



Videojuego serio para la enseñanza del esqueleto de la cabeza humana

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor:

Ariel Padrón Gómez
Roberto Elías Pérez Ozete

Tutor:

MSc. Omar Correa Madrigal

Cotutor:

Ing. Loyda Cárdenas Rey

La Habana, junio del 2013
Año del 54 aniversario de la Revolución

“Todo en la vida comienzan con el primer paso”

DECLARACIÓN DE AUTORÍA

Declaramos ser únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los _____ días del mes de junio del año 2013.

Autores

Tutor

DATOS DE CONTACTO

- Ariel Padrón Gómez, estudiante de la carrera Ingeniería en Ciencias Informáticas (UCI). Perteneciente al Centro Informática Industrial (CEDIN) de la facultad 5. Línea de trabajo Realidad Virtual y Visualización 3D.
 - Teléfono personal: 5 815 8300
 - Teléfono fijo: 032 812002
 - E-mail: whixerpro@gmail.com
 - Dirección particular: Pepito Tey # 94 / Grabiél Garcia y Victoria, Guáimaro, Camaguey.

- Roberto Elías Pérez Ozete, estudiante de la carrera de Ciencias Informáticas (UCI). Perteneciente al Centro Informática Industrial (CEDIN) de la facultad 5. Actualmente trabaja en la línea de Realidad Virtual y Visualización 3D.
 - Teléfono personal: 5 244 0239
 - Teléfono fijo: 045 527448
 - E-mail: roberto.ozete@gmail.com
 - Dirección particular: Calvo # 1158 / 24 y 25, Cárdenas, Matanzas.

AGRADECIMIENTOS

Agradecer a los grandes hombres de ciencia Frank Netter, Andrew Rolling y Dave Morris por acompañar esta tesina desde sus inicios, ya que sus documentos han sido de uso imprescindible para esta investigación.

- Agradecer fundamentalmente la confianza que han depositado mis padres en todo momento de mi carrera estudiantil, apoyándome siempre en todas las decisiones que he tomado y enseñándome el lado positivo de las cosas, también a Betty por ser la mejor hermana, compañera y amiga en el mundo, siempre apoyarme y aconsejarme a pesar de su corta edad.

Agradecer a mi primo Frank por su ayuda prestada en disímiles momentos y a mi gran amigo Andy por su gran apoyo incondicional, su comprensión y su confianza en todos estos años, siempre es bueno saber que existe un hombro firme en el uno se puede apoyar y que este siempre va a estar ahí incondicionalmente.

Agradecerle a Luna por estar siempre ahí pendiente de mi situación, dándome fuerzas cuando mi ánimo decae y cuando no también, te has ganado un lugar muy importante en mi vida. Y no por últimos menos importantes a todos mis amigos de la universidad que han jugado un gran papel todos estos años, siempre ahí aguantando y apoyando mis locuras.

Y de forma general quiero darle un gran agradecimiento a todo el que de una forma u otra ha puesto su granito de arena en la elaboración de “Aventura Anatómica”, mi padre siempre me dice que todo en la vida comienza con el primer paso, pues ahora mírame, lo estoy dando.

Roberto.

- Agradezco la dicha que me ha dado la vida de tener a los padres que tengo, a mi hermano, mi tía Ana y mi sobrino; que han sido los motores impulsores de esta obra. Agradezco a toda mi familia por el entorno de armonía en que he vivido, a Chela y Nenita por estar siempre cerca y ser unas madres para mí. Agradezco al Ruso, Deny y Emily por sus consejos y por la amistad incondicional de toda la vida, a Osvy y Meisel por los años de los años y ser quienes son a pesar del tiempo y la distancia. Agradezco a Yami por su apoyo y preocupación en cada momento de mi carrera. A Ernesto, Raudy y Osmany por guiarme en mis años de estudio en la Universidad y más que todo por los momentos vividos y la fiel amistad. A Lisbett por todas las noches de mortificaciones antes de cada examen. Agradezco a mis compañeros de apartamento, al Fide, al Negro, a Raul, Orlando, Syndi, Adrian y a mi amigo y compañero de tesis Roberto por confiar y tener los mismos ideales y sueños de alcanzar las más grandes metas en la vida. Agradezco a todos, por formar parte de mi universo.

Ariel

DEDICATORIA

El presente trabajo de diploma va dedicado a mi papá, que una vez quiso que fuera médico; hoy soy Informático y pienso con este trabajo hacerle algún aporte a la medicina. A mi madre, por ser mi ángel y guiar cada paso que camino en esta vida. A mi hermano, por ser una pieza clave en la familia y a mi tía Ana Luisa por su amor, dedicación y ser su hijo menor.

Ariel

El presente trabajo de diploma va dedicado fundamentalmente a todo el esfuerzo realizado por mis padres para que hoy pudiese alcanzar este lugar y a la memoria de mi abuelo Gustavo por todo su esfuerzo para verme salir bien en los estudios. A todos, gracias.

Roberto

RESUMEN

El estudio de la anatomía humana hace uso de diferentes sistemas informáticos, dígame multimedias, foros y sitios web. La manera de abordar estos contenidos de manera lúdica no es muy frecuente. Con este trabajo se crea un videojuego que permitirá el estudio de la anatomía humana, específicamente del esqueleto de la cabeza.

En la confección de este videojuego serio se mantiene el equilibrio entre juego y el contenido a aprender. Se emplean recursos psicológicos y formas de estructuración del conocimiento en los estudiantes para diseñar el videojuego. Se utilizan además diseños gráficos en tres dimensiones, sonidos y un guión de aventura que guía la historia del videojuego. El proceso de desarrollo se realizó utilizando la metodología Huddle y la herramienta principal de desarrollo fue el motor de videojuego Unity. Esta combinación permitió terminar el videojuego en el periodo de 9 meses, con un equipo de desarrollo de dos personas. Las pruebas realizadas, dígame, de rendimiento (tiempo real) y jugabilidad arrojaron resultados satisfactorios.

Palabras Clave: anatomía humana, recursos psicológicos, videojuegos serios.

Índice

INTRODUCCIÓN	10
CAPITULO 1: Fundamentación Teórica	13
1.1 Juego Serio (Serious Game)	13
1.2 Clasificación de los videojuegos serios.	14
1.3 Teoría psicológica en el diseño del videojuego.	14
1.3.1 Factores cognitivos:.....	16
1.3.2 La memoria subconsciente:.....	16
1.4 Bases del plan de estudio de la Morfología Humana, aspectos relacionados con el esqueleto de la cabeza humana.	16
1.5.1 Proceso de Desarrollo	17
1.5.2 Fase de Pre-producción	18
1.5.3 Fase de Producción.....	19
1.6 Metodología de desarrollo de Software.	21
1.7 Herramientas utilizadas	23
1.7.1 Gráficos 2D	23
1.7.2 Gráfica 3D	24
1.7.3 Edición de Sonido Adobe Audition CS6	24
1.7.4 Motores de videojuegos	24
CAPÍTULO 2: Solución Técnica.....	30
2.1 Relación entre el diseño del videojuego y los elementos psicológicos.	43
CAPÍTULO 3: Ingeniería del Sistema	45
3.1 Requisitos Software.....	45
3.1.1 Requisitos funcionales.....	46
3.1.2 Requisitos no funcionales del Videojuego	51
3.2 Casos de Uso.....	52
3.3 Diagrama de clases.....	60
3.4 Implementación	61
3.4.1 Componentes	62
3.4.2 Scripts	65
3.5 Prueba.....	70
3.5.1 Pruebas de Rendimiento.	70

3.5.2 Pruebas de Jugabilidad	72
Conclusiones	74
Recomendaciones.....	75
Bibliografía	76
Glosario de términos:	78

INTRODUCCIÓN

Los niños desde que nacen aprenden a conocer el mundo a través de los juegos, estos les permite socializarse y relacionarse con su entorno. Jugar para ellos es una capacidad innata (no enseñada) por medio de la cual es posible representar sus sentimientos y vivencias. A lo largo de la vida, el ser humano ha utilizado los videojuegos como instrumento de entretenimiento, para adquirir habilidades y destrezas que le permitan desempeñarse en su existencia, facilitando su integración, su creatividad y ayudándolo en el aprendizaje físico-motriz.

En los videojuegos se incluyen uno o varios usuarios llamados *players* o jugadores, lo cuales mantienen una interacción constante con varias interfaces, como pueden ser los *joysticks* o controles, teclado, mouse y un dispositivo de video. Los videojuegos pueden recrear situaciones virtuales de cualquier índole (fantasía, acción, lógica, deporte, terror), entornos en donde las reglas de la física pueden romperse y distorsionarse, mundos fantásticos o situaciones con un realismo extraordinario.

Esto se logra gracias a la conjunción de numerosas artes que estimulan los sentidos y encienden los sentimientos de los jugadores. Así es como un videojuego, a través de determinadas reglas, sumerge al jugador en una situación donde se deben resolver problemas, eliminar obstáculos, cumplir objetivos y superar metas para llegar al final y obtener algún tipo de recompensa.

Los videojuegos como medio de entretenimiento, han evolucionado de manera abrupta en los últimos años, convirtiéndose en una industria multimillonaria, superando en ocasiones a la industria del cine, en casos de superproducciones de varios años de desarrollo que cuentan con cientos de programadores, artistas gráficos y de sonido, narración, marketing, leyes, inclusive actores. (Jesús Arce, 2011).

Desarrollar un videojuego implica conocer disciplinas y poseer habilidades, pasando por la creatividad y recorriendo varias ciencias formales y sociales, cuyo proceso va mucho más allá de un típico desarrollo de software. Así es como los videojuegos unifican el arte, la ciencia y la tecnología. Hoy en día debido a la gran interoperabilidad y las funciones que brindan los motores de juegos, se les han dado otros usos, como son la realización de sistemas informáticos más enfocados en las empresas, el marketing, la salud, la educación y las ciencias militares.

La presente investigación está enmarcada en el campo de la salud, más específicamente dentro del campo de la Anatomía Humana, debido a que esta materia en la formación de un médico, en todos los tiempos ha constituido una de las bases esenciales de su malla curricular y tradicionalmente su estudio se hace a través del sistema de conferencias, clases prácticas y seminarios.

La sociedad actual "sociedad de la información y el conocimiento", ofrece soportes digitales donde el estudiante debe interactuar con la información científica, la teleclase, la clase taller, el seminario integrador y la consulta docente, como nuevos retos de formación.

A nivel mundial investigadores e instituciones responsables de educación, están de acuerdo en la necesidad de adaptar la universidad a la "sociedad del conocimiento" y en destacar el rol que las

Tecnologías de la Información y las Comunicaciones (TIC) juegan en la construcción de medios de enseñanza (Barajas M, 2003), constituyendo un nuevo paradigma de trabajo académico para la educación superior.

Estos conceptos implican una integración entre las formas tradicionales de educación y las nuevas tecnologías. En Cuba el estudio de la anatomía se basa en el uso de las formas tradicionales de la educación, empleando los altos volúmenes de información, los atlas anatómicos y varios sistemas informáticos haciendo uso de las tecnologías. Las mismas en ocasiones con bajo nivel tecnológico, poco atractivas y poco motivantes antes los ojos del alumnado, factores que influyen en que la curva de aprendizaje tienda a decaer, motivo que conllevó a resaltar la poca existencia de medios tecnológicos que permitan el aprendizaje, en este caso, del esqueleto de la cabeza humana desde un enfoque lúdico. Enfoques lúdicos que han tenido sus efectos y muestras fehacientes en el contexto internacional como la escuela de Medicina de la Universidad Pedagógica y Tecnológica de Colombia, donde se evaluó un Entorno Virtual de Aprendizaje (EVA) como apoyo para medir el impacto en el rendimiento académico y la percepción de los estudiantes (Meléndez, 2009); así como el trabajo realizado por *Sandra Bucarey*, primera titulada en el Magíster en Modelado del Conocimiento para Entornos Educativos Virtuales (Bucarey, 2006).

En la provincia de Villa Clara en el año 2004 se hizo una investigación que tenía por nombre "Propuesta didáctica ante los problemas cognitivos de la Anatomía Humana", la misma llevó una revisión analítica de documentos rectores de la carrera de Medicina que arrojaron los siguientes problemas cognoscitivos:

- La falta de **herramientas metodológicas**, terminología y referencias conceptuales para iniciar los estudios en la Anatomía Humana.

En cuanto a la parte pasiva del aparato locomotor (Piug, 2001) se tiene como principales problemas:

- Identificar en piezas anatómicas los distintos huesos del esqueleto y sus partes principales, así como las articulaciones que se establecen entre ellos.
- Interpretar los principios de la mecánica articular, a partir del número y las formas de caras articulares que conforman una unión ósea específica.

Además de los problemas éticos y bioéticos que conlleva el uso de materias vivas para las disecciones cuando existen otros medios que permiten estas visualizaciones.

Teniendo en cuenta las necesidades y la situación existente en el área de desarrollo de sistemas informáticos para el estudio de la anatomía humana se plantea el **problema científico**: ¿Cómo desarrollar un sistema informático que sirva como complemento para el aprendizaje del esqueleto de la cabeza humana?

Este trabajo tiene como **objeto de estudio**, los videojuegos como medio de aprendizaje.

Como **campo de acción** los videojuegos para el aprendizaje de la Anatomía Humana.

El **objetivo general** desarrollar videojuego para el aprendizaje de la Anatomía Humana.

Para alcanzar los objetivos propuestos se realizarán las siguientes **tareas investigativas**:

- Análisis y elaboración del guión, medios médicos a utilizar y ambientes con los que se va a trabajar.
- Análisis del estado del arte, las herramientas y metodologías para la realización de un videojuego con fines educativos.
- Desarrollo de los artefactos ingenieriles necesarios para la los flujos de trabajo de Análisis y Diseño.
- Implementación de la solución.
- Pruebas de la aplicación.

Métodos teóricos

Histórico - Lógico: se empleó para la fundamentación y sistematización de los aspectos teóricos contemplados en el desarrollo de la investigación acerca de los videojuegos, las metodologías a utilizar, los planes de estudios y demás elementos relacionados con el contenido del trabajo.

Analítico - Sintético: durante el proceso de investigación permitió descubrir los elementos esenciales, concepciones y conceptos en torno al objeto de investigación.

Entrevista o Encuesta: este método permitió establecer una comunicación con los usuarios escogidos para la realización de las pruebas de jugabilidad del videojuego.

Métodos Empíricos

Observación: Se empleó como método referencial al observar distintos videojuegos que sirvieron como objeto de análisis y comparación para establecer las características y elementos fundamentales que debía cumplir la propuesta que plantea el autor.

CAPITULO 1: Fundamentación Teórica

En el presente capítulo se hace un estudio de las diferentes clasificaciones de videojuegos serios existentes así como los recursos psicológicos a utilizar durante la implementación del mismo. Se hace un análisis del plan de estudio de la Morfología Humana, además de un estudio de las tecnologías, herramientas y metodología a utilizar en el proceso de desarrollo de la aplicación.

1.1 Juego Serio (Serious Game)

En la actualidad se hace complejo definir el género de un videojuego, debido a que cada vez estos se interrelacionan y mezclan sus temáticas. La presente investigación se centra en el estudio de los videojuegos serios ya que son estos los que permiten representar los contenidos de una materia de manera divertida y dinámica.

En 2005, Mike Zyda abordó de una forma actualizada los conceptos referentes a que es un juego, un videojuego y un videojuego serio en un artículo publicado en la revista *"Computer"* de la *IEEE Computer Society* (Zyda, 2005).

Juego: una prueba física o mental, llevada a cabo de acuerdo con unas reglas específicas, cuyo objetivo es divertir o recompensar al participante.

Videojuego: una prueba mental, llevada a cabo frente a una computadora de acuerdo con ciertas reglas, cuyo fin es la diversión o esparcimiento, o ganar una apuesta.

Videojuego serio: una prueba mental, de acuerdo con unas reglas específicas, que usa la diversión como modo de formación gubernamental o corporativa, con objetivos en el ámbito de la educación, sanidad, política pública y comunicación estratégica.

Otra definición de renombrados profesionales del ámbito en cuanto a los videojuegos serios es la siguiente: "Un videojuego serio es un juego en el que la educación (en sus diversas formas) es el principal objetivo, en lugar de entretenimiento." (Clark C, 1987).

Esta definición ayuda a aclarar la contradicción aparente, pero sin duda también genera confusión en algunas personas (tanto desarrolladores de videojuegos como educadores) que ven incompatibles al entretenimiento y la educación. Pues no sólo la educación y el entretenimiento no son conflictivas entre sí, sino que hay muchos lugares donde los dos conceptos se superponen y donde cada parte puede utilizar las herramientas de la otra para lograr sus objetivos.

1.2 Clasificación de los videojuegos serios.

Aunque la clasificación de los videojuegos serios es algo que todavía tiene que consolidarse, existen sin embargo una serie de términos que se han comenzado a utilizar en la actualidad los cuales se muestran a continuación.

Advergaming: publicidad y juego, es la práctica de usar videojuegos para publicitar una marca, producto, organización o idea.

Edutainment: este es un término que resulta de la unión de *education* (educación) y *entertainment* (entretenimiento), es decir, educación y entretenimiento o diversión. Se aplica a los programas que enseñan mediante el uso de recursos lúdicos.

Aprendizaje basado en juegos: tienen como objetivo mejorar el aprendizaje. Están diseñados en general manteniendo un equilibrio entre la materia y la jugabilidad teniendo en cuenta la capacidad del jugador para retener y aplicar dicha materia en el mundo real. Es utilizado en el mundo empresarial para mejorar las capacidades de los empleados en temas como la atención al público y negociaciones.

Edumarket Games: cuando un juego serio combina varios aspectos (por ejemplo, los propios del *advergaming* y del *edutainment* u otros relacionados con la prensa y la persuasión), se dice que la aplicación es un juego de tipo *edumarket*, término que resulta de la unión de *education* (educación) y *marketing* (publicidad). Un ejemplo es *Food Force*, un juego con objetivos en el ámbito de las noticias, la persuasión y el *edutainment*.

News Games: son juegos periodísticos (del inglés *news*, es decir, noticia) que informan sobre eventos recientes o expresan un comentario editorial. Un ejemplo es *September 12th*, un juego creado por Gonzalo Frasca que denuncia el uso de la violencia para resolver el problema del terrorismo.

Simuladores o videojuegos de simulación: son juegos que se emplean para adquirir o ejercitar distintas habilidades o para enseñar comportamientos eficaces en el contexto de situaciones o condiciones simuladas. En la práctica, son muy usados los simuladores de conducción de vehículos, coches, trenes, aviones, etc., como por ejemplo *FlightGear*), los simuladores de gestión de compañías (por ejemplo, *Transport Tycoon*) y los simuladores sobre negocios en general, que ayudan a desarrollar el pensamiento estratégico y enseñan a los usuarios los principios de la micro y macroeconomía y de la administración de empresas (por ejemplo, *Virtonomics*).

Militainment: este es un término que resulta de la unión de *military* y *entertainment*, es decir, militar y entretenimiento o diversión. Son juegos financiados por el ejército o que, de lo contrario, reproducen operaciones militares con un alto grado de exactitud.

1.3 Teoría psicológica en el diseño del videojuego.

Todo videojuego implica el uso de factores psicológico en mayor o menor medida para atrapar al jugador en su historia, para hacer que el usuario se enamore y quiera jugar el videojuego hasta el final. En la presente investigación se hace un estudio de varios de estos factores pero particularmente en los que influyen directamente en la adquisición del conocimiento.

El ser humano almacena el conocimiento en estructuras semánticas. Las estructuras se organizan a partir de la interacción con el mundo externo usando los procesos perceptuales. Esta información se almacena en diferentes tipos de memoria y se rechaza o se mantiene según diversos criterios, lo que se puede llamar "aprendizaje". Existen dos modelos que permiten estudiar el aprendizaje de un determinado concepto por parte de una persona:

Modelo mental: A partir de las experiencias y el aprendizaje realizado por la persona, éste tiene un modelo inexacto y sólo conoce relaciones sencillas del concepto estudiado ("si se cumple esta condición, ocurre esto"). Por ejemplo, sabe que si presiona el acelerador de un coche, éste se mueve, pero no sabe por qué (no entiende su principio mecánico).

Modelo conceptual: El estudio del concepto se hace de manera detallada y se conocen todos los mecanismos que actúan sobre él. Es el modelo detallado de un concepto. Siguiendo con el ejemplo anterior, un ingeniero mecánico conoce al detalle cómo está construido un coche.

En el caso de la programación de aplicaciones, el modelo que tiene el programador sobre su aplicación sería un modelo conceptual, mientras que el modelo por el que se forman los diferentes usuarios sería un modelo mental. Para una misma aplicación, los dos modelos deben asemejarse tanto como sea posible. Cuanta más divergencia haya entre ellos, más probable será el rechazo por parte del usuario. Si se quiere que el videojuego sea intuitivo, se debe pensar en cómo crear un modelo conceptual para que se asemeje al del usuario.

De igual manera se hizo necesario revisar los estilos de aprendizajes, que no son más que los rasgos cognitivos, afectivos y fisiológicos que sirven como indicadores relativamente estables de cómo los discentes perciben, interactúan y responden a sus ambientes de aprendizaje.

1.3.1 Factores cognitivos:

Entre estos factores están:

- Las modalidades sensoriales.

Las modalidades sensoriales preferidas por cada sujeto es, sin duda, otro elemento que debe analizarse. Los individuos se apoyan en distintos sentidos para captar y organizar la información, de forma que algunos autores la esquematizan de la siguiente forma:

- Visual o icónico lleva al pensamiento espacial.
- Auditivo o simbólico lleva al pensamiento verbal.
- Cinético o inactivo lleva al pensamiento motor.

1.3.2 La memoria subconsciente:

A diario llega al cerebro mucha información a través de los órganos sensoriales; en realidad se percibe la mayoría de esta información, existiendo diferentes maneras de manejarlas. Las personas pueden ver cosas no solo de manera consciente, mientras más conscientes somos del entorno el cerebro percibe muchas más cosas de su alrededor puesto que a través de la *visión periférica* (Cascales, 2007) se recibe mucha información del entorno que nos rodea siendo el procesamiento preconsciente el encargado de procesar toda la información en el subconsciente o mente interior antes de que sea registrada conscientemente, todo esto sucede gracias a la visión periférica. No son solo las cosas que se ven las que nos afectan, las cosas que se ven de manera no consciente a menudo tienen más impacto que las que se observan conscientemente.

1.4 Bases del plan de estudio de la Morfología Humana, aspectos relacionados con el esqueleto de la cabeza humana.

En el presente apartado luego del estudio de los diferentes tipos de videojuegos serios y la importancia que implica el uso de los factores psicológicos en su implementación, se necesita hacer una revisión del plan de estudio de la Anatomía Humana en Cuba. Esto permitirá organizar la información a representar en el videojuego además de constatar la importancia del estudio del esqueleto de la cabeza humana, debido a su complejidad entre las demás partes que componen el aparato locomotor.

La disciplina de Morfología Humana que se imparte en el primer año de la especialidad de Licenciatura en Enfermería y Tecnología de la Salud en las Facultades de Ciencias Médicas de Cuba, se caracterizan por los contenidos de las Ciencias Morfológicas interrelacionados, con el inconveniente hasta ahora de utilizar como libros de textos básicos, los correspondientes a las disciplinas de Anatomía, Histología y Embriología, que se imparten de forma independiente de la especialidad de Medicina.

En el plan de estudio se tratan los aspectos esenciales de las estructuras que componen el organismo humano desde el punto de vista macroscópico y del desarrollo, de forma que permitan determinar las características generales de cada sistema orgánico y precisar las características particulares más importantes de los órganos que los componen; por lo tanto se evita el exceso de detalles. Además, se incorporan aspectos de la filogenia, anatomía de superficie y radiológica, morfología funcional y aplicada de importancia clínica.

También se trata de utilizar una redacción lo más clara, sencilla y armónica posible, con el empleo de la terminología morfológica internacional, y se agregaron los sinónimos más usuales. Las ilustraciones están basadas principalmente en cuadros sinópticos y dibujos esquemáticos en cantidad suficiente, para que puedan ser de utilidad a los estudiantes en las actividades prácticas y el estudio independiente. El libro consta de dos tomos para facilitar su manipulación por los estudiantes, que se corresponden con los contenidos de las dos asignaturas que componen la disciplina de Morfología Humana. En el tomo I se desarrollan los temas de las generalidades de la Morfología Humana, del nivel celular y tisular y del período prenatal, así como los sistemas somáticos (tegumentario y osteomioarticular). El tomo II abarca los temas de la esplacnología, que comprende los aparatos digestivo, respiratorio, urinario, reproductor y endocrino; además, incluye el aparato circulatorio, el sistema nervioso y los órganos de los sentidos. (Piug, 2001).

1.5 Fases del proceso de Diseño y Desarrollo de un videojuego.

A lo largo de este apartado se comentará cómo se diseña un videojuego y cómo se conforma el equipo de desarrollo. Se describirá la metodología de Ingeniería del Software usada y qué actividades se realizan para crear un videojuego. Podríamos definir diseño y desarrollo de un videojuego como las actividades multidisciplinares en la cual se diseña y se produce un videojuego, desde su concepto inicial, hasta su versión final.

1.5.1 Proceso de Desarrollo

Aunque los videojuegos se consideran un tipo de software más, no existe una metodología común y propia para su diseño y posterior desarrollo. Se puede asegurar, que realizar un videojuego no es una tarea sencilla y que requiere de mecanismos y metodologías de desarrollo de software propio, sin embargo, no existe una metodología estándar que rija dichos procesos y asegure su calidad. Son las propias compañías las que fijan cuál será su filosofía de trabajo a lo largo de la creación de un juego. En lo que sí coinciden diversos autores como Rollings (Rollings&Morris, 2003) y especialmente Bethke (Bethke, 2003), es que el desarrollo del juego, a lo largo de su ciclo de vida, se puede asemejar al de una película de cine, pudiéndose segmentar en tres fases ampliamente diferenciadas: Pre-Producción, Producción y Post-producción, cada una con sus etapas características.

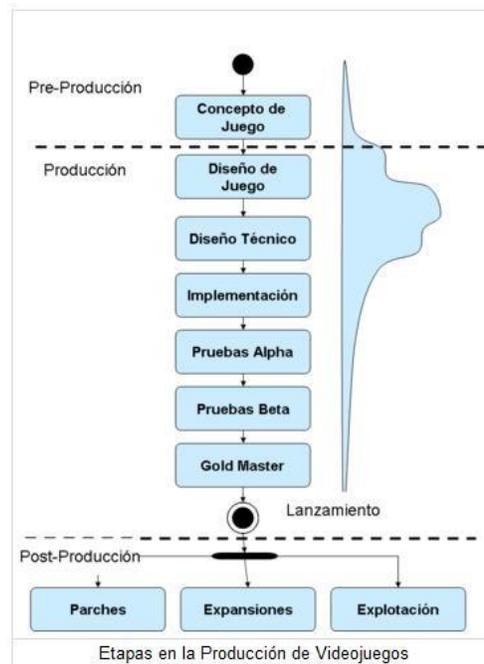


Ilustración 1: Etapas en la producción de videojuegos.

1.5.2 Fase de Pre-producción

La fase de Pre-producción es la etapa inicial del proceso de desarrollo, se caracteriza sobretodo porque se realiza la concepción de la idea del videojuego, es decir, los aspectos fundamentales que conformarán el videojuego a lo largo de su construcción, a continuación se definen cada una de las tareas a realizar en esta etapa:

Género: Se debe especificar el género o los géneros al que pertenece el juego para así establecer las características básicas para su posterior diseño.

Historia: Se debe realizar un esbozo de la trama o historia a desarrollar por el juego, indicando qué se quiere contar y cómo se quiere contar (storyline y storytelling).

Bocetos: Se crean bocetos o diseños preliminares de los personajes y dónde transcurrirá la acción del juego, ya sean decorados, ambientaciones, ropaje, música, movimientos entre otros.

Gameplay: Es la parte más importante de este proceso de concepción del juego. El gameplay es un concepto amplio y difuso, que se define de manera diferente en cada juego. Se podría definir como la esencia, grado o naturaleza del videojuego, incluyendo parte de la interactividad. Es aquí donde vamos a definir cómo se va a jugar, qué cosas se pueden hacer en el juego y cómo va reaccionar el entorno del juego a las acciones del jugador a través del personaje. A su vez estableceremos cómo será la curva de aprendizaje del jugador. Todo esto sin entrar en detalles gráficos, sonoros o de historia. (Zimmerman,

2003).

Una vez cerrada la fase de Pre-producción se debe crear la primera versión del documento de diseño del videojuego (del inglés Game Design Document, GDD) (Bethke, 2003). En esta versión se muestran las primeras especificaciones del juego, plasmando los puntos anteriores. Es la base fundamental para comenzar la fase de Producción, en especial la de diseño del videojuego, pues fija todo lo que se trabajará en ella.

1.5.3 Fase de Producción

Es la fase del proceso que más tiempo demora, es donde los mayores esfuerzos se invierten en las etapas de diseño del videojuego y del diseño técnico. Es en estas etapas donde más personas colaborarán, pues al equipo de diseño de videojuego inicial se le incorporará el resto de la plantilla asociada a la producción del juego. Concluida todas las etapas que componen la etapa de producción se tiene una primera versión del producto final. A continuación, se describen cada una de las etapas.

Diseño de Juego: se detallan todos los elementos que compondrán el juego, dando una idea clara a los miembros del grupo desarrollador de cómo son. Se termina el GDD diseñando en profundidad los aspectos anteriormente especificados.

Diseño Artístico: se crea la Biblia de la Historia donde se recogen todas las historias de los personajes del mundo donde sucede el videojuego, de su pasado y de los personajes secundarios que aparecen, creando el hilo argumental completo, con todos los detalles.

Sonido: Se diseñan todos los elementos sonoros del videojuego: voces, ambiente, efectos y música. Se comienza el motor de sonido.

Interfaz: Se describe la forma en que se verán los elementos GUI (del inglés Graphical User Interface) y HUD (del inglés Head-Up Display), mediante los cuales el usuario interactuará con el juego.

Gráficos: Dependiendo de si el juego es 2D o 3D se diseñarán los elementos gráficos como los *sprites*, modelos 3D, cámaras y luces a utilizar. Se da comienzo al motor gráfico.

Diseño Mecánico: Se diseña cómo se va a interactuar en el juego y las reglas que rigen a éste y las comunicaciones que deben darse en caso de jugarse on-line. Además, se diseña el comportamiento de los personajes y del mundo que les rodea, así como sus habilidades y otros detalles. Se comienza a diseñar la Inteligencia Artificial (IA) del juego y el motor asociado a ella. También se diseña el Motor Físico, encargado de generar diversos aspectos físicos de los elementos y del mundo donde se lleva a cabo el juego (explosiones, disparos, caídas).

Motor del Juego: Es el elemento más importante, y se puede decir que es la parte fundamental de este proceso tan costoso y duradero. El Motor del Juego hace referencia a una serie de rutinas que permiten la representación de todos los elementos del videojuego (Bethke, 2003) (Rollings&Morris, 2003). Del mismo modo en que la carrocería, la pintura y los exteriores no andan sin un motor, el arte y los guiones del juego no funcionan sin un motor del juego. Es aquí donde se debe controlar cómo se representan los elementos del juego y cómo se interactúa con ellos. Se gestionará la IA del videojuego, los sonidos asociados a cada elemento y todos los aspectos gráficos asociados a éstos, incluida la *cinemática* de éste. Se puede decir que el Motor del Juego equivale a una conjunción del Motor Gráfico, Motor de Sonido, Gestor de IA y Motor Físico, más las reglas necesarias para crear el universo completo del juego.

Diseño Técnico: Esta es la etapa más relacionada con el diseño del Software, pues es donde se tratará el videojuego como un verdadero producto de software. Esta etapa describirá cómo se llevará a cabo la implementación del videojuego en una máquina real a través de una determinada metodología y mediante un lenguaje concreto. Se generarán los diagramas que describan el funcionamiento estático y dinámico, la interacción con los usuarios y los diferentes estados que atravesará el videojuego como software.

Esta fase incluye la planificación del juego y en ella se identifican las tareas necesarias para desarrollarlo, repartiéndolas entre los distintos componentes del equipo desarrollador.

Implementación: La etapa de implementación consiste en "montar todas las piezas" descrito anteriormente utilizando el Motor del Juego. Se finalizan todos los contenidos del juego: misiones, scripts, efectos e IA. Este proceso tiene pocas innovaciones y es donde se pulen algunos errores detectados en el diseño inicial. No existen dependencias y el ritmo de trabajo debe ser constante por parte del equipo de desarrollo. Se pueden desechar ideas que son buenas, pero que no encajan al final del juego y corregir imprevistos.

Pruebas Alpha: Durante las pruebas Alpha o Code Complete se tiene un producto terminado. Este producto es probado por un equipo pequeño, que ha estado involucrado en el diseño y desarrollo del juego, en busca de errores para su refinamiento. Uno de los aspectos a probar es la Jugabilidad y de la forma de probarla se hablará a lo largo de este trabajo.

Pruebas Beta: En las pruebas Beta o Content Complete se terminan todas las variaciones del contenido (decorado de misiones, gráficos, textos en diferentes idiomas, doblaje, etc.). Estas pruebas se realizarán por un equipo externo al equipo de desarrollo, ya sean externos a la empresa o pertenecientes al

proyecto, intentando conseguir que el videojuego vea la luz con la menor cantidad posible de defectos.

Gold Master: Es exactamente el juego que se publicará y se enviará a la fábrica para su producción con todo el contenido de arte (diseño de portada, caja, etc.) y manuales de usuario.

1.6 Metodología de desarrollo de Software.

La industria del videojuego se aferró muchos años a utilizar la metodología cascada (Keith, 2009) para guiar los procesos de desarrollo, esta metodología no era la más óptima pues no estaba orientada a la construcción de videojuegos, lo cual hizo necesario la búsqueda de otras alternativas. En principio hay que considerar que un videojuego es una aplicación de software, esto originó la búsqueda de procesos ya existentes para el desarrollo de software; se analizaron en torno a sus actividades, roles y artefactos para ver qué tan plausibles eran para guiar proyectos de videojuegos. En la siguiente tabla se hace un análisis de algunas de las metodologías ya existentes:

Tabla 1: Metodologías de Software.

Procesos Viables al Desarrollo de Videojuegos		
Metodología	Descripción	Valoración
Modelo en Cascada	Proceso de desarrollo de software especializado secuencial en el cual el desarrollo se basa en el modelo Cascada a través de las fases de concepción, iniciación, análisis, diseño, construcción y pruebas (Flood, 2003).	El producto final se demora más de lo esperado ya que cualquier problema que se presente en una de las etapas se tiene que regresar a una anterior para corregirlo. Requiere hacer muchos cambios a los documentos y regresarse a etapas anteriores propicia a que se vuelva un proceso muy desordenado. Sin embargo es uno de los procesos más populares en la industria de los videojuegos.

Rational Unified Process	Es un proceso de desarrollo de software iterativo, es adaptable y entallable para satisfacer las necesidades del equipo del proyecto (Kruchten, 2004).Comúnmente sigue una metodología pesada.	Proceso muy completo pero no está enfocado al desarrollo de videojuegos. Demasiada documentación permite realizar un buen producto con lo que ello conlleva (tiempo, dinero y personas involucradas). Sin embargo esta metodología genera mucha documentación innecesaria para el desarrollo de un videojuego.
Scrum Framework	Scrum es un framework de desarrollo de software iterativo-incremental utilizado en el desarrollo de software ágil. El trabajo está estructurado en ciclos conocidos como sprints. Durante cada sprint los equipos toman los requerimientos de una lista ordenada por prioridades conocidas como historias de usuario. Al terminar cada sprint, se tiene una versión potencialmente final del producto (Scrum_Alliance, 2009).	Scrum facilita la iteración, permite a los equipos entregar características pulidas para probar la calidad del juego a lo largo de su desarrollo y así incorporar la retroalimentación de jugadores. Scrum no es solo para programadores, involucra a muchas personas a un solo proyecto. Es útil debido a que los videojuegos hoy en día se vuelven más complejos e involucran a personas multidisciplinarias. Por dichas razones, se considera Scrum ideal para el desarrollo de videojuegos.

Después de un estudio de las diferentes metodologías, se desprende que no existen procesos específicos para el desarrollo de videojuegos de manera pública. En tal sentido se publica poca información, modelos, plantillas y herramientas, por lo tanto puede considerarse que no existe una metodología estándar para el desarrollo de videojuegos.

Huddle es una de las pocas metodologías de desarrollo que está al alcance de todos, ya que cada empresa de videojuego define su propio estilo de trabajo. La misma tiene las siguientes características: es ágil, óptimo para equipos multidisciplinarios de 5 a 10 personas, iterativo, incremental y evolutivo. Huddle, sin embargo, puede utilizarse en equipos de menos de cinco elementos. Permite guiar proyectos de videojuegos, la cual está fundamentada siguiendo la analogía de SCRUM y su base es proveniente de las fases de desarrollo propuestas por Rollings y Morris en el 2003, anteriormente tratadas en el epígrafe 1.5.

Todo el proceso se divide en 3 fases:

- Preproducción.
- Producción.
- Postmortem.

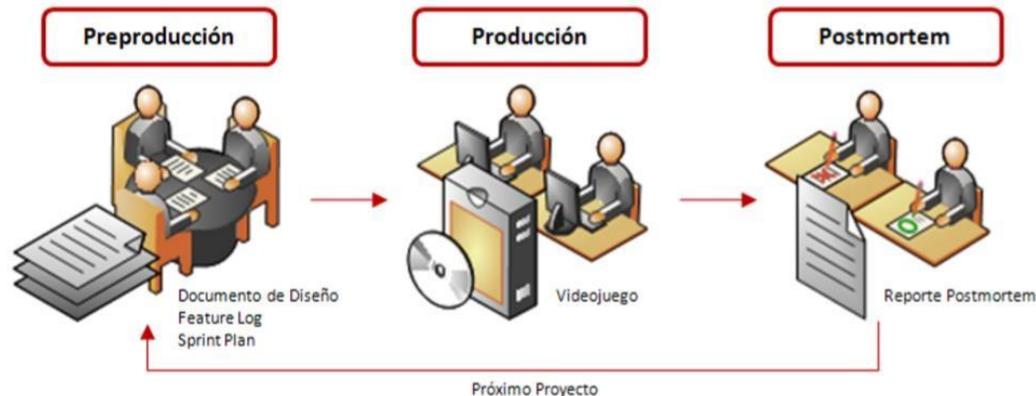


Ilustración 2: Las 3 fases del proceso de Huddle.

1.7 Herramientas utilizadas

A continuación se dará una breve explicación de cada uno de los programas utilizados durante la realización del proyecto; ya haya sido de forma exhaustiva o simplemente para momentos puntuales.

1.7.1 Gráficos 2D

▪ Inkscape

Inkscape es un editor de gráficos vectoriales de código abierto, con capacidades similares a Illustrator, Freehand, CorelDraw o Xara X, usando el estándar de la W3C: el formato de archivo Scalable Vector Graphics (SVG). Las características soportadas incluyen: formas, trazos, texto, marcadores, clones, mezclas de canales alfa, transformaciones, gradientes, patrones y agrupamientos. Inkscape también soporta meta-datos Creative Commons, edición de nodos, capas, operaciones complejas con trazos, vectorización de archivos gráficos, texto en trazos, alineación de textos, edición de XML directo y mucho más. Puede importar formatos como Postscript, EPS, JPEG, PNG, y TIFF y exporta PNG así como muchos formatos basados en vectores. Esta herramienta será utilizada para crear los Game User Interface (GUI) o interface de usuario del juego.

- **Adobe Photoshop CS6**

El software Adobe Photoshop CS6 proporciona aún más potencia de edición de imágenes, nuevas opciones creativas y el motor Adobe Mercury Graphics Engine para lograr un rendimiento increíblemente rápido. Retoca con las nuevas funciones basadas en el contenido y crea diseños y películas fascinantes mediante los nuevos flujos de trabajo y herramientas rediseñadas. (Adobe, 2013).

1.7.2 Gráfica 3D

- **Autodesk 3ds Max**

Autodesk 3ds Max y Autodesk 3ds Max Design proporcionan potentes herramientas integradas de modelado, animación y renderización en 3D que permiten a los artistas y los diseñadores dedicar más energía a la creatividad en lugar de a las dificultades técnicas. Aunque ambos productos comparten la tecnología principal, uno ofrece herramientas especializadas a los desarrolladores de juegos, creadores de efectos visuales, diseñadores de gráficos de movimiento y otros profesionales de la creatividad que trabajan en el diseño de medios, mientras que el otro está concebido específicamente para los arquitectos, diseñadores, ingenieros y especialistas en visualización. (Autodesk, 2013)

- **ZBrush**

ZBrush es un programa de modelado y pintura digital que ha revolucionado la industria de 3D con sus potentes funciones e intuitivos flujos de trabajo. Construido dentro de una elegante interfaz, ZBrush ofrece las herramientas más avanzadas del mundo para los artistas digitales. Con un arsenal de características que se han desarrollado enfocadas a la usabilidad, ZBrush crea una experiencia de usuario que se siente muy natural y al mismo tiempo potencia la inspiración del artista. Con la capacidad de esculpir miles de millones de polígonos, ZBrush te permite crear con un único límite. Tu imaginación. (3DEOLOGIC, 2013).

1.7.3 Edición de Sonido Adobe Audition CS6

El software Adobe Audition CS5 proporciona las herramientas profesionales que necesita para dotar del mejor sonido a sus producciones de vídeo y audio. Gestiona de forma eficaz una amplia gama de tareas de producción de audio, incluidas la grabación, mezcla y restauración de sonido.

1.7.4 Motores de videojuegos

Existe actualmente una amplia gama de motores de videojuegos, con diferentes tipos de licencias y orientados a cumplir distintos tipos de propósitos. Se puede encontrar motores comerciales y gratuitos, con metodologías 2D o 3D, inclusive que brindan soluciones de juegos a variadas plataformas (Windows, Linux, android, etc.).

En el sitio web Mod DB (Mod-DB, 2013) que presenta una extensa lista de aproximadamente 247 motores, que pueden ser tomados en cuenta a la hora de desarrollar un videojuego. Uno de los propósitos de este trabajo es seleccionar un Motor de Juego de última generación para examinar sus características principales y realizar un videojuego con dicho motor. Pero se debe tener en cuenta que, actualmente, las grandes empresas de videojuego trabajan con tecnologías de vanguardia, las cuales pueden costar miles de dólares. Afortunadamente existe una nueva tendencia por parte de algunas empresas que desarrollan motores de videojuegos, que los impulsa a colocar en el mercado versiones gratuitas (generalmente limitadas en algún aspecto) de sus herramientas para que las personas que deseen aprender, puedan hacerlo sin tener que comprar una versión full. Entre estas empresas se puede encontrar las conocidas Epic Games (Epic, 2013) y Unity Technologies (UnityTechnologies, 2013), las cuales ofrecen versiones gratuitas y descargables de sus famosos motores de videojuegos con los cuales se establecerá a continuación una comparación entre ambos motores para decidir con cual se trabajará en la aplicación.

Tabla 02: Unity vs UDK

Parámetros	Unity 3D	UDK
Plataformas	PC, X360, Wii, PS3, Linux, Mac, iPhone, iPad, Android, Web (plugin)	PC, X360, PS3, iPhone, iPad
Compañía	Unity Technologies	Epic Games
Sitio Oficial	Unity3d.com	www.udk.com
Licencia	Existen dos tipos de licencias principales: Unity y Unity-Pro. La primera es gratuita (con algunas limitaciones) y está orientada a estudiantes y personas que deseen aprender a trabajar con este motor. La versión Pro no es gratuita, pero tiene características adicionales. Tanto Unity como Unity-Pro incluyen el entorno de desarrollo, tutoriales, ejemplos de proyectos y el contenido, el apoyo a través de la comunidad de foros, wiki, y las futuras de actualizaciones de la versión.	UDK es gratuito para propósitos no comerciales (por ejemplo para fines educativos). No obstante existen licencias comerciales particulares y a nivel de compañía.

Valoración	8.9 (153 votos desde el sitio Mod DB)	9.5 (182 votos desde el sitio Mod DB)
Completo de entorno edición	Entorno integrado de desarrollo con jerarquía, edición visual, inspector de propiedades y vista previa del videojuego.	Unreal Engine 3 ofrece un entorno plenamente integrado de edición a través del famoso Unreal-Editor (todas las herramientas clave del motor son accesibles a través de él). El nuevo Unreal-Content-Browser permite gestionar activos como las mallas, materiales, sonidos y animaciones, más fácil que nunca.
Poder de renderizado	Para la versión 3 de Unity se reescribió la prestación de renderizado, logrando más flexibilidad y potencia. Esto significa que la prestación es en general un 50% más rápido que antes.	UDK le trae Gemini, el sistema de renderizado multihilo. Gemini crea una realidad visual exuberante y proporciona la potencia necesaria para realizar simulaciones realistas. Se pueden obtener resultados impresionantes con este procesador flexible y altamente optimizado. UDK posee características de color de 64 bits HDR rendering pipeline, gamma-corrector, procesador de espacio de color lineal que proporciona una imaculada precisión de color. Además soporta una amplia gama de efectos post-procesamiento, tales como el desenfoque de movimiento, la profundidad de campo, bloom, oclusión de ambiente y materiales definidos por el artista.

Animación avanzada	El sistema de animación de Unity permite tratar a los objetos tanto genéricos como humanoides, en este último caso permite el uso de cada parte del cuerpo por separado para la animación de mismo bajo la configuración IK.	El sistema de animación de Unreal le da control total sobre cada detalle, cada músculo y cada hueso.
Scripting de gran alcance	Unity soporta tres lenguajes de Scripting: JavaScript, C#, y una versión modificada de Python llamada Boo. Los tres son igual de rápidos e interoperables. Además pueden utilizar librerías .NET con soporte a base de datos, expresiones regulares, XML, acceso a ficheros y funciones de red. El Scripting frecuentemente es considerado limitado y lento. Pero en Unity los scripts son compilados a código nativo y se ejecuta casi tan rápido como C++.	UDK presenta a Unreal-Script, un sistema totalmente integrado de alto nivel, que implementa un lenguaje de programación orientado a objetos, y Unreal-Kismet, una solución avanzada de scripting visual. Las herramientas permiten a cualquier persona crear un juego de gran alcance con un control prácticamente ilimitado.
Físicas del mundo real	Llena de vida a los objetos del juego con las funcionalidades provistas por el motor de NVIDIA's PhysX.	UDK le da peso a todas sus creaciones a través del NVIDIA's PhysX System. Puede visualizarse una física intensa utilizando Unreal- PhAT y entornos destructibles que sumergirán a los jugadores.
Iluminación y Sombras	Unity ofrece un sistema de iluminación altamente optimizado con lightmaps y sombras de tiempo real.	Las funciones de iluminación y sombra más avanzadas, incluyendo la iluminación global de Unreal- Lightmass. Las características son

		fácilmente añadidas a las creaciones a través de Unreal-Editor.
Cinematografía	Unity permite utilizar las cámaras, las partículas y las cinemáticas en cada escena ejecutándolas en tiempo real, además de tener varios efectos de imágenes que brinda un realismo sensacional a la visualización del juego.	Unreal-Matinee inyecta escenas estilo de película en su juego, lo que permite colocar cada cámara, objeto y explosión dentro del mundo e incluso reproducirlo en tiempo real.
Terreno	Puede crear increíbles paisajes con densa vegetación que funcionarán sin problemas en hardware de gama baja, además de ocupar muy poco espacio en disco.	El sistema de terrenos de UDK le permite personalizar el lugar y la vegetación, las estructuras y los innumerables puntos de interés en el mundo de su juego. El terreno en UDK es fácil de editar y modificar para crear niveles y mundos.
Red	Posee funcionalidades multijugador como chat y puntuaciones en línea. Unity brinda solución a todas las necesidades de red para sus videojuegos.	UDK proporciona una arquitectura de red flexible y de alto nivel adecuada para variados géneros de juegos, así como también para proyectos de simulación.
Materiales y Shaders en tiempo real	El Unity-Shader-System combina facilidad de uso, flexibilidad y rendimiento. Todos los Shaders incorporados se integran perfectamente con cualquier tipo de luz, con o sin cookies. Además puede escribir sus propios Shaders con el potente lenguaje de Unity: ShaderLab con Cg y GLSL.	UDK viene equipado con un sistema de materiales de gran alcance, con una interfaz visual que le permitirá crear Shaders en tiempo real y de arbitrariedad compleja sobre la marcha.
Audio	Mezcla en tiempo real gráficos en 3D con streaming de audio y vídeo.	UDK está diseñado para ofrecer a los desarrolladores la libertad creativa y un control total sobre el diseño de sonido.

Efectos de Partículas	Posee un sistema de partículas avanzadas donde se pueden crear cualquier tipo de efecto dependiendo de la imaginación del programador.	UDK permite crear efectos dinámicos de partículas, como lluvia o nieve, que dan vida a sus juegos.
Editor de vegetación	El TreeCreator permite crear vegetación que interactuará con el medio en tiempo real, soportando efectos de viento y colisiones.	SpeedTree le permite generar vegetación en tiempo real, animación de árboles con el viento, exuberantes selvas, bosques espesos y mucho más.
Requerimientos mínimos de Hardware	Windows: XP SP2; Mac OS X: Intel CPU & "Leopard" 10.5. Tarjeta Gráfica con 64 MB. El uso del Occlusion Culling requiere GPU. El resto solo depende de la complejidad del proyecto.	Windows Vista Procesador de 2.0+ GHz 2 GB RAM DX11 NVidia: serie 400 ATI: 5000 series 3 GB de espacio libre en disco

Como se puede observar, los motores propuestos cumplen ampliamente las expectativas. Pero para llevar a cabo este proyecto sólo debe seleccionarse uno y existen varias características que aclaran el proceso de selección.

Tanto Unity como UDK son multiplataforma, pero Unity tiene una ventaja que podría ser explotada en el marco de esta investigación o futuros proyectos, ya que permite publicar los juegos en un navegador web, a través de un *plugin* especial creado para tal propósito. Esto significa que se puede desarrollar un videojuego y posteriormente colocarlo en un sitio web para que los usuarios lo prueben de forma online. Esto es similar a lo que se puede hacer con la tecnología Flash y su reproductor Flash Player 22, pero con la solidez interna de un motor de videojuegos de última generación. El *plugin* es totalmente gratuito y puede descargarse desde el sitio web de Unity. Además de que el lenguaje de programación que utiliza es conocido y de fácil implementación como es el caso del JavaScript, C# y Boo los cuales permiten una mejor estructura de clases a la hora de su implementación. Otra característica que hace a Unity especial para llevar a cabo este proyecto, es que tiene menos requisitos de hardware que UDK. Tras esta comparación queda propuesto el Unity 3D como motor de juego a usar para la elaboración del videojuego.

CAPÍTULO 2: Solución Técnica

El presente capítulo contiene todas las especificaciones necesarias para comenzar el proyecto, éstas van desde el tema principal del videojuego hasta el número de niveles que tendrá. En el mismo se asienta toda la información detallada para la composición del guión técnico siguiendo las características de la metodología Huddle. Esta información queda registrada en un artefacto ingenieril llamado documento de diseño (tabla 03), artefacto esencial de la metodología Huddle. Se podrá encontrar un epígrafe con la relación de los elementos psicológicos y la manera de interactuar con los contenidos de la asignatura de Anatomía (esqueleto de la cabeza humana). Todos estos aspectos forman parte del proceso de Pre-Producción.

Tabla 03: Documento de diseño

Concepto	
Título	Aventuras Anatómicas
Estudio/Diseñadores	Roberto Elías Pérez Ozete y Ariel Padrón Gómez
Género	Juego Serio - Aventura de Plataforma en Tercera Persona
Plataforma	Computadora Personal
Versión	1
Sinopsis de Jugabilidad y Contenido	<p>Aventura Anatómica comienza en la tranquila ciudad de Baracoa en el año 1845 cuando el ex arqueólogo Miguel de Torreira parte hacia Belice en una investigación arqueológica en las ruinas Mayas, pues su principal objetivo es entregarle al museo de La Habana un esqueleto reconstruido de un sacerdote maya.</p> <p>Resulta que cuando entra en las ruinas encuentra una tumba pero sus huesos no estaban. Se encontraban esparcidos por todo el interior de esta, lo que lleva a que el desarrollo del juego es tras la búsqueda de estos huesos para reconstruir el esqueleto, pero para esto tendrá que atravesar por peligrosas trampas que le harán su camino un poco más angosto.</p>

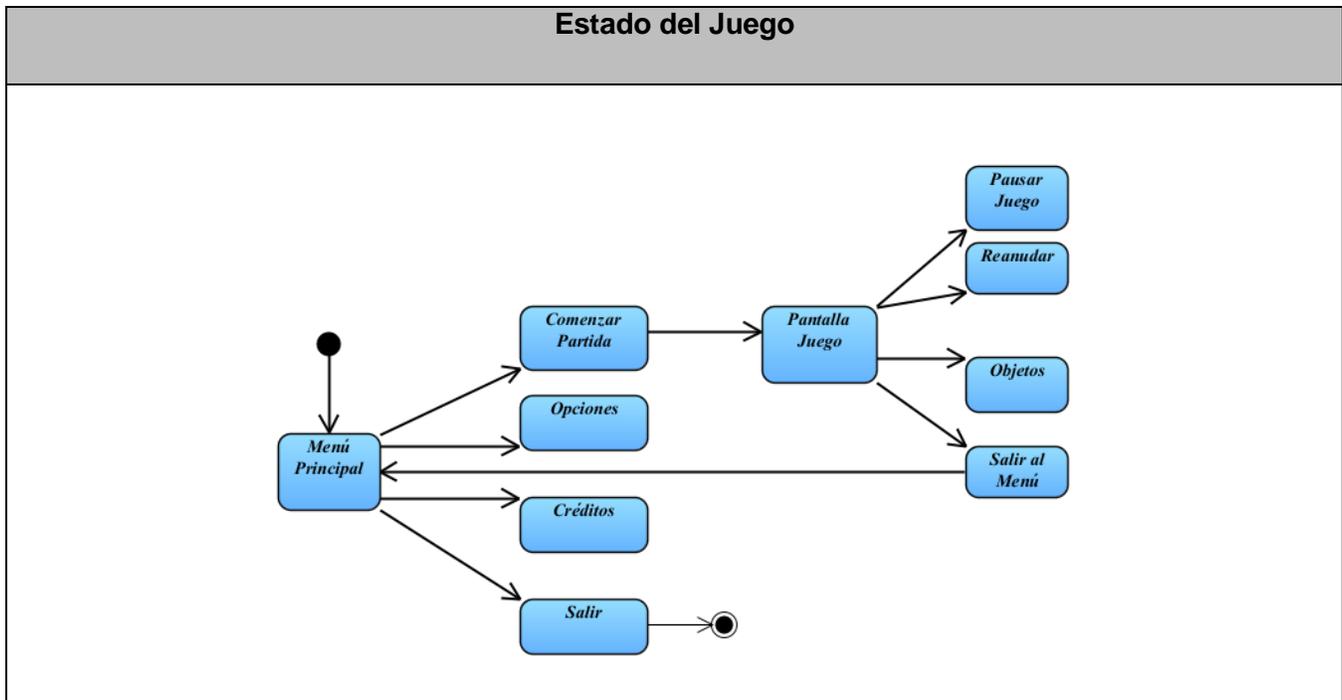
Categoría	-
Mecánica	El jugador puede caminar libremente por el ambiente o correr si así lo desea, también puede coger objetos específicos de la escena y colocarlos en determinados lugares para completar puzles o desactivar trampas.
Tecnología	<ul style="list-style-type: none"> • Hardware (Requerimientos mínimos) <ul style="list-style-type: none"> - Tarjeta Gráfica con 64 MB. - El uso del Occlusion Culling requiere GPU • Software: <ul style="list-style-type: none"> - Gráficos 2D: - Inkscape - Photoshop CS6 - Gráficos 3D: - Autodesk 3DMax - Autodesk Maya - Zbrush - Edición de Sonido: - Adobe Audition CS6 - Motor de Videojuego: - Unity 3D • Lenguaje de Programación: <ul style="list-style-type: none"> - JavaScript
Público	Está dirigido fundamentalmente a estudiantes universitarios que cursan la carrera de medicina aunque puede ser jugado por cualquier persona mayor de 16 años.
Historial de Versiones	
Versión 1 (13-1-2012)	

Visión General del Juego

"Aventuras Anatómicas" surge como una herramienta de apoyo a las formas tradicionales y medios de enseñanza de la Anatomía. Basado en recursos psicológicos y técnicas de diseño de videojuegos logrando estructurar y organizar el contenido haciendo uso de animaciones y recursos tridimensionales, recreando todo un ambiente inmersivo para la enseñanza de esta materia. Con él se aporta un recurso educativo-motivante para los estudiantes de medicina. Con el mismo se logra identificar una estrategia para el estudio de la anatomía, a partir de la explotación de recursos psicológicos para reforzar las habilidades afines con la adquisición del conocimiento de esta materia.

Mecánica del Juego

Mecánica del Juego	
Cámaras	Cámara 3D en 3era Persona
Periféricos	Teclado y Mouse
Controles	Para moverse se usará los Axis de Control(Up, Down, Left y Right) Para cambiar en estado de caminar a correr la tecla Shift Para llevar a cabo las acciones con la tecla Enter Para ver el menú pause la tecla Esc
Puntajes	La forma de puntuación esta dado en el transcurso del videojuego en el que el personaje recoge los huesos del cráneo, con cada hueso va actualizando su diario.
Guardad/Cargar	-



Interfaces

Menú Principal

Nombre de la Pantalla	Menú Principal
Descripción de la Pantalla	Es la interfaz inicial del juego, en ella el jugador tiene acceso a : Comenzar Partida: para iniciar el juego. Opciones: Para configurar las opciones de gráfico. Créditos: Para ver los créditos del videojuego Salir: Para salir del juego
Imagen	 <p style="text-align: center;">Ilustración 3: Menú principal</p>

Menu Pause	
Nombre de la Pantalla	Menu Pause
Descripción de la Pantalla	<p>Es la interfaz de cuando el juego está en pause(Tecla Esc), en ella el jugador tiene acceso a :</p> <p>Reanudar: Sale del menú pause y regresa al juego.</p> <p>Objetos: Para ver los objetos desbloqueados.</p> <p>Opciones: Para configurar las opciones de gráfico.</p> <p>Salir al Menú: Para salir al Menú Principal del juego.</p>
Imagen	 <p>Ilustración 4: Menú pause.</p>
Niveles	
Baracoa	
Título del Nivel	Baracoa
Encuentro	Este es el primer escenario en el que comenzara el juego.
Descripción	Un pequeño pueblo construido cercano a la costa por lo que tiene buen flujo marítimo. Las acciones se llevaran a cabo en este nivel en horas de la noche.

Objetivos	Buscar la forma de desembarcar hacia Belice.
Progresos	Cuando el jugador logra buscar el permiso para el desembarco y es entregado al capitán se proyecta una cinemática y pasa al escenario en las junglas de Belice.
Ítems	No se utiliza ningún objeto en este nivel.
Personajes	Capitán del Barco y Gobernador del Pueblo
Música y Efectos de Sonido	La música utilizada en este nivel recrea un ambiente tranquilo ya que las acciones son de noche. También se utilizó un sonido ambiente de costas y otro de montes.
Referencias de BGM y SFX	Rakujitsu: Música de fondo. The Sea – Calm: Sonido ambiente de costas.
Jungla	
Título del Nivel	Jungla
Encuentro	El jugador llega a esta escena después de que habla con el capitán en el puerto de Baracoa y este lo lleva a Belice. Ya en esta se adentra en la jungla en la búsqueda de las ruinas mayas.
Descripción	Una típica jungla llena de diversa vegetación. En su interior se encuentran las ruinas mayas, una reliquia de siglos de existencia.
Objetivos	El objetivo es encontrar las ruinas mayas y entrar en los dos niveles que incluye.
Progresos	El jugador después que encuentra la primera entrada a las ruinas y recoge todos los objetos de esta regresa a esta escena para entrar en la otra parte de las ruinas y completar el juego. Después de que entra en la segunda ruina ya deja atrás este escenario.
Ítems	No se utiliza ningún objeto en este nivel.

Personajes	No existe ningún intercambio con otros personajes en este nivel.
Música y Efectos de Sonido	La música utilizada en este nivel recrea un ambiente de jungla con el sonido de las aves, los grillos y otros animales. También se usa una música de fondo para enfatizar el ambiente.
Referencias de BGM y SFX	The Forest: Sonido ambiental. Nunmu: Música de fondo.
Primera entrada a las ruinas	
Título del Nivel	La Cara del Fantasma
Encuentro	Este nivel es la primera parte del interior de las ruinas mayas. Se llega a esta escena después de encontrar la puerta que accede a este en las ruinas que están en la jungla.
Descripción	Es un salón inmenso con muchos corredores, inicialmente el jugador encuentra la tumba donde se suponía que estuvieran los huesos pero estos están dispersos por todo el nivel donde se encuentran peligrosas trampas y puzles que dificultaran el camino del jugador y lo mantendrán entretenido.
Objetivos	Recoger los 5 huesos que están en el interior de este nivel.
Progresos	Después de tener los 5 huesos que están en este nivel el jugador debe regresar a la jungla para buscar la segunda entrada y buscar los huesos que faltan para completar la cabeza.
Ítems	Objetos recolectables: <ul style="list-style-type: none"> • Hueso Frontal • Hueso Esfenoide • Mandíbula Superior • Mandíbula Inferior • Hueso Vumer

	<p>Objetos secundarios:</p> <ul style="list-style-type: none"> • Gema del Jaguar • Ojo de Lalok
Personajes	No existe ningún intercambio con otros personajes en este nivel.
Música y Efectos de Sonido	<p>En todo en interior del nivel se escucha el sonido del fuego quemando en aceite de las lámparas, en lugares específicos el agua que cae de una fuente y sonido de voces fantasmales. Cada puerta posee su sonido propio al abrirse al igual que las trampas que están presente en este nivel.</p> <p>La música de fondo cambia según la situación en la que se encuentre el personaje en ese momento para resaltar el sentimiento de tranquilidad, velocidad o miedo.</p>
Referencias de BGM y SFX	<p>Música de Fondo:</p> <ul style="list-style-type: none"> • The Dungeon • Adrenaline

Progreso del Juego

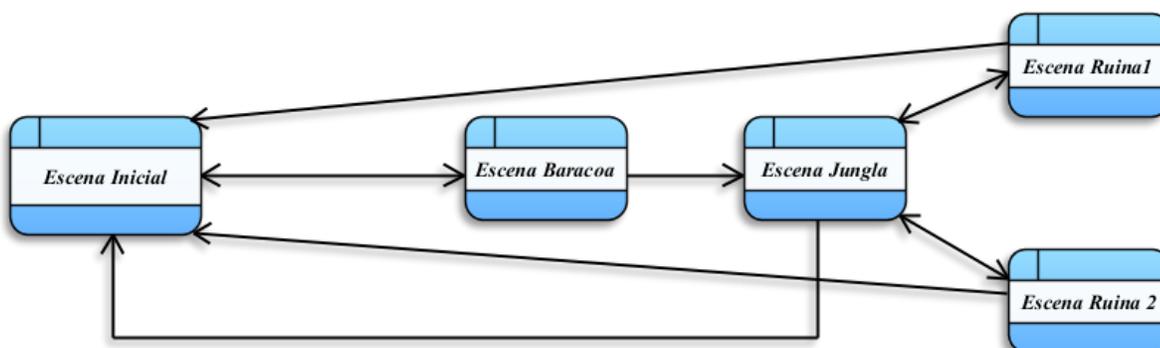


Ilustración 5: Diagrama de estado del videojuego.

Personajes	
Miguel de Torreira	
Nombre del Personaje	Miguel de Torreira
Descripción	El personaje es un humano de 60 años de edad y estudió arqueología en Europa en su juventud. Posee una camisa naranja y un pantalón corto de color naranja oscuro. Unas pantuflas de la época bastante largas y un sombrero inglés de expediciones. Posee una gran barba con corte inglés y unos anteojos pequeños sin patas.
Imagen	 <p>Ilustración 6: Personaje principal.</p>
Concepto	Miguel de Torreira es el personaje principal del juego Aventura Anatómica, es un anciano aventurero que tuvo una fuerte vida de expediciones en su juventud. Se asentó en Baracoa a pasar su vejez tranquilamente hasta que se despierta su sentido de aventurero cuando se entera del descubrimiento de unas ruinas mayas en Belice y decide ir a investigarlas en persona.
Encuentro	Miguel entra en escena desde el principio pues toda la historia gira en entorno a su vida y sus acciones.
Habilidades	Posee las mismas habilidades de un ser humano. Puede caminar, correr, coger y poner objetos.

Capitán del Barco	
Nombre del Personaje	Capitán Bartolomé Eltimba
Descripción	Un capitán de barco que posee una casaca azul al estilo antiguo y un enorme cinturón cruzándole el pecho. Posee una barba negra pequeña y un sombrero de capitán color piel.
Imagen	 <p>Ilustración 7: Capitán del barco.</p>
Concepto	El capitán Eltimba ama el mar y a su barco por eso siempre está cercano a este en el puerto para evitar que los aldeanos le hagan algún daño. Mantiene buenas relaciones con Miguel pero es estricto con las leyes de la ciudad.
Encuentro	El capitán entra en escena cuando Miguel le pide que lo lleve a Belice.
Personaje No-Jugable	Es un personaje secundario que su propósito es llevar a Miguel al puerto de Belice para que continúe la expedición.
Gobernador de la ciudad de Baracoa	
Nombre del Personaje	José Bermúdez, gobernador de la ciudad de Baracoa.

Descripción	Personaje perteneciente a la alta clase de la sociedad. Tiene un traje español de gala. Su corte de pelo es corto y una pequeña barba en forma de candado bien afeitada.
Imagen	 <p>Ilustración 8: Gobernador.</p>
Concepto	Bermúdez es un gobernador bastante conocido y querido por toda la ciudad. Siempre se encuentra ocupado en su casa procurando que todo en el pueblo marche en orden.
Encuentro	Bermúdez entra en escena cuando a Miguel necesita un permiso del gobernador para poder abordar el barco del capitán Eltimba, pues este se lo pide para legalizar su viaje.
Personaje No-Jugable	El gobernador es un personaje secundario de la primera escena cuyo propósito es darle un permiso a Miguel para poder viajar a Belice.

Música y Sonido

Música de fondo:

MF001: Rakujitsu (Música que enfatiza la noche y sus sonidos)

MF002: Nunmu (Música relajante para el recorrido por la jungla)

MF003: The Dungeon (Usada en el interior de las ruinas que enfatiza el misterio)

MF004: Adrenaline (Brinda sensación de velocidad y de peligro)

Sonidos ambientales:

SE001: The Sea – Calm(Sonido ambiente de costas)

SE002: Hot Summer Sounds (Sonido ambiente de montes)

SE003: The Forest (Sonido ambiente de jungla activa)

SE004: FireBurn (sonido de fuego)

SE005: Puerta Principal (sonido de puerta abriéndose)

SE006: DoorUnlock (sonido de puerta que se abre por desbloqueo)

SE007: RotateUnlocker(sonido de engranes rotando y abriendo puerta)

SE008: WaterFall(Agua callendo)

Imágenes de Concepto

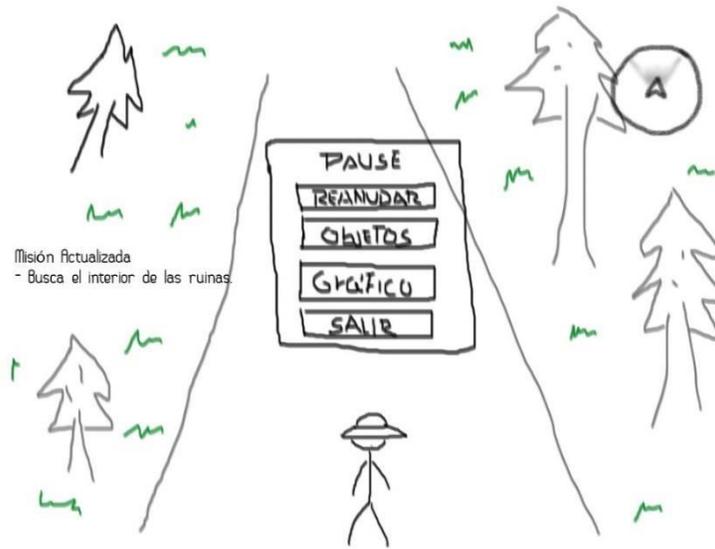


Ilustración 9: interfaz principal



Ilustración 10: Ruinas Mayas

Miembros de Equipo

Diseñador y desarrollador: Roberto E. Pérez Ozete

Diseñador y desarrollador: Ariel Padrón Gómez

Detalles de Producción

Fecha de inicio 15/11/2012

Fecha de Terminación 1/05/2013

2.1 Relación entre el diseño del videojuego y los elementos psicológicos.

La problemática fundamental en el desarrollo de videojuegos serios siempre es lograr el equilibrio entre el contenido y la diversión. Para dar respuesta a esta problemática se elaboró una estrategia de aprendizaje en donde se evidencia la influencia de los recursos psicológicos en los factores cognoscitivos y su relación con los elementos que conforman la jugabilidad.

Dado el caso del cúmulo de información y siguiendo el plan de estudio de la Anatomía Humana en Cuba (Piug, 2001), para facilitar el estudio del esqueleto de la cabeza humana, esta se puede dividir en dos partes, el viscerocráneo y el neurocráneo.

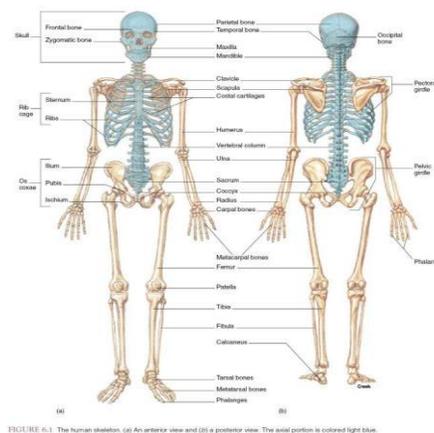


Ilustración 11: Viscerocráneo y neurocráneo.

De esta misma forma se estructura el videojuego, haciendo corresponder cada nivel a cada parte del cráneo. Existiendo una armonía entre el contenido del plan de estudio y el modo de representarlo en el videojuego. Con esta característica el estudiante se familiariza con el contenido y puede ir creando una base organizada del conocimiento que le permite identificar cada objeto de la escena. Para esto se apela al uso de la memoria subconsciente, a través de la información visual que se le ofrece mediante los sistemas de trampas presentes en cada nivel del juego, formándose de esta manera un **modelo mental** en los discentes.

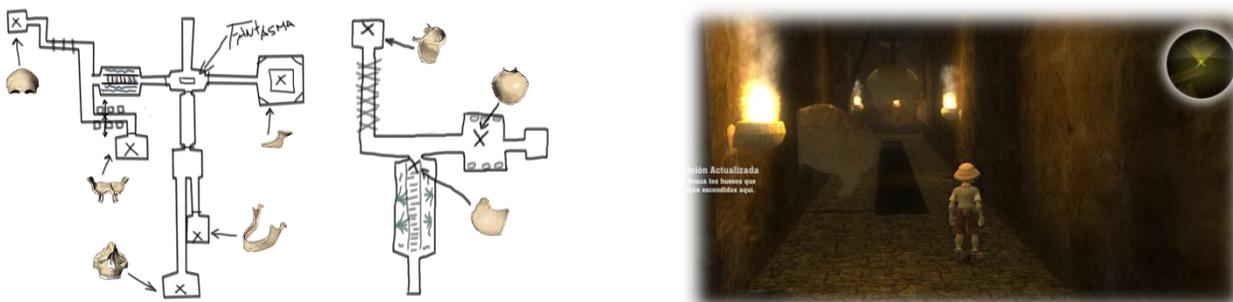


Ilustración 12: Organización de la información y diseño de trampas.

En la medida que el jugador se adentra en cada nivel, se visualiza cada parte del cuerpo según la etapa del juego en que se encuentre, logrando identificar y obteniendo información adicional del hueso en cuestión, haciendo uso de la memoria consciente, que junto al sonido y la información visual presente en la escena, refuerzan fundamentalmente la atención del jugador, logrando así, mantener todos los sentidos del mismo en función de una elevada inducción del conocimiento (**modelo conceptual**).

Una vez transitados todos los sistemas de trampas y recogidos los huesos correspondientes a su estructura ósea, el jugador puede armar dicha estructura, consolidando todo lo identificado con anterioridad e interrelacionando cada fase, obteniendo como resultado mayores niveles cinéticos lo cual conlleva a un mayor pensamiento motórico (**modelo conceptual**).

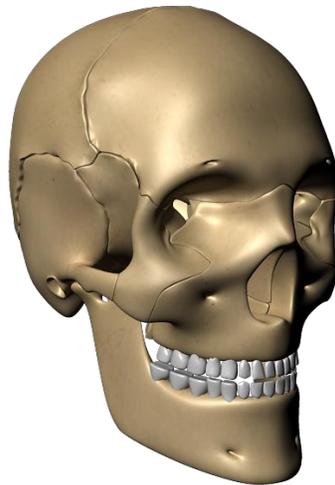


Ilustración 13: Esqueleto de la cabeza.

CAPÍTULO 3: Ingeniería del Sistema

En el proceso de Producción, siguiendo la metodología Huddle, cada rol ejecuta sus tareas atendiendo a su especialidad. Los elementos recogidos en el siguiente capítulo son los relacionados con los ingenieros informáticos.

A continuación se describen los procesos ingenieriles a realizar para la construcción del videojuego, los cuales permiten identificar las necesidades del producto mediante la obtención de los requisitos funcionales y no funcionales, ayudando al desarrollador a comprender la naturaleza del producto a construir, así como detectar las funcionalidades requeridas. El desarrollo de este capítulo comenzará con la extracción de los requisitos del software, funcionales y no funcionales que se han identificado, posteriormente se continuará con el análisis detallado de los casos de uso y la creación de los diagramas de clases que guiarán la fase de implementación.

3.1 Requisitos Software

A continuación se describen los requisitos software del videojuego, estos requisitos se han concretado en las diferentes reuniones establecidas según la metodología. Con el fin de facilitar la tarea de comprensión por parte del lector, en primer lugar se explica la estructura de cada uno de los requisitos así como los campos por los que está compuesta su definición, y posteriormente mostrar la manera en la que están agrupados.

Cada requisito estará representado por una tabla de la siguiente forma:

Identificación							
Título							
Descripción							
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	Esencial		Deseable		Opcional		
Fuente							

Tabla 04: Tabla ejemplo de requisitos

Donde cada campo significa lo siguiente:

- **Identificación:** Código que identifica de forma unívoca cada uno de los requisitos.
- **Título:** Título descriptivo del requisito.
- **Descripción:** Explicación clara y concisa del requisito que se está especificando.
- **Prioridad:** Orden de cumplimiento de un requisito. A mayor prioridad, mayor urgencia para realizarlo.
- **Estabilidad:** Probabilidad de cambio de un requisito. A mayor estabilidad, menor posibilidades de

que dicho requisito se vea modificado.

- **Claridad:** Cataloga la no ambigüedad de un requisito. Cuanta mayor claridad tenga, menor ambigüedad contendrá.
- **Verificabilidad:** Facilidad para comprobar que el sistema cumple un requisito.
- **Necesidad:** Esencialidad. Interés del cliente en la realización de un requisito.
- **Fuente:** Origen del requisito ya sea una persona o documentación.

Por otro lado, con ánimo de ofrecer una estructura que facilite la realización y comprensión de los requisitos, se van a organizar en dos categorías:

- **Requisitos de software funcionales:** especifican qué tiene que hacer el software. Definen el propósito del software.
- **Requisitos de software no funcionales:** especifican como el software debe hacer las cosas y que debe cumplir.

3.1.1 Requisitos funcionales

Las siguientes tablas se corresponden con los requisitos funcionales que se han extraído del videojuego a desarrollar:

Identificación: RSF-VG-001							
Título	Activar cambio de escena						
Descripción	El sistema permitirá iniciar una nueva escena siempre que el usuario lo desee.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 05: RSF-VG-001 – Activar cambio de escena.

Identificación: RSF-VG-002							
Título	Activar efectos de audio						
Descripción	La aplicación ejecutará todos los efectos de audio que se encuentran en la escena interactuando en ese momento.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 06: RSF-VG-002 – Activar efectos de audio.

Identificación: RSF-VG-003							
Título	Activar créditos						
Descripción	La aplicación mostrará los créditos en la pantalla de inicio de forma ascendente.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		<u>Deseable</u>		Opcional		
Fuente	Autor del proyecto						

Tabla 07: RSF-VG-003 – Activar créditos.

Identificación: RSF-VG-004							
Título	Activar música de fondo						
Descripción	La aplicación ofrecerá un instrumental de fondo que acompañará el juego todo el tiempo en el que el jugador permanezca en el escenario.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 08: RSF-VG-004 – Activar música de fondo.

Identificación: RSF-VG-005							
Título	Activar el GUI del menú						
Descripción	La aplicación ofrecerá una gama de opciones que se mostrarán en la pantalla al iniciar la aplicación.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 09: RSF-VG-005 – Activar el GUI del menú.

Identificación: RSF-VG-006							
Título	Cambiar sistema gráfico						
Descripción	La aplicación dispondrá de un grupo de opciones que permitirán que el usuario modifique el gráfico del videojuego.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 10: RSF-VG-006 – Cambiar sistema gráfico.

Identificación: RSF-VG-007							
Título	Ver prestaciones de la PC						
Descripción	La aplicación dispondrá de un menú que mostrará la disponibilidad en que se encuentra la PC para correr la aplicación.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 11: RSF-VG-007 – Ver prestaciones de la PC.

Identificación: RSF-VG-008							
Título	Cerrar el proceso correctamente.						
Descripción	Cuando el usuario desee cerrar la aplicación el proceso que la está ejecutando debe finalizarse correctamente.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 12: RSF-VG-008 – Cerrar el proceso correctamente.

Identificación: RSF-VG-009							
Título	Desactivar el menú pause						
Descripción	El usuario al dar clic en la opción reanudar cerrará el menú pause e irá nuevamente a la interfaz del juego.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 13: RSF-VG-009 – Desactivar el menú pause.

Identificación: RSF-VG-010							
Título	Activar y desactivar mapa						
Descripción	La aplicación bloqueará la visibilidad del mapa al cambiar de escena o al mostrar las cinemáticas así mismo cuando estas terminen de interactuar el mapa se activará de nuevo.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 14: RSF-VG-010 – Activar y desactivar mapa.

Identificación: RSF-VG-011							
Título	Activar y desactivar misión						
Descripción	La aplicación actualizará en la interfaz la misión en la que se encuentra el personaje mientras está jugando. También se activará la visibilidad de este panel o no en dependencia de la acción que se realice.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 15: RSF-VG-011 – Activar y desactivar misión.

Identificación: RSF-VG-012							
Título	Cambiar al GUI de objetos						
Descripción	El usuario podrá activar la interfaz de objetos sin tener otras interfaces que dificulten su visión.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 16: RSF-VG-012 – Cambiar al GUI de objetos.

Identificación: RSF-VG-013							
Título	Mostrar información de los objetos.						
Descripción	El videojuego mostrará clara y legiblemente toda la información de los objetos que se encuentran en escena.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 17: RSF-VG-013 – Mostrar información de los objetos.

Identificación: RSF-VG-014							
Título	Cambiar de escena al inicio.						
Descripción	El videojuego permitirá desde cualquier escena volver al inicio.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 18: RSF-VG-014 – Cambiar de escena al inicio.

Identificación: RSF-VG-015							
Título	Activar animaciones del personaje.						
Descripción	El videojuego activará la animación del personaje en dependencia de la acción que el jugador desee hacer.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 19: RSF-VG-015 – Activar animaciones del personaje.

Identificación: RSF-VG-016							
Título	Activar movimientos en los ejes X y Z.						
Descripción	La aplicación permitirá que el personaje se pueda mover en los ejes X y Z en dependencia de los axis de control.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 20: RSF-VG-016 – Activar movimientos en los ejes X y Z.

Identificación: RSF-VG-017							
Título	Activar animaciones de puertas y trampas.						
Descripción	El videojuego activará las animaciones correspondientes a puertas y trampas.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 21: RSF-VG-017 – Activar animaciones de las puertas.

Identificación: RSF-VG-018							
Título	Detectar colisión de la mano						
Descripción	Los objetos detectarán la colisión existente en la mano del avatar para realizar una determinada acción.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 22: RSF-VG-018 – Detectar colisión de la mano.

Identificación: RSF-VG-019							
Título	Habilitar otra cámara dejando la actual deshabilitada						
Descripción	El videojuego en cinemáticas o transiciones de escena activará o desactivará la cámara principal para darle visibilidad a otra cámara.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 23: RSF-VG-019 – Habilitar la otra cámara dejando la actual deshabilitada.

Identificación: RSF-VG-020							
Título	Activar rocas para lanzar						
Descripción	El jugador presionando la tecla R del teclado activará la opción de lanzar rocas.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 24: RSF-VG-020 – Activar rocas para lanzar.

Identificación: RSF-VG-021							
Título	Activar animaciones del objeto						
Descripción	Los objetos mostrarán las animaciones que poseen.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 25: RSF-VG-021 – Activar animaciones del objeto.

3.1.2 Requisitos no funcionales del Videojuego

Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos. Las siguientes tablas se corresponden con los requisitos no funcionales del videojuego:

Identificación: RNSF-VG-001							
Título	Rendimiento						
Descripción	El ordenador tiene que estar acordes a los requerimientos mínimos para que la aplicación pueda ejecutarse correctamente.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Desarrollador						

Tabla 26: RNSF-VG-001 – Rendimiento.

Identificación: RNSF-VG-002							
Título	Compatibilidad						
Descripción	El sistema deberá ser compatible con las diferentes versiones del sistema operativo Windows y con las diferentes resoluciones de pantalla.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Desarrollador						

Tabla 27: RNSF-VG-002 – Compatibilidad.

Identificación: RNSF-VG-003							
Título	Usabilidad						
Descripción	El videojuego puede ser jugado por cualquier persona sin tener en cuenta su nivel académico, solo con tener un conocimiento previo de informática puede disfrutar del mismo.						
Prioridad	Alta	Media	Baja	Estabilidad	Si	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Desarrollador						

Tabla 28: RNSF-VG-003 – Usabilidad.

3.2 Casos de Uso

En esta sección se van a detallar los diferentes casos de uso que se han definido durante la fase de análisis del proyecto. Los casos de uso cubren las distintas funcionalidades que podrán realizar los usuarios que interactúen con el producto.

La funcionalidad principal de la aplicación desarrollada es el aprendizaje de la anatomía humana mediante un videojuego, consiguiendo así el entretenimiento y la formación del usuario. En el siguiente apartado se definirá el actor que interactuará con el videojuego.

El actor detectado en el videojuego es evidentemente un jugador.

Jugador: Se trata del rol más común a la hora de utilizar un videojuego. Cualquier persona puede desempeñar este rol ya que no se requiere de ninguna capacidad ni conocimiento concreto. El juego está dirigido a estudiantes de medicina, aunque puede ser utilizado por otro usuario con algún conocimiento previo de anatomía o ninguno.

Especificaciones de Casos de uso

En este apartado se va a detallar el formato de las tablas utilizadas para definir los diferentes casos de uso que se extraen de las dos aplicaciones desarrolladas. Para ello se hará uso de la siguiente tabla:

Identificador	
Nombre	
Descripción	
Actor	
Precondiciones	
Postcondiciones	
Escenario	

Tabla 29: Tabla ejemplo de Casos de usos.

Donde cada campo significa lo siguiente:

- **Identificador:** Código que identifica de forma unívoca cada caso de uso.
- **Nombre:** Nombre descriptivo del caso de uso.
- **Descripción:** Explicación clara y concisa del caso de uso que se está especificando.
- **Actor:** tipo de usuario de la aplicación.
- **Precondiciones:** Condiciones que se deben cumplir previamente para poder realizar una determinada operación.
- **Postcondiciones:** Estado que presenta el sistema tras la ejecución de una determinada operación.
- **Escenario:** Ejecución del caso de uso en el escenario.

Casos de uso del Jugador

El jugador, una vez iniciada la aplicación, podrá realizar un grupo de funcionalidades las cuales fueron recogidas mediante casos de uso que conforman el sistema. En la ilustración 13, 14 y 15 se puede observar estas funciones que puede realizar el actor (*jugador*) en los diferentes estados del juego:

- Menú principal.
- Menú pause.
- Escenarios del juego.

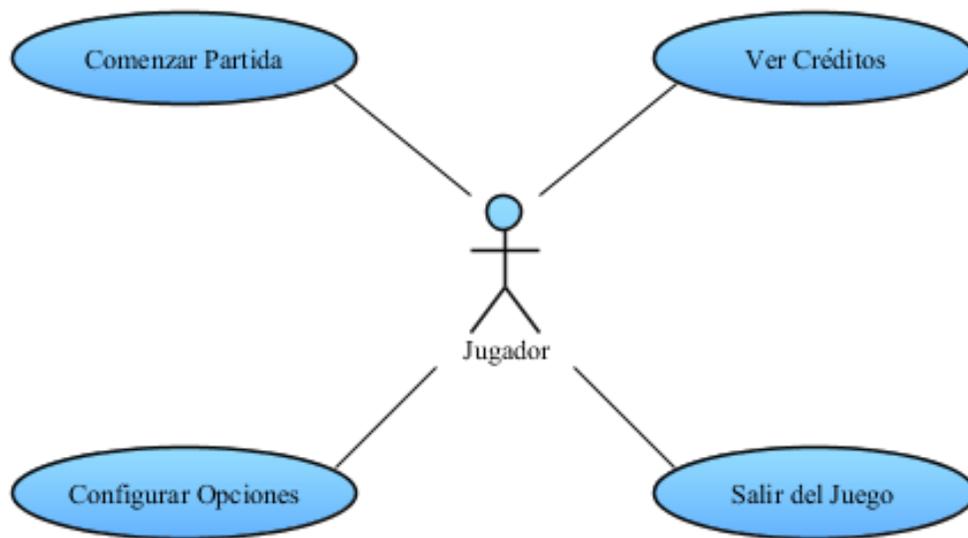


Ilustración 14: Casos de uso desde menú principal.

Identificador: CU0001	
Nombre	Comenzar Partida
Requisitos funcionales asociados	<ul style="list-style-type: none"> ► Activar cambio de escena. ► Activar efecto de audio.
Descripción	El jugador podrá iniciar una partida en cualquier momento, inclusive si está en medio del juego.
Actor	Jugador
Precondiciones	Ninguna
Postcondiciones	Comienza la partida.
Escenario	El usuario inicia la aplicación y después de pulsar el botón "comenzar partida" entra en la primera escena del juego.

Tabla 30: CU0001-Comenzar Partida.

Identificador: CU0002	
Nombre	Configurar opciones
Requisitos funcionales asociados	<ul style="list-style-type: none"> ➤ Cambiar el GUI del menú. ➤ Cambiar sistema gráfico. ➤ Ver prestaciones de la PC.
Descripción	El jugador podrá configurar las opciones graficas del juego según las características de su ordenador.
Actor	Jugador
Precondiciones	Ninguna
Postcondiciones	El sistema gráfico del juego se adecua a las opciones escogidas por el jugador.
Escenario	El jugador después de iniciada la aplicación puede acceder a este botón en el cual se encontrarán una serie de opciones gráficas configurables para el rendimiento o calidad visual del juego.

Tabla 31: CU0002-Configurar opciones.

Identificador: CU0003	
Nombre	Ver créditos
Requisitos funcionales asociados	<ul style="list-style-type: none"> ➤ Activar créditos. ➤ Activar música de fondo.
Descripción	El jugador podrá ver el equipo de desarrollo del videojuego.
Actor	Jugador
Precondiciones	Ninguna
Postcondiciones	El sistema muestra una nueva ventana en la que se muestran los nombres y funciones del equipo de trabajo.
Escenario	El jugador tras pulsar en el botón Créditos se cargará una nueva pantalla con los datos correspondientes.

Tabla 32: CU0003-Ver créditos.

Identificador: CU0004	
Nombre	Salir del juego
Requisitos funcionales asociados	<ul style="list-style-type: none"> ➤ Cerrar el proceso correctamente. ➤ Desactivar música de fondo.
Descripción	El jugador podrá abandonar el juego en cualquier momento pulsando en el botón de salir.
Actor	Jugador
Precondiciones	El juego debe de estar iniciado

Postcondiciones	Se saldrá de la aplicación
Escenario	El usuario inicia la aplicación y tras pulsar en el botón salir desde el menú principal se cerrará la aplicación.

Tabla 33: CU0004-Salir del juego.

En la *ilustración 14* se puede observar los casos de uso que puede realizar el actor *jugador* cuando se encuentra en la pantalla de menú pause del videojuego.

Las diferentes acciones que puede realizar el jugador son reanudar la partida, regresando al estado del juego, ver los objetos que posee en la mochila y salir al menú principal.

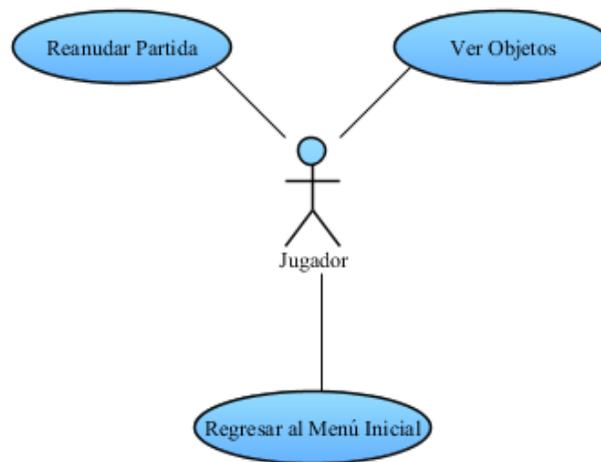


Ilustración 15: Casos de uso desde menú pause.

Identificador: CU0005	
Nombre	Reanudar partida
Requisitos funcionales asociados	<ul style="list-style-type: none"> ➤ Desactivar el menú pause. ➤ Activar y desactivar mapa. ➤ Activar y desactivar misión. ➤ Activar efecto de audio.
Descripción	El jugador podrá continuar una partida.
Actor	Jugador
Precondiciones	Se debe de haber iniciado una partida.
Postcondiciones	Se mostrará el juego en el mismo lugar donde se activó el menú pause.
Escenario	El usuario inicia la aplicación y tras pulsar en el botón Reanudar el juego sigue en el mismo lugar.

Tabla 34: CU0005-Reanudar partida.

Identificador: CU0006	
Nombre	Ver objetos
Requisitos funcionales asociados	<ul style="list-style-type: none"> ▶ Cambiar al GUI de objetos. ▶ Mostrar información de los objetos.
Descripción	El jugador podrá obtener una documentación más amplia sobre los huesos que ha recolectado hasta el momento.
Actor	Jugador
Precondiciones	Se debe de haber iniciado una partida.
Poscondiciones	El sistema mostrará la información asociada a cada hueso recogido en la escena hasta el momento, los demás se mostrarán bloqueados.
Escenario	El jugador después de iniciada la aplicación puede acceder a este botón en el cual se encontrará toda la información de los huesos que componen el aparato locomotor.

Tabla 35: CU0006-Ver objetos.

Identificador: CU0007	
Nombre	Regresar al menú inicial
Requisitos funcionales asociados	<ul style="list-style-type: none"> ▶ Cambiar de escena al inicio. ▶ Activar efecto de audio.
Descripción	El jugador podrá salir al menú principal.
Actor	Jugador
Precondiciones	Se debe de haber iniciado una partida
Poscondiciones	El sistema mostrará la pantalla del menú principal.
Escenario	El jugador después de iniciada la aplicación puede acceder a este botón con el cual puede acceder al menú inicial.

Tabla 36: CU0007- Regresar al menú inicial.

La ilustración 15 se corresponde con las acciones que puede realizar el jugador cuando se encuentra en la pantalla del juego, es decir, cuando está jugando.

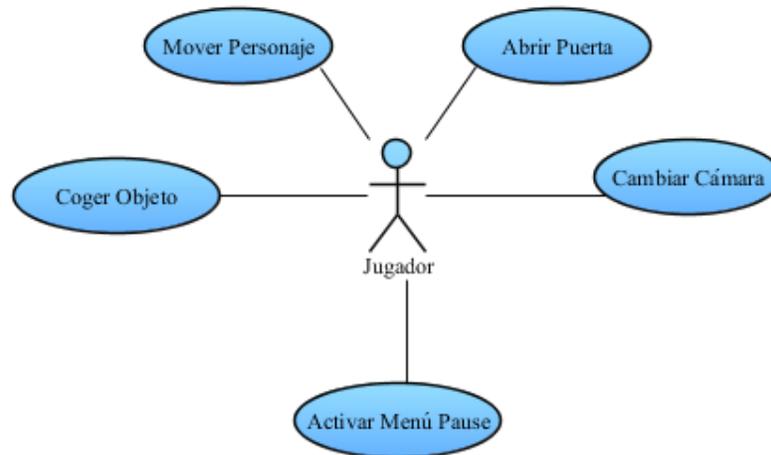


Ilustración 16: Casos de uso desde pantalla del juego.

Identificador: CU0008	
Nombre	Mover personaje
Requisitos funcionales asociados	<ul style="list-style-type: none"> ► Activar animaciones del personaje. ► Activar movimiento en los ejes X y Z.
Descripción	El usuario utilizará las teclas Axis de Control (Up, Down, Left y Right) del teclado para mover al personaje por la pantalla.
Actor	Jugador
Precondiciones	El jugador debe de encontrarse en la pantalla del juego.
Postcondiciones	El personaje se desplazará por la pantalla.
Escenario	El usuario desde el juego deberá pulsar las teclas Up, Down, Left y Right del teclado.

Tabla 37: CU0008- Mover personaje.

Identificador: CU0009	
Nombre	Abrir puerta
Requisitos funcionales asociados	<ul style="list-style-type: none"> ► Activar animaciones de puertas y trampas. ► Detectar colisión de la mano.
Descripción	El jugador podrá abrir las puertas de dos formas, una por proximidad y la otra usando llaves.
Actor	Jugador
Precondiciones	Debe de encontrarse dentro de la partida.

Poscondiciones	El sistema mostrará la animación de las puertas abriéndose.
Escenario	El usuario desde la escena deberá mover al personaje hasta acercarse a las puertas.

Tabla 38: CU0009- Abrir puerta.

Identificador: CU0010	
Nombre	Coger objetos
Requisitos funcionales asociados	<ul style="list-style-type: none"> ▶ Activar animaciones del personaje. ▶ Detectar colisión de la mano.
Descripción	El jugador podrá recoger los huesos acercándose a ellos y presionando la tecla Z del teclado.
Actor	Jugador
Precondiciones	Debe de encontrarse dentro de la partida.
Poscondiciones	El sistema almacenará los huesos en su diario anatómico.
Escenario	El usuario desde la escena deberá mover al personaje hasta acercarse a los huesos.

Tabla 39: CU0010- Coger objetos.

Identificador: CU0011	
Nombre	Activar menú pause
Requisitos funcionales asociados	<ul style="list-style-type: none"> ▶ Activar y desactivar mapa. ▶ Activar y desactivar misión.
Descripción	El usuario tiene la posibilidad de pulsar la tecla Esc. del teclado para acceder al menú pause del juego.
Actor	Jugador
Precondiciones	El jugador debe de encontrarse en la pantalla del juego y pulsar la tecla Esc. del teclado.
Poscondiciones	Se expondrá un menú con las opciones de: Reanudar, Opciones y Salir al menú.
Escenario	El usuario desde la pantalla del juego deberá pulsar la tecla Esc. del teclado.

Tabla 40: CU0011- Activar menú pause.

Identificador: CU0012	
Nombre	Cambiar cámara
Requisitos funcionales asociados	<ul style="list-style-type: none"> ▶ Habilitar la otra cámara dejando la actual deshabilitada. ▶ Activar rocas para lanzar.
Descripción	El jugador podrá cambiar la configuración de la cámara, presionando la tecla R para activar la cámara en 1ra persona y T para activar la cámara en 3ra persona.
Actor	Jugador
Precondiciones	Se debe de haber iniciado una partida
Poscondiciones	El sistema mostrará una de las configuraciones de las cámaras al presionar las teclas R o T del teclado.
Escenario	El usuario desde la escena del juego deberá pulsar las teclas R o T del teclado.

Tabla 41: CU0012- Cambiar cámara.

3.3 Diagrama de clases

El diseño de los diagramas que se han realizado pretende establecer una definición más específica de cada una de las estructuras utilizadas en este videojuego, las cuales están compuestas por *gameObject* similares, estos funcionan como contenedores donde se les adjuntan componentes y script que son los que dictan el funcionamiento de los mismos convirtiéndose en parámetros modificables de forma visual. Todos estos *gameObject* se comunican entre ellos de forma pública a través de MonoBehaviour que es el núcleo de Unity 3D.

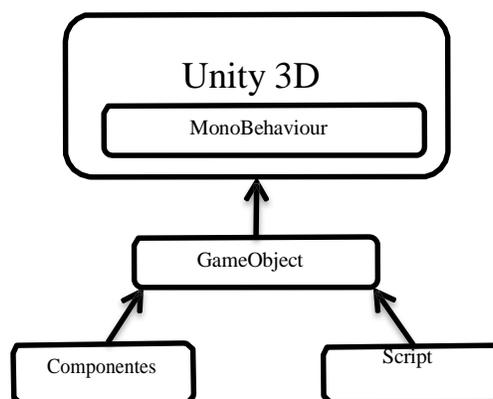


Ilustración 17: Relación entre el núcleo de Unity y los objetos que componen la escena.

De esta forma quedará compuesto el diagrama de clases del sistema, donde los gameObject se muestran siguiendo la analogía de los diagramas UML en forma de clases, conteniendo los script y componentes para formar objetos del juego que interactuarán entre sí en tiempo real dentro del videojuego. Siguiendo esta estructura podrán quedar clases aisladas una de otras pero implícitamente están integradas a MonoBehaviour.

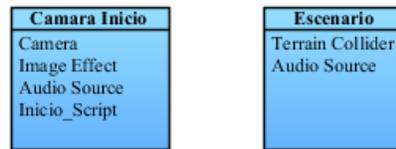


Ilustración 18: Diagrama de clases de Menú Inicial.

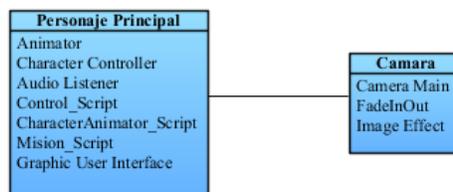


Ilustración 19: Diagrama de clases de Menú Pause.

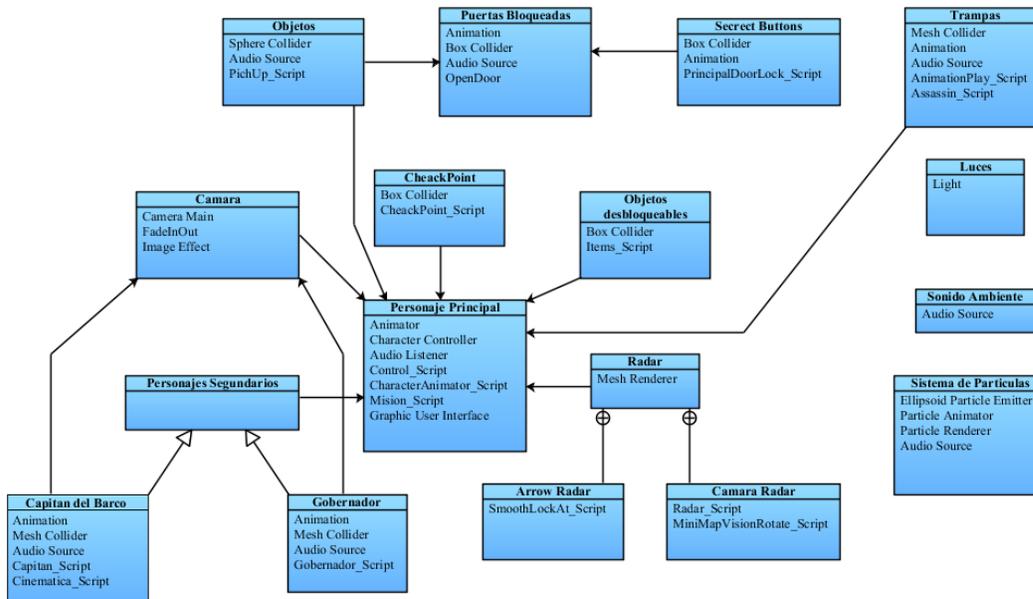


Ilustración 20: Diagrama de clases de escenarios.

3.4 Implementación

Para la implementación del videojuego se utilizó fundamentalmente javascript como lenguaje de programación primario atendiendo a que el mismo es el lenguaje nativo de Unity aunque también existen algunos script implementados en c# que vienen incluidos en el paquete inicial del motor. Estos script están clasificados en base a la función que cumplen en el videojuego conectándose entre sí y dando vida

a los objetos que interactúan en los escenarios. Los GameObject están compuestos por un grupo de componentes que brinda Unity para darle determinadas propiedades a estos objetos y los script que dictarán el comportamiento de cada uno en los diferentes casos. A continuación se expondrán los componentes y script usados en la implementación del videojuego "Aventura Anatómica".

3.4.1 Componentes

Los componentes que provee Unity 3D para la interacción en las escenas se muestran en formas diversas. Pueden ser para crear comportamiento, definiendo apariencia, e influenciando otros aspectos de la función de un objeto en el juego. Los componentes comunes de producción de juego vienen contruidos dentro del Unity, desde el Rigidbody, hasta elementos más simples, como luces, las cámaras, los emisores de partículas y otros, que permiten estructurar de forma rápida un escenario.

Componentes de visión

- **Camera**

Las cámaras son los dispositivos que capturan y muestran el mundo al jugador. Mediante la personalización y la manipulación de cámaras, se puede hacer la presentación del juego verdaderamente única. El número de cámaras a utilizar en una escena es ilimitado y la configuración puede hacerse en cualquier orden, en cualquier lugar de la pantalla, o sólo ciertas partes de la pantalla.

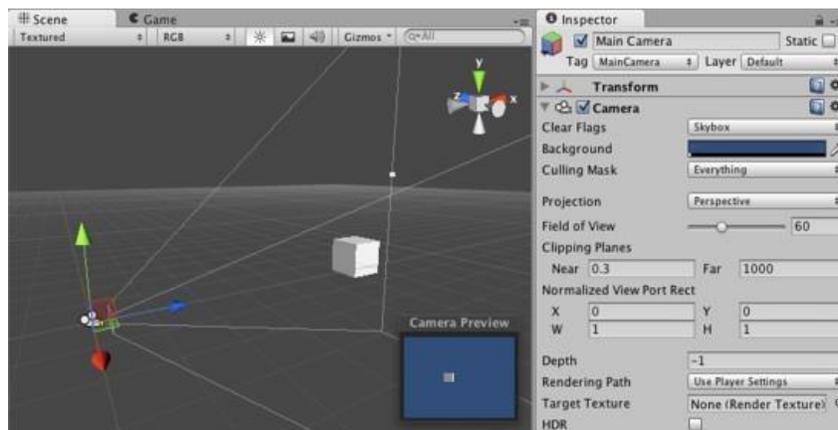


Ilustración 21: Componentes de visión.

Componentes de audio

- **Audio Source:**

La fuente de audio reproduce un clip de audio en la escena. Si el clip es un clip de audio 3D, la fuente se reproduce en una posición determinada y se atenúan con la distancia. El audio se puede transmitir a cabo entre los altavoces (estéreo a 7.1) (Spread) y se transformó entre 3D y 2D (*PanLevel*). Esto puede ser controlado con la distancia con las curvas de difuminación. Además, si el oyente está dentro de uno o múltiples zonas de reverberación, reverberaciones se aplica a la fuente. (Sólo PRO) filtros individuales se pueden aplicar a cada fuente de audio para una experiencia de audio aún más rica.

- **Audio Listener**

El *Audio Listener* actúa como un dispositivo de entrada de micrófono. Se recibe información desde cualquier fuente de audio que figura en la escena y reproduce sonidos a través de los altavoces del ordenador. Para la mayoría de las aplicaciones que tiene más sentido para conectar al oyente a la cámara principal. Si un detector de audio está dentro de los límites de una zona de reverberación (*Reverb*) se aplica a todos los sonidos audibles en la escena. Por otra parte, los efectos de audio se pueden aplicar a la escucha y se aplicarán a todos los sonidos audibles en la escena.

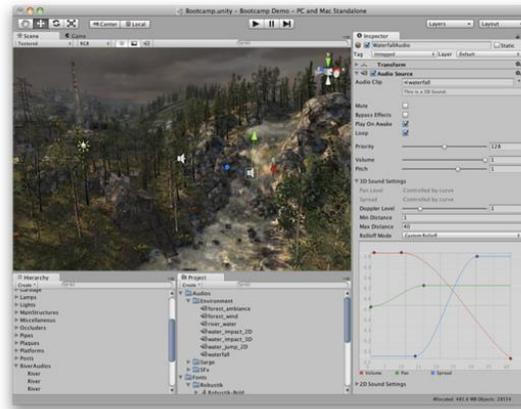


Ilustración 22: Componentes de audio.

Componentes de colisiones

- **Box Collider**

El *Box Collider* una colisión básica en forma de cubo primitivo. El *Box Collider* puede cambiar de tamaño en diferentes formas de prismas rectangulares. Funciona muy bien para las puertas, paredes

plataformas, etc. También es eficaz como un torso humano en un muñeco de trapo o un casco de coche en un vehículo. Por supuesto, funciona perfectamente para las cajas.

- **Capsule Collider**

El *Capsule Collider* se compone de dos medias esferas unidas por un cilindro. Se trata de la misma forma que la cápsula primitiva. Puede ajustar radio y la altura del *Capsule Collider* independientemente unos de otros. Se utiliza en el controlador de caracteres y funciona bien para los polos, o se puede combinar con otros Colliders para formas inusuales.

- **Sphere Collider**

El *Sphere Collider* una colisión básica en forma de esfera primitiva. El *Sphere Collider* puede ser redimensionar a escala uniforme, pero no a lo largo de cada eje. Funciona muy bien para la caída de piedras, pelotas de ping-pong y otros cuerpos en forma esférica.

- **Character Controller**

El *Character Controller* se utiliza principalmente para el control del reproductor de tercera persona o en primera persona que no haga uso de física *Rigidbody*. Se trata simplemente de una cápsula que se puede contar para moverse en una dirección de un guión. El controlador entonces llevar a cabo el movimiento, sino ser limitado por colisiones. Se deslizará a lo largo de las paredes, subir las escaleras (si es que son más bajos que el Paso Offset) y caminar en pendientes dentro del límite de la pendiente.

- **Rigidbody**

El *Rigidbody* permiten a los *GameObjects* a actuar bajo el control de la física. Este puede recibir fuerzas y a la par para que los objetos se mueven de una manera realista. Cualquier *GameObject* que contenga un *Rigidbody* va a estar influenciado por la gravedad, este actúa bajo fuerzas a través de secuencias de comandos, o interactúa con otros objetos a través del motor de física *PhysX* de *NVIDIA*.

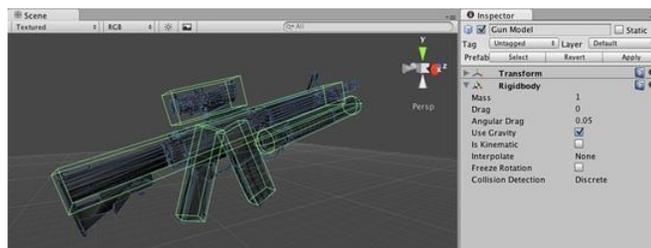


Ilustración 23: Componentes de colisión.

Componentes de animación

- **Animation**

Cualquier *GameObject* que tiene un avatar también tendrá un componente animador, que es la relación entre el carácter y su comportamiento. Este componente referencia a componentes de un controlador animador que se utiliza para configurar el comportamiento del personaje. Esto incluye la instalación de máquinas de estado, se mezclan árboles, y los eventos que deben ser controlados desde el guión.



Ilustración 24: Componentes de animación.

3.4.2 Scripts

El Scripting es una parte esencial ya que define el comportamiento del juego (o las normas) en Unity. El lenguaje de programación recomendado para Unity es JavaScript, aunque C# o Boo pueden ser igualmente usados.

Scripts de la cámara.

Nombre
CameraController
Descripción
Controla el comportamiento de la cámara la cual perseguirá en todo momento al personaje siendo el centro de su enfoque. También detecta las colisiones con los objetos de etiqueta "paredes" para evitar que estos se interpongan entre la cámara y el personaje.

Nombre
Fadeinout
Descripción
Este script permite que una textura seleccionada funcione como transición de escena haciendo uso de su componente alfa para disolver la textura.

Nombre
InicioScript
Descripción
Este script es el que le da inicio al juego, creando una interfaz para comenzar la partida, modificar las opciones gráficas, ver créditos o para salir de la aplicación.

Tabla 42: Scripts de la cámara.

Scripts del personaje.

Nombre
Control
Descripción
Este script controla el movimiento del personaje, el control de la muerte de este y las acciones que realiza para las interacciones con el medio que lo rodea.

Nombre
CharacterAnimator
Descripción
Este script controla todas las animaciones que el personaje debe realizar en cada acción que el usuario decida hacer y la mezcla de las mismas.

Nombre
GraphicUserInterface
Descripción
Este script se encarga de crear toda la interfaz de usuario referente a las escenas en las que el personaje posea acción de la aplicación así como guardar los objetos y mostrar sus características.

Nombre
Mision_Script
Descripción
Este script controla las misiones que debe realizar el jugador en todo el transcurso de juego y mostrar en la pantalla la que está realizando en ese momento.

Nombre
TirarPiedra
Descripción
Este script permite lanzar un <i>GameObject</i> previamente puesta la visión en primera persona.

Nombre
CogerItems
Descripción
Este script permite coger objetos de la escena y que el personaje conozca la existencia de este. El objeto recogido se colocará en la misma posición de la mano.

Tabla 43: Scripts del personaje.

Scripts de personajes secundarios.

Nombre
Fantasma_Script
Descripción
Este script pertenece a un personaje secundario con forma de fantasma. Permite activar la voz del fantasma y una cámara para componer una pequeña cinematográfica.

Nombre
Talk_Gobernador
Descripción
Este script permite la interacción entre el gobernador de la ciudad y el personaje principal. Activa la voz del mismo, un diálogo de respuestas opcionales y una cámara para componer una pequeña cinematográfica.

Nombre
Capitan_Change_Scene_Jungla
Descripción
Este script permite la interacción entre el capitán del barco y el personaje principal. Activar la voz del mismo, un diálogo se respuestas opcionales y una cámara para componer una pequeña cinematográfica. Después de que el personaje termina la misión encomendada por este se activa una video y se cambia de escenario.

Tabla 44: Scripts de personajes secundarios.

Scripts de objetos.

Nombre
PickUp
Descripción
Este script pertenece a los objetos que funcionan como llaves de puertas. En él se conoce que puerta es la que se va a abrir.

Nombre
RotationObject
Descripción
Este script hace que el objeto que lo posea se mantenga rotando en un lugar de la escena y que el personaje pueda obtenerlo y almacenarlo en su mochila de objetos.

Nombre
Open_Gobernador_Door
Descripción
Este script hace que el personaje cambie su posición actual y se sitúe en una prevista en este script.

Nombre
Open_Door
Descripción
Este script activa una animación del objeto al que pertenece dependiendo de la distancia en la que se encuentra el personaje.

Nombre
Move_Galeon
Descripción
Este script hace que el objeto que lo posea se mueva hacia adelante.

Tabla 45: Scripts de objetos.

Scripts de trampas.

Nombre
Asesinato
Descripción
Este script chequea que si la etiqueta del <i>GameObject</i> con el que este colisione coincide con la predeterminada por este, el otro <i>GameObject</i> active la función de morir.

Nombre
OpenPorton01
Descripción
Este script pertenece a los objetos que funcionan como puertas. En él se conoce la llave que la va a abrir y si está colocada en la posición correcta. También activa la animación de abrir la misma.

Nombre
MandibulaDetect
Descripción
Este script pertenece a una trampa que se activa según la posición del personaje con referencia a este.

Nombre
ActivateRock
Descripción
Este script pertenece a una trampa que se activa según la posición del personaje con referencia a este, señala la posición en la que va a caer y se mueve hacia abajo hasta destruirse.

Nombre
BolasDeFuego
Descripción
Este script pertenece a una trampa de forma circular. Permite que esta rote en el eje z y se desplace hacia adelante y después de pasado un tiempo se destruye.

Nombre
LanzasTrampa
Descripción
Este script activa sus funcionalidades cuando el personaje se acerca a una distancia predeterminada al <i>GameObject</i> que lo posee. Activa una cámara que proyecta una cinemática y una imagen en la pantalla que hace función de tutorial. Conoce si el personaje realizó las acciones pertinentes para desactivarla o no. Resetea todas las posiciones de la escena si el personaje muere intentando desactivarla.

Tabla 46: Scripts de trampas.

Scripts del mapa.

Nombre
Radar
Descripción
Este script se encarga de mostrar en el mapa la posición del personaje con referencia al lugar en el escenario en la que se encuentra.

Nombre
MiniMap_ArrowRotation
Descripción
Este script permite visualizar en el minimapa hacia donde está mirando el personaje.

Nombre
SmoothLookAt
Descripción
Este script permite mostrar la dirección en la que se encuentra el objetivo de la misión actual.

Tabla 47: Scripts del mapa.

3.5 Prueba

No existe un método estándar para las pruebas de juego, y la mayoría de las metodologías son desarrolladas por desarrolladores de videojuegos individuales. Las metodologías se están continuamente refinando y las pruebas se llevan a cabo según el tipo de juegos. El sub-proceso de prueba o el testing es uno de los más importantes dentro del desarrollo de videojuegos, ya que a través de este se controla la calidad del videojuego. La función principal de las pruebas es el descubrimiento y documentación de los defectos de software (JuegaLibre, 2013).

Después de concluido el proceso de desarrollo y la aplicación se encuentra en estado funcional se procede a ser probada por un equipo pequeño, que ha estado involucrado en el diseño y desarrollo del juego, en busca de errores para su refinamiento. Algunos de los aspectos a probar son la jugabilidad y el rendimiento.

3.5.1 Pruebas de Rendimiento.

Una vez concluido el videojuego queda propuesto para su correcto funcionamiento un ordenador con los siguientes requerimientos mínimos de hardware:

- Plataforma: Windows XP, Vista, 7 o 8.
- Memoria RAM: 1 Gb.
- Tarjeta Gráfica: Integrada de 64 MB.
- Tarjeta de audio compatible con los formatos mp3 y OGG.

El equipo de desarrollo realizó pruebas en ordenadores con estas prestaciones y en otras de mayor procesamiento, con el objetivo de verificar el rendimiento del videojuego. Se tomó en cuenta la cantidad de fotogramas por segundo a los que corría la aplicación sobre las diferentes computadoras para verificar la velocidad con que el sistema procesaba la información y las posibles causas de deficiencia. A continuación se muestran los resultados obtenidos en esta prueba.

Prestaciones de la PC	Frames/seg obsevados	Frames/seg total
<ul style="list-style-type: none"> • Intel Atom a 1.66 Gh • 1GB de RAM • Resolución: 1366x768 	19 FPS	30 FPS
Observaciones		

El sistema de iluminación usado en la escena de Baracoa disminuye considerablemente el rendimiento de la aplicación y el ordenador disminuye su rendimiento al utilizarse efectos de imagen avanzada.
Recomendación
Según las observaciones obtenidas de esta primera prueba se hacen necesario una mejora en el sistema de iluminación del primer escenario pues el exceso de uso de esta hace que disminuya el rendimiento, de igual modo el uso permanente de efectos de imágenes avanzado hace que el rendimiento también disminuya. En la segunda iteración se procedió al uso de lightProbes para mejorar el rendimiento de las luces en la escena y se implementó el uso de efectos de imagen de forma opcional.

Prestaciones de la PC	Frames/seg obsevados	Frames/seg total
<ul style="list-style-type: none"> • Dual Core a 2.1 Gh • 2GB de RAM • Resolución: 1024x768 	30 FPS	30 FPS
Observaciones		
Las imágenes de transición entre escenarios se visualizan comprimidas debido a la resolución de pantalla.		
Recomendación		
Según las observaciones obtenidas existe poca compatibilidad con varias resoluciones de pantalla por lo que en la segunda iteración se procedió a crear un script que se encargara de adaptar las imágenes de transición a la resolución de pantalla que posea el ordenador.		

Prestaciones de la PC	Frames/seg obsevados	Frames/seg total
<ul style="list-style-type: none"> • Intel Atom a 1.66 Gh • 1GB de RAM • Resolución: 1366x768 	30 FPS	30 FPS
Observaciones		
El sistema acepta perfectamente la aplicación, siendo ejecutada a nivel medio de gráficos y la interfaz se adapta bien a la resolución de pantalla.		
Recomendación		
-		

Tabla 48: Pruebas de Rendimiento.

Una vez culminadas las pruebas de rendimiento y detectadas las deficiencias, se pasa a una segunda revisión del videojuego para corregir los errores tomados, y se vuelve a comenzar otra iteración de prueba para verificar la veracidad de las correcciones.

3.5.2 Pruebas de Jugabilidad

Con el fin de llevar a cabo las pruebas de jugabilidad se tomaron una muestra de 3 estudiantes para que prueben el funcionamiento del videojuego, usando como método la entrevista. Las preguntas formuladas para estos estudiantes fueron las siguientes:

¿Cuáles fueron los aspectos más interesantes en el videojuego?

¿Te parecen cómodos los controles?

¿Qué se pudiera mejorar en la jugabilidad?

- Jugador 01: Dagmar Mir Leyva, 5to año de la Facultad 5.

Respuesta: El juego resulta muy interesante y atractivo ya que tiene un nivel gráfico muy bueno con respecto a los demás realizados en la universidad. De las cosas más interesantes que pude apreciar fue el poder aprender sobre los huesos que componen la cabeza humana además de los entornos que recrea el juego. Los controles son bastante fáciles de usar además de los tutoriales que brinda la aplicación que son muy intuitivos para las personas que juegan por primera vez. En sentido general todo está muy bien pero me gustaría que después que se tomen los objetos que indican las misiones se desactivaran las trampas para facilitar el paso pues si pierdes en una tienes que regresar de nuevo a recoger lo que perdiste.

- Jugador 02: Ernesto Toledo Fundora, 5to año de la Facultad 5.

Respuesta: La jugabilidad del juego es excelente, lo más emocionante es como este juego te lleva a un mundo completamente diferente en el cual se aprecia la originalidad de los creadores. El uso de los controles básicos de movimientos es bastante fluidos y operables, y crean una muy buena comodidad para el jugador. Como aspecto a mejorar para futuras versiones recomiendo el mejor manejo de las cámaras y un mapa más interactivo que sea capaz de dar una mejor orientación al jugador. En general la operatividad del juego es una de las mejores que he probado.

- Jugador 03: Aiman Mothac Abdul Rajman Valdés, 5to año de la Facultad 5.

Respuesta: El juego está muy bien concebido, sobre todo el enfoque que los autores le dan a este pues es impresionante el nivel de detalle que posee el nivel de la jungla y las ruinas que en ella se puede apreciar, sobre todo resaltar la calidad gráfica del mismo. Los controles me parecen muy cómodos a la hora de jugar pues todos están en posiciones en el teclado accesibles para los dedos. Algo que me gustaría que mejoraran es los botones para abrir las puertas que se encuentran muy distantes de ellas y en muchas ocasiones el jugador puede no darse cuenta de ellos. En sentido general me ha gustado muchísimo la idea de este juego.

Después de haber realizado las entrevistas a los jugadores se pudieron obtener algunas recomendaciones que estos apreciaron en el juego:

- Desactivar las trampas una vez alcanzado el objetivo de la misión.
- Mejorar el movimiento de la cámara.
- Mejorar la interactividad del mapa.
- Dar mejor visibilidad a los botones que abren las puertas.

Tomadas estas recomendaciones hechas por los probadores se pasó a una segunda iteración de la aplicación donde fueron corregidas el 75% de estas. Posteriormente se volvió a probar con las mismas personas y estas tuvieron un alto grado de aprobación al ver las mejoras implementadas.

Conclusiones

A partir del trabajo realizado se puede concluir que la unión de la metodología Huddle y el motor de videojuegos Unity, permitió un rápido flujo de trabajo. Terminando el videojuego en tan solo 9 meses y con un equipo de trabajo de 2 personas.

Por otra parte el uso de los diseños de interiores basados en los elementos psicológicos como la memoria subconsciente; los modelos mentales y los modelos conceptuales presentes a lo largo del videojuego y representados en los sistemas de trampas, propicia una mayor probabilidad de que el videojuego sea intuitivo y logre transmitir el contenido de la anatomía del esqueleto de la cabeza humana de manera más aceptada.

Las pruebas de jugabilidad llevadas a cabo permitieron conocer el nivel de aceptación del videojuego por parte de los usuarios, resultando motivante, de fácil maniobrabilidad en los controles, intuitivo e innovador durante toda su etapa de juego.

Se concluye que el uso de tecnologías de última generación para el desarrollo del videojuego permitió crear un producto con un rendimiento aceptable y escalable en cuanto a la calidad gráfica.

Recomendaciones

Con la elaboración de este proyecto se abren varias líneas de investigación y desarrollo, que contribuyan a la mejora de la aplicación presentada para el desarrollo de nuevos videojuegos destinados a la educación. Entre las mejoras posibles que se pueden realizar al videojuego desarrollado se destacan:

- Ampliar el número de tipos de trampas a usar. Actualmente el videojuego consta de algunas diseñadas pero pudiera buscarse una mejora de las mismas, en busca de una mayor asimilación del conocimiento a cuenta de la memoria subconsciente.
- Mejorar la interacción entre el avatar y el medio que lo rodea.
- Dar un mejor refinamiento al script de la cámara.

Bibliografía

- 3DEOLOGIC. 2013.** ZBrush Información general. *Pixologic*. [En línea] 2013. <http://pixologic.es>.
- Adobe. 2013.** Productos Adobe. *Adobe Photoshop*. [En línea] 2013. <http://www.adobe.com/es/products/photoshop.html>.
- Autodesk. 2013.** Autodesk 3ds Max. *Autodesk*. [En línea] 2013. <http://www.autodesk.es/adsk/servlet/pc/index?siteID=455755&id=14626995>.
- Bethke, Erik. 2003.** *Wordware Publishing*. s.l. : Inc., 2003.
- Bucarey, Sandra. 2006.** *Metodología de construcción de objetos de aprendizaje para la enseñanza de la Anatomía Humana en cursos integrados*. Chile : s.n., 2006.
- Barajas M, y. o. (2003).** *La Tecnología educativa en la enseñanza superior. Entornos virtuales de aprendizaje*. Madrid: McGraw-Hill/Interamericana de España.
- Cascales, Ferrer. 2007.** Atención. "PROCESOS PSICOLÓGICOS BÁSICOS". s.l. : Universidad de Alicante, 2007.
- Clark C, Abt. 1987.** *Serious Games*. New York : Viking Press, 1987.
- Epic. 2013.** *Epic Games*. [En línea] 2013. <http://www.epicgames.com/>.
- Flood, K. 2003.** Game Unified Process. *GameDev*. [En línea] 2003. <http://www.gamedev.net/reference/articles/article1940.asp>.
- Gamasutra. 2009.** "Gamasutra's Best of 2009: Top 5 Game Companies". *Gamasutra*. [En línea] 2009. http://www.gamasutra.com/view/news/26243/Gamasutras_Best_Of_2009_Top_5_Game_Companies.php.
- Gonzalez, Julia. 2003.** *Tunnig Educational Structures in Europe*. España : s.n., 2003.
- Jesús Arce, Luis. 2011.** Desarrollo de Videojuego. 2011.
- Journal., The Wall Street. 2013.** "The WSJ Technology Innovation Award Winners, Category by Category". *The Wall Street Journal*. [En línea] 2013. <http://online.wsj.com/article/SB10001424052748703904304575497473735761294.html>.
- JuegaLibre. 2013.** Prueba de juego. *JuegaLibre*. [En línea] 2013. <http://juegalibre.virtual.uniandes.edu.co/index.php/2013/03/07/testing-prueba-de-juego/>.
- Jugabilidad. 2010.** Desarrollo de Videojuegos. *Jugabilidad y Experiencias del Jugador*. [En línea] 2010. <http://jugabilidad.wikispaces.com>.
- Keith, C. 2009.** Waterfall Game Development,. *Agile Game Development*. [En línea] 2009. <http://www.agilegamedevelopment.com/2009/01/in-dawn-of-video-game-development.html>.
- Kruchten, P. 2004.** *The Rational Unified Process: An Introduction*. Estados Unidos: Addison-Wesley : s.n., 2004.

MCGAGHIE, WILLIAM C. 1987. *Introducción a la preparación de planes de estudios de la medicina basados en la competencia.* 1987.

Meléndez, Bernardo F. 2009. *ENTORNOS VIRTUALES COMO APOYO AL APRENDIZAJE DE LA ANATOMÍA EN MEDICINA.* Colombia : s.n., 2009.

Mod-DB. 2013. *Mod-DB.* [En línea] 2013. <http://www.moddb.com/engines>.

Piug, Rosell. 2001. *Morfología Humana I.* La Habana : Editorial Ciencias Médicas, 2001.

Rollings&Morris. 2003. *Game Architecture and Design: A New Edition.* California, : New Riders Publishing, 2003.

Scrum_Alliance. 2009. *What is Scrum? Scrum Alliance.* [En línea] 2009.
http://www.scrumalliance.org/pages/what_is_scrum.

UnityTechnologies. 2013. *Unity 3D. Unity Technologies.* [En línea] 2013. <http://unity3d.com>.

Zimmerman, Salen &. 2003. *Rules of Play: Game Design Fundamentals.* s.l. : MIT Press, 2003.

Zyda, Mike. 2005. *From Visual Simulation to Virtual Reality to Game.* s.l. : "Computer" de IEEE Computer Society, 2005.

Glosario de términos:

Cinemática: En un videojuego, cinemática (en inglés cutscene) se refiere a una secuencia de vídeo a través de la cual el jugador no tiene o tiene un control limitado, rompiendo el juego y se utiliza para avanzar en la trama, fortalecer el personaje principal, presentar personajes enemigos, y proporcionar información de fondo: la atmósfera, el diálogo, y las pistas. Estas escenas a menudo utilizan los gráficos del juego para crear eventos programados.

Occlusion Culling: Del español exclusión selectiva de oclusión, es una opción que deshabilita el procesamiento de los objetos cuando no se ven actualmente por la cámara, ya que están tapadas por otros objetos

Plugin: es una aplicación informática que interactuando con otra aplicación le aporta una función específica a ésta última, como por ejemplo servir como driver (o controlador) en una aplicación con el fin de hacer funcionar un dispositivo en otro software.

Visión periférica: La Visión Periférica es aquella que te permite abarcar hasta casi 180° con tu vista. Se ve todo, aunque no nítidamente.

GameObject: Cada objeto en el juego es un GameObject. Sin embargo, GameObjects no hacen nada por su cuenta. Ellos necesitan propiedades especiales antes de que puedan convertirse en un personaje, un entorno, o un efecto especial.

Game Engine: Del español 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.

MonoBehaviour: MonoBehaviour es la clase base de todos los scripts derivados de Unity 3D. Usando Javascript cada guión se deriva automáticamente de MonoBehaviour. Cuando se utiliza C # o Boo tiene que derivar explícitamente MonoBehaviour.

Frame: Se denomina frame en inglés, a un fotograma o cuadro, una imagen particular dentro de una sucesión de imágenes que componen una animación

FPS: Las imágenes por segundo (fotogramas por segundo o cuadros por segundo) (en inglés frames per second, FPS) es la medida de la frecuencia a la cual un reproductor de imágenes genera distintos fotogramas (frames).