQUINA VIRTUAL

En Quake III una máquina virtual se denomina QVM: Tres de ellas están cargados en cualquier momento, vea fig 3.56.

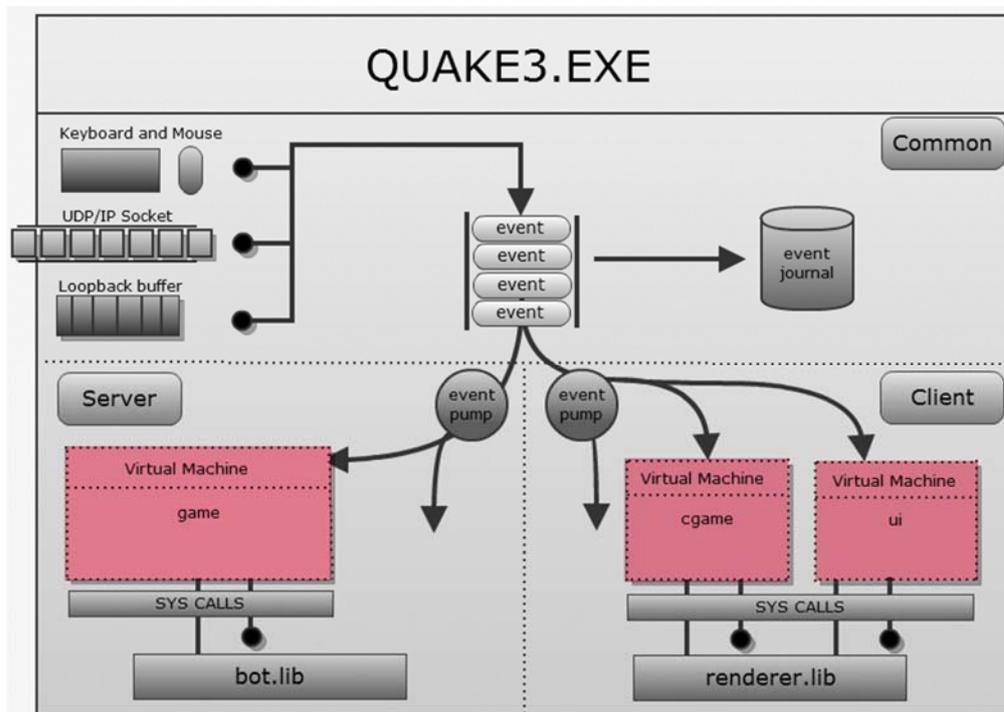


fig. 3.56

- Lado del cliente: Dos máquina virtual se cargan. Mensaje son enviados uno u otro dependiendo del gamestate:

Cgame: Recibe mensajes durante las fases de batalla. Realiza recortes de entidad, predicciones y activación de renderer.lib.

q3_ui: Recibe mensajes durante las fases de menú. Utiliza el sistema llamadas para elaborar menús.

- Lado del servidor:

game: recibe siempre el mensaje: Realice gamelogic e impulsa bot.lib para realizar la IA.

3.12.14 QVM INTERNAS

Antes de describir cómo se utilizan las QVMS, vamos a ver cómo se genera el código de bytes, vea fig. 3.57.

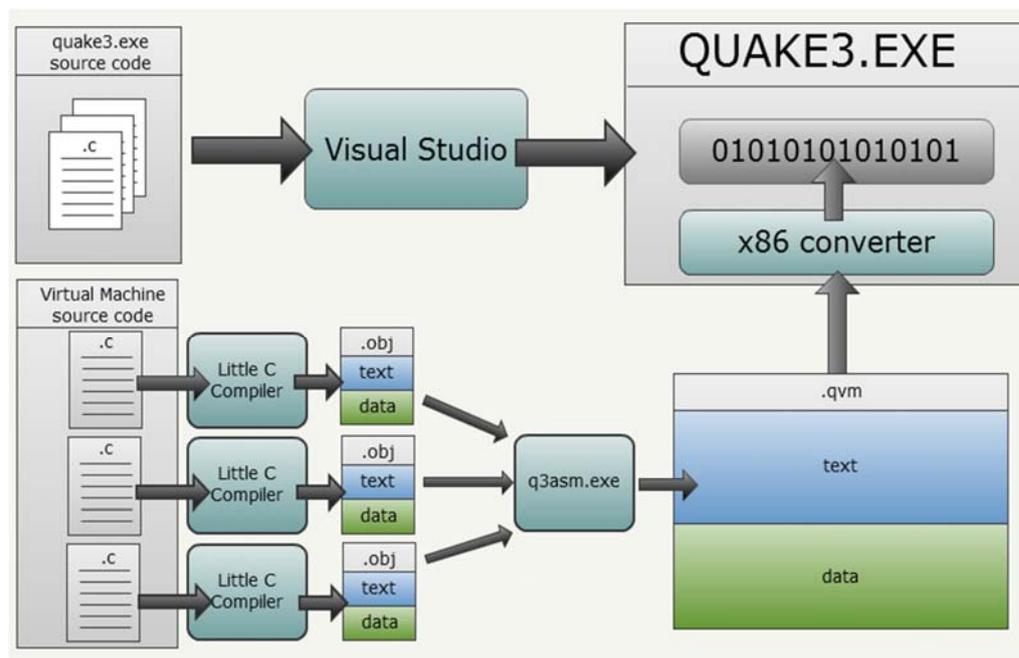


fig. 3.57

quake3.exe y su intérprete bytecode son generados a través de Visual Studio, pero el bytecode VM toma un camino muy diferente:

1. Cada archivo c (unidad de traducción) se compila de forma individual a través de LCC.
2. LCC se utiliza con un parámetro especial por lo que no emitirá un PE (Windows Portable Ejecutable), sino que es representación intermedia conjunto del texto basado de la pila de la máquina. Cada archivo producido presenta una sección de texto, datos y bss con las exportaciones e importaciones de símbolos.
3. Una herramienta especial del Software q3asm.exe de id toma todos los archivos de texto del ensamblado y los ensambla en un solo archivo de qvm. También transformar todo, desde texto a binario (para la velocidad en caso de que el nativo convertido no pueda surtir efecto). q3asm.exe también reconoce cuales métodos son llamadas de sistema y da a ellos un número de símbolo negativo.

4. Después de cargar bytecode binario, quake3.exe se convierte en instrucciones x86.

LCC INTERNOS

Aquí está un ejemplo concreto a partir de una función que queremos ejecutar en la máquina virtual:

```
extern int variableA;

int variableB;

int variableC=0;

int fooFunction(char* string){

    return variableA + strlen(string) }


```

Guardado en la unidad de traducción de module.c, lcc.exe es llamado con una etiqueta especial para evitar que se genere un objeto de Windows PE, sino más bien la salida de la Representación Intermedia. Este es el LCC. Obj salida de comparación de la función C superior.

```
data
export variableC
align 4
LABELV variableC
byte 4 0
export fooFunction
code
proc fooFunction 4 4
ADDRFP4 0
INDIRP4
ARGP4
ADDRLP4 0
ADDRGP4 strlen
CALLI4
ASGNI4
ARGP4 variableA
INDIRI4
ADDRLP4 0
INDIRI4
ADDI4
RETI4
LABELV $1
endproc fooFunction 4 4
import strlen
bss
export variableB
align 4
LABELV variableB
skip 4
import variableA


```

Algunas observaciones :

- El código de bytes se organiza en secciones (marcado en rojo): se puede ver claramente el bss (variables sin inicializar), datos (variables inicializadas) y el código (normalmente llamado texto, o lo que sea ..)
- Las funciones se definen a través de proc , sandwich endproc (marcado en azul).
- La Representación Intermedia de LCC es una máquina de pila: Todas las operaciones se realizan en la pila sin suposiciones hechas sobre registros del CPU .
- Al final de la frase LCC tenemos un montón de archivos de importación/exportación de variables/funciones.
- Cada instrucción comienza con un tipo de operación (es decir: ARGP4, ADDRGP4, CALLI4 ...). Todos los parámetros y el resultado se pasan en la pila.

Tal un archivo de texto se genera para cada uno. C en el módulo VM .

3.12.15 Q3ASM.EXE INTERNOS

q3asm.exe toma los archivos de texto LCC representación intermedios y los ensambla juntos en un archivo qvm, vea fig. 3.58:

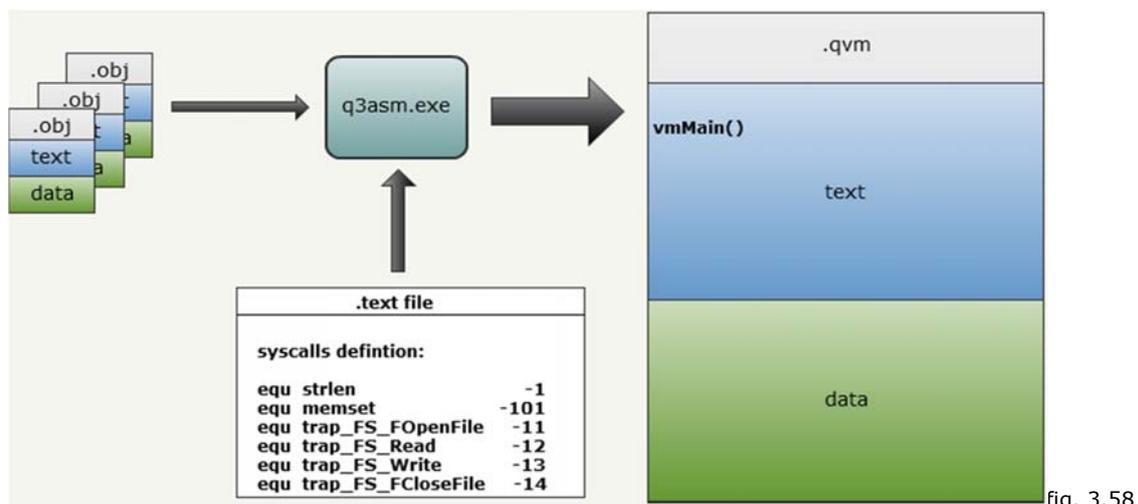


fig. 3.58

Varias cosas a destacar:

- q3asm da sentido a cada uno de los símbolos de importación/exportación a través de archivos de texto.
- Algunos métodos están predefinidos a través de un archivo de texto llamado sistema. Se puede ver [syscall for the client VM](#) y [for the Server VMs](#). El sistema de símbolos de llamada se les atribuye un valor entero negativo para que puedan ser identificados por el intérprete.

- q3asm Cambio la representación de texto a binario con el fin de ganar espacio y velocidad, pero eso es más o menos que, sin optimizaciones.
- El primer método para ser ensamblado DEBE ser vmMain ya que es el distribuidor de mensajes de entrada. Además debe estar ubicado en 0x2D en el segmento de texto del código de bytes.

3.12.16 COMO FUNCIONA QVM

- el dibujo ilustra el primer punto de entrada único y punto de salida único que actúe como despachador, vea fig. 3.59:

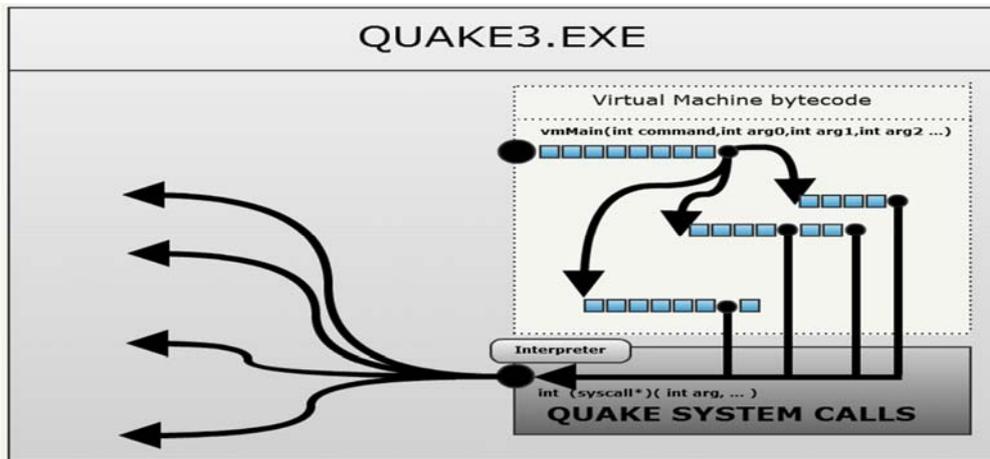


fig. 3.59

Algunos detalles:

Mensajes (Quake3 ->VM) se envían a la máquina virtual de la siguiente manera:

- Cualquier parte de Quake3 puede llamar VM_Call (vm_t * vm, int callnum, ...).
- VMCall toma hasta 11 parámetros y escribir cada valor de 4 bytes en el código de bytes de máquina virtual (vm_t * vm) desde 0x00 hasta 0x26.
- VMCall escribe el mensaje de identificación en 0x2A.
- El intérprete comienza a interpretar los códigos de operación 0x2D (donde vmMain fue colocada por q3asm.exe).
- vmMain actúan como un despacho y pasa el mensaje al método bytecode apropiado.

Se puede encontrar la lista de mensajes que se pueden enviar a la máquina virtual cliente y servidor VM (en la parte inferior de cada archivo).

Las llamadas al sistema (VM -> Quake3) salen de esta manera:

- El intérprete ejecuta los códigos de operación VM, uno tras otro (VM_CallInterpreted).

- Cuando se encuentra un código de operación CALLI4 comprueba el índice método int.
- Si el valor es negativo, entonces se trata de una llamada al sistema.
- El "sistema llamada función a puntero" (int (* llamada al sistema) (int * parms)) es llamado con los parámetros.
- La función a la que apunta llamada al sistema actúa como un despacho y ruta de la llamada al sistema a la parte derecha de quake3.exe

3.12.17 TEMA DE LA PRODUCTIVIDAD Y LA SOLUCIÓN

Con una larga cadena de herramientas tales, desarrollar código VM era difícil, vea fig. 3.60:

- La cadena de herramientas era lenta.
- La cadena de herramientas no se integró a Visual Studio.
- La construcción de un QVM implicó el uso de herramientas de línea de comandos. Era engorroso e interrumpió el flujo de trabajo.
- Con tantos elementos de la cadena de herramientas que era difícil identificar qué parte que tuvo la culpa en caso de errores.

Así idTech3 también tiene la capacidad de cargar un archivo DLL nativo para las partes VM y resuelto todo:

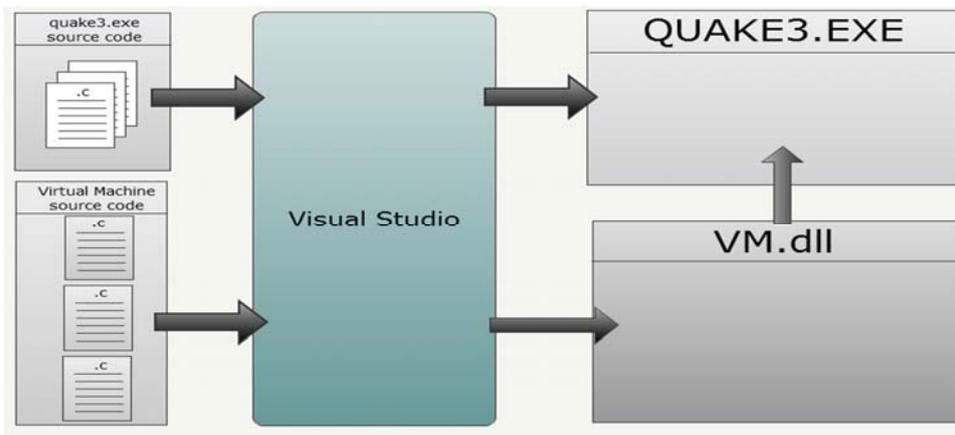


fig. 3.60

En general, el sistema VM es muy versátil ya que una máquina virtual es capaz de correr:

- Interprete de bytecode
- compilacion de instrucciones x86 de bytecode
- Código compilado de Windows

3.12.18 INTELIGENCIA ARTIFICIAL

La comunidad de modding escribió el sistema bot para cada versión de motores anteriores IDTECH. Dos de ellos eran muy famosos en su momento:

- Quake1 tenía Omicron.
- Quake2 tenía Gladiador.

Pero para idTech3 el sistema bot fue fundamental por que se iba a desarrollar en la empresa y estar disponible fuera de la caja. Pero las cosas no sucedieron exactamente como se esperaba durante el desarrollo: Para empeorar las cosas, un ingrediente fundamental del juego - los bots - faltaba. Motores de búsqueda eran personajes controlados por el ordenador. Un buen bot sería mezclarse con la acción y la carne la escena como un extra de robótica, así como interactuar con el jugador. Para Quake III, un combate a muerte único juego, los robots eran esenciales para la acción de un solo jugador. Eran implícitamente complejos porque tenía que comportarse como seres humanos. Carmack había decidido, por primera vez, a delegar el trabajo de creación de estos robots a otro programador en la empresa. Pero él no hizo un seguimiento. Una vez más, Carmack asume incorrectamente que todo el mundo era tan auto-motivado y hábil como él. Estaba equivocado. Cuando Graeme luchaba por contener en el trabajo, se descubrió que los robots fueron completamente ineficaces.

3.12.19 ARQUITECTURA DE INTELIGENCIA ARTIFICIAL

Al final, Esto probablemente explica por qué esta parte del código del servidor se encuentra en su propio proyecto: bot.lib, vea fig. 3.61:

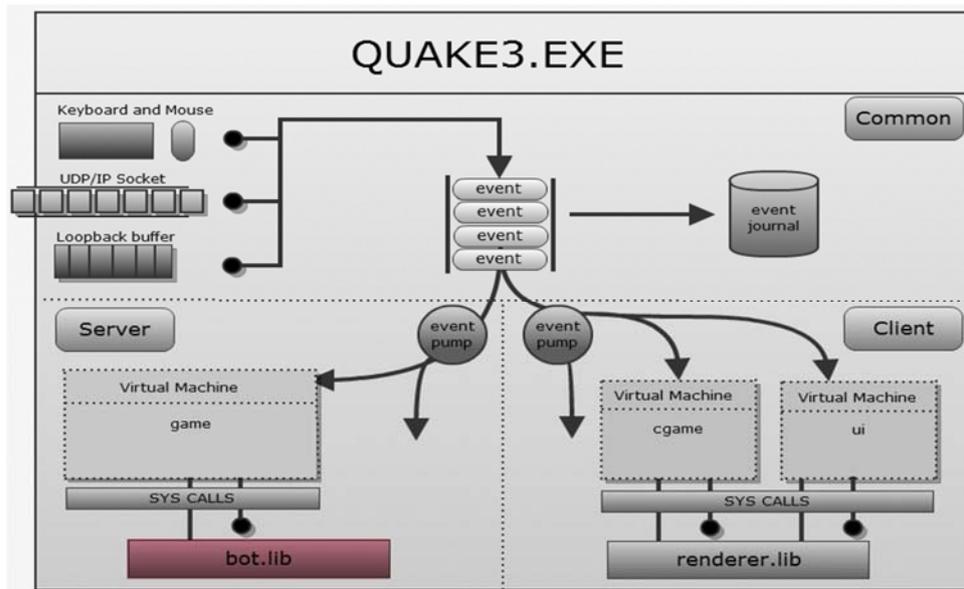


FIG. 3.61

3.13 HARDWARE Y PERIFERICOS

Todo juego sin importar si es físico o virtual tiene un hardware, y nos referimos a con que vamos a jugar. Un juego es un conjunto de ideas, con reglas y premios además de sus respectivos castigos pero que usaremos para que estas ideas puedan ser visualizadas y comprendidas pues fácil, elementos de hardware. Conocerlos a todos sería imposible, mejor utilizaremos las ideas más prácticas para conocerlos. Con la llegada del puerto serie universal al mercado de los dispositivos de computadora, se ha estandarizado todos los elementos de hardware y la compatibilidad es un estándar actual.

Muchos periféricos aplicados a los videojuegos comparten esa raíz y su costo se abarato exponencialmente durante la primera década del siglo XXI, es muy notorio este auge de compatibilidad por ser una estrategia de mercado que libra muchas barreras de exclusividad entre compañías competidoras.

No es rentable crear hardwares específicos para videojuegos por que se presta a ser un fracaso rotundo en las ventas por esa trampa de la exclusividad.

Dentro de los periféricos más comunes tenemos una categoría muy grande en diversidad que son los periféricos de control directo como:

Son dispositivos de hardware diseñados específicamente para el entretenimiento, son computadoras de propósito específico, las cuales sus componentes están enfocados a correr aplicaciones interactivas.

En este caso son los videojuegos.

Por razones históricas solo se mencionan algunas consolas por ser las primeras como lo fueron Atari®, Nintendo® que en los años 70's y 80's del siglo XX fue el posicionamiento de los videojuegos como una nueva forma de entretenimiento además de las famosas MAQUINITAS, consolas de videojuegos que con depositar una moneda uno adquiere el derecho a jugar.

Están disponibles en el mercado hardwares de 64 bits con la compatibilidad de correr aplicaciones de 32 bits.

Por mencionar algunas marcas son:

Game Boy®

Game Boy Advance®

GameCube®

Nintendo 64®

PC

PlayStation®

PlayStation 2®

PSP®

PlayStation 3®

PlayStation Portable®

Sega Dreamcast®

Sega Saturn®

Xbox®

Nintendo DS®

Un gamepad es un dispositivo de entrada usado para interactuar con un videojuego ya sea para consola o PC. El gamepad o control de mando permite moverse e interactuar con los elementos del juego para realizar las diversas acciones necesarias para cumplir los objetivos. La creación del primer gamepad vino acompañada del desarrollo de la primera consola de videojuegos en la década de los sesenta, La Primera consola que conto con un Gamepad fue la Nintendo Entertainment System. El desarrollo de un sistema de entretenimiento electrónico a base de imágenes interactivas en un televisor creó la necesidad de

contar con un dispositivo por medio del cual se logrará esta interacción. Así Ralph Baer, creador del primer sistema de videojuegos, acompañó su creación, la consola Odyssey, con un par de Joysticks (Palanca de mando) para el juego entre dos participantes. Los gamepads originalmente nacieron para las consolas en Forma de Palanca pero al llegar los videojuegos a las computadoras personales estos llegaron a ser también un dispositivo indispensable como el mismo teclado o el ratón. Un gamepad se caracteriza por ser un tablero con una o varias palancas o crucetas que pueden ser analógicas o digitales diseñadas para usarse con el dedo pulgar y una serie de botones generalmente colocados de su lado derecho cada uno con una función específica.

Desde la creación del Dual Shock (Segunda versión del controlador de PlayStation) se usaron controles llamados de dicha forma, así hasta ahora, el PlayStation 3 usa el DualShock 3 Sixaxis.

Kinect® para Xbox 360®, o simplemente Kinect (originalmente conocido por el nombre en clave «Project Natal»), es «un controlador de juego libre y entretenimiento» creado por Alex Kipman, desarrollado por Microsoft para la videoconsola Xbox 360, y desde junio del 2011 para PC a través de Windows 7 y Windows 8.3 Kinect permite a los usuarios controlar e interactuar con la consola sin necesidad de tener contacto físico con un controlador de videojuegos tradicional, mediante una interfaz natural de usuario que reconoce gestos, comandos de voz, y objetos e imágenes. El dispositivo tiene como objetivo primordial aumentar el uso de la Xbox 360®, más allá de la base de jugadores que posee en la actualidad. En sí, Kinect compite con los sistemas Wiimote con Wii MotionPlus® y PlayStation Move®, que también controlan el movimiento para las consolas Wii® y PlayStation 3®, respectivamente.

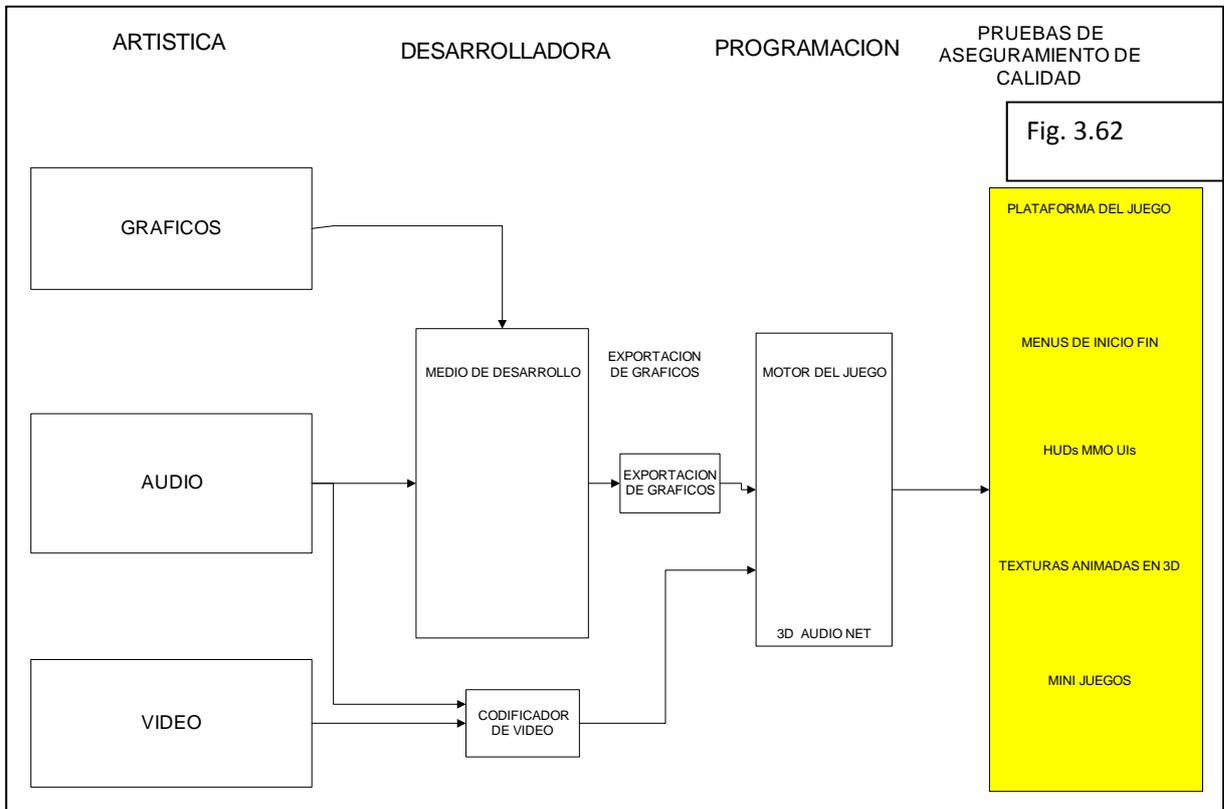
Kinect® fue lanzado en Norteamérica el 4 de noviembre de 2010 y en Europa el 10 de noviembre de 2010. Fue lanzado en Australia, Nueva Zelanda y Singapur el 18 de noviembre de 2010, y en Japón el 20 de noviembre de ese mismo año. Las opciones de compra para el sensor Kinect® incluyen: un paquete con el dispositivo mismo y la consola Xbox 360®, ya sea con una de 4 GB o 250 GB y el juego Kinect Adventures®, un paquete que incluye el sensor con el juego Kinect Adventures® y otro (por tiempo limitado) que incluye el dispositivo Kinect® con el juego Kinect Adventures® y un código que permite descargar el juego Child of Eden®. Este dispositivo fue creado por la necesidad de agregar la cultura del ejercicio físico en los jugadores. En los últimos 10 años ha habido un incremento de obesidad en la población en general por diversas causas.

El kinect surge como estrategia de marketing para ofrecer al mercado meta la posibilidad de ejercitarse mientras se divierte con una consola de juegos de Xbox 360®.

Esta etapa final corresponde generalmente a pruebas de aseguramiento de calidad en las diferentes configuraciones de hardware y compilaciones en las diferentes plataformas disponibles que existen para el videojuego.

Como son PC, GAMECUBE, PS3, PSP, XBOX360

La fig. 3.62 muestra la etapa final de pruebas de aseguramiento de calidad en el videojuego



3.13.1 DISPOSITIVOS MÓVILES

Se enfocan principalmente a teléfonos móviles, normalmente cada compañía produce su propio sistema operativo como NOKIA®, SONY ERICSON®, MOTOROLA®, SAMSUNG®, LG® entre otros.

Actualmente en la guerra de los sistemas operativos se busca estandarizar la plataforma general de telefonía móvil basada en el sistema operativo android.

Es muy complicado decir que sucederá con el futuro de los dispositivos móviles cuando la frontera entre las laptops y los teléfonos móviles se está volviendo difusa. Con una constante evolución de aplicaciones y plataformas los juegos siempre encontraran un lugar en cada dispositivo móvil que sea capaz de reproducirlos. Hoy en día la conectividad, la versatilidad, la migración de un sistema operativo a otro vuelve a las aplicaciones obsoletas en un santiamén así como los dispositivos móviles.

Sin embargo, la estrategia para resolver esta volatilidad del mercado se soluciona lanzando continuamente nuevos dispositivos con mayor versatilidad en su hardware.

4 MODELO DE NEGOCIOS

4.1 PLAN DE NEGOCIOS

El plan de negocios consiste en un instrumento para pedir financiamiento, es un documento que proyecta los alcances de un producto o servicio así como sus pros y sus contras, también muestra las fortalezas y debilidades en el escenario mundial.

Describiendo el producto se pretende crear un videojuego, la temática es un paseo en bicicleta en primera persona y con las opciones de multijugador, sus objetivos son capturar la imaginación del usuario mostrando escenarios reales de la localices visitadas, dentro de sus alcances se busca el realismo y la interactividad con el escenario, fomentar el ejercicio y la cultura del deporte en todo el mundo

El negocio de este producto es generar recursos a través de la publicidad promocionada en el contenido del producto. Los anunciantes estarán en tiempo real promocionando sus productos con sus respectivos logos, ofreciendo sus productos y servicios. También se promocionaran vestuarios, bicicletas, motocicletas, autos, camiones, aviones, barcos, paseos, sitios turísticos, playas, sitios de interés, entre otras muchas cosas que la ruta vaya mostrando como hoteles, restaurantes. Cada anuncio que sea clickeado mostrara lo que el patrocinador ofrece en tiempo real, y ahí es donde el juego ganara un centavo de dólar por cada clic que sea activado.

En una segunda etapa que estará en función del impacto del producto se pretenderá vender un periférico que actuara como simulador para montar una bicicleta real y dentro de sus características están mostrar imágenes de una experiencia real en bicicleta, con sus respectivos arneses para montar cualquier bicicleta estándar, enlazado este videojuego a una consola o pc controlara la mecánica del tour.

Una proyección del futuro cercano son las mejoras y abaratamiento en costos supondrá la evolución de la tecnología.

Este producto busca posicionarse en el mercado por ser de distribución gratuita y operar en todas las plataformas disponibles así como computadoras, consolas y dispositivos móviles. Además de ofrecer entretenimiento y oportunidades de negocio a terceros. Lo más importante es posicionarlo como el juego que vende a través de paseos en bicicleta virtuales por todo el mundo. Se pide apoyo para financiar la parte de software.

4.1.1 RESUMEN EJECUTIVO

Overlands es un software , consiste en un videojuego de paseos y recorridos virtuales por México.

En su etapa inicial será solo en las principales vialidades de la cd de México, si hay aceptación se expandirá hacia otras ciudades.

La experiencia del juego consiste en una simulación del paseo con todos los eventos que se pueden generar en una experiencia real.

El usuario podrá hacer su aportaciones para mejorar la experiencia del juego

La finalidad es ofrecer entretenimiento a todo público que le guste pasear y ver lo que México ofrece. Cada tour cubre un recorrido donde se observaran y en algunos casos interactuar con el escenario, con la finalidad de hacer una experiencia realista e interactiva.

Con gráficos de alta calidad y fotografías reales de los lugares visitados mostrara la parte virtual de escenarios conocidos, así como paseos a pie de los lugares visitados como instalaciones deportivas, playas, centros comerciales, mostrara toda la gama de oportunidades para anunciarse en tiempo real. Con la conectividad permanente en la nube se podrán ofrecer productos y servicios y sus respectivos precios en tiempo real.

El sistema de negocios propone cobrar una comisión por cada anuncio que el jugador active, así aunque lo vea y no lo active no se cobrara, así todos ganamos. La particularidad de este sistema de negocios es que el cliente verdadero son todos los anunciantes y el cliente secundario son los jugadores . a ambos se les ofrecerá la mejor experiencia visual e interactiva en juegos de simulación virtual.

Todos los anunciantes que deseen participar deberán tener un sistema de información que soporte esta plataforma para que su publicidad este vigente y sus productos y servicios tengan valides en el momento de que el usuario les de clic.

El juego mostrara todo tipo de publicidad verdadera, siempre que el anunciante tenga contrato para estar en OVERLANDS,

Este producto plantea la oportunidad de negocios ofreciendo espacios publicitarios en tiempo real con participación directa del jugador.

Durante el recorrido muestra sitios de interés común

Descripción de la compañía, GAMESKAPE ENTERTAINMENT, es una compañía de software dedicada al entretenimiento, ofreciendo los mejores productos para el entretenimiento y publicidad. Usando como oportunidad de negocio la evolución de la conectividad y el abaratamiento del WIFI, y la proliferación de los dispositivos permanentemente conectados a internet.

No se requieren instalaciones físicas, utilizamos la modalidad de tele trabajadores.

Este producto es gratis, su distribución se hara a través de torrentz o del mismo sitio web, www.gameskape.com.

4.1.2 GESTIÓN DE LA CORPORACIÓN

El plan de operaciones consiste en las características del negocio, todos son tele trabajadores, vea fig. 4.1.

1. planeación con visión a largo plazo
2. Aumento de la calidad del producto.
3. Disminución de costos
4. Capacitación constante de la mano de obra.
5. Monitoreo constante de las necesidades cliente
6. Proyección financiera y estudio del medio ambiente externo de la empresa.

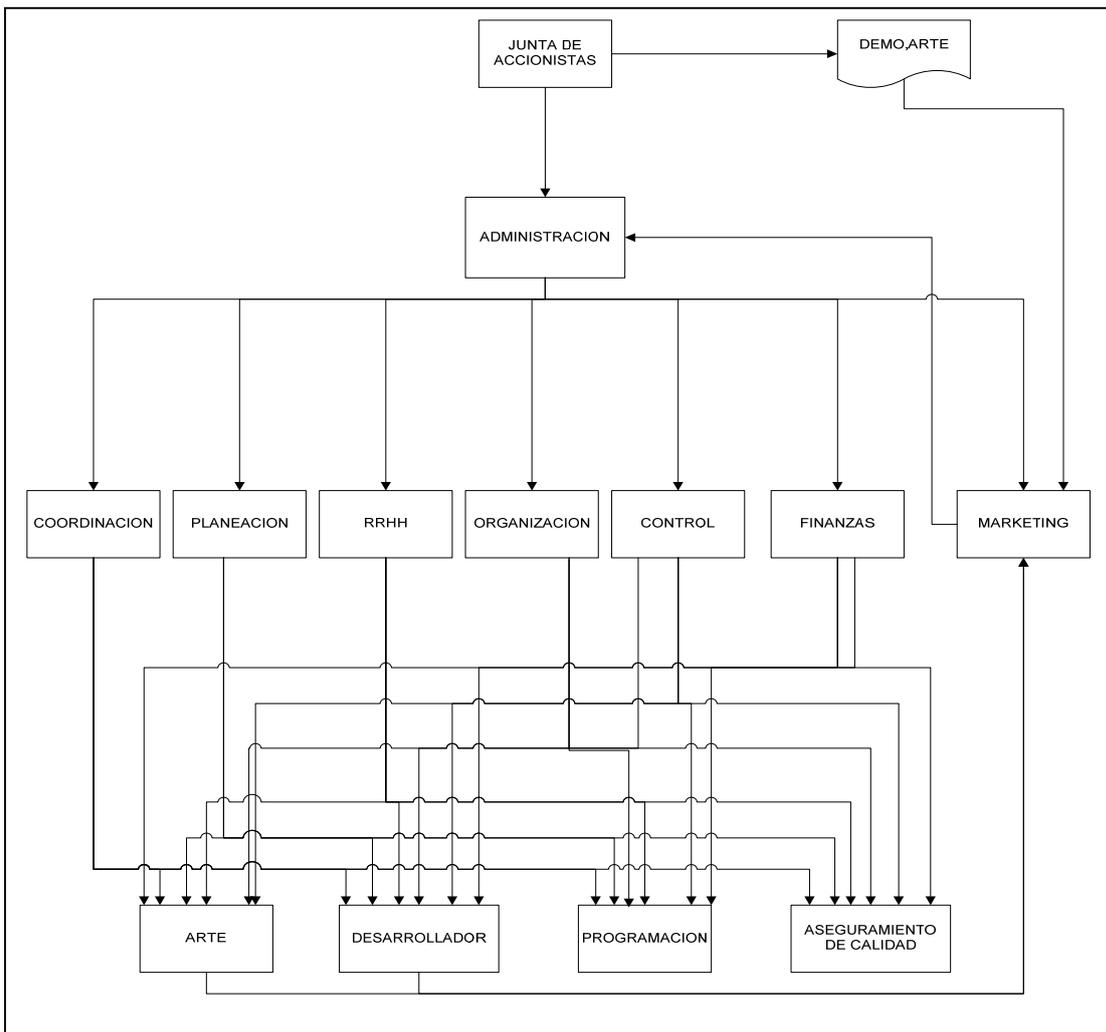


Fig. 4.1 muestra a los grupos de trabajo y sus respectivos responsables. Cada grupo de trabajo está representado por un responsable que entregara resultados a otros grupos de trabajo y a sus superiores.

La finalidad de este mapa jerárquico es proyectar el enfoque sistémico de la funcionalidad de la empresa a través de su capital humano.

DESCRIPCIÓN DEL PRODUCTO

Es un videojuego en la modalidad de primera persona, multijugador, para jugarse en línea permanentemente.

El videojuego consiste en montar una bicicleta y hacer recorridos en las modalidades de paseo, tour, competencia en ciudades, pueblos, montañas, colinas, playas, mares y autopistas, todo con sus respectivos accidentes geográficos así como la naturaleza propia de la zona.

Mostrará situaciones reales de tráfico, peligros, climas, fallas mecánicas.

La interface gráfica mostrara datos en tiempo real de las condiciones del recorrido como distancia, velocidad, aceleración, altitud, coordenadas, muestra escenarios reales de alta definición y calidad de fotografía.

Estos recorridos tienen planeación al iniciar el juego

El patrocinio del juego estará soportado por todos los anunciantes que deseen participar de forma estática o dinámica.

La publicidad vendrá Fabricantes de:

automóviles, autobuses, transportes de carga, aviones

Espectaculares

Centros comerciales, tiendas, anuncios particulares

Restaurantes

Motocicletas

Bicicletas

También tendrá la particularidad de que cada fabricante podrá enviar su producto virtual para ser colocado en la plataforma de juego con su respectivo plan de venta.

Cada producto anunciado aquí tendrá esa particularidad de anunciarse con un respectivo sistema de ventas.

El videojuego será gratis en su descarga y uso. La condición es que este conectado a internet para que funcione el sistema de publicidad en tiempo real.

Su etapa inicial cubrirá las principales vialidades de la cd de mexico y su área metropolitana.

Otra particularidad será que los usuarios podrán contribuir con la mejora del escenario haciendo sus propia publicidad, aportando nuevos vehículos automotores, realismo del paisaje en general, aportando noticias con respecto a lo que acontece en determinadas zonas, incrementando el esquema de detalle de las rutas.

Se pretende un mundo virtual en bicicleta con todas las características del mundo real.

4.1.3 PLAN DE MERCADOTECNIA

El plan de mercadotécnica consiste en promocionar el producto a través de todos los motores de búsqueda de internet disponibles en la www. Estos motores de búsqueda en internet tienen casi las mismas políticas de pagos, estos motores de búsqueda son los siguientes disponibles hasta febrero de 2014, vea fig. 4.2.

A9.com, Accona, Alexa Internet, Alleba, AltaVista, Ansearch, Apache Droids, Archie, Ask.com, AskMeNow, Aspseek, Astalavista.box.sk, Ayna (buscador), Baidu, BASE, Become.com, Bing, Bing Health, Bing Noticias, Bing Shopping, Bing Travel, Bing Video, Bing xRank, Blackle, Blekko, Brainboost, Google, Business.com, Búsqueda multimodal, Búsqueda por voz, Carrot2, ChaCha (motor de búsqueda), ChemXSeer, CiteSeer, CiteSeerX, Citysports.de, Cuil, Cunoticias, Dandelon.com, Dialnet, Digital.CSIC, Directory of Open Access Journals, Dogpile, DuckDuckGo, Everything software, Exalead, Goo, Guruji, Hotbot, Inktomi, Ixquick, Karlsruher, Virtueller Katalog, Knowledge Graph, Lycos, Medes, Metabuscador, Miner.hu, Motor de búsqueda, Buscador visual, Ms. Dewey, Multibuscador, Naver, Nestoria, NexTag, Nutch, Obtención de preferencias, Olé (buscador), Palabra clave (motor de búsqueda), Personalización de contenidos, PolySpot, PubSub, Quaero, Query by Humming, Rambler (portal), Redalyc.org, Rediff, Search based applications, Seekport, Shazam, Sphinx (buscador), Splog, Sproose, Tame.it, Technorati, Teoma, Trovator, TrustRank, Virgilio.it, Volunia, Walla!, WebCrawler, Wikia Search, Wikiseek, Wikiwix, WorldCat, YaCy, Yahoo! Search, Yasni

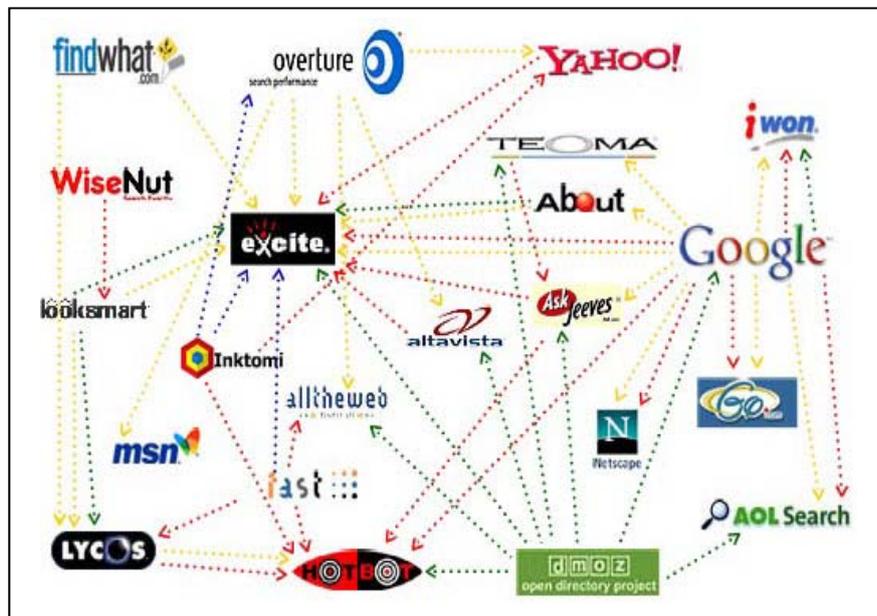


Fig.4.2

Por ejemplo google adwords ofrece planes de mercadotecnia completos y ajustados a la medida del producto.

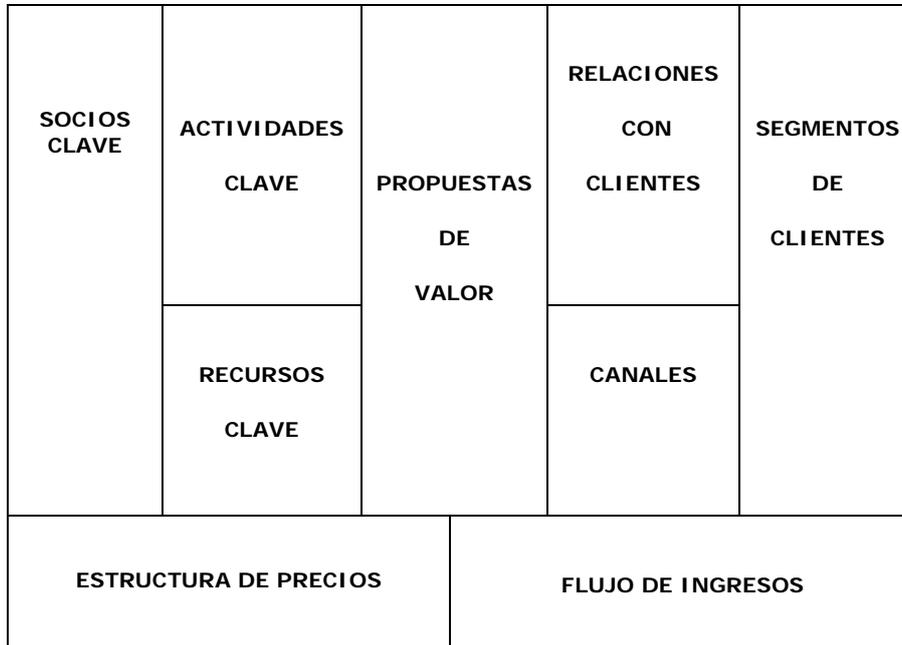
4.2 MODELO DE NEGOCIOS

El jugador no es el cliente final, es el cliente secundario pero habrá que satisfacerlo con producto de calidad en todos sus aspectos porque aquí es donde entra el cliente principal que es el anunciante que pague por anunciar sus productos y servicios dentro de la plataforma de juego, el anunciante solo pague por cada click que el usuario haga para ver su publicidad y así todos ganamos. Esta es la idea principal del modelo de negocios, cuidar y ofrecer la máxima calidad en servicio de este producto. Como es un producto de distribución gratuita con las máximas prestaciones, su única restricción es que deberá estar permanentemente conectado a la nube para que pueda funcionar. Los socios clave brindaran toda la infraestructura tecnológica para que la plataforma funcione. Las actividades clave son la publicidad de productos y servicios de terceros. Los recursos clave son la evolución constante de la tecnología y el abaratamiento de la nube. Flujos de ingreso, es indeterminado, sin embargo por cada sesión del juego y que el jugador vea la publicidad de algún anunciante, al darle clic, el juego ganara un centavo de dólar que el anunciante pague. Las actividades clave son la perfecta ejecución del juego con publicidad en tiempo real.

Los aliados clave son todos los participantes en este negocio con sus respectivas unidades de negocio como son: finanzas, relaciones públicas, asuntos de negocio, mercadotecnia, ingeniería, patrocinadores.

El principal activo de este negocio son los atractivos visuales y la jugabilidad que dispone la plataforma misma.

Fig.4.3



Este diagrama muestra las relaciones de las unidades estratégicas y su relación, fig. 4.3.

4.2.1 RESUMEN EJECUTIVO

“Un estilo, una actitud, un talento que se asume y se aprende desde pequeño para descubrir que es tan importante ganar como respetar al adversario, y que hay que ganar resistiendo ante la adversidad y buscando la belleza”.

Que hay que ser elegantes ante la competencia. Hay que saber ganar pero también hay que saber asumir la derrota. Que hay que ser solidarios y sacrificados.

En concreto la visión del corporativo es crear el concepto del siglo XXI. Un proyecto; ser líderes en el terreno del entretenimiento, tener los productos y una economía acorde con el prestigio de la marca y consolidar su imagen”. “lo más importante que tiene el corporativo es su imagen y prestigio y esa imagen de GAMESKAPE, de sentimiento que esta fomentada por el espíritu de competir en un mundo cambiante. Una identidad que no debemos de perder, que debemos fomentar y desde esos objetivos tan entrañables, es un hecho palpable” ser el concepto del siglo XXI, se consigue con el éxito corporativo y económico, logrando ganar mercados y siendo solvente.

Pues como se dijo anteriormente, estamos en un mundo que evoluciona rápidamente, donde es fácil caer por la competencia, pero nuestro espíritu de lucha es el que nos mantiene a la vanguardia en nuevos conceptos.

Sin visión y misión una empresa no puede existir, es como estar sin rumbo y es fácil perderse, porque no se sabe hacia dónde ir, es como perder el tiempo.

La idea es crear el concepto del entretenimiento digital del siglo XXI, y se logra creando la marca.

La visión de GAMESKAPE es ser el concepto en entretenimiento digital del siglo XXI, con productos innovadores inspirados de la imaginación del cliente.

La misión es ofrecer los mejores productos para el entretenimiento digital, pensando en ti como nuestra principal responsabilidad, creando los productos más innovadores en su clase.

Las estrategias se enfocan a estos rubros, a los productos, a la promoción, al tipo de cliente, al desarrollo de innovaciones. Estas estrategias son un conglomerado de directrices obtenidas de la visión y la misión de la empresa. Aquí se busca el crecimiento de la empresa para posicionarla como el concepto del entretenimiento. Cada estrategia tiene como finalidad ganar el aprecio del cliente y permanecer como la mejor opción en su clase. El punto inicial para poder diseñar un modelo de BSC es la definición de la visión y estrategias, no es sencillo, sin embargo, normalmente la gente llega a acuerdos en este sentido, pues son muy genéricos los objetivos de crecimientos que se persiguen. Donde la mayoría tropieza es en la interpretación de las estrategias. Por ejemplo, el crecimiento de la empresa difícilmente sería cuestionable, pero para algunos, el crecimiento puede significar aumento geográfico, para otros mayores ingresos o incremento de empleados. Llegar a consensos sobre las estrategias lleva a establecer tanto objetivos como indicadores que midan los objetivos.

El entendimiento compartido de lo que es o ha de ser el corporativo no requiere mucha discusión. Este entendimiento facilita la toma de decisiones. En su implementación puede haber divergencia de opiniones. Pero no hay que

confundir las decisiones estratégicas (el que) con las implementaciones (el cómo). La imagen que pretende dar el corporativo se podría concretar como la de una empresa con clase, que le ha llevado a ser considerada como el mejor corporativo de entretenimiento y en el que es un desafío construir una historia de valor a base de marca y contenido, pero con unos valores universales que van mucho más allá del entretenimiento, como así lo pretende reflejar. El espíritu de superación y el respeto por la competencia son los valores principales del corporativo.

Entendida la visión y estrategias de la empresa es posible determinar los objetivos que hay que cumplir para lograr la estrategia y aterrizarlos en indicadores. Es importante que los indicadores no controlen la actividad pasada solamente, los indicadores deben reflejar los resultados muy puntuales de los objetivos, pero también deberán informar sobre el avance para alcanzar esos objetivos. Esto es, la mezcla balanceada entre indicadores de resultados e indicadores de actuación es lo que permitirá comunicar la forma de conseguir los resultados y, al mismo tiempo, el camino para lograrlo. Como resaltan Kaplan y Norton: "Resultados son los indicadores históricos, indicadores de la actuación son indicadores previsionales."

Otro aspecto que hay que resaltar es el número de indicadores que deberán formar parte del escenario principal o maestro. Según Kaplan y Norton, un número adecuado para esta labor es de 7 indicadores por perspectiva y si son menos, mejor. Se parte de la idea de que un Tablero con más de 28 indicadores es difícil de evaluar cabalmente, además de que el mensaje que comunica el BSC se puede difuminar y dispersar los esfuerzos en conseguir varios objetivos. El ideal de 7 indicadores por perspectiva no significa que no pueda haber más, simplemente pueden consultarse en caso de hacer una análisis más a conciencia, pero los mencionados serán aquellos que se consulten frecuentemente y puedan, verdaderamente, indicar el estado de salud de la compañía o área de negocio.

La definición de balance Scorecard (BSC, Kaplan y Norton, 1996) es un instrumento que traduce la misión y la estrategia de la organización en una amplia colección de métricas e indicadores. Posteriormente proporciona la estructura necesaria para servir como instrumento de control y de sistema de implantación de la estrategia.

4.2.2 VISIÓN, MISIÓN Y ESTRATEGIAS DE GAMESKAPE

."Un estilo, una actitud, un talento que se asume y se aprende desde pequeño para descubrir que es tan importante ganar como respetar al adversario, y que hay que ganar resistiendo ante la adversidad y buscando la belleza".

Que hay que ser elegantes ante la competencia. Hay que saber ganar pero también hay que saber asumir la derrota. Que hay que ser solidarios y sacrificados.

En concreto la visión del corporativo es crear el concepto del siglo XXI. Un proyecto; ser líderes en el terreno del entretenimiento, tener los productos y una economía acorde con el prestigio de la marca y consolidar su imagen". "lo más importante que tiene el corporativo es su imagen y prestigio y esa imagen de GAMESKAPE, de sentimiento que esta fomentada por el espíritu de competir en un mundo cambiante. Una identidad que no debemos de perder, que debemos fomentar y desde esos objetivos tan entrañables, es un hecho palpable" ser el concepto del siglo XXI, se consigue con el éxito corporativo y económico, logrando ganar mercados y siendo solvente.

Pues como se dijo anteriormente, estamos en un mundo que evoluciona rápidamente, donde es fácil caer por la competencia, pero nuestro espíritu de lucha es el que nos mantiene a la vanguardia en nuevos conceptos.

Sin visión y misión una empresa no puede existir, es como estar sin rumbo y es fácil perderse, porque no se sabe hacia dónde ir, es como perder el tiempo.

La idea es crear el concepto del entretenimiento digital del siglo XXI, y se logra creando la marca.

La visión de GAMESKAPE es ser el concepto en entretenimiento digital del siglo XXI, con productos innovadores inspirados de la imaginación del cliente.

La misión es ofrecer los mejores productos para el entretenimiento digital, pensando en ti como nuestra principal responsabilidad, creando los productos más innovadores en su clase.

Las estrategias se enfocan a estos rubros, a los productos, a la promoción, al tipo de cliente, al desarrollo de innovaciones. Estas estrategias son un conglomerado de directrices obtenidas de la visión y la misión de la empresa. Aquí se busca el crecimiento de la empresa para posicionarla como el concepto del entretenimiento. Cada estrategia tiene como finalidad ganar el aprecio del cliente y permanecer como la mejor opción en su clase. El punto inicial para poder diseñar un modelo de BSC es la definición de la visión y estrategias, no es sencillo, sin embargo, normalmente la gente llega a acuerdos en este sentido, pues son muy genéricos los objetivos de crecimientos que se persiguen. Donde la mayoría tropieza es en la interpretación de las estrategias. Por ejemplo, el crecimiento de la empresa difícilmente sería cuestionable, pero para algunos, el crecimiento puede significar aumento geográfico, para otros mayores ingresos o incremento de empleados. Llegar a consensos sobre las estrategias lleva a establecer tanto objetivos como indicadores que midan los objetivos.

El entendimiento compartido de lo que es o ha de ser el corporativo no requiere mucha discusión. Este entendimiento facilita la toma de decisiones. En su implementación puede haber divergencia de opiniones. Pero no hay que

confundir las decisiones estratégicas (el que) con las implementaciones (el cómo). La imagen que pretende dar el corporativo se podría concretar como la de una empresa con clase, que le ha llevado a ser considerada como el mejor corporativo de entretenimiento y en el que es un desafío construir una historia de valor a base de marca y contenido, pero con unos valores universales que van mucho más allá del entretenimiento, como así lo pretende reflejar. El espíritu de superación y el respeto por la competencia son los valores principales del corporativo.

Entendida la visión y estrategias de la empresa es posible determinar los objetivos que hay que cumplir para lograr la estrategia y aterrizarlos en indicadores. Es importante que los indicadores no controlen la actividad pasada solamente, los indicadores deben reflejar los resultados muy puntuales de los objetivos, pero también deberán informar sobre el avance para alcanzar esos objetivos. Esto es, la mezcla balanceada entre indicadores de resultados e indicadores de actuación es lo que permitirá comunicar la forma de conseguir los resultados y, al mismo tiempo, el camino para lograrlo. Como resaltan Kaplan y Norton: "Resultados son los indicadores históricos, indicadores de la actuación son indicadores previsionales."

Otro aspecto que hay que resaltar es el número de indicadores que deberán formar parte del escenario principal o maestro. Según Kaplan y Norton, un número adecuado para esta labor es de 7 indicadores por perspectiva y si son menos, mejor. Se parte de la idea de que un Tablero con más de 28 indicadores es difícil de evaluar cabalmente, además de que el mensaje que comunica el BSC se puede difuminar y dispersar los esfuerzos en conseguir varios objetivos. El ideal de 7 indicadores por perspectiva no significa que no pueda haber más, simplemente pueden consultarse en caso de hacer una análisis más a conciencia, pero los mencionados serán aquellos que se consulten frecuentemente y puedan, verdaderamente, indicar el estado de salud de la compañía o área de negocio.

La definición de balance Scorecard (BSC, Kaplan y Norton, 1996) es un instrumento que traduce la misión y la estrategia de la organización en una amplia colección de métricas e indicadores. Posteriormente proporciona la estructura necesaria para servir como instrumento de control y de sistema de implantación de la estrategia.

4.2.3 PROPUESTAS DE VALOR

Construir una empresa es todo un reto, pero transformarla en una empresa de espectáculos es un concepto más amplio de negocios, lo que trae un incremento en los beneficios una nueva versión del liderazgo y la reinención del concepto de competitividad.

Para crear una marca que se transforme en el concepto va más allá del conocimiento usual, es impactar al mercado con productos que capturen la imaginación del cliente, es abordar la psique de cada comprador convenciéndolo de que nuestro producto valdrá aun después de su uso, que tendrá más valor que cuando lo compro.

Construir una marca es crear valor agregado con el paso del tiempo, en el mundo del videojuego es muy posible que después del lanzamiento del producto quizá pocos recuerden el título, pero los productos de mercadotecnia como juguetes, figuras coleccionables, enseñanzas filosóficas entre otros productos secundarios sean recordados por los usuarios.

La idea es permanecer en los recuerdos del comprador con un buen sabor de boca.

Esto es muy difícil de comprender por qué cada cabeza es un mundo, entonces estandarizar es lo más fácil y confiable para obtener resultados. Por ejemplo, de los videojuegos que han hecho historia fueron STREET FIGHTER®, principalmente el 2, los títulos posteriores solamente fueron presentados para llenar el hueco en un mercado virgen donde renacía el interés por el videojuego.

Cabe mencionar que una marca es sinónimo de prestigio, de orgullo, por sus productos y esto se ve reflejado en sus ventas. El éxito en el entretenimiento es parte de una filosofía enfocada al cliente como principal valor de ser de la empresa, ofreciendo lo mejor para todos los compradores. Es poniendo atención a los gustos de los clientes y darles lo que buscan, ofrecerles un producto que llene sus expectativas y capturando su mente, así es como la ilusión se crea.

Ser solventes es todo un reto, es todavía más difícil porque los inversionistas cuidan su dinero y quieren ver resultados tangibles traducidos en ingresos y no en estadísticas o cifras manipuladas. Para ser solvente se tienen que tener otras entradas de dinero para amortiguar las fluctuaciones del mercado, estas entradas provienen de fuentes como mercadotecnia de productos, conferencias, exposiciones, productos secundarios.

4.2.4 CANALES PARA LLEGAR A ESOS CLIENTES

Hoy en día la internet ha revolucionado al mundo en la última década, ha sido la más grande explosión de la evolución humana en relación al tiempo.

Internet para tener fuerza ha tenido que ofrecer elementos gratuitos para su difusión y aceptación.

Queremos decir que la aceptación es la gran variedad de recursos que ofrece para todo. Uno de esos recursos es que es un medio masivo y relativamente barato para ventas. Como canal de ventas ha transformado a los esquemas tradicionales como son por catálogo, anuncios en medios impresos, radio, tv, entre los más comunes y conocidos.

Para configurar una campaña de marketing se tiene que plantear algunos elementos básicos como son los siguientes. Esto normalmente corresponde al departamento de marketing porque son ellos los que manejan los canales de ventas, sin embargo no está por demás conocer la estructura de marketing. Estos puntos señalados abajo corresponden a las herramientas que google adwords ofrece, sin embargo son similares a los de cualquier otro motor de búsqueda como es TERRA®, YAHOO®, BING®, ASK®, LYCOS®, ALTAVISTA®, MSN®, TEOMA®, EXCITE®, ALEXA®, FAST®, LOOKSMART®, OVERTURE®, NETSCAPE®, HOTBOT® y los que vengan en el futuro.

Crear anuncios y campañas:

Configurar una campaña nueva

Configurar grupos de anuncios

Anuncios de texto

Anuncios gráficos

Anuncios de vídeo

Anuncios para móviles

Otros tipos de anuncios

4.2.5 MODELO DE NEGOCIOS Y EL ANÁLISIS DE MERCADO

Iniciamos de una preponderancia; ¿qué es lo que hacemos?, ¿qué es lo que hicimos? y ¿qué es lo que haremos?

El modelo de negocios es proporcionar al mercado lo que más se venda y sacar los productos con rapidez, en esta época el crédito es el modelo de negocios mundial, entonces toda persona que tenga una tarjeta de crédito debito podrá comprar y obtener su producto instantáneamente (eso dependerá de la velocidad y ancho de banda de su conexión a internet).

Como empresa desarrolladora de productos para el entretenido digital, tales como videojuegos buscamos ser del gusto del público ofreciendo productos de clase mundial. Con temáticas variadas y para todos los públicos.

La idea es conservar al cliente con productos que promuevan valores como la salud, el desarrollo personal, incentiven la inteligencia, promuevan el liderazgo, la cultura hacia las artes, el aprecio a la tecnología, las ciencias naturales, a las ciencias aplicadas, a las ciencias sociales, a la filosofía, a los valores morales, estética, música, equilibrio, armonía. Es un todo lo que involucra el diseño de productos enfocados al público

Esta gama de estándares busca el posicionamiento, el precio, la plaza, el producto, la promoción. Entonces el modelo de negocios ya tiene una razón de ser, ya tiene misión y visión, tiene estrategia basada en valores hacia el cliente. Entonces todos los esfuerzos están enfocados a retener al cliente con productos de clase mundial, ese es el modelo de negocios, pero como funciona esto.

Funciona lanzando al mercado muchos videojuegos con especiales características que los diferencian para observar a cuales segmentos de mercado impactan más.

Calidad total.

Servicios de calidad al cliente.

Responsabilidad total.

4.2.6 RESULTADOS DE UN TRABAJO DE EQUIPO

¿Qué es trabajar en equipo?, existirían varias definiciones, sin embargo es la más intuitiva, diríamos que es filosofía y también que no hay definición clara, si en algo estamos de acuerdo es que cada persona tiene su definición personal, entonces no tenemos nada más que palabras, entonces que sigue, pues bien fácil, trabajar en equipo son actitudes y acciones, somos lo que hacemos y no lo que decimos, entonces hablamos de algo más profundo que inicia con nosotros mismos y es porque tenemos consciencia de lo que queremos y deseamos lograrlo. La idea se ha transformado en hechos, porque nos preguntamos hacia dónde vamos en nuestro camino hacia el final de nuestros días, la pregunta es cómo queremos morir, si de rodillas pidiendo clemencia o de pie. Esto es lo que nos da la fuerza de continuar y enfrentarnos a la vida por un futuro mejor. Ahora encontrar esas personas que también comparten ese ideal es lo difícil mas no imposible, pero porque todo esto, trabajar en equipo es una sinergia de compromisos y hechos que se cumplen en tiempo y forma donde las ideas conforman al grupo de poder para crear la empresa y darle fuerza, es el reflejo de la actitud para enfrentar los retos como son posicionarse en el mercado, la competencia desleal, la piratería, el espionaje industrial, el plagio de ideas, prototipos, modelos, empleados desleales, apatía, ansiedad, inseguridad, reajustes económicos, nuevas legislaciones, entre otras cosas.

Y el reto verdadero es si caes volver a levantarte, y si no caes, es crecer.

Es muy importante saber lo que somos para tener la visión correcta de lo que queremos, así es como se mejoran las infraestructuras, de ahí parte la evolución de lo que queremos y de lo que tenemos, las infraestructuras empiezan con nosotros mismos, la disciplina y el trabajo continuo, aprender de los errores.

En este contexto, en el mundo de los negocios y la libre empresa el ideal es incrementar los ingresos a nuestra compañía, eso como lo hacemos, pues es fácil teniendo la idea de cómo pero como cada empresa es diferente no se puede estandarizar los protocolos, sino que estos se diseñan en función a como mejor resulte en el escenario mundial, incrementar las ganancias es el resultado de trabajar en equipo es prueba y error, es conocer a la competencia y conocernos a nosotros mismos, nuestras fortalezas y debilidades, los ingresos son parte del reflejo de lo que hacemos, es la forma de manejar nuestra actitud y que se ve en el manejo de la compañía.

Con este esquema filosófico es como se consigue atraer a los mejores del mundo, en todos los campos del conocimiento para el entretenimiento.

Con una estrategia enfocada al cliente, que la marca sea sinónimo de calidad y de clase mundial.

4.2.7 EL ARTE DE CONSTRUIR UNA MARCA

Construir una empresa es todo un reto, pero transformarla en una empresa de espectáculos es un concepto más amplio de negocios, lo que trae un incremento en los beneficios una nueva versión del liderazgo y la reinención del concepto de competitividad.

Para crear una marca que se transforme en el concepto va más allá del conocimiento usual, es impactar al mercado con productos que capturen la imaginación del cliente, es abordar la psique de cada comprador convenciéndolo de que nuestro producto valdrá aun después de su uso, que tendrá más valor que cuando lo compro.

Construir una marca es crear valor agregado con el paso del tiempo, en el mundo del videojuego es muy posible que después del lanzamiento del producto quizá pocos recuerden el título, pero los productos de mercadotecnia como juguetes, figuras coleccionables, enseñanzas filosóficas entre otros productos secundarios sean recordados por los usuarios.

La idea es permanecer en los recuerdos del comprador con un buen sabor de boca.

Esto es muy difícil de comprender por qué cada cabeza es un mundo, entonces estandarizar es lo más fácil y confiable para obtener resultados. Por ejemplo, de los videojuegos que han hecho historia fueron STREET FIGHTER®, principalmente el 2, los títulos posteriores solamente fueron presentados para llenar el hueco en un mercado virgen donde renacía el interés por el videojuego.

Cabe mencionar que una marca es sinónimo de prestigio, de orgullo, por sus productos y esto se ve reflejado en sus ventas. El éxito en el entretenimiento es parte de una filosofía enfocada al cliente como principal valor de ser de la empresa, ofreciendo lo mejor para todos los compradores. Es poniendo atención a los gustos de los clientes y darles lo que buscan, ofrecerles un producto que llene sus expectativas y capturando su mente, así es como la ilusión se crea.

Ser solventes es todo un reto, es todavía más difícil porque los inversionistas cuidan su dinero y quieren ver resultados tangibles traducidos en ingresos y no en estadísticas o cifras manipuladas. Para ser solvente se tienen que tener otras entradas de dinero para amortiguar las fluctuaciones del mercado, estas entradas provienen de fuentes como mercadotecnia de productos, conferencias, exposiciones, productos secundarios.

4.2.8 EL MAPA ESTRATÉGICO DE GAMESKAPE

El mapa 5.21 muestra los puntos estratégicos que se deben cuidar para crear la imagen que deseamos de GAMESKAPE, buscamos lo mejor, y esto es una filosofía es un conglomerado de ideas interconectadas que son:

Ganar reconocimiento

Ser la mejor marca del siglo XXI

Ser solvente

Tener capital humano competitivo

Incrementar el número de seguidores

Incrementar ingresos

Controlar gastos

Controlar el flujo de efectivo (cash flow)

Contar con el mejor capital humano del mundo

Tener modelo de reclutamiento

Crear ilusión

Relacionar costos de egresos con ingresos

Pagar más que otros corporativos

Contar con los mejores expertos del mundo

Mejorar el patrocinio (sponsorship)

Incrementar audiencias

Efectividad

Contar con valores de identidad

Fortalecer la marca internacional

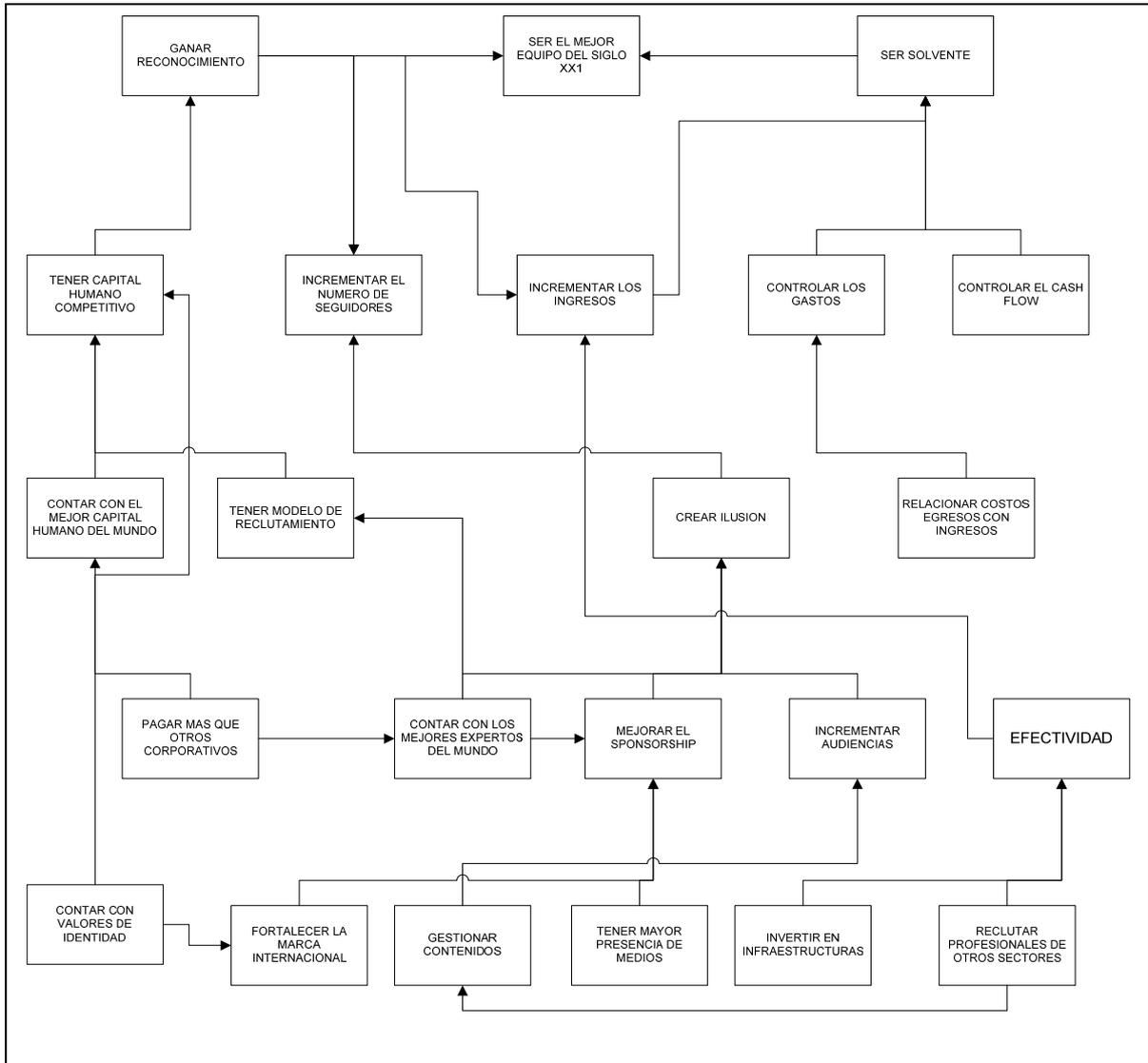
Gestionar contenidos

Tener mayor presencia de medios

Invertir en infraestructura

Reclutar profesionales de otros sectores.

Este grupo de valores estratégicos se comprende mejor su interrelación observando el mapa siguiente, fig. 4.4.



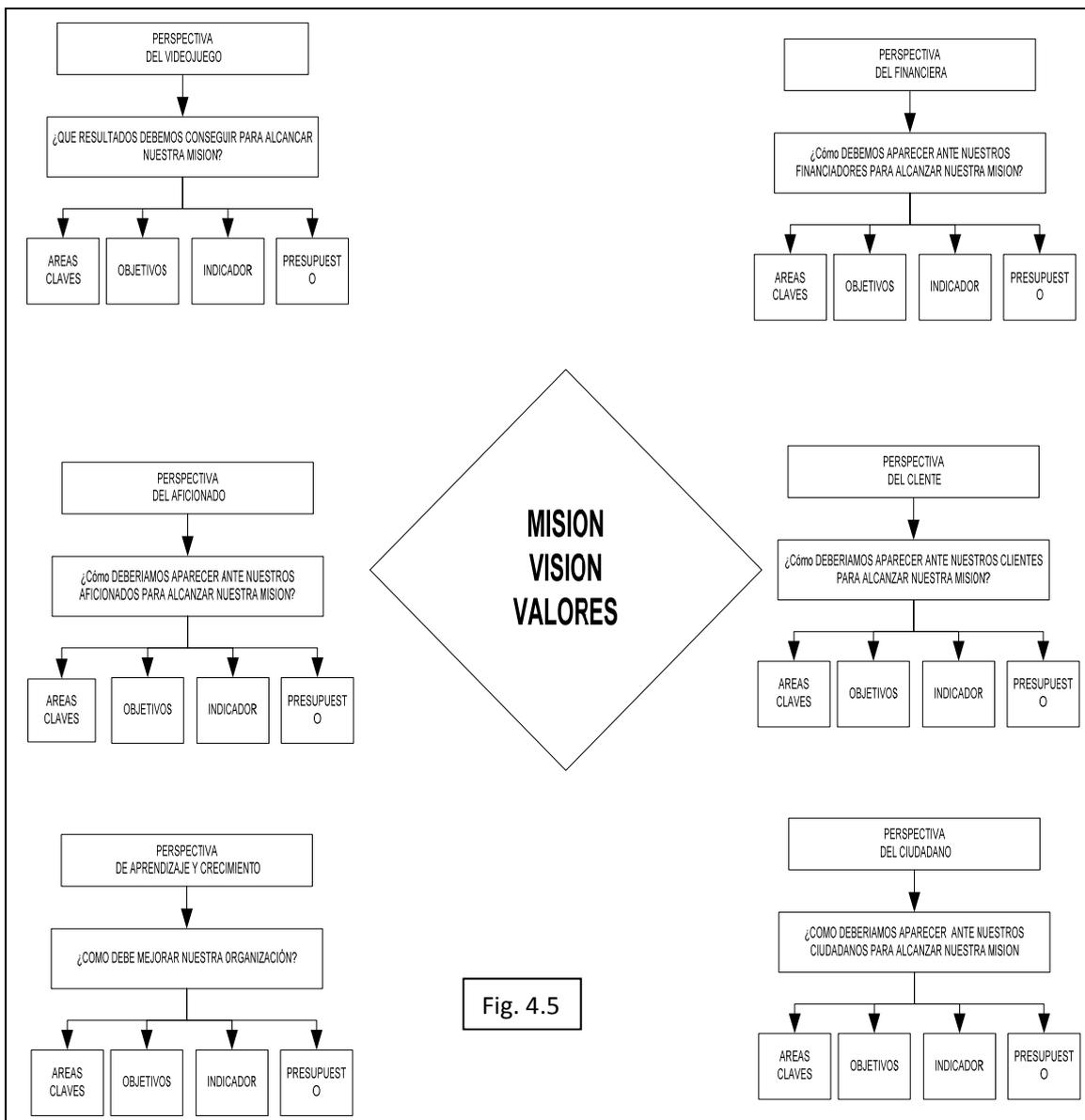
Mapa estratégico de GAMESKAPE.

Fig. 4.4

4.2.9 BALANCE SCORE CARD PARA GAMESKAPE

La fig. 4.5 visualiza el balance Scorecard para GAMESKAPE, está diseñado para no perder de vista los criterios fundamentales del producto y la compañía en congruencia con la misión, visión y valores a través de las seis perspectivas básicas del BSC.}

- Perspectiva del videojuego
- Perspectiva financiera
- Perspectiva del aficionado
- Perspectiva del cliente
- Perspectiva del ciudadano
- Perspectiva de aprendizaje y crecimiento



4.2.10 FEEDBACK Y APRENDIZAJE ESTRATÉGICO

El sistema de feedback utilizado para comprobar las hipótesis sobre las que se basa la estrategia. La fig. 4.6 muestra **el feedback que sugiere aprendizaje estratégico, que es la capacidad de formación de la organización a nivel ejecutivo**, es el poder "aprender la forma de utilizar el Balanced Scorecard como un sistema de gestión estratégica."

El desarrollo de la estrategia es un proceso continuo.
Feedback y formación estratégicos.

Objetivos típicos de un BSC

Obtener claridad y consenso alrededor de la estrategia (FMC Corporation).

Alcanzar enfoque (Chemical Bank).

Desarrollar liderazgo (Mobil).

Intervención estratégica (Sears Roebuck).

Educar a la organización (Cigna P&C).

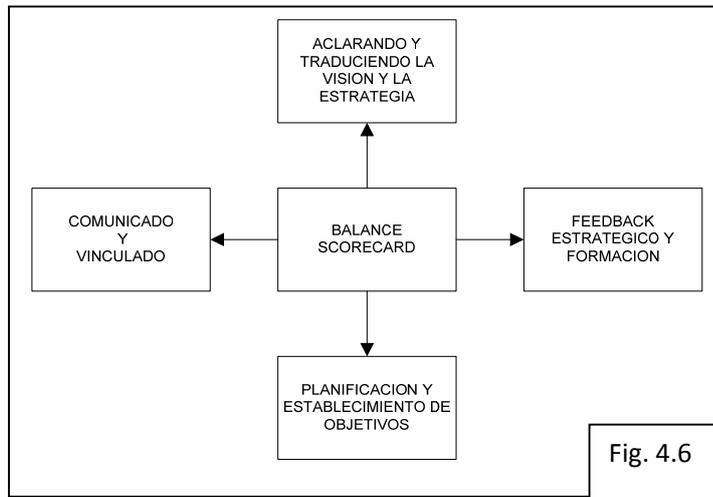
Fijar metas estratégicas (Rockwater).

Alinear programas e inversiones (United way).

Para enlazarlo al sistema de incentivos (EPM).

Mejorar el sistema de indicadores actuales (Transporte-Ecopetrol).

Mantenernos enfocados estratégicamente y evaluar la gestión estratégica (Penta D.O.)



En uno de los aspectos más innovadores, BSC no termina en el análisis de los indicadores. Es un proceso permanente en el que puede haber feedback de un bucle, que consiste en corregir las desviaciones para alcanzar los objetivos fijados y feedback de doble bucle, donde los estrategas cuestionan y reflexionan sobre la vigencia y actualidad de la teoría planteada en un inicio, y su posible adecuación.

4.2.11 ASUNTOS DE NEGOCIO

Se refiere a como el negocio debe crecer y mantenerse. En un escenario internacional, es tener presente todos los recursos informativos en tiempo real para interpretar adecuadamente los datos.

La información obtenida se emplea para tomar decisiones sobre los productos y la empresa misma.

De los elementos que se presta atención son:

- También es observar a la competencia.**
- Analizar productos semejantes**
- Aplicar ingeniería inversa**
- Aprobar Campañas de marketing**
- Aprobación de productos**
- Aprobación de conceptos visuales**
- Análisis de propuestas y alianzas con otras compañías.**
- Manejar disputas legales.**
- Absorber otras compañías.**
- Fusionarse con otras empresas.**
- Analizar nuevos productos.**
- Externar recomendaciones en el análisis de propuestas.**
- Inferir en las decisiones del corporativo.**
- Expandir al corporativo.**
- Y en dado caso, vender la compañía.**

Entre otras cosas y más el área de asuntos de negocio gestiona la política de la empresa con el mundo exterior.

5 EJEMPLO DE APLICACIÓN

5.1 EJEMPLO DE APLICACIÓN

Desarrollo de videojuegos comprende tres objetivos fundamentales que son:

Proyectar la oportunidad de negocios A TRAVÉS DE UN plan de negocios y un modelo de negocios. Ambos plantean la oportunidad de éxito por las circunstancias que el mercado ofrece, en el plan de negocios se proyecta el nacimiento de la compañía en relación a un innovador sistema de negocios ofrecido a través de un producto llamado OVERLAND que es un videojuego de primera persona que simula la experiencia de un paseo real en bicicleta por la ciudad de México. En el modelo de negocio se proyectan valores fundamentales que una organización ya consolidada y comprometida con sus clientes tendrá que ofrecer para mantener el liderazgo en un mercado muy competido.

Mostrar el desarrollo conceptual de LA idea y la organización del sistema productivo esta fórmula es la que se emplea para llevar a buen término un proyecto, esta parte comprende dos ideas estructurales que son: que es lo que quiero y como lo hago.

Mostrar el funcionamiento de un videojuego y su organización como software, como parte fundamental de esta empresa de software se muestra como a través un kit de desarrollo se puede lograr de forma básica un software que trate de mostrar estas cualidades.

El enfoque sistémico que se pretende lograr es hacer fácil la aplicación del conocimiento en el desarrollo de videojuegos como una oportunidad de negocios rentable. Estamos en una era de la WIFI, la conectividad, la aldea global y la colectividad humana. Una era de las redes sociales, el legado de NAPSTER, así como el intercambio libre de contenidos multimedia, una era de libertad informática pero a la vez de paranoia por la seguridad.

Hablar de videojuegos es porque la misma tecnología se ha prestado a que se abaraten los dispositivos donde pueden ejecutarse, hoy en día, cualquier persona por muy humilde que sea tiene un teléfono móvil y por ende tiene juegos ya sea en versión trial o demos, cualquiera tiene un rato de ocio por qué puede pasar ese esparcimiento jugando.

5.1.1 LA PROPUESTA A PRODUCIR

La selección de una propuesta en un negocio no se busca que sea una elección democrática sino práctica donde los elementos que favorezcan su éxito de ventas en el mercado sean tangibles y medibles. De un conjunto de propuestas para su posible realización se elige aquella que tenga las mejores opciones de éxito.

5.1.2 OVERLANDS

Es una aventura de recorrer México en bicicleta por autopistas y carreteras, visitando lugares de interés como pueblos, playas, zonas arqueológicas y sus respectivas peripecias, se juega en primera persona.

Entre más experiencias de viaje se acumule es como se formaran la galería de recuerdos.

De los sitios de interés visitados son valorados por la fama y sus cualidades locales como son la comida, arquitectura, historia, relieve, su gente.

Entre las cosas que se recuerdan será como fue la experiencia del viaje, todo lo que se tuvo que hacer para llegar al destino.

Y lo más importante que cada viaje es toda una experiencia de determinación.

Se desarrolla en nuestros 20´s cuando estábamos solteros, cuando nuestra única preocupación es la escuela, aprobar las materias, pasar extraordinarios, cuando México está inmerso en un reajuste económico y hay poco dinero en nuestros bolsillos, y no nos gusta el alcohol ni otros vicios comunes, entonces decidimos a dar la vuelta en bicicleta para olvidar nuestros problemas y descubrir un nuevo mundo de opciones donde pondríamos a prueba nuestra determinación en llegar a donde otros buscaban pretextos para no hacer el paseo.

Siendo testigos de paisajes de excepcional belleza, de pisar por primera vez carreteras nuevas, recorrer ciudades enteras, cruzar pendientes, bajar a toda velocidad montañas, pasear por playas, selvas, de las inclemencias del tiempo, de accidentes.

Participa en una épica historia de juventud, en la que se mezclan determinación, ocio, curiosidad, audacia, y redención.

La idea se considera buena porque el ciclismo es una cultura y se le pueden agregar esquemas comerciales para vender más productos e inspirar a otros con viajes a lugares dentro del territorio nacional.

Esta idea está respaldada por que es un paseo en bicicleta.

La economía de mundo tiende a reducir el uso de la quema de combustibles fósiles para uso automotor.

Se fomenta la cultura de la salud a través del deporte.

Mundialmente hay un masivo aumento del sobrepeso y se busca la forma de promover la cultura de la salud.

Vender bicicletas

Crear la cultura del respeto a ciclista.

Se seleccionó México porque es el país donde se escribió esta tesis por su amplia variedad de criterios se busca promover varios conceptos inherentes como el

turismo, la salud, la venta de bicicletas de marca, cultura del ciclismo como estilo de vida, para todo público.

Enfocado al ecoturismo como mercado meta.

Darle promoción mundial a sitios de interés como son sitios arqueológicos, pueblos,

Entusiastas y aficionados al turismo en bicicleta.

Compartir links con la comunidad online a través de redes 4G para intercambiar experiencias que se podrán almacenar y tener presentes en todo momento cada vez que el usuario entre al juego en línea.

Establecer relaciones comerciales con marcas para anunciarse dentro del juego y crear una publicidad virtual en tiempo real.

Fomentar el turismo de aventura de forma virtual y así crear un nuevo estilo de impulsar el turismo a través de escenarios reales en ambientes virtuales.

Incorporar temas de moda durante la partida del juego como parte de las conversaciones entre dos ciclistas.

Hacer alianzas con google earth® y otras compañías para establecer el uso de GPS en tiempo real a los recorridos.

Fomentar el respeto al ciclista, que se le vea como una persona positiva y no asociarlo con delincuencia, porque hay mucha gente maniática que su visión es muy limitada.

5.1.3 TRANSFORMACIÓN DE UN CONCEPTO A UN VIDEOJUEGO

Leyendo la historia y la trama de la idea, se estandarizan en criterios. Los criterios se diseñan para crear métricas y generar un esquema de metas, progresos y puntuación, con las cuales se logran objetivos. Para alcanzar estos objetivos se tiene que hacer un recorrido que mantenga la atención del jugador, esta es la esencia del videojuego.

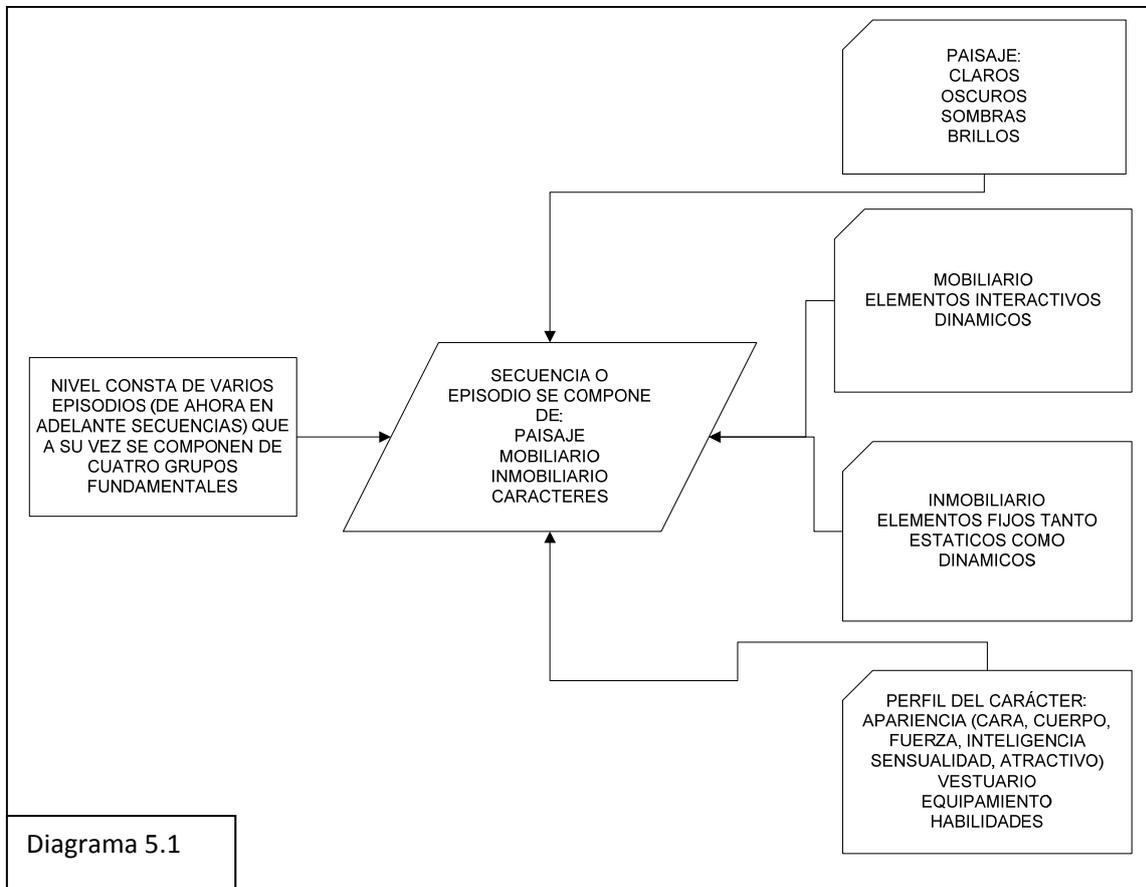
Es un videojuego de primera persona y multijugador que busca promocionar una cultura del deporte a través de admirar las bellezas naturales de México en viajes en bicicleta a través del ecoturismo, narrado por dos personajes básicos, cuenta las aventuras de dos ciclistas de carretera que buscaron encontrarse a sí mismos con lo que la vida y el mundo ofrecía a través de recorrer México en bicicleta.

Con gráficos de gran detalle y escenarios conocidos entre sus características esta la evolución del clima de los paisajes, condiciones climáticas en función al día y la época del año, sitios de interés de México como son paisajes vistos desde la carretera, zonas arqueológicas, pueblos mágicos, playas.

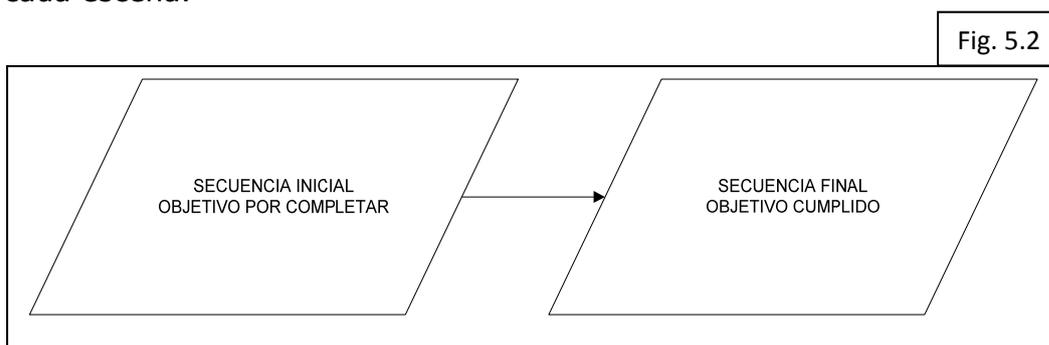
Con niveles llenos de aventuras y condiciones aleatorias pretende capturar la imaginación de una nueva generación de personas dispuestas a disfrutar la vida, con la determinación para llegar a la meta RIDE THE WILD WIND busca capturar la imaginación con atracciones verdaderas y que cualquier persona puede lograr con esfuerzo.

5.1.4 ARTE Y GUIÓN GRÁFICO DEL VIDEOJUEGO

Constituido por escenarios naturales y reales



La figura 5.1 describe y contabiliza los elementos básicos que están visibles en cada escena.



la figura 5.2 ilustra de forma general la razón de cambio con respecto al tiempo desde un evento inicial a un evento final en cada secuencia.

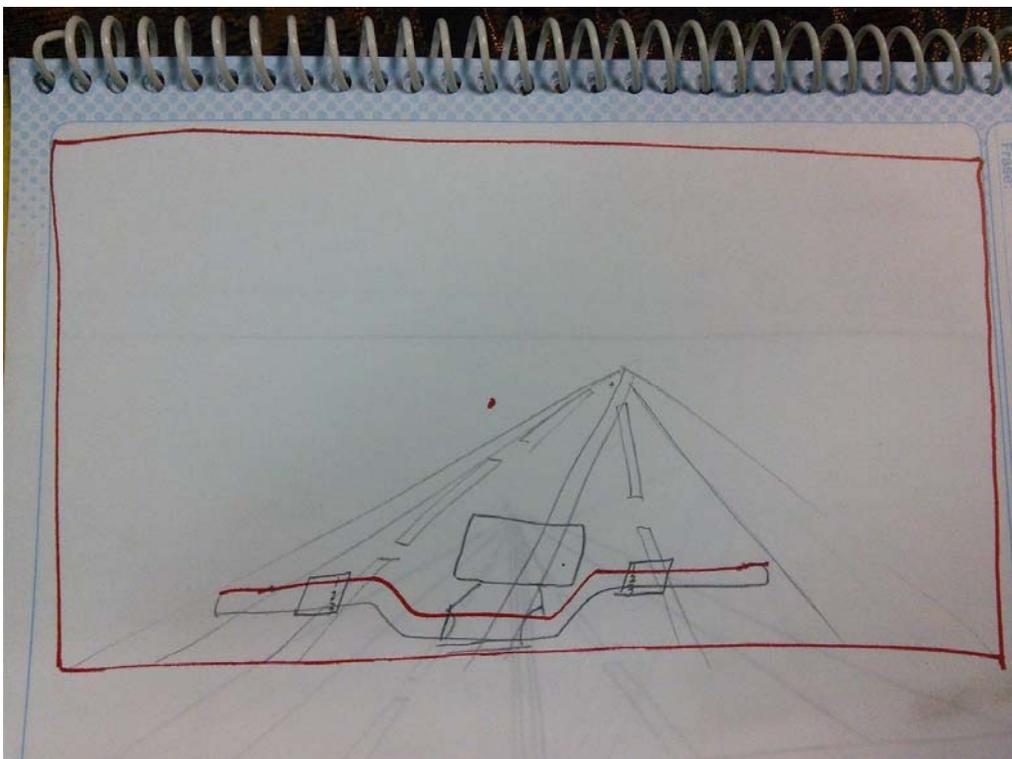
5.1.5 BOCETOS Y PROTO ARTE DE OVERLANDS

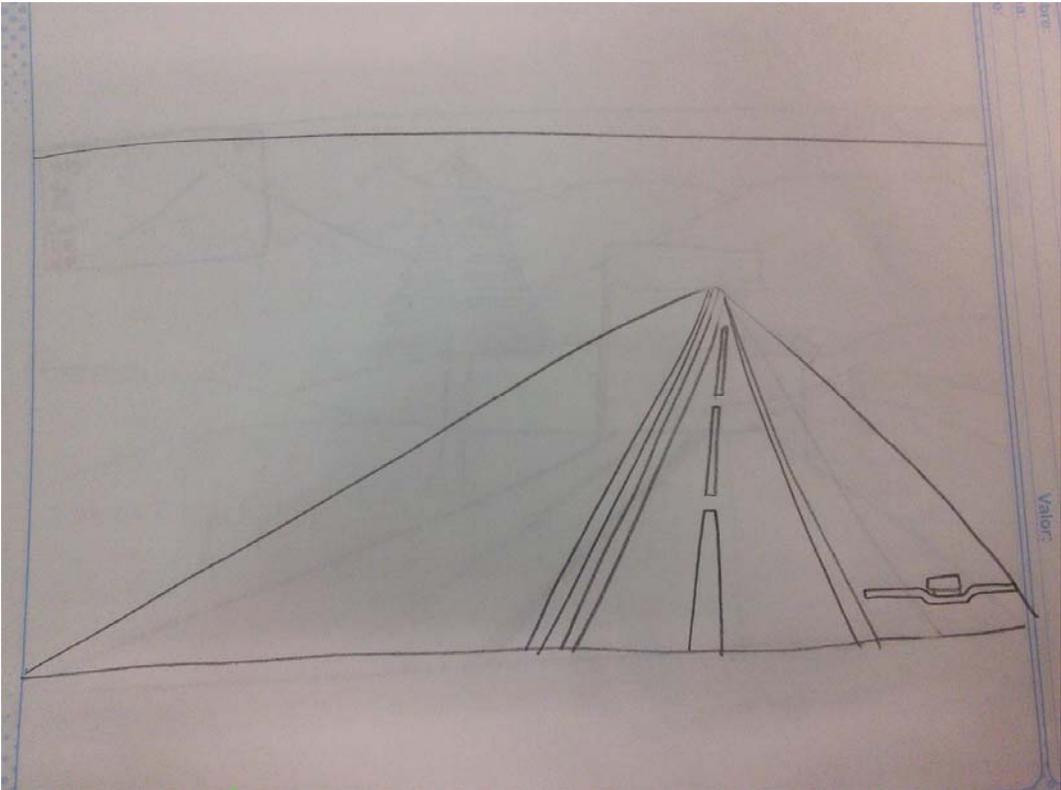
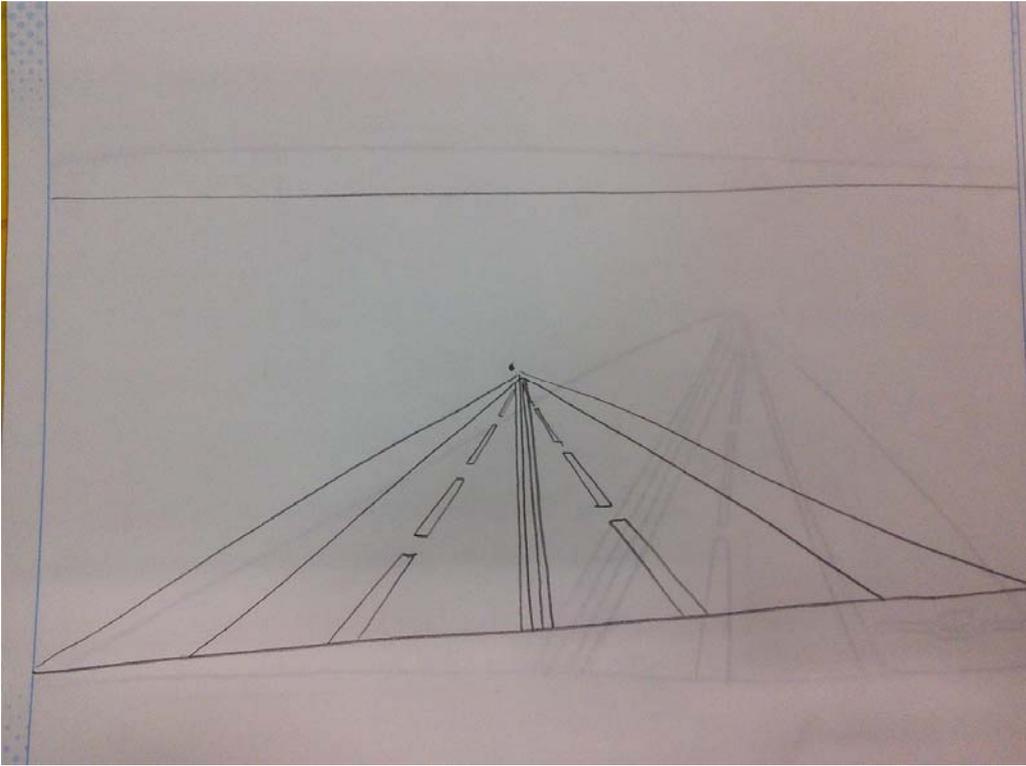
Estos dibujos y bocetos muestran las ideas básicas de la propuesta de OVERLANDS, consiste en un paseo en bicicleta por el municipio de Nezahualcóyotl, describiendo una ruta por las principales vialidades y algunas calles, donde se involucra todo tipo de elementos propios de un tour, donde hay peligros estáticos y dinámicos, obstáculos, focos rojos, paisaje, accidentes geográficos, accidentes urbanos, peatones, perros, automóviles estacionados y en movimientos, postes, publicidad de todo tipo, desviaciones en la ruta, clima, lluvias, viento en contra, a favor, robos, asaltos, temperaturas, tráfico. Luz, noche.

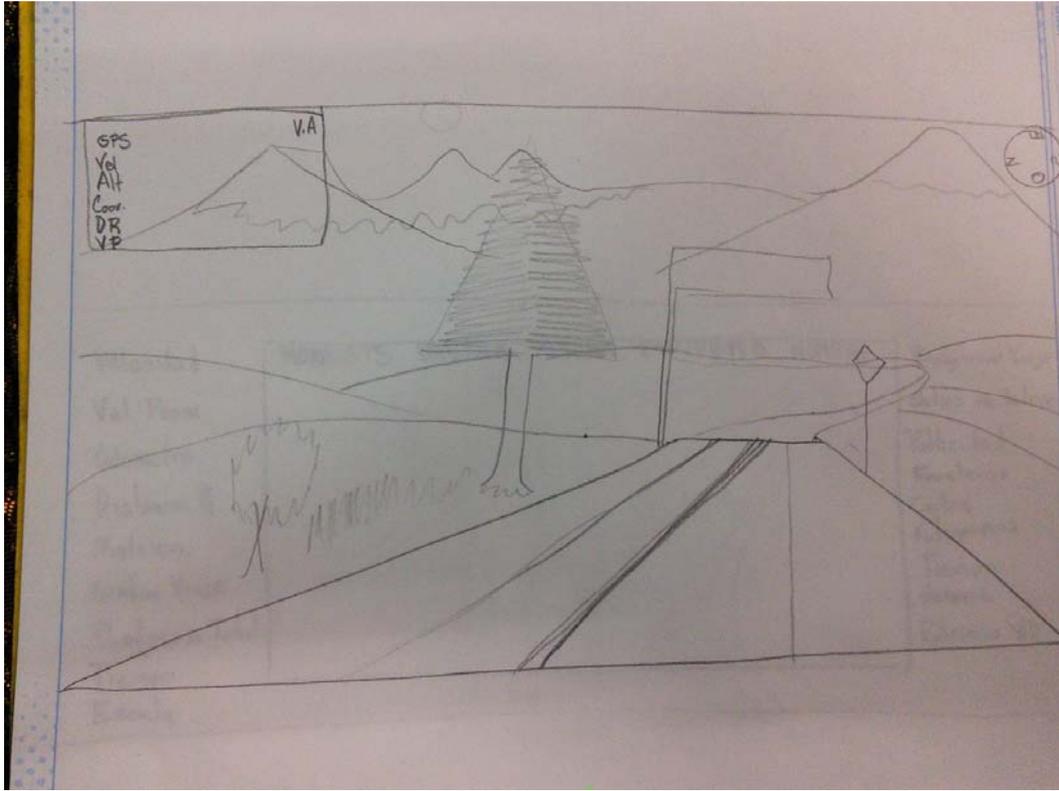
En el trayecto hay un amplio panorama visual que está compuesto de comercios, espectaculares, vehículos automotores etc. Animales, personas, incluso aviones, helicópteros, flora y fauna típica de la región.

El walkthrough está soportado por la idea de llegar al objetivo marcado al inicio de la sesión. El gameplay es la experiencia de un paseo en bicicleta real, y su jugabilidad está compuesta por los niveles de experiencia; miedoso, precavido, ordinario, amateur, biker, maniático.

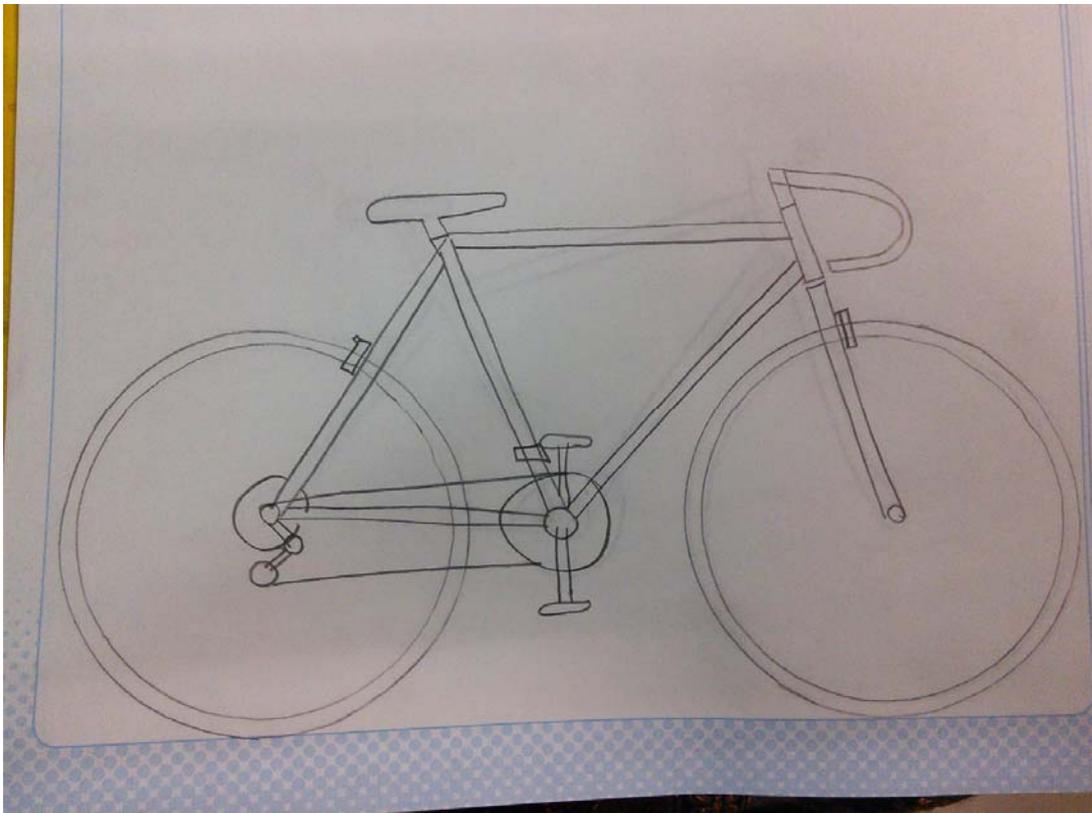
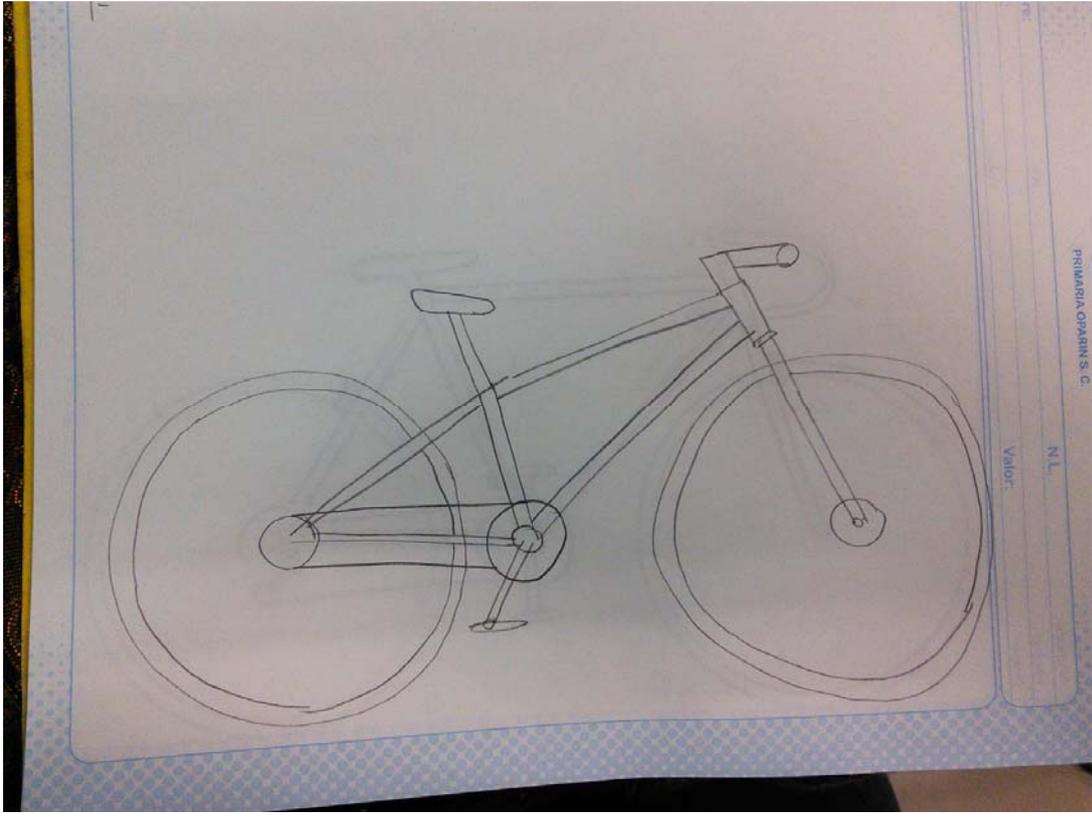
Estos niveles de experiencia se componen por todos los eventos dinámicos y estáticos donde la diferencia radicara en la aleatoriedad y rapidez con que los eventos sucedan y la rapidez para resolver las situaciones de forma asertiva.

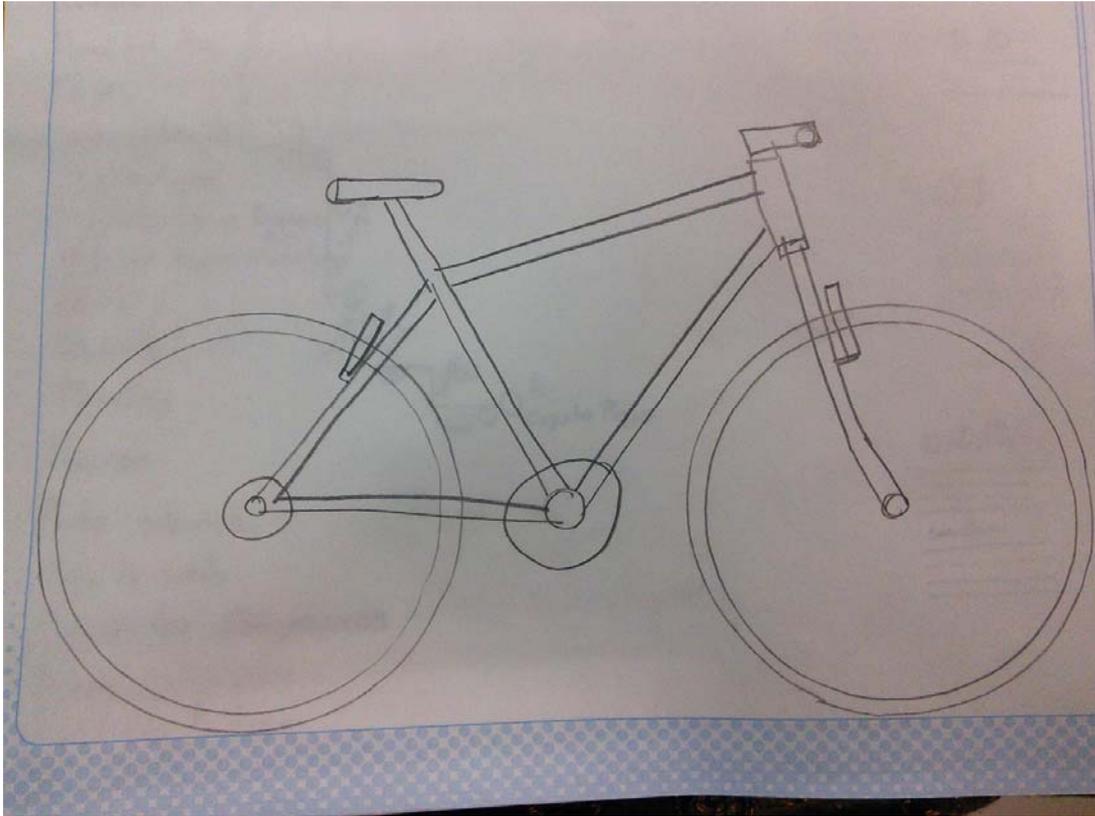






	HORA	GPS	BRUJULA	ALTURA	LATITUD	LONG	NOMBRE	
Velocidad								Programar Viaje
Vel Prom								Sitios de Interes
Odometro								Poblacion
Distancia R								Fuenteras
Metrica								Calles
Grabar Viaje								Fotografias
Distancia Total								Tiempo
Tiempo								Galeria
Escala								Edificios 3D





Ruta

N
O S E

Distancia Recorrida _____
 Sitios de Interés _____
 Nombres de las calles _____
 Hora del día _____
 Clima _____
 Inmuebles B _____
 Objetos fijos _____
 Objetos fijos Dúcam _____
 Objetos fijos Estator _____
 Viento _____
 Animales _____
 Plantas _____
 Tráfico _____
 Ruta coloreada _____
 Tipo de suelo _____
 Accidentes Geográficos _____
 Postes _____
 Alumbrado Público _____

⑦, ⑧

⑤, ⑥

①, ②, ③,

cuella n

Nombre: _____

Fecha: _____

Frases: _____

N.L. _____

Valor: _____

OPARIN

PRIVADA OPARIN S. C.

Flora y Fauna : Arboles, Pastos, Arbustos, Peatones, Pajaros, Perros

Trafico : Microbuses, Autobuses rurales, Autobuses Urbanos, bicicletas,
Automoviles Particulares, Taxis, Colectivos, Motocicletas

Objetos Fijos : Arboles

5.1.6 ESQUEMA DE DISTRIBUCIÓN DEL PRODUCTO

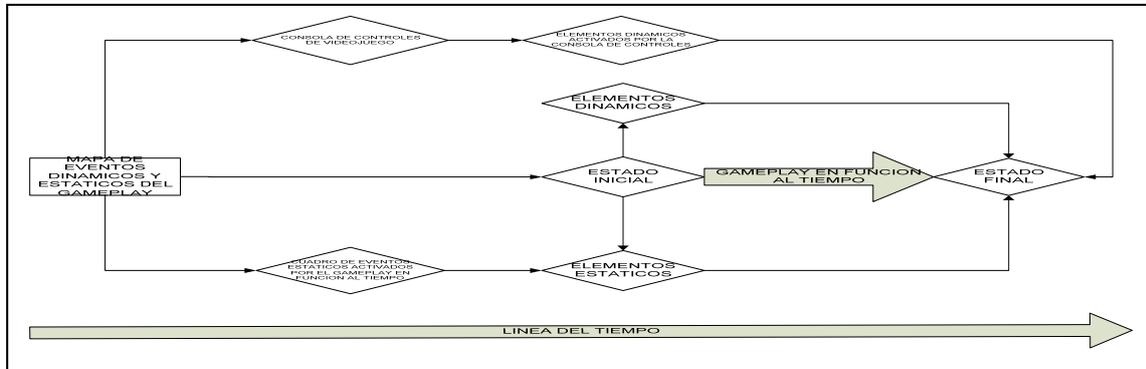


Fig.5.3

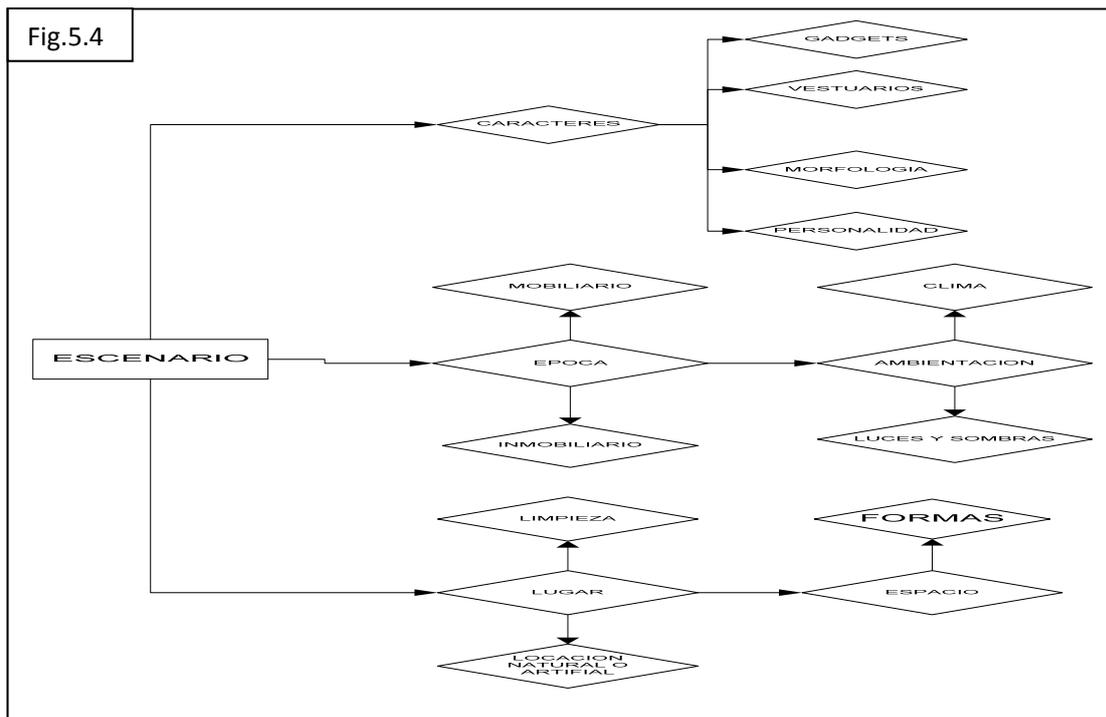


Fig.5.4

La figura 5.3 y 5.4 muestran cómo se desarrolla un evento y sus cambios que experimenta en función al tiempo mientras el juego se está ejecutando. Se pueden observar estados iniciales y estados finales con sus respectivos elementos estáticos y dinámicos, como son fondos, mobiliario, inmobiliario, luces así como la ilusión del transcurso del tiempo en la escena.

5.1.7 ORGANIZACIÓN DEL VIDEOJUEGO

En su lanzamiento solo será por las principales vialidades de la ciudad de México, donde pueden trazarse rutas compuestas en función al destino y configurables en función al tiempo para ajustarlas a la época del año.

En función a la fecha y hora del año se autoajusta el sistema del videojuego para que en el momento que el usuario inicie partida las condiciones climáticas del juego sean reales si se está conectado a internet pero si esta offline, la inteligencia artificial del juego creara un escenario aproximado al real tomando como referencia archivos de la época del año para aproximar las condiciones climatológicas del momento.

Las condiciones aleatorias de la ruta como son baches, accidentes, robos, etc., y también en función al nivel de dificultad, también se auto configuran a través de internet u offline.

El basado en la clase popular donde los recursos son escasos pero el entusiasmo y las ganas son mayores que cualquier obstáculo.

Los protagonistas de esta aventura son dos jóvenes universitarios de clase social popular.

Viviendo su juventud en sus 20 's, con ideas de alcanzar a conocerse a si mismos como meta final.

Aquí se busca explorar la determinación de un individuo a través de la curiosidad y la indagación de hasta donde se puede llegar.

Y es buscar respuestas.

La narración de la historia plantea una narración del viaje de mostrar lo que los ojos ven y como se vivió la experiencia.

Es un juego en primera persona

5.1.8 SEGMENTACIÓN Y ETAPAS DEL VIDEOJUEGO

Se auto configura en función al objetivo. Esta autoconfiguración maneja dos criterios básicos que son fijas y aleatorias. Estos criterios se ajustan en función a la dificultad del nivel elegido. Los criterios fijos son el camino mismo con sus respectivas condiciones orográficas, subidas, bajadas, rectas, curvas, puentes, carreteras libres, autopistas, sitios de interés como zonas arqueológicas, pueblos, playas, estas condiciones no cambian con el nivel de dificultad. Los criterios aleatorios se auto ajustan por la fecha y época del año para auto configurar de forma aleatoria y en función a la hora tales como el clima en función a la ruta del recorrido, como son calor, humedad, lluvia, frio, nieve, así como las condiciones del camino tales como tráfico, estado de la ruta, baches, suciedad, robos, descomposturas entre otros.

Condiciones climatológicas

Sol
Lluvia
Viento
Contaminación
Frio
Humedad
Calor
Día
Noche

Condiciones del camino

Trafico
Distancia
Pendientes positivas
Pendientes negativas
Baches
Suciedad

Condiciones aleatorias

Baches
Caídas
Descomposturas
Robos
Lesiones
Accidentes
Perros

Condiciones para ganar puntos y trofeos

Cumplir el recorrido
Cumplir a contra reloj
Aplicar técnicas de ciclismo eficientes
Tomar excelentes fotografías del viaje

Describe un viaje de mochileros en bicicleta, un paseo en México por sus autopistas con todos los elementos a favor y en contra de un viaje, donde conoces más amigos, gente amable y no tanto.

Inicia con los preparativos previos al viaje que son:

Destino
Tiempo de recorrido
Sitios de interés
Lugares para recargar provisiones
Conocer el camino a través de un mapa
Indagar sobre las costumbres del lugar.
Condiciones de tráfico
Llevar dinero para cualquier eventualidad

Condiciones del clima
Condiciones del camino

La salida es en cualquier ciudad, pueblo, playa, domicilio, de la república mexicana, con una planeación que involucra, la elección de un buen recorrido depende de condiciones favorables como el clima, la cantidad de dinero, condiciones de la bicicleta, condición física de uno mismo, condición de la vestimenta.

El objetivo del juego es llegar a la meta propuesta por uno mismo como usuario, planearla bien para llegar a buen termino

Durante el paseo:

Es tener ritmo
Mantener actitud positiva
Conservar el entusiasmo
Alejarse de problemas
Evitar maniobras extrañas o peligrosas
Evitar manejar durante la noche.

Al finalizar el paseo:

Prepararse para la siguiente partida
Comentar las experiencias
Comentar los sitios de interés

5.1.9 ELEMENTOS DEL VIDEOJUEGO

CARACTERES

Dos caracteres básicos con su propio esquema y personalidad

ESCENARIOS

Escenarios naturales de México, con vista de calle impulsados por google earth.

OBJETOS DINÁMICOS

Vehículos como bicicletas, motocicletas, automóviles, pick ups, autobuses, transporte urbano de pasajeros, taxis, vehículos extraño.

OBJETOS ESTÁTICOS

Carpeta asfáltica, banquetas, calles, casas, edificios, arboles, monumentos, colinas, montañas, ríos, puentes, monumentos naturales, nubes, sol, luna, Animales, pájaros, peces.

AUDIO

Audio de fondo compuesto por tráfico,
Sonidos mecánicos procedentes de la bicicleta.
Voces de los caracteres cuando estén hablando en modo multijugador.

CINEMÁTICAS

Conformadas por películas de escenarios naturales y urbanos.
Presentadas después de llegar algún sitio de interés.

CONDICIONES CLIMÁTICAS DE LOS ESCENARIOS

Lluvias, viento, sol, huracanes, humedad, noche, atardeceres, amaneceres.

Dividido en niveles de dificultad:

- Ciclista de ocasión
- Ciclista de juguete
- Ciclista equilibrado
- Ciclista maniático

Cada nivel es una ruta con objetivos de tipo:

- Naturaleza
- Zonas arqueológicas
- Ciudades
- Pueblos
- Playas

Bicicletas

- Básica montaña de 27 velocidades
- Estándar de montaña de 27 velocidades
- Premium de montaña de 27 velocidades
- Ruta estándar de 18 velocidades

Ruta básica de 18 velocidades
Ruta Premium de 20 velocidades
Turismo de 1 velocidad
Turismo de 3 velocidades
Turismo Premium de 8 velocidades.

Accesorios

Bicicletas
Casco
Guantes
Accesorios
Ropa
Dinero
Herramientas
Alimentos
Repuestos
Mapas
GPS

MENÚ INTERACTIVOS DEL VIDEOJUEGO

Consiste en la estructuración gráfica de todos los criterios de operación y jugabilidad (gameplay) del videojuego. Se divide en:

PARTIDA

INICIO
SELECCIONAR PARTIDA
GUARDAR
CRÉDITOS
CARACTERÍSTICAS ESPECIALES
CONFIGURACIÓN
SALIR

Bicicleta configuración:

MODELO
NUMERO DE VELOCIDADES
EQUIPO ADICIONAL
GPS
INSUMOS
LUCES

MANDOS OPERACIONALES

DIRECCIÓN DERECHA E IZQUIERDA
LUCES
CAMBIO DE VELOCIDADES TRASERO
CAMBIO DE VELOCIDADES FRONTAL
FRENADO

CICLISTA ATUENDO

VESTIMENTA SUPERIOR
VESTIMENTA INFERIOR
CASCO
GUANTES
PROTECCIONES
GAFAS
CALZADO

HERRAMIENTAS

CICLISTA FÍSICO Y MENTAL
FORTALEZAS

FUERZA
INTELIGENCIA
DETERMINACIÓN
ACROBACIA
AGILIDAD
VISIÓN
RESISTENCIA
SALUD
ATENCIÓN

DEBILIDADES

FRIO
MALA ALIMENTACIÓN
ENFERMEDAD
CANSANCIO
DESVELO
SOBREPESO

MAPA DE MENÚS DE CONTROL DEL VIDEOJUEGO

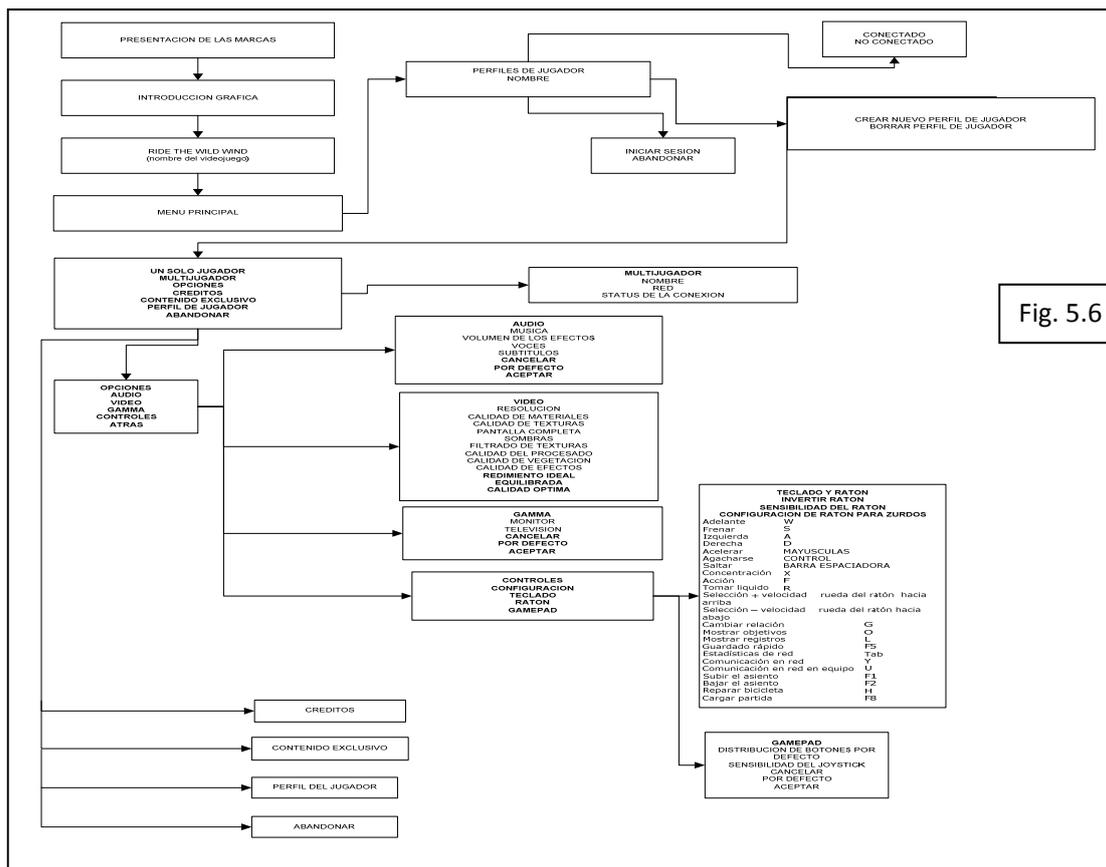


Fig. 5.6

La figura 5.6 ilustra el mapa de menús que aparecerán en el videojuego. Dividido por jerarquías de menú principal y menús secundarios se puede conformar.

5.1.10 DISEÑO DE PARTIDA

Está diseñado en primera persona.

Presentación
Título del juego

Inicio
Crear perfil
 Configurar bicicleta
 Configurar ciclista
 Configurar ruta
 Configurar nivel
 Seleccionar dificultad
 Salir

Iniciar partida
 Seleccionar partida
 Borrar partida
 Guardar partida
 Cargar partida

Créditos
Salir

Características especiales
 Cinemáticas
 Anunciar
 Agregar contenido
 Salir

Opciones
 Video
 Audio
 Controles
 Contenido descargable
 salir

UN JUGADOR

Modo Historia: empieza una nueva partida o retoma el juego desde donde lo guardaste la última vez. Antes de empezar una nueva partida, elige uno de los tres niveles de dificultad:

Fácil, medio o difícil. Recuerda que también puedes empezar la partida desde cualquiera de los niveles que habías alcanzado la última vez.

Secretos: explora las fotografías y dibujos secretos que hayas encontrado durante la partida.

Estadísticas personales: tus estadísticas; tiempo de juego, puntería, etc.

Cargar: cargar una partida guardada anteriormente.

MULTIJUGADOR

Desde este menú puedes crear o unirse a un combate utilizando la conexión a Internet o una red de área local.

INTERNET

Partida rápida: busca y únete automáticamente a una partida existente en Internet. No se necesitan más ajustes.

Partida personalizada: selecciona el tipo de partida que quieres buscar para jugar. Puedes ordenar los servidores disponibles en función de tus preferencias.

Crear partida: personaliza todas las condiciones que quieras mientras actúas como host.

Cuando hayas seleccionado los ajustes, inicia la partida como el host.

Desbloquear personajes: utiliza el dinero que consigas en el juego para desbloquear nuevos

Personajes multijugador.

Estadísticas: lista de los mejores jugadores con diferentes tipos de estadísticas del juego.

LAN

Buscar una partida: busca y únete a una partida.

Crear una partida: crea una partida.

OPCIONES

Audio: cambia el volumen de la música, los efectos de sonido y los diálogos.

Vídeo: ajusta la configuración del vídeo para obtener una calidad y un rendimiento óptimos.

Gamma: ajusta el nivel gamma de la pantalla y selecciona el dispositivo de pantalla.

Controles: modifica los controles del juego predeterminados, incluida la sensibilidad del ratón, los accesos directo del teclado, invierte el ratón, recargar automáticamente, sistema de cobertura dinámico y el mando de juego alternativo (por ej. un joystick).

CRÉDITOS

Selecciona esta opción para ver los créditos del juego.

CONTENIDOS EXCLUSIVOS

Introduce el código de contenidos exclusivos.

PERFIL DE JUGADOR

Gestiona tus perfiles de jugador. Interfaz del juego

SALIR

Salir del juego y volver a Windows.

MENÚ DEL JUEGO

Puedes acceder al menú del juego pulsando la tecla Esc mientras juegas.

Reanudar: volver a la partida.

Reiniciar capítulo: reiniciar el nivel actual.

Guardar: guardar el progreso de la partida.

Cargar: cargar una partida guardada anteriormente.

Opciones: modificar las opciones del juego.

Salir al menú: volver al menú principal. Perderás todos los progresos que no hayas guardado.

INTERFAZ DEL JUEGO

1. Texto del objetivo: puedes mostrarlo siempre que pulse la tecla O.
2. Indicador de cooperación: marca la posición actual del personaje.
3. Icono de acción: muestra la acción disponible en la actualidad, por ejemplo, subirse a una bicicleta, saltar un bache, cambiar velocidad, acelerar, etc.
4. Indicador de objetivos: muestra la dirección de otro objetivo.
5. Mira: se pone roja cuando hay un posible peligro cerca.
6. Indicador de acción de cooperación: indica una acción conjunta
7. Menú de velocidades: te permite elegir una de las relaciones disponibles con la rueda del ratón.
8. Indicador de dinero: muestra la cantidad de dinero a disposición del jugador.
9. recursos en uso: muestra el número de insumos o el número de recursos que puedes utilizar.
10. Reservas de insumos: el número de reservas restantes para cada tipo de reto.
11. bicicleta: nombre de la bicicleta en ese momento.
12. Estadísticas de la bicicleta: estadísticas del equipo en ese momento.
13. Indicador de recarga: aparece cuando hay que volver a cargar energía (tecla R) cuando ya no queda más insumos para el tipo de reto actual.
14. Indicador de modo Concentración: muestra el porcentaje de energía y recursos para lograr el reto dentro del tiempo y condiciones programadas.

RECURSOS Y ARTILUGIOS

- Bicicletas
- Cascos
- Guantes
- Accesorios
- Ropa
- Dinero
- Herramientas
- Alimentos
- Repuestos
- Mapas
- GPS

COMO JUGAR

- Movimientos
- Cambiar de velocidad
- Uso de técnicas de pedaleo
- Respiración y técnicas anti cansancio
- Modo concentración para romper marca
- Reparar bicicleta
- Acciones conjuntas
 - Uso de mapa
 - Uso de GPS
 - Uso de alimentos
 - Uso de hidratantes
 - Uso de dinero
 - Uso de ropa
 - Uso de herramientas
 - Uso de casco
 - Uso de guantes
 - Uso de bloqueador solar
 - Uso de descanso
- Uso de gafas
- Uso de accesorios

MULTIJUGADOR

El modo multijugador se basa en un sistema original de puntuación y marcas de los jugadores.

Los jugadores ganan dinero en el juego y lo utilizan para mejorar parámetros, comprar nuevas clases y desbloquear logros basados en ellas. Los jugadores ganan dinero eliminando a competidores y completando los objetivos del juego. Hay una recompensa por rebasar a cada jugador hasta el final de la marca. Cuando rebasas a un jugador, ganas la cantidad de dinero equivalente a su recompensa.

Hay puntos de reabastecimiento como tiendas, puestos de comida en la carretera, talleres para reparación.

MODOS MULTIJUGADOR

Ganas rebasando al pelotón evitando que ellos te bloqueen y que tú los bloquees, rebásalos y mantén la marca para ganar puntos, dinero y experiencia. Gana el equipo con más puntos. Coordina tus movimientos con el resto de tu equipo para ganar este combate.

Persecución: hay dos equipos y un jugador de Se busca; el mejor jugador de uno de los equipos. Si el de Se busca sobrevive durante 60 segundos, el equipo puntúa.

Leyendas de la carretera: uno de los equipos debe completar varios objetivos; algunos pueden completarse de varias maneras y otros son opcionales. El equipo contrario intenta evitar que se completen los objetivos antes de que se acabe el tiempo asignado. Los niveles cuentan con un gran contenido temático y se basan en acontecimientos históricos reales.

CLASES DE ELEMENTOS MULTIJUGADOR

Hay trece clases de elementos diferentes, cinco de las cuales están disponibles desde el principio. El resto pueden desbloquearse utilizando el dinero obtenido en el juego.

Cada elemento ha sido diseñado para adecuarse a un tema determinado; para permitir un estilo y unas tácticas determinadas, así como para adecuarse al paisaje del camino y actuar como parte de la historia.

- Bicicleta básica montaña de 27 velocidades
- Bicicleta estándar de montaña de 27 velocidades
- Bicicleta Premium de montaña de 27 velocidades
- Bicicleta de ruta estándar de 18 velocidades
- Bicicleta de ruta básica de 18 velocidades
- Bicicleta de ruta Premium de 20 velocidades
- Bicicleta de turismo de 1 velocidad
- Bicicleta de turismo de 3 velocidades
- Bicicleta de turismo Premium de 8 velocidades.

2 niveles des bloqueables conforme se alcancen los créditos y habilidades.

5.1.11 DISEÑO DE PRODUCCIÓN

La contabilización de todo el proyecto y quien lo hará.
La parte dinámica que se conforma de las características dinámicas del recorrido y la parte estática que se conforma de la interface del jugador en primera persona así como sus respectivos intercambios de información con otros usuarios.

Parte dinámica fija. (Vista de calle)

32 niveles. (1 por cada estado de la república)

Por estado número de:

- Carreteras por estado
- Sitios arqueológicos
- Pueblos por estado
- Sitios de interés
- Ríos
- Lagunas
- Montañas
- Colinas
- Accidentes geográficos
- Caminos de terracería
- Hoteles
- Restaurantes
- Gasolineras
- Espectaculares
- Arboles
- Monumentos
- Puentes
- Miradores
- Pendientes positivas
- Pendientes negativas
- Curvas
- Semáforos
- Casetas
- Señalamientos
- Cruceros de ferrocarril
- Trenes
- Tiendas
- Topes
- Líneas en el pavimento
- Derrumbes
- Incendios
- Desviaciones

parte dinámica activa (elementos móviles), esta estará sujeta a la configuración inicial del perfil que automáticamente ingresara la fecha del año para que de forma automática e inteligente se auto configure y conforme el perfil de la partida en tus 4 modos operativos a elección con la finalidad de que sea lo mas cercana a un viaje real en bicicleta.

Parte dinámica activa:

- Trafico aleatorio de automóviles
- Tráfico aleatorio pesado
- Trafico de bicicletas
- Embotellamientos
- Accidentes
- Perros
- Descomposturas
- Sol
- Lluvia
- Cansancio
- Viento a favor
- Viento en contra
- Humedad
- Lluvia
- Terremotos
- Enfermedad
- Fuerza
- Desvelo

Parte estática (jugador en primera persona, consola de mando e interface visual)

- Bicicleta
- Mandos
- Relaciones de velocidad
- GPS
- Dinero
- Hora
- Fecha
- Mandos
- MMORG
- HUDs

Con estos tres bloques estructurales del videojuego se agrega el sonido. Todas estas unidades están numeradas y se pasaran a su respectivo grupo de trabajo.

5.1.12 ETAPAS DEL PROYECTO

Este mapa muestra de forma sencilla a los grupos de trabajo que intervienen en la producción del videojuego.

La fig. 5.7 observándola de izquierda a derecha y de arriba hacia abajo se observa que el proyecto inicia con la etapa artística conformada por gráficos, audio y video que son los activos del videojuego.

Posteriormente pasa a la etapa desarrolladora donde se integra el medio de desarrollo y la codificación del video, en esta etapa es donde se establece la interactividad de los elementos y la interface de exportación de gráficos a la siguiente etapa del videojuego.

En la etapa de programación se integran las APIs que son el motor de juego, el audio dinámico/estático, 3d y la red, esta etapa es el juego en si, sometiéndose a rudas pruebas de ejecución y QA..

Finalmente todo el producto casi listo se le agregan MENÚS, HUDs, MMORG, texturas, se compila para otras plataformas y se ejecutan pruebas de calidad.

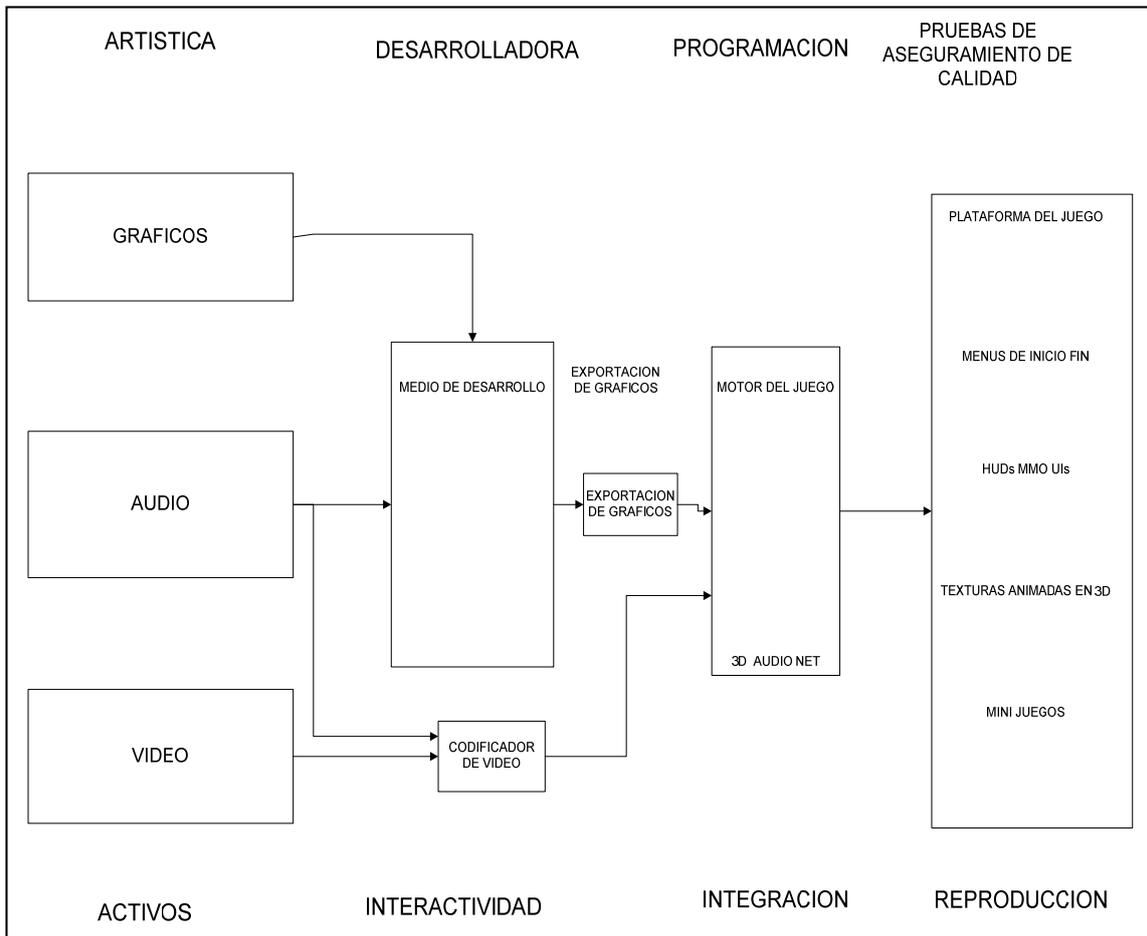
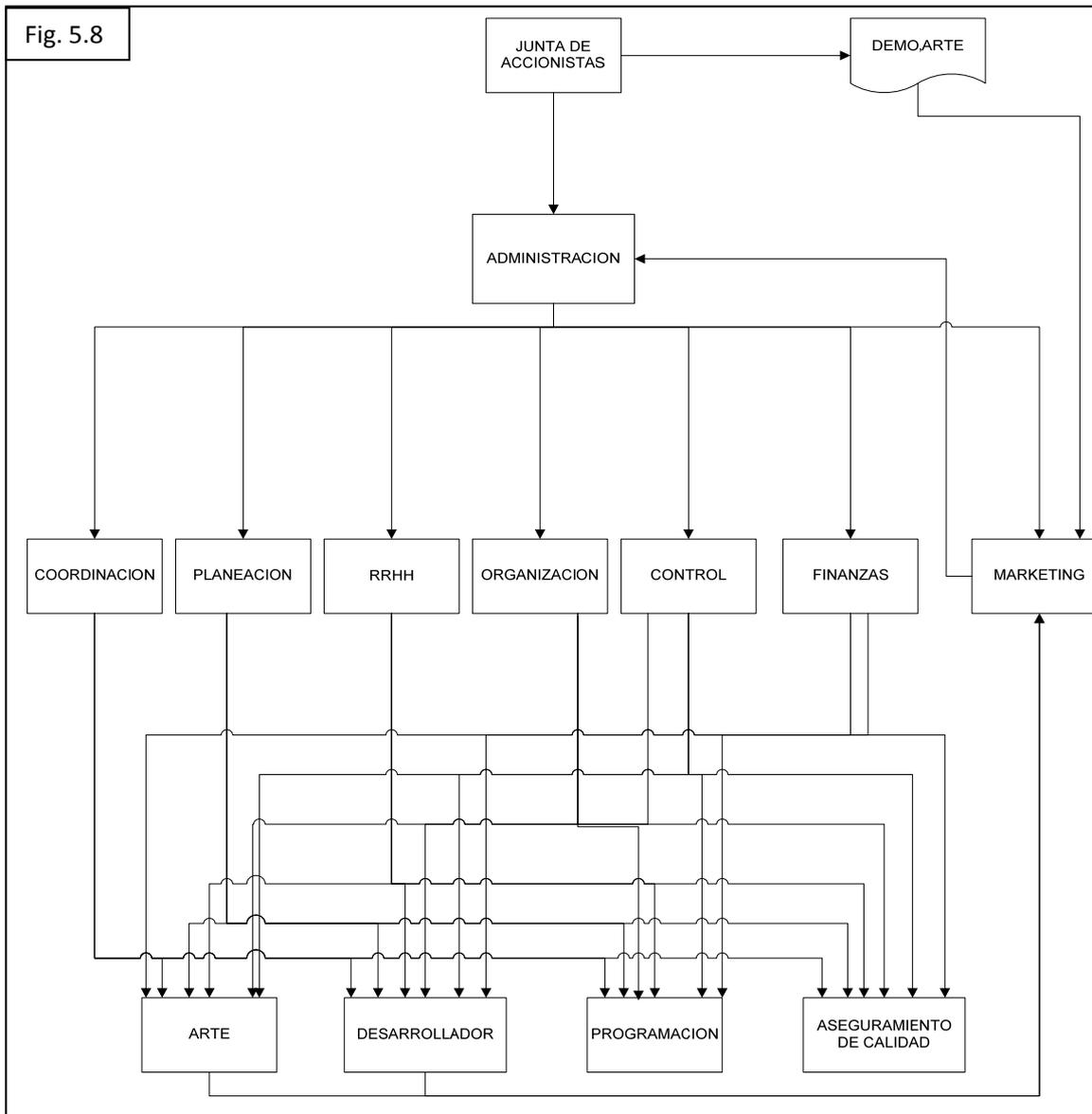


Fig. 5.7

5.1.12.1 GRUPOS DE TRABAJO

La fig. 5.8 muestra a toda la producción, ordenada por jerarquías se aprecia las interrelaciones entre grupos de trabajo. Los resultados pasan de un area a otra donde las actividades simultáneas, secuenciales, están monitoreadas para dar cumplimiento a los requerimientos de planeación. El diseño de producción es la parte más importante del proyecto porque el producto ya es medible y sujeto a aplicaciones de todas las herramientas disponibles para obtener el resultado en tiempo y forma. Cada grupo de trabajo tiene sus tareas y responsables.



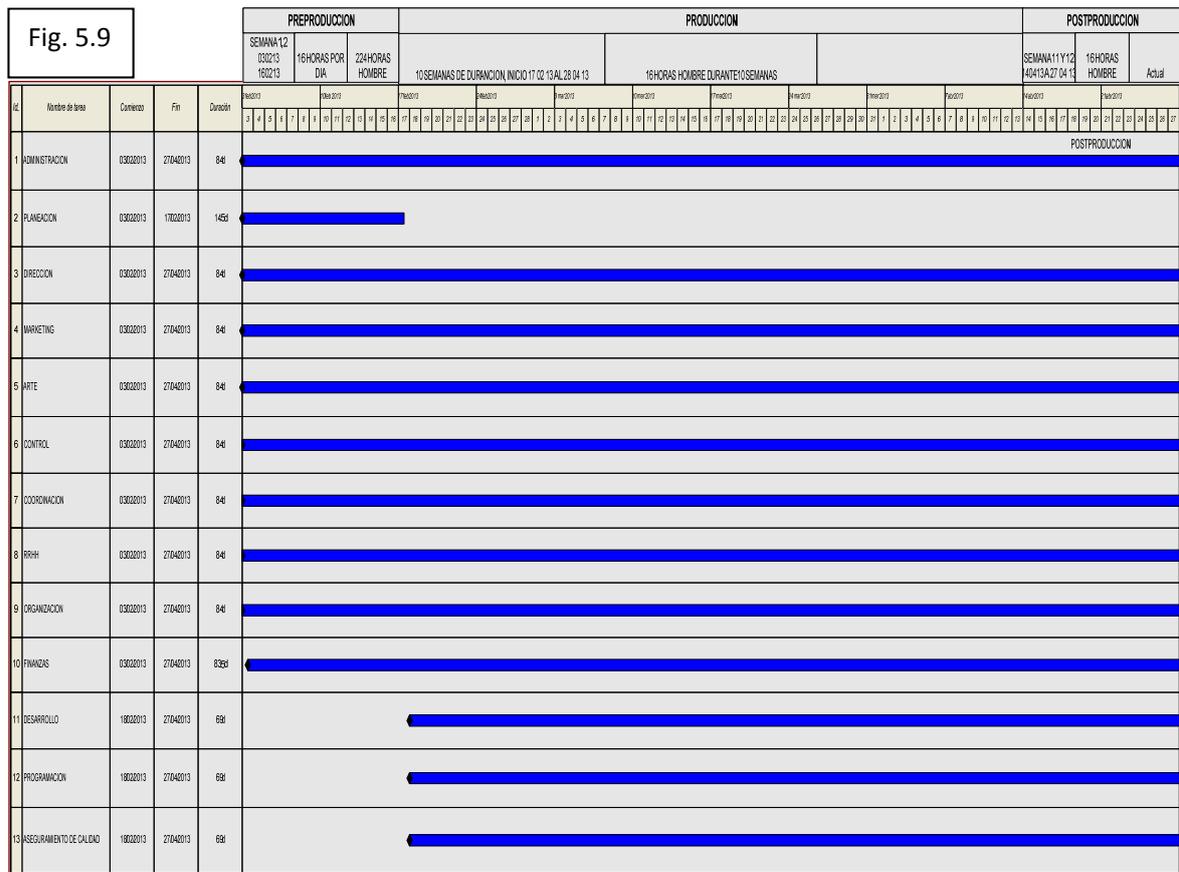
5.1.12.2 CRONOGRAMA DE PRODUCCIÓN

Producción solo se puede explicar de esta forma, que cada persona involucrada en el proyecto ya tiene una tarea y un tiempo para cumplirla en función a los requerimientos de los otros grupos de trabajo.

La normalización de los procesos y sus respectivas fases en el desarrollo del producto establece la estrecha relación entre producción y planeación.

Las tareas no se pueden atrasar porque se compromete toda la producción generando pérdidas.

La fig. 5.9 representa el cumplimiento de las tareas en función al tiempo.



5.1.13 ENSAMBLE DEL VIDEO JUEGO Y PRUEBAS BETA

El ensamble del videojuego se va efectuando conforme avanzan los procesos terminados en sus respectivas etapas y dominios, consiste en que el ensamble del videojuego es continuo y las pruebas de calidad también son simultáneas como resultado de que el tiempo presiona.

El videojuego ha sido terminado y probado, alcanzando los objetivos de Planeación. Ahora viene la siguiente etapa que son las pruebas beta. Antes de su lanzamiento oficial se harán innumerables pruebas de ejecución que su finalidad será buscar fallas y notificarlas al respectivo grupo de trabajo.

Estas pruebas beta están estructuradas estadísticamente en función a las condiciones dadas durante la ejecución, al ser notadas se analizara e investigara donde se originan estos procesos defectuosos. Como es un proceso estadístico completo, este automáticamente envía muestras de fallas y las condiciones que los originaron. Producción hasta aquí llega. El video juego se ha convertido en un producto terminado y operativo. El cual funciona bien sin embargo someterlo a pruebas de trabajo pesado es para garantizar su funcionalidad.

5.1.13.1 POSTPRODUCCIÓN Y ASEGURAMIENTO DE CALIDAD

Durante la producción del videojuego se van efectuando simultáneamente rutinas de aseguramiento de calidad de software, la finalidad es tener bajo control todos los procesos.

Un proceso bajo control es un protocolo confiable que da inicio al siguiente proceso. En consecuencia se manejan cadenas de procesos confiables.

En el contexto de OVERLANDS se han contabilizado 32 procesos importantes (que son cada nivel, por lo que cada esquema tiene sus propias características cíclicas)

Y en un mundo tan competido no se puede dar el lujo de cometer errores y perder el mercado por falta de calidad del producto. La calidad es una idea muy beneficiosa porque nos proporciona confianza en los productos, es gratis por la calidad no cuesta nada, es parte fundamental de todo producto de clase mundial, es una filosofía de vida, y un videojuego es como un producto más, que tiene que tener calidad para que sea confiable.

En realidad la postproducción es un conjunto de protocolos, normas, procedimientos, procesos, métricas, métodos aplicados a los diferentes bloques que conforman el videojuego, se aplican de forma inmediata una vez que cada bloque se termina y los beta testers se aseguran de que la funcionalidad sea total dentro de lo posible.

Durante la creación del videojuego suelen aparecer errores que se van corrigiendo con las métricas apropiadas, sin embargo, durante la ejecución del producto por parte de algún usuario se puede dar el caso de fallas del software debido al hardware particular o a la plataforma donde se esté corriendo el producto. Entonces para esto se crean los parches que corregirán estas deficiencias no contempladas durante las pruebas beta del producto.

Todo esto se hace por que el prestigio de la marca está en juego, y no es recomendable dejar las cosas así.

Como se ha dicho, la calidad no cuesta, la calidad es un valor intrínseco al producto, no es valor agregado es solamente una calidad, y es la mejor que se puede hacer. Con los protocolos disponibles, se aplican a todos los procesos participantes en la creación del producto y del manejo de la empresa.

Estas filosofías de calidad son el reflejo del compromiso de la empresa con su clientela.

5.1.13.2 ASEGURAMIENTO DE CALIDAD DE LOS BETA TESTERS

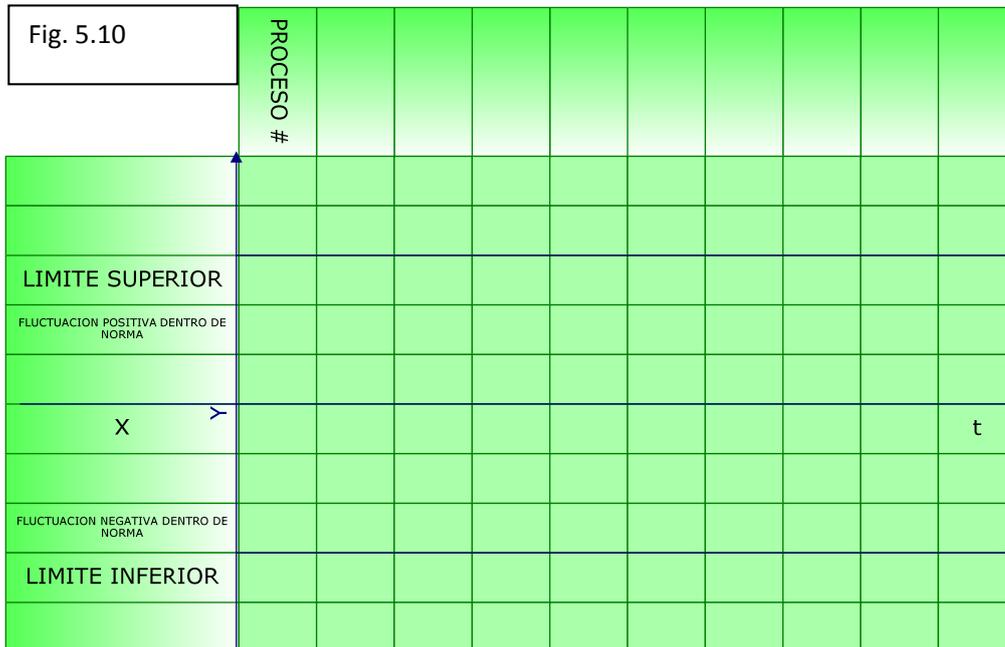
Consiste en graficar el comportamiento en línea del producto a través de pruebas extremas de uso por parte de jugadores profesionales bajo diferentes configuraciones de hardware. La idea es generar errores y registrar las circunstancias que los propiciaron.

Estas condiciones se analizan por otros expertos y se cuantifican de forma sistemática de cada cuando y bajo qué condiciones se originaron. Estos errores estandarizados se clasifican y se solucionan porque ya se tienen los criterios de cómo se originaron por que se analizaron los procesos que intervinieron en su generación.

La fig. 5.10 muestra un criterio sencillo de operación donde interviene el tiempo, el número de proceso y un límite superior e inferior como marcas para estandarizar el proceso si está dentro de parámetros o fuera de control.

Esta grafica sirve de modelo para analizar fallas por unidad de procesos en función al tiempo. Por ejemplo, si un error se genera unas 600 000 veces cada segundo pero sin exceder sus límites inferiores o superiores y sin afectar otros procesos de forma sustancial significaría que está dentro de parámetros confiables, tomando como referencia que cada proceso consiste en unos 50 000 de iteraciones por segundo.

GRAFICO BASICO DE CONTROL DE PROCESOS CON LIMITES EN FUNCION AL TIEMPO



5.1.13.3 ASEGURAMIENTO DE CALIDAD DE COMPATIBILIDAD DE HARDWARE

Esta etapa del proceso consiste como parte de los beta testers pero conformada por otro modelo estadístico que se alimenta de los errores generados y enviados en línea. Durante la ejecución del producto se generan errores originados por alguna incompatibilidad. Este modelo está diseñado del mismo modo que el modelo de aseguramiento de calidad de los beta testers que consiste en el número de procesos en función al tiempo con sus respectivos límites de operación, se registran los eventos y se someten a análisis para ser resueltos.

5.1.13.4 ASEGURAMIENTO DE CALIDAD DE COMPATIBILIDAD DE RED

Este criterio de calidad es similar a los anteriores, donde las condiciones son las mismas pero se aplican en otros bloques del producto. En este caso corresponde al bloque de LAN Y NET, donde se maneja el criterio de la administración de hardware y el envío y recepción de datos, así como la conectividad y el manejo de recursos. Este modelo está diseñado para funcionar en redes que ofrezcan calidad de servicios y no experimenten fluctuaciones desconocidas, que también ese es otro parámetro a considerar, porque las fluctuaciones estandarizadas es donde el prototipo del videojuego ha funcionado y ofrecen métricas y esquemas conocidos. En situaciones donde se desconocen las fluctuaciones es imposible estandarizar el error y no es posible solucionarlo pro que esta fuera de parámetros operacionales conocidos.

5.1.13.5 CONTROL ESTADÍSTICO DEL PRODUCTO EN LÍNEA

Este criterio es muy importante y que debe de funcionar sin importar si el videojuego es original o es una copia ilegal. Consiste en él envío de datos referentes a los hábitos del usuario cada vez que usa el producto, de forma sistemática se estandarizan este modelo de hábitos y se consideran los datos para el diseño de nuevos productos. Este criterio es valioso por la información que proporciona del usuario desde su nombre y edad hasta cada cuando usa el producto así como día, mes, hora, año, que se le dificulta, que se le facilita, entre otros detalles que ponen en evidencia su lado B como usuario.

5.1.13.6 SUPERVISIÓN DE CONTENIDO

Consiste en supervisar todo el producto terminado. Estas supervisiones consisten en observar el comportamiento de gameplay en sus diferentes capas de aplicación. Es como una revisión de calidad pero de contenido visual. Aquí se revisan los errores visuales producidos por algún fallo en la comunicación con ARTE, DESARROLLO, PROGRAMACIÓN. Estas revisiones normalmente no arrojan resultados negativos pero de haberlo, se turna la observación al grupo correspondiente. Esta supervisión visual del gameplay es un cotejo entre el guion escrito del proyecto y el producto final. Todos los contenidos del videojuego así como sus características extras.

Hablando de censura y la clasificación correcta de un videojuego se estandariza a través de este criterio

5.2 RESUMEN EJECUTIVO PLAN DE NEGOCIOS

Overlands es un software , consiste en un videojuego de paseos y recorridos virtuales por México.

En su etapa inicial será solo en las principales vialidades de la cd de México, si hay aceptación se expandirá hacia otras ciudades.

La experiencia del juego consiste en una simulación del paseo con todos los eventos que se pueden generar en una experiencia real.

El usuario podrá hacer su aportaciones para mejorar la experiencia del juego

La finalidad es ofrecer entretenimiento a todo público que le guste pasear y ver lo que México ofrece. Cada tour cubre un recorrido donde se observaran y en algunos casos interactuar con el escenario, con la finalidad de hacer una experiencia realista e interactiva.

Con gráficos de alta calidad y fotografías reales de los lugares visitados mostrara la parte virtual de escenarios conocidos, así como paseos a pie por donde no se pueda transitar en bicicleta de los lugares visitados como instalaciones deportivas, playas, centros comerciales, mostrara toda la gama de oportunidades para anunciarse en tiempo real. Con la conectividad permanente en la nube se podrán ofrecer productos y servicios y sus respectivos precios en tiempo real.

El sistema de negocios propone cobrar una comisión por cada anuncio que el jugador active, así aunque lo vea y no lo active no se cobrara, así todos ganamos. La particularidad de este sistema de negocios es que el cliente verdadero son todos los anunciantes y el cliente secundario son los jugadores . a ambos se les ofrecerá la mejor experiencia visual e interactiva en juegos de simulación virtual.

Todos los anunciantes que deseen participar deberán tener un sistema de información que soporte esta plataforma para que su publicidad este vigente y sus productos y servicios tengan valides en el momento que el usuario dé clic.

El juego mostrara todo tipo de publicidad verdadera, siempre que el anunciante tenga contrato para estar en OVERLANDS,

Este producto plantea la oportunidad de negocios ofreciendo espacios publicitarios en tiempo real con participación directa del jugador.

Durante el recorrido muestra sitios de interés común

Descripción de la compañía, GAMESKAPE ENTERTAINMENT, es una compañía de software dedicada al entretenimiento, ofreciendo los mejores productos para el entretenimiento y publicidad. Usando como oportunidad de negocio la evolución de la conectividad y el abaratamiento del WIFI, y la proliferación de los dispositivos permanentemente conectados a internet.

No se requieren instalaciones físicas, utilizamos la modalidad de tele trabajadores.

Este producto es gratis, su distribución se hará a través de torrentz o del mismo sitio web, www.gameskape.com.

5.3 RESUMEN EJECUTIVO MODELO DE NEGOCIOS

“Un estilo, una actitud, un talento que se asume y se aprende desde pequeño para descubrir que es tan importante ganar como respetar al adversario, y que hay que ganar resistiendo ante la adversidad y buscando la belleza”.

Que hay que ser elegantes ante la competencia. Hay que saber ganar pero también hay que saber asumir la derrota. Que hay que ser solidarios y sacrificados.

En concreto la visión del corporativo es crear el concepto del siglo XXI. Un proyecto; ser líderes en el terreno del entretenimiento, tener los productos y una economía acorde con el prestigio de la marca y consolidar su imagen”. “lo más importante que tiene el corporativo es su imagen y prestigio y esa imagen de GAMESKAPE, de sentimiento que esta fomentada por el espíritu de competir en un mundo cambiante. Una identidad que no debemos de perder, que debemos fomentar y desde esos objetivos tan entrañables, es un hecho palpable” ser el concepto del siglo XXI, se consigue con el éxito corporativo y económico, logrando ganar mercados y siendo solvente.

Pues como se dijo anteriormente, estamos en un mundo que evoluciona rápidamente, donde es fácil caer por la competencia, pero nuestro espíritu de lucha es el que nos mantiene a la vanguardia en nuevos conceptos.

Sin visión y misión una empresa no puede existir, es como estar sin rumbo y es fácil perderse, porque no se sabe hacia dónde ir, es como perder el tiempo.

La idea es crear el concepto del entretenimiento digital del siglo XXI, y se logra creando la marca.

La visión de GAMESKAPE es ser el concepto en entretenimiento digital del siglo XXI, con productos innovadores inspirados de la imaginación del cliente.

La misión es ofrecer los mejores productos para el entretenimiento digital, pensando en ti como nuestra principal responsabilidad, creando los productos más innovadores en su clase.

Las estrategias se enfocan a estos rubros, a los productos, a la promoción, al tipo de cliente, al desarrollo de innovaciones. Estas estrategias son un conglomerado de directrices obtenidas de la visión y la misión de la empresa. Aquí se busca el crecimiento de la empresa para posicionarla como el concepto del entretenimiento. Cada estrategia tiene como finalidad ganar el aprecio del cliente y permanecer como la mejor opción en su clase. El punto inicial para poder diseñar un modelo de BSC es la definición de la visión y estrategias, no es sencillo, sin embargo, normalmente la gente llega a acuerdos en este sentido, pues son muy genéricos los objetivos de crecimientos que se persiguen. Donde la mayoría tropieza es en la interpretación de las estrategias. Por ejemplo, el crecimiento de la empresa difícilmente sería cuestionable, pero para algunos, el crecimiento puede significar aumento geográfico, para otros mayores ingresos o incremento de empleados. Llegar a consensos sobre las estrategias lleva a establecer tanto objetivos como indicadores que midan los objetivos.

El entendimiento compartido de lo que es o ha de ser el corporativo no requiere mucha discusión. Este entendimiento facilita la toma de decisiones. En su implementación puede haber divergencia de opiniones. Pero no hay que confundir las decisiones estratégicas (el que) con las implementaciones (el cómo). La imagen que pretende dar el corporativo se podría concretar como la de una empresa con clase, que le ha llevado a ser considerada como el mejor corporativo de entretenimiento y en el que es un desafío construir una historia de valor a base de marca y contenido, pero con unos valores universales que van mucho más allá del entretenimiento, como así lo pretende reflejar. El espíritu de superación y el respeto por la competencia son los valores principales del corporativo.

Entendida la visión y estrategias de la empresa es posible determinar los objetivos que hay que cumplir para lograr la estrategia y aterrizarlos en indicadores. Es importante que los indicadores no controlen la actividad pasada solamente, los indicadores deben reflejar los resultados muy puntuales de los objetivos, pero también deberán informar sobre el avance para alcanzar esos objetivos. Esto es, la mezcla balanceada entre indicadores de resultados e indicadores de actuación es lo que permitirá comunicar la forma de conseguir los resultados y, al mismo tiempo, el camino para lograrlo. Como resaltan Kaplan y Norton: "Resultados son los indicadores históricos, indicadores de la actuación son indicadores previsionales."

Otro aspecto que hay que resaltar es el número de indicadores que deberán formar parte del escenario principal o maestro. Según Kaplan y Norton, un número adecuado para esta labor es de 7 indicadores por perspectiva y si son menos, mejor. Se parte de la idea de que un Tablero con más de 28 indicadores es difícil de evaluar cabalmente, además de que el mensaje que comunica el BSC se puede difuminar y dispersar los esfuerzos en conseguir varios objetivos. El ideal de 7 indicadores por perspectiva no significa que no pueda haber más, simplemente pueden consultarse en caso de hacer un análisis más a conciencia, pero los mencionados serán aquellos que se consulten frecuentemente y puedan, verdaderamente, indicar el estado de salud de la compañía o área de negocio.

La definición de balance Scorecard (BSC, Kaplan y Norton, 1996) es un instrumento que traduce la misión y la estrategia de la organización en una amplia colección de métricas e indicadores. Posteriormente proporciona la estructura necesaria para servir como instrumento de control y de sistema de implantación de la estrategia.

"Un estilo, una actitud, un talento que se asume y se aprende desde pequeño para descubrir que es tan importante ganar como respetar al adversario, y que hay que ganar resistiendo ante la adversidad y buscando la belleza".
Que hay que ser elegantes ante la competencia. Hay que saber ganar pero también hay que saber asumir la derrota. Que hay que ser solidarios y sacrificados.

En concreto la visión del corporativo es crear el concepto del siglo XXI. Un proyecto; ser líderes en el terreno del entretenimiento, tener los productos y una economía acorde con el prestigio de la marca y consolidar su imagen". "lo más importante que tiene el corporativo es su imagen y prestigio y esa imagen de GAMESKAPE, de sentimiento que esta fomentada por el espíritu de competir en un mundo cambiante. Una identidad que no debemos de perder, que debemos fomentar y desde esos objetivos tan entrañables, es un hecho palpable" ser el concepto del siglo XXI, se consigue con el éxito corporativo y económico, logrando ganar mercados y siendo solvente.

Pues como se dijo anteriormente, estamos en un mundo que evoluciona rápidamente, donde es fácil caer por la competencia, pero nuestro espíritu de lucha es el que nos mantiene a la vanguardia en nuevos conceptos.

Sin visión y misión una empresa no puede existir, es como estar sin rumbo y es fácil perderse, porque no se sabe hacia dónde ir, es como perder el tiempo.

La idea es crear el concepto del entretenimiento digital del siglo XXI, y se logra creando la marca.

La visión de GAMESKAPE es ser el concepto en entretenimiento digital del siglo XXI, con productos innovadores inspirados de la imaginación del cliente.

La misión es ofrecer los mejores productos para el entretenimiento digital, pensando en ti como nuestra principal responsabilidad, creando los productos más innovadores en su clase.

Las estrategias se enfocan a estos rubros, a los productos, a la promoción, al tipo de cliente, al desarrollo de innovaciones. Estas estrategias son un conglomerado de directrices obtenidas de la visión y la misión de la empresa. Aquí se busca el crecimiento de la empresa para posicionarla como el concepto del entretenimiento. Cada estrategia tiene como finalidad ganar el aprecio del cliente y permanecer como la mejor opción en su clase. El punto inicial para poder diseñar un modelo de BSC es la definición de la visión y estrategias, no es sencillo, sin embargo, normalmente la gente llega a acuerdos en este sentido, pues son muy genéricos los objetivos de crecimientos que se persiguen. Donde la mayoría tropieza es en la interpretación de las estrategias. Por ejemplo, el crecimiento de la empresa difícilmente sería cuestionable, pero para algunos, el crecimiento puede significar aumento geográfico, para otros mayores ingresos o incremento de empleados. Llegar a consensos sobre las estrategias lleva a establecer tanto objetivos como indicadores que midan los objetivos.

El entendimiento compartido de lo que es o ha de ser el corporativo no requiere mucha discusión. Este entendimiento facilita la toma de decisiones. En su implementación puede haber divergencia de opiniones. Pero no hay que confundir las decisiones estratégicas (el que) con las implementaciones (el cómo). La imagen que pretende dar el corporativo se podría concretar como la de una empresa con clase, que le ha llevado a ser considerada como el mejor corporativo de entretenimiento y en el que es un desafío construir una historia de valor a base de marca y contenido, pero con unos valores universales que van mucho más allá del entretenimiento, como así lo pretende reflejar. El

espíritu de superación y el respeto por la competencia son los valores principales del corporativo.

Entendida la visión y estrategias de la empresa es posible determinar los objetivos que hay que cumplir para lograr la estrategia y aterrizarlos en indicadores. Es importante que los indicadores no controlen la actividad pasada solamente, los indicadores deben reflejar los resultados muy puntuales de los objetivos, pero también deberán informar sobre el avance para alcanzar esos objetivos. Esto es, la mezcla balanceada entre indicadores de resultados e indicadores de actuación es lo que permitirá comunicar la forma de conseguir los resultados y, al mismo tiempo, el camino para lograrlo. Como resaltan Kaplan y Norton: "Resultados son los indicadores históricos, indicadores de la actuación son indicadores previsionales."

Otro aspecto que hay que resaltar es el número de indicadores que deberán formar parte del escenario principal o maestro. Según Kaplan y Norton, un número adecuado para esta labor es de 7 indicadores por perspectiva y si son menos, mejor. Se parte de la idea de que un Tablero con más de 28 indicadores es difícil de evaluar cabalmente, además de que el mensaje que comunica el BSC se puede difuminar y dispersar los esfuerzos en conseguir varios objetivos. El ideal de 7 indicadores por perspectiva no significa que no pueda haber más, simplemente pueden consultarse en caso de hacer una análisis más a conciencia, pero los mencionados serán aquellos que se consulten frecuentemente y puedan, verdaderamente, indicar el estado de salud de la compañía o área de negocio.

La definición de balance Scorecard (BSC, Kaplan y Norton, 1996) es un instrumento que traduce la misión y la estrategia de la organización en una amplia colección de métricas e indicadores. Posteriormente proporciona la estructura necesaria para servir como instrumento de control y de sistema de implantación de la estrategia.

6 CONCLUSIONES

El objetivo de este trabajo fue crear una plantilla para elaborar proyectos de videojuegos, dividiéndolo en dos partes fundamentales, que son, la parte conceptual y la parte creativa.

La primera es experimental y subjetiva, la segunda es medible y objetiva, ambas integran el desarrollo de videojuegos.

Este trabajo de investigación está inspirado en la búsqueda de respuestas, la mayoría se encontró y desde varios puntos de vista se comprendió que la principal motivación del ser humano es el dinero. El dinero impulsa a descubrir nuevos caminos, conquistar, crear y conservar lo logrado.

Este proyecto es el conocimiento aplicado para crear cualquier videojuego, sin importar su tamaño o alcances.

Para lograr la conclusión satisfactoria de un proyecto lo más importante es la planeación y la división del trabajo, con estos dos elementos presentes y claros es como responsabiliza a los involucrados para la ejecución de todas las actividades. El desarrollo del videojuego plantea métricas, estándares, métodos, procesos, metodologías, filosofías, todo esto tiene criterios fácilmente medibles en función al tiempo, por lo cual se puede esperar un resultado tangible y satisfactorio.

En la parte conceptual también es medible y tangible pero es experimental por lo que se puede experimentar un éxito relativo. Cuando se toma la decisión de elegir una propuesta entre un grupo de ellas es el resultado de una decisión objetiva respaldada por los criterios del estudio de mercados y su enfoque al mercado meta, pero también se consideran en la elección los criterios del consejo de administración que son los que toman la decisión final sobre darle luz verde al proyecto o cancelarlo. Por parte del consejo de administración todo proyecto se decide en base a la objetividad pero no siempre es así, siempre habrá intereses creados que limiten o desvíen el objetivo del proyecto, algunas veces se corrompe y no alcanza el objetivo propuesto y raras veces se transforma en un éxito sin precedentes.

Finalmente, un videojuego es dinero, es una gran oportunidad de negocios, es arte expresado en forma gráfica, es crear mundos virtuales, es darle vida a lo muerto, es entrenamiento, es aprendizaje, es entretenimiento. Los videojuegos son la expresión máxima del entretenimiento digital en la era de las comunicaciones, se han ganado el lugar a pulso en un mundo muy competido, donde las oportunidades de negocio son efímeras. Lo más importante para crear un proyecto es dividir el trabajo. Los videojuegos son un conglomerado de ideas transformadas en audio y video dinámicos, son la prueba fehaciente del ingenio humano, son la demostración multidisciplinaria de la técnica y el arte vivo. Es una demostración de la psique humana reflejada en forma donde uno puede interactuar en un mundo virtual y no solamente ser un espectador.

Una idea lo es todo, pero buscar esas personas que pueden ayudarnos a traer esa idea al mundo real es lo verdaderamente difícil.

7 GLOSARIO

Assets (Activos)

Son los modelos, animaciones, sonidos, IA, físicas. Son los elementos que forman el juego en sí, el código hace funcionar los assets.

Application Programming Interface (Interfaz de Programación de Aplicaciones)

Es un sistema de rutinas, de protocolos y de herramientas para desarrollar programas de aplicación. Un buen API hace más fácil desarrollar un programa proporcionando todos los bloques del desarrollo del programa. El programador pone los bloques juntos.

Entre estos los más importantes son el DirectX (de Microsoft) y el OpenGL (que trabaja con la mayoría de los sistemas operativos).

Antialiasing:

El anti-aliasing revisa los polígonos y difumina los bordes y vértices, para que los bordes no se vean como dentados. Esta técnica se puede hacer de dos maneras. La primera se realiza de modo individual, entremezclando polígonos para sobreponerlos unos delante de otros.

La segunda manera se hace por medio de tomar todo el marco y quitarle los bordes dentados, pero esto requiere de mucha memoria.

BSP Tree Hierarchy (BSP Árbol de Jerarquía)

Es un método para determinar qué superficies de un mundo, y qué objetos, están realmente en la escena en momento dado, dada su localización en el mundo. Esto se utiliza a menudo para los objetos del desecho, y también para entresacarlos para reducir el proceso del AI (Inteligencia Artificial) y de la animación.

Bump Mapping:

Técnica vieja de texturas que tratan de mostrar como la luz se refleja en el objeto. Solo hasta hace poco se volvió a retomar.

Culling

Codificado que logra que los objetos que no se ven en determinado cuadro de la animación por causa de objetos que los obstaculizan (como una pared) no tomen tiempo de renderizado. Así se reduce la cantidad de trabajo del motor. El Culling es más fácil de implementar en juegos en donde la visión es controlada como los RTS en comparación con lo FPS. Un método de Culling puede ser por "Árboles BSP"

Depth Testing (prueba de profundidad):

Con esto se empieza a eliminar los píxeles ocluidos y se pone en práctica el concepto de sobre dibujado. La prueba de profundidad es una técnica utilizada para determinar que objetos están delante de otros en la misma localización del píxel.

Sobre Dibujado: es la cantidad de veces que se ha dibujado un píxel en un frame. Se basa en la cantidad de elementos existentes en la tercera dimensión (profundidad).

Forma de Juego Emergente

Consiste en programar al AI con un conjunto de reglas que le permitan al programa adherir situaciones que el programador no previera.

Flat Shading Lighting (Iluminación de Sombreado Plano)

Consiste en que cada polígono represente un valor leve que se pase al polígono completo que genere una imagen plana del mismo, a esta imagen también se le asigna un color determinado.

Higher-order surfaces (superficies de alto orden)

Renderizado matemáticamente, usado en las tarjetas gráficas más recientes y potentes. También llamado: Patches (parches).

Iluminación (lighting)

Distintos APIs proveen diferentes tipos de iluminación

Inteligencia Artificial (IA o AI)

Es la característica más importante que se le atribuye a un motor al lado de la representación de modelos o Render. La IA provee de estímulo al juego. Es crítico en la parte de la forma de juego (gameplay).

La inteligencia artificial de determinado juego puede tornarse muy compleja, primero se debe definir la línea base del comportamiento de los NPC (Non Player Characters - Personajes no Jugables), primero debe definirse qué hace el NPC (patrulla, guarda, etc.), luego se delimita su "visión del mundo", que es lo que el NPC puede ver del mundo del juego; se debe tomar en cuenta que el personaje no sólo estará en medio del mundo del juego sino que también interactuará con él, después vienen las rutinas de Toma de Decisión: si el NPC está patrullando, y hay un sonido, ¿debe darle importancia o no?, ¿investiga su origen o no?, etc. Es un sistema de reglas para las acciones que responden (o inician) y que el jugador debe responder, esto es un concepto más general de IA.

Light Map Generation (Generación del mapa de luz): se usa una segunda capa de textura (mapa de luz) que dará el efecto de iluminación a los modelos, es un efecto excelente pero debe tomarse antes del renderizado pero si se tienen Luces Dinámicas (o sea luces que se mueven, encienden o apagan sin intervención de programa) se debe estar regenerando los mapas en cada Frame de animación lo que toma mucha cantidad de memoria (pero son de render rápido).

LOD (level of detail, nivel de detalle):

El sistema de nivel de detalle está relacionado con la complejidad geométrica de los modelos. Algunos sistemas necesitan que se hagan múltiples versiones del modelo, para que dependiendo de cuán cerca se esté del modelo así será su cantidad de polígonos. Otros sistemas ajustan dinámicamente esta característica pero en este caso da más carga al CPU

Mapeo MIP:

Consiste en pre procesar las texturas creando múltiples copias del mismo cada una la mitad del anterior, esto porque si la textura solo es pegada al polígono cada textura es a cada píxel y tomara más tiempo de render; así cada Texel (elemento de Textura) toma menos espacio.

Music Tracks in Games (pistas de audio)

Hay dos formas de manejar el sonido. Uno es por medio de archivos .wav (o similares), lo cual emite un muy buen sonido, pero se requiere de mucha memoria. Por otro lado se puede utilizar archivos midi, esto reduce la necesidad de memoria, pero los sonidos no son tan buenos.

NURBS B-splines racionales no uniformes o (acrónimo inglés de non-uniform rational B-spline) es un modelo matemático muy utilizado en la computación gráfica para generar y representar curvas y superficies.

Patches (Parches)

Son perfectos para describir geometrías, sobre todo cuando se trata de curvas, pues la expresan mediante fórmulas matemáticas logrando colocar puntos en el mundo del juego.

Three-point polygon (polígonos de 3 puntos-Triángulos)

El más usado por las tarjetas aceleradoras 3D.

Objetos 3D

Los objetos se almacenan por puntos en un mundo 3D, llamados vértices. Los vértices van formando polígonos; cuanto más polígonos posea un objeto, más complicado se hace, lleva más tiempo de procesamiento pero es más detallado. El juego no necesita saber cuántos objetos hay en memoria o cómo el Render va a mostrarlos, sólo le interesa que el render despliegue en la forma correcta, y que el modelo esté en el cuadro correcto de la animación.

Phong Shading:

Es similar al Gouraud Shading, trabajan con la textura, solo que el Phong Shading usa a los píxeles en lugar de los vértices.

El Phong Shading toma más tiempo de procesamiento que el Vertex Shading pero sus resultados son mucho mejores en cuestión de suavizado de texturas.

Render (Renderización)

Es la parte del código que pone en pantalla los ambientes y objetos.

Retessellation

Técnica usada por la característica de TruForm de ATI que consiste en tomar un modelo basado en triángulos y transformarlo en uno de High-Order Surfaces para alisarlo y de nuevo pasarlo a un modelo de Triángulos.

Stencil Shadowing (Plantilla de Sombreado):

La idea es renderizar una vista de un modelo desde la perspectiva de la fuente de luz y después utilizar esto para crear o para generar un polígono con la forma de esta textura sobre las superficies afectadas por el modelo. Así se obtiene una iluminación que parece real. Pero es costosa, porque usted está creando texturas "en vuelo", y hace múltiple render de la misma escena.

El manejo del cache de textura es imprescindible para que el juego se desarrolle rápido (y para cualquier motor), ya que si se presenta un constante swapping de las texturas en la tarjeta el juego se verá lento y tedioso, algunos APIs descargan cada textura cuando esto pasa, pero eso haría que en cada cuadro se refresquen las texturas dando más lentitud. Todo se trata de cargar la menor cantidad de veces una misma textura, pero eso también depende del API que se utilice. Otra técnica es la compresión de texturas, comprimir texturas es como comprimir MP3, los algoritmos de compresión logran una relación 4:1 que no es mucho pero ayuda.

Scripting Systems(Sistemas de scripting)

Pre-scripted Cinemáticas: usada normalmente en una situación que necesita la explicación en una manera controlada. Para presentar las escenas de la historia, ahora se utiliza el cortar-escenas que presenta la historia en vídeo digital y luego por medio de transiciones se pasa a las gráficas reales del juego.

El scripting le permite al diseñador tomar mando de la escena y manipularla, como colocar objetos o eventos que el jugador no controla. En muy complicado, se necesita de una mente muy metódica y lógica, la mayoría de estos scripts se basan en lenguaje C.

Visual Scripting Systems: como lo dice su nombre, permite manejar el script en un ambiente gráfico en lugar de un código escrito, se maneja un carácter real en un ambiente del juego real.

Sonido

Creative Labs ahora ha proporcionado sus extensiones manejadores de sonido EAX para DirectX, y la nueva iniciativa de OpenAL (biblioteca audio abierta). OpenAL, como suena, es un API para los sistemas de los sonidos de la misma manera que OpenGL es un API.

Para el procesado de sonido es muy similar al procesado de los modelos, muchas veces un software los procesa antes de pasar al hardware respectivo, por ejemplo DirectSound hace al sonido para la Tarjeta de sonido lo que Direct3D hace al modelado antes de llegar a la Tarjeta 3D. Esto es llamado "pre mezcla" en el software.

Textura:

Es esencial para que las escenas 3D se vean reales, en si las texturas son imágenes que se rompen en los distintos polígonos del modelo, muchas imágenes tomarán mucho espacio en la memoria por eso se debe usar técnicas de compresión:

Texturas Múltiples:

Requiere múltiples renderizados por lo que para obtener buen resultado se necesita una tarjeta con Acelerador de Gráficos, provee mejor calidad que el simple mapeo. Se puede colocar una imagen sobre otra (más transparente) para dar el sentido de movimiento pulso o hasta sombra.

Vertex Lighting

Se determinan cuantos polígonos cruzan el vértice, se toma el total de todas las orientaciones de los polígonos (Normal) y se asigna la normal al vértice. Para cada vértice, un polígono dado reflejará la iluminación en una forma levemente distinta. La ventaja es que al hardware le toma menos tiempo el procesarlo, pero este tipo de iluminación no produce sombras.

Vertex Shading (Sombreado de Vértice, Gouraud shading):

Solicita al motor de renderizado un color para cada vértice, luego por medio de interpolación se renderiza cada píxel por la distancia en relación con su respectivo vértice.

Vertex and Pixel Shaders (Vértices y Sombreo de Pixeles):

Con este método se pueden extraer y utilizar directamente las características y facilidades de la tarjeta de video, sin tener que utilizar mucho la API. Pero no es utilizable en todas las tarjetas.

MOTORES DE JUEGO LICENCIADOS

3D GameStudio	EMotion FX	Offset Engine
3DSTATE 3D Engine	Esperient Creator	Open Game Engine
Agar	Eternity Engine	OGRE
Allegro library	euphoria	PLIB
Antiryad GX	Explorations	Power Render
APOCALYX	Exult	ProQuake
Aurora Engine	FPS Creator	PySoy
Axiom Engine	Frostbite Engine	Quantum3
Baja Engine	Gamebryo Element	Realm Crafter
Blender Game Engine	GZDoom	RealmForge
Blender Publisher	Genesis3D	Refractor Engine
Boom	GemRB	RelentENGINE
BRender	GTGE	RenderWare
Build engine	Hedgehog Engine	Retribution Engine
BYOND	INSANE	Revolution3D
C4 Engine	JADE engine	SAGE engine
Catmother	JAMAGIC	Serious Engine
CheapHack	Jedi	Shark 3D
Coldstone game engine	JGame	ShiVa
CPAL3D	jMonkeyEngine	Silent Storm engine
CEL	Jogre	Sith
CELstart	KallistiOS	Source engine
Crazyball Engine	Kaneva Game	Sploder
Cube	Platform	Telejano
Dagor Engine 3	KjAPI	Tile Studio
DarkBasic	Knight Free 3D Suite	Torque Game Engine
DarkPlaces	KonsolScript	Torque Game Engine Advanced
Daybreak motor	Lithtech	Truevision3D
Dead Souls	Medusa	Unigine
Delta3d	Monumental	Vengeance engine
DGD	Technology Suite	Vicious Engine
Dim3	M.U.G.Y.N	Visual3D.NET
Dungeon Craft	Multimedia Fusion 2	Virtools
DXFramework	Multiverse Network	White Engine
DX Studio	Nebula Device	Wintermute Engine
Earth-4 Engine	NeoAxis Engine	World Builder
Electron engine	Nvidia Perfkit	XnGine
Elflight Engine	Odyssey Engine	Zillions of Games
	OctLight	

MOTORES DE JUEGO GRATUITOS

Adventure Game Studio	KallistiOSKnight Free
Agar	3D Suite
Allegro library	KonsolScript
Antiryad GX	M.U.G.Y.N
APOCALYX	Multiverse Network
Axiom Engine	Nebula Device
Baja Engine	OctLight
Blender Game Engine	Open Game Engine
Blender Publisher	OGRE
Boom	Panda3D
Build engine	PLIB
BYOND	ProQuake
Catmother	Pygame
CheapHack	PySoy
CEL	RealmForge
CELstart	Retribution Engine
Crystal Space	Ren'Py
Cube	RPG Toolkit
DarkPlaces	Sauerbraten
Daybreak motor	Sploder
Dead Souls	Stratagus
Delta3d	Telejano
DGD	Tenebrae
Dim3	Tile Studio
Doom engine	vbGORE
Dungeon Craft	Wintermute Engine
DXFramework	World Builder
DX Studio	XtremeWorl
Epee Engine	
Eternity Engine	
Ethanon Engine	
Exult	
FIFE	
Game Maker	
GQ	
GTGE	
GZDoom	
Genesis3D	
GemRB	
Horde3D	
Irrlicht	
JGame	
jMonkeyEngine	
Jogre	

BIBLIOGRAFIA

Drawing, Author; Andrew Loomis, ed. Viking Press

El héroe de los mil rostros, Autor Joseph Campbell, ed FCE 1947, 1959

El arte de la guerra, autor sun tzu, Ed, fondo de cultura económica

Padre rico, padre pobre, autor; Robert kiyosaki, ed. Aguilar 1997

Cuadrante del flujo del dinero, autor; Robert kiyosaki, ed. Aguilar 1998

Guía para invertir autor; Robert kiyosaki, ed. Aguilar 2001

La conspiración de los ricos autor; Robert kiyosaki, ed. Aguilar 2006

Mi lucha, autor Adolfo Hitler, ed. Editorial del partido nacional socialista de america latina. 1926

Inteligencia Emocional, Autor Daniel Goleman, ed. Vergara. 2001

El mapa estratégico del real Madrid, Autor; florentino Pérez, análisis de Ignacio Urrutia de hoyos, Kimio Kase, Carlos Marti Sanchis, Magdalena Opazo Breton, centro de sport business management-iese business school

BSC- Autor; Kaplan y Norton 1996

Programación para Juegos

Mastering Unreal technology Volume 1, introduction to level design with Unreal Engine 3

Unreal Develoment Kit

Unreal

C++ FOR DUMIES

MICROSOFT VISUAL STUDIO. NET

PHYSICS FOR GAME PROGRAMMERS

FUNDAMENTALS OF MATH AND

PHYSICS FOR GAME PROGRAMMERS

GAME ENGINE ARCHITECTURE

JASON GREGORY

ED. A.K PETERS

2009

Inteligencia Artificial

Programming Game AI by Example, Author Mat Buckland, Editor: Wordware Publishing Inc. (1 de octubre de 2004)

AI Game Engine Programming, Author Brian Schwab, Editor: Charles River Media; Edición: 2nd Revised edition (21 de enero de 2009)

Scripting

Game Scripting Mastery, John Romero (Prólogo), Alex Varanese (Author), Editor: Premier Press; Edición: Pap/Com (18 de diciembre de 2002)

Programming in Lua, Author Roberto Ierusalimschy, Publisher: Lua.org; 2 edition (March 5, 2006)

Networking

Networking and Online Games: Understanding and Engineering Multiplayer Internet Games

Grenville Armitage (Author), Mark Claypool (Author), Philip Branch (Author)

Algorithms and Networking for Computer Games Author(s): Jouni Smed, Harri Hakonen
Published Online: 21 JUN 2006

Sonido/Música

Creating Music and Sound for Games, Author G.W, Editor: Course Technology Inc; Edición: 1 (17 de noviembre de 2006

Audio Programming for Interactive Games, Author Martin D. Wilde, Editor: Focal Press; Edición: Pap/Cdr (11 de marzo de 2004 FOCAL PRESS

Business plans handbook
Volume 22
Michelle leel
Project editor
GALE CENGAGE LEARNING
2012

How to write a business plan
Mike McKeever
NOLO, 8th edition, 2007

MESOGRAFIA

REVISTAS

GAME PRO®
PODER Y NEGOCIOS®
CINE PREMIER®
TOP FILM®
MAXIM®
NATIONAL GEOGRAPHIC
CONOZCA MÁS
QUO
DARK HORSE COMICS®

VIDEOJUEGOS

ASSASSINS CREED®
ALAN WAKE®
ALIENS®COLONIAL MARINES
ALIENS VS PREDATOR®
ALIENS VS PREDATOR 2010®
ALICE®
ALICE RETURN TO MADNESS®
ALPHA PROTOCOL®
BLACK SHARK®
BLADES OF TIME®
BATMAN® ARKHAM CITY
BATMAN® ARKHAM ASYLUM
BATMAN® ARKHAM ORIGINS

BATTLE FIELD® BAD COMPANY 3
BIG GAME HUNTER
CALL OF JUAREZ® BLOOD
REDEMPTION
GUNSLINGER

CALL OF DUTY®
CALL OF DUTY® GHOST
COMPANY OF HEROES®
DARK SIDERS 2, DEATH LIVES®
DEAD SPACE 3®
DOOM®
GHOST RECON®
HITMAN®: ABSOLUTION
INJUSTICE GOD AMONG US®
MARS WAR LOGS®
METRO LAST NIGHT®
MEDAL OF HONOR® WARFIGHTER
ALLIED
ASSAULT
AIRBORNE

REMEMBER ME®
TOMRIDER®
MAFIA 2®
OPERATION FLASH POINT®
STAR TREK® THE BIRTH OF
FEDERATION
LEGACY
SNIPER ELITE® V2
SNIVER ELITE® 2
SNIPER ELITE® NAZI ZOMBIE ARMY
2
SNIPER GHOST WARRIOR®

SNIPER GHOST WARRIOR 2®
SUPER STREET FIGHTER IV ARCADE
EDITION®
STEEL TIDE®
TAKE ON HELICOPTERS®
TAKE ON MARS®
TERMINATOR SALVATION®
THE MATRIX® PATH OF NEO

SPARTACUS®
IRON MAN®
ALIENS®
PREDATOR®
THE SOPRANOS®
NEON GENESIS
EVANGELION®

THE WALKING DEAD® SURVIVAL
INSTINT
THE AMAZING SPIDER MAN®
THE SIMPSONS ®
THE KLUB 17®
TOTAL WAR® ROME
TOTAL WAR® ROME 2
TOTAL WAR® NAPOLEON
TOTAL WAR® VIKING INVASION
PROCYCLING 2013®
PHOENIX® FLIGHT SIMULATOR®
JAMES BOND®

LEGENDS®
QUANTUM OF

SOLACE®

BLOOD STONE®
GOLDEN EYE®

JURASSIC PARK®: THE GAME
SPIDER MAN® SHATTERED
DIMENSIONS
SPLINTER CELL® BLACK LIST
STAR WARS® THE FORCE
UNLEASHED

THE FORCE

UNLEASHED 2

JEDI ACADEMY
BATTLE FRONT 2
THE OLD

REPUBLIC

KNIGHTS OF THE

OLD REPUBLIC
WANTED®

FILMOGRAFIA

SKYFALL®
PROMETHEUS®
HAGIME NO IPPO®
BEN HUR®

WWW.

GOOGLE.COM
GOOGLEADWORS.COM
WIKIPEDIA.COM
esrb.org/index-js.jsp
AUTODESK.COM
AUTODESK.MX/
[http://fabiensanglard.net/quake2/
unocero.com/2013/04/15/que-tan-
alta-es-tu-capacidad-de-asombro/](http://fabiensanglard.net/quake2/unocero.com/2013/04/15/que-tan-alta-es-tu-capacidad-de-asombro/)

[http://www.unocero.com/2013/04/
19/alas-girl-scouts-en-estados-
unidos-estrenaran-una-medalla-
para-fomentar-la-programacionlas-
girl-scouts-en-estados-unidos-
lanzan-medalla-para-fomentar-la-
programacion/](http://www.unocero.com/2013/04/19/alas-girl-scouts-en-estados-unidos-estrenaran-una-medalla-para-fomentar-la-programacionlas-girl-scouts-en-estados-unidos-lanzan-medalla-para-fomentar-la-programacion/)

SOFTWARE

AUTODESK GAMEWARE
AUTODESK humanIK
AUTODESK SCALEFORMS
AUTODESK 3D MAX
AUTODESK MAYA
ADOBE FLASH
ADOBE PHOTOSHOP
ADOBE
UNREAL ENGINE 3
CRYWARE

**TODAS LAS MARCAS SON DE
SUS RESPECTIVOS
PROPIETARIOS**