



## RED FOREST

Memoria del trabajo del Máster de Creación de Videojuegos realizado por Jordi Arenas, Carlos Chacón, Marc Cuscallola, Alexandre Spyropoulos, David Torres

Bellaterra, Octubre de 2012



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Requerimientos del proyecto . . . . .	1
<b>2. Documento de diseño</b>	<b>3</b>
2.1. Game concept . . . . .	3
2.2. Descripción ampliada . . . . .	3
2.2.1. Género . . . . .	3
2.2.2. Propósito y público objetivo . . . . .	4
2.2.3. Plataforma . . . . .	5
2.3. Mecánicas . . . . .	5
2.3.1. Punto de vista . . . . .	5
2.3.2. Desarrollo general . . . . .	5
2.3.3. Escenario . . . . .	5
2.3.4. Sistema de juego . . . . .	6
2.3.5. Elementos interactivos . . . . .	8
2.3.6. Armas . . . . .	9
2.3.7. Ítems . . . . .	9
2.3.8. Acciones . . . . .	9
2.3.9. Interfaz . . . . .	12
2.3.10. Control . . . . .	13
2.4. Arte . . . . .	15
2.4.1. Caperucita . . . . .	15
2.4.2. El Lobo . . . . .	16
2.4.3. Enemigos . . . . .	16
2.4.4. Armas . . . . .	17
2.4.5. El bosque . . . . .	17
2.4.6. La cabaña . . . . .	19
2.4.7. La cueva . . . . .	19
2.5. Historia . . . . .	20
2.6. Tecnología . . . . .	21
2.7. Sonido . . . . .	22

<b>3. Desarrollo</b>	<b>23</b>
3.1. Arte . . . . .	23
3.1.1. Escenario . . . . .	23
3.1.2. Personajes . . . . .	28
3.2. Programación . . . . .	37
3.2.1. Gráficos . . . . .	38
3.2.2. Física . . . . .	40
3.2.3. Inteligencia Artificial . . . . .	42
3.2.4. Scripting . . . . .	44
3.2.5. Audio . . . . .	44
3.2.6. Tools . . . . .	45
<b>4. Resultados</b>	<b>49</b>
<b>5. Conclusiones</b>	<b>51</b>
<b>Bibliografía</b>	<b>53</b>
<b>Apéndice</b>	<b>55</b>
<b>A. Manual de usuario</b>	<b>55</b>

# Índice de figuras

2.1.	Ambientación oscura de la noche . . . . .	4
2.2.	Idea de Caperucita . . . . .	8
2.3.	Esbozo del menú de pausa . . . . .	13
2.4.	Ejemplo de estilo gráfico cell shading . . . . .	15
2.5.	Idea de la cara de Caperucita . . . . .	16
2.6.	Idea del Lobo . . . . .	16
2.7.	Idea de los enemigos . . . . .	17
2.8.	Idea de armas . . . . .	18
2.9.	Idea del bosque . . . . .	18
2.10.	Idea de la cabaña . . . . .	19
2.11.	Idea de la cueva . . . . .	19
3.1.	Bocetos sobre el escenario inicial . . . . .	24
3.2.	Investigación del escenario y movimientos en el Laberint d'Horta	24
3.3.	Proceso de creación del escenario (de izquierda a derecha y de arriba a abajo) . . . . .	25
3.4.	Textura de 4096x4096 aplicada como primera prueba) . . . .	26
3.5.	Escenario en 3ds Max con la textura de definición del terreno)	26
3.6.	Iluminación del escenario. Arriba izquierda: tonalidad amarillenta. Arriba derecha: Amarillenta con rojizo. Debajo: Tonalidad final rojiza) . . . . .	27
3.7.	Textura del skydome final . . . . .	27
3.8.	roca en high poly y low poly . . . . .	28
3.9.	Todos los personajes del juego . . . . .	29
3.10.	Referencias de arte sobre Caperucita . . . . .	30
3.11.	Proceso de creación de Caperucita . . . . .	30
3.12.	Proceso de creación del Lobo . . . . .	32
3.13.	Referencias para el Lobo . . . . .	32
3.14.	Referencias para el conejo . . . . .	33
3.15.	Proceso de creación del conejo . . . . .	34
3.16.	Referencias para el ciervo . . . . .	36
3.17.	Proceso de creación del ciervo . . . . .	36
3.18.	Escena final con las sombras . . . . .	38
3.19.	Mapas de sombras creados con Cascaded Shadow Mapping . .	39

3.20. Diferentes mapas de sombras aplicados a la escena. El rojo es el mapa con más calidad y el azul el que tiene menos calidad.	39
3.21. Efecto de lens flare en el juego . . . . .	41
3.22. Escena de test de físicas . . . . .	42
3.23. Escena de test de de inteligencia artificial . . . . .	44
3.24. Editor de Wwise con el proyecto del juego . . . . .	45
3.25. Particle Editor . . . . .	46
3.26. GUI Editor . . . . .	47

# Capítulo 1

## Introducción

Este documento presenta el trabajo realizado durante todo el año de duración del Máster de creación de videojuegos de la UAB. El objetivo del máster es aprender a técnicas del desarrollo de un videojuego, tanto de la parte de arte, como la de diseño y programación. Finalmente, y para poner en funcionamiento todos los conocimientos aprendidos, se desarrolla un pequeño videojuego.

Antes de empezar en un videojuego cualquiera, se hicieron varias sesiones de *brainstorming* para exponer ideas de juegos y así confeccionar proyectos a poder realizar. De estas reuniones salieron dos ideas.

La primera idea era un juego *hack 'n' slash* en tercera persona ambientado en una catedral con un aspecto realista. El juego tomaba fuerza con el espíritu del personaje, un ser que atrapado en el limbo. El personaje, sin piel, siempre estaba perdiendo sangre, y tenía que matar a los enemigos para recuperar la sangre perdida y así poder vivir.

La segunda idea que surgió fue otro juego *hack 'n' slash* en tercera persona, pero esta vez ambientado en el cuento de Caperucita Roja. El objetivo planteado era ir a matar al Lobo porqué se había comido a la abuela de la protagonista. El juego se ambientaba en un bosque nocturno y nevado, y tendría un estilo más *cartoon*.

Finalmente, se eligió el juego de Caperucita Roja, inicialmente llamado como Caperucita Loca.

Una vez seleccionado el proyecto a realizar, se empezó a buscar información y referencias sobre lo que podría contener el juego. Se definieron tres escenarios distintos donde pasaría la acción: el bosque, la cabaña de la abuela y la cueva del Lobo feroz.

### 1.1. Requerimientos del proyecto

El proceso de creación del proyecto ha estado marcado por diferentes fases, unas más marcadas que otras.

1. Diseño inicial. Se planteó toda la idea del videojuego, así como el diseño del escenario, enemigos, personaje principal y demás elementos que conforman el juego.
2. Prototipo.
  - Arte. Creación de un esbozo de escenario, el personaje principal y un enemigo.
  - Programación. Implementación del motor del juego y una idea del *gameplay* del juego.
3. Desarrollo.
  - Arte. Creación del escenario, de los personajes (principal como enemigos) y demás elementos que conforman el juego.
  - Programación. Mejorar el motor con nuevas técnicas y herramientas, y programación completa del *gameplay* y de la inteligencia artificial.
4. Integración. Juntar el trabajo de arte y programación para tener el juego listo.
5. Testeo. Fase de pruebas, para poder corregir errores no detectados anteriormente.

En el capítulo 2 de la memoria se muestra el documento de diseño inicial. En el capítulo 3 se muestra todo el proceso de desarrollo del juego en sus tres vertientes: diseño, arte y programación. En el capítulo 4 se presentan los resultados obtenidos y en el capítulo 5 las conclusiones sacadas de todo el proceso. Finalmente, en el apéndice se entrega un pequeño manual del juego.

# Capítulo 2

## Documento de diseño

### 2.1. Game concept

Juego en 3D de tipo *hack 'n' slash* y aventura en tercera persona, donde se inspira en el universo de los cuentos para niños llevándolo a una versión *gore* y bastante violento.

### 2.2. Descripción ampliada

La historia transcurre en un mundo tidimensional con una atmósfera oscura y de terror (figura 2.1), con cierta visión *cartoon*, donde Caperucita lo recorre bajo la luz de la luna buscando la venganza de la muerte de su abuela. Para ello deberá encontrar al malvado Lobo pero antes tendrá que combatir con decenas de monstruos que han sido convocados por el Lobo, cortándolos en diferentes trozos con su hoz destructora y sus cuchillos cortantes.

En esta aventura, caperucita deberá recorrer el bosque y la casa donde habita su abuela matando a todos los enemigos que se encuentren a su paso. Como su abuela estará muerta por el lobo, Caperucita irá a la cueva del lobo en busca de venganza. ¿Logrará Caperucita matar al lobo feroz que gobierna el bosque y que mató a su abuela?

#### 2.2.1. Género

Este juego es de tipo *hack 'n' slash* en combinación con aventura de acción y el subgénero *beat them all*. Se caracteriza por combinar elementos de aventura con elementos de suspense y acción, dentro de una atmósfera de terror psicológico donde lo que prima es la supervivencia.

Está basado en varios juegos ya existentes como *American McGee's Alice Madness Returns* [1], *Fairytales Fights* [2], *God of War* [3] y *Darksiders* [4], entre otros.



*Figura 2.1: Ambientación oscura de la noche*

### 2.2.2. Propósito y público objetivo

El principal objetivo de Caperucita Loca es ofrecer a los jugadores un entorno vistoso, tridimensional y con mucha acción al estilo gore. Es un juego destinado básicamente a gente adulta mayores de 18 años y enfocado a la audiencia masculina. La gente que prefiere estos tipos de juegos son gente joven comprendida entre los 15 y 30 años debido a que está situado en un ambiente de aventura, misterio y suspense que permite liberar adrenalina mientras juegan.

Los jugadores jugarán a este juego por esa motivación de vivir experiencias o escenas de charcos de sangre algo prohibitivas durante la etapa adolescente y poder experimentar ese sentimiento de terror y de masacre de cuerpos.

Dentro del mercado, podríamos situar este juego cercano a otros existentes como *Alice Madness Returns* en cuanto a las dimensiones de los escenarios, de la protagonista y los enemigos, y del *Fairytales Fights* aunque con mayor grado de acción gore y sin tanta fantasía. También se pude parecer en algunas mecánicas a *Darksiders* en cuanto a la captura de almas o bien a los tipos de secuencia de acciones (combos) o también a la cantidad de enemigos que acechan al protagonista en un mismo momento. A diferencia de estos juegos, Caperucita no se parece a uno de ellos en concreto sino que tiende a mezclar todo lo anteriormente comentado aumentando la acción con la aparición de un mayor número de enemigos en cada batalla en la que se vea envuelta intentando sobrevivir con un conjunto de ataques y combos.

Comparado a otros juegos, Caperucita es un concepto original porque todavía no existen juegos que mezclan el tipo *hack 'n' slash* y el cuento de la Caperucita Roja. Es un concepto de juego que tiene su propia identidad. Por otra parte, el tipo de juego *hack 'n' slash* no es un estilo muy común en comparación con la gran cantidad de juegos de tipo *FPS (First Person Shooter)*, *MMORPG (Massively Multiplayer Online Role-Playing Game)* o

deporte.

### 2.2.3. Plataforma

El juego se ha desarrollado para PC con sistema operativo Windows [9]. El juego se podría transportar a otras plataformas como *PlayStation 3* [10] o *Xbox 360* [11].

## 2.3. Mecánicas

### 2.3.1. Punto de vista

El juego será en tercera persona con la cámara en perspectiva detrás del personaje para tener una buena visibilidad del escenario, de los enemigos y de los ítems existentes.

Podría existir la acción de rotación de cámara mediante teclas o el ratón, pero con la principal característica de pivotar sobre el personaje principal.

### 2.3.2. Desarrollo general

El juego es lineal combinando zonas de combate y plataformas. Estará ambientado en un bosque de noche al aire libre pero limitado, la cabaña de la abuela y la cueva que hace de madriguera del Lobo, el enemigo principal.

El jugador pasará por diferentes zonas (2 o 3) de combate y plataformas, separadas adecuadamente.

### 2.3.3. Escenario

Idealmente se harán tres escenarios contiguos:

1. El bosque. La historia empieza en el bosque, permitiendo al jugador aprender cual es las mecánicas básicas, como debe manipular al personaje y familiarizarse con el control. Así pues, se mostraran los controles para que el jugador practique y aprenda los posibles movimientos y combinaciones de ataque y defensa. Una vez pasada esta primera parte del bosque, el jugador estará preparado para los combates posteriores.

En un segunda parte en el bosque, el jugador empezará a encontrar enemigos, básicamente animales del bosque.

El bosque estará formado por árboles sin hojas, debido a que es otoño o invierno. Las ramas de los árboles estarán caídas dando la sensación de ser árboles muertos y rotos. También contendrá arbustos, rocas y elementos característicos de los bosques. Habrá sensación de viento.

Existiran zonas pequeñas donde la habilidad del salto sea necesaria. Estas zonas no contendrán enemigos.

2. La cabaña. La cabaña estará situada en medio dle bosque y estará visiblemente iluminada para que el jugador llegue y vea lo sucedido. Se podrá acceder dentro de la cabaña. El interior estará formado por un gran comedor con chimenea y un par de habitaciones, la de la abuela y otra de invitados. En caso de existir más habitaciones, estas no serán accesibles por el jugador.

La cabaña, externamente, estará formada por piedra y madera. Puede haber un jardín con algunas flores en tiestos.

En el interior habrá una cinemática con parte de historia, donde se explicará como el Lobo se come a la abuela.

3. La cueva. Estará formada por un pasillo lardo y poco ancho, desenvolviendo a una sala más grande donde reside el enemigo final, el Lobo. La cueva será de roca y estará iluminada mediante velas.

Para acceder a esta parte del escenario, se hará a través de una trampilla que hay dentro de la cabaña, o bien por el bosque siguiendo un camino lineal.

#### **2.3.4. Sistema de juego**

##### **Objetivos**

El objetivo principal del juego es derrotar el Lobo que habita el bosque. El jugador deberá matar a todos los enemigos que se encuentre en su camino y que han sido enviados por el lobo para, finalmente, enfrentarse a él en un combate final.

Para finalizar cada nivel hay que llegar al final de este, derrotando a cualquier enemigo que se interponga en el camino del jugador.

La primera parte del bosque será un tutorial de las mecánicas del juego. La segunda parte, será para aprender a combatir y a defenderse. A medida que el jugador avance por el nivel, la cantidad de enemigos irá en aumento hasta la cabaña.

En la parte de la cabaña la acción será más agresiva matando a más enemigos. Una vez matados todos, se mostrará una cinemática o mediante mensajes, un poco de historia.

En el momento de luchar, el personaje puede realizar varias secuencias de ataques diferentes por cada arma que lleve equipada, lo que llamamos *combos*.

Finalmente, Caperucita irá a la cueva para matar al Lobo, donde estará lleno de enemigos más fuertes. Al final del nivel, habrá la batalla contra el enemigo más feroz del juego, el Lobo.

### Reglas

- El desarrollo de una partida es bastante sencillo y lineal. El personaje avanza y salta en el entorno y así puede moverse en todas las direcciones con libertad.
- Puede explorar el entorno para encontrar elementos beneficiosos para el jugador.
- Para finalizar cada nivel, se tiene que llegar al final de este derrotando a cualquier enemigo que se interponga en el camino.
- Dentro del escenario del bosque hay unos límites de escenario para que el jugador no salga.
- El personaje no puede salir de una plataforma caminando.
- La cueva es un entorno cerrado y, por lo tanto, ya limitado marcando un único camino a seguir.
- Caperucita tiene una barra de vida y la partida se termina cuando se pierde toda la vida.
- Si Caperucita pierde toda la vida, el juego se reiniciará al último punto de control.
- Durante una partida no hay límites de tiempo.
- Los puntos de control serán al inicio de cada nivel.
- Los enemigos mueren cuando no tengan vida.
- Para derrotar a un enemigo, el jugador deberá hacer uso de las habilidades de ataque, intentando que estos queden si vida.
- Cuando un enemigo muere, puede dejar objetos que el jugador podrá recoger.
- Los combos estarán limitados a n golpes. Una vez finalizados, se empezará el combo de nuevo.
- Si el jugador está en defensa, los golpes no restarán vida.
- El Lobo será el enemigo más fuerte y con más vida.
- Los enemigos también tendrán modo defensa donde no se le bajará la vida.
- El juego hará guardados automáticos. Se hará en punto específicos.



*Figura 2.2: Idea de Caperucita*

- En cualquier momento, el usuario podrá acceder al menú de la partida pulsando una tecla.
- En el menú se podrán configurar diferentes apartados del juego.

### 2.3.5. Elementos interactivos

Como elementos interactivos dentro del juego hay:

- El personaje principal: Caperucita.
- El enemigo principal: el Lobo.
- Enemigos secundarios.
- Algunas armas, como la ballesta.

Todos estos elementos tendrán animaciones propias.

#### Personaje principal

Caperucita. Está basada en los cuentos de *Caperucita* pero dimensionada hacia una atmósfera totalmente *gore* asemejándose a la chica protagonista de la película *Hard Candy* [12] de David Slade. Mide unos 1.55m y unos 40kg de peso. Tiene una fisonomía muy delgada y su cara destaca por sus ojos grandes y luminosos (figura 2.2).

Lleva un vestido con falda y se caracteriza por su capa roja que la cubre enteramente, protegiéndola del frío del bosque.

#### Enemigos

Durante el combate, irán apareciendo enemigos constantemente, pero sólo habrá 10 enemigos en batalla. Los enemigos son:

- El Lobo. El principal enemigo del juego. Este puede llamar a otros monstruos de la noche para matar al jugador. Es bastante más grande que el personaje principal. Destaca por tener un cupero musculoso y muy peludo, lo que le hace muy rápido y fuerte. También se caracteriza por sus grandes manos, una grandísima boca con unos colmillos muy afilados y grandes ojos.
- Rata. Pequeña y peluda con una gran boca y dientes pequeños pero afilados. En sus mordiscos lleva una gran cantidad de veneno.
- Murciélagos. Tiene un cuerpo pequeño y grandes alas con unas garras pequeñas pero afiladas, permitiendo desgarrar cualquier trozo de carne de sus presas. Vislumbra perfectamente en la noche a sus presas con sus ojos brillantes.
- Conejo. Lleva una zanahoria con punchos. Tiene un ataque a distancia que consiste en lanzar zanahorias.

#### 2.3.6. Armas

El jugador podrá tener dos armas básicas.

- Hozes. Caperucita tiene dos hozes, una en cada mano que le permiten realizar golpes circulares y verticales, cortando enemigos por la mitad.
- Ballesta. Es una arma a distancia. Se apunta con la misma orientación que el personaje y se recarga automáticamente.

#### 2.3.7. Ítems

Los ítems son variados y modifican alguna característica del jugador. Algunos ítems son permanentes mientras que otros temporales. Estarán situados por el escenarios y se activan cuando el jugador pasa por encima de ellos para recogerlos.

- Manzana. Aumentan la vida del personaje. Se encuentra por el suelo o se desprende de los enemigos al morir.
- Setas. Aumentan la fuerza del jugador temporalmente.

#### 2.3.8. Acciones

En esta sección se describen las acciones de los personajes.

- Caperucita. El personaje podrá realizar las siguientes acciones:
  - Moverse por el escenario tridimensional.

- Saltar en todas direcciones y sobre si mismmo, es decir, en vertical.
- Atacar.
- Defender.
- Cada arma tendrá unos movimientos o secuencias de ataque diferentes. En algunas armas se podrán hacer consecutivamente una secuencia de movimientos. Si la secuencia ha tenido éxito, el siguiente golpe será un golpe especial para rematar al enemigo. El combo se desactivará cuando alguno de los movimientos no toquen al enemigo.
- Recoger ítems. Se recogerán al pasar por encima del objeto y se aplicarán en ese mismo instante.
- El Lobo. El enemigo principal tendrá ciertos movimientos como los siguientes:
  - Ataque melée (cuerpo a cuerpo).
  - Golpes con las garras.
  - Coger a Caperucita y morderla.
  - Rugir. Después del rugido, aparecen nuevos seres de la noche para ayudar al lobo.
- Rata.
  - Ataque melée.
  - Morder.
  - Inyecta veneno. Se cura con un antídoto.
- Murciélagos.
  - Ataque aéreo.
  - Morder.
  - Inyecta veneno. Se cura con un antídoto.
- Conejo.
  - Ataque melée.
  - Defensa.

### Animaciones

A continuación se detallan las animaciones de los diferentes enemigos y del personaje principal.

- Caperucita.
  - Estado de reposo (Idle).
  - Correr.
    - Ir de frente.
    - Desplazarse lateralmente.
    - Ir hacia atrás.
  - Defenderse.
  - Bloquear un ataque mientras se defiende.
  - Saltar.
    - De forma vertical.
    - en una dirección dependiendo de la tecla presionada.
  - Derrumbarse / Levantarse.
  - Morir.
  - Atacar. Dependiendo del arma que lleve equipada Caperucita tendrá unas animaciones u otras.
    - Hoz.
      - Armarla.
      - Golpe inicial.
      - Golpe final.
    - Ballesta.
      - Armarla.
      - Disparar / Recarga automática.
- El Lobo
  - Idle.
  - Correr.
  - Caminar.
  - Golpes con las garras.
  - Coger a Caperucita y morderla.
  - Rugido.
  - Defenderse con las patas delanteras.
  - Atacar. Tendrá más de un ataque, así que tendrá más de una animación en este aspecto.

- Rata.
  - Idle.
  - Correr.
  - Caminar.
  - Morder.
- Murciélagos.
  - Idle.
  - Correr.
  - Caminar.
  - Ataque aéreo.
  - Morder.
- Conejo.
  - Idle.
  - Correr.
  - Caminar.
  - Ataque con garras.
  - Se protege con las garras y agachándose.

### 2.3.9. Interfaz

La interfaz del juego está compuesta por dos partes, la que pertenece al juego en sí, el HUD, y la otra todo el sistema de menús.

#### HUD de juego

El HUD del juego se forma en dos partes distintas:

- **HUD(*Head-Up Display*)**. Estará compuesto por una barra de vida que irá variando según la vida del jugador. Será una barra horizontal. También contendrá un ícono mostrando qué arma lleva el personaje en ese instante. Se podrá mostrar una lista con las armas disponibles en ese momento y seleccionar una concreta. Por defecto esto estará oculto y sólo se mostrará cuando se pulse una tecla.
- Menú de pausa. Desde dentro del juego se podrá acceder a un menú simple. Su función será de pausa del juego y contendrá las siguientes acciones (figura 2.3):
  - Continuar partida. Permitirá volver al juego y continuar jugando.



*Figura 2.3: Esbozo del menú de pausa*

Opciones. Se podrá modificar parámetros del juego.

Finalizar partida. La partida se terminará y se irá al menú principal.

### Menús externos

Cuando se ejecute por primera vez el juego o se salga de la partida, se irá al sistema de menús.

- Menú principal. Es la primera ventana que sale de todo el menú. Contendrá las siguientes opciones:

Empezar partida. Permite al jugador empezar el juego desde el inicio.

Continuar partida. Continua la partida con la última partida guardada.

Opciones. Permite configurar el juego.

Créditos. Se muestra la información de todos los componentes del grupo.

Salir. Se finaliza la ejecución del juego.

- Menú de opciones. Permite configurar diferentes opciones del juego.

- Pantalla de información. Se mostrará información sobre los controles y el objetivo del juego.

#### 2.3.10. Control

El juego funcionará con teclado y ratón. Las acciones generales son:

- Activar habilidad: Botón medio del ratón (BTM).

- Cambio de armas: Rueda del ratón (BR).

### **Movimiento de desplazamiento**

- Desplazarse.
  - A la izquierda: A.
  - Hacia abajo: S.
  - A la derecha: D.
  - Adelante: W.
- Andar: Shift.
- Acelerar: Alt.
- Saltar: Barra espaciadora.
  - Doble salto: Barra espaciadora + barra espaciadora.

Combinando con alguna acción de teclado o ratón puede ejecutar un golpe específico.

### **Movimientos de ataque**

- Atacar: Botón izquierdo del ratón (BTI).
- Acelerar + ataque. Se usará para atacar y empujar a un enemigo.
  - Choque y golpeo: Alt + BTI + Dirección.
  - Choque y doble rajada: Alt + BTI + BTI + Dirección.
  - Salto, choque y rajada: Alt + Barra espaciadora + BTI + Dirección.
- Patada del dragón (combo).
  - Golpe simple: Dirección + BTI.
  - Golpe con giro: Dirección + BTI + BTI.
  - Golpe final: Dirección + BTI + BTI + BTI.
- Cuchillos (combo).
  - Corte: BTI.
  - Doble corte: BTI + BTI.
  - Triple corte mortal: BTI + BTI + BTI.
  - Golpe final: BTI + BTI + BTI + BTI.



Figura 2.4: Ejemplo de estilo gráfico *cell shading*

- Hoz (combo).
- Corte: Dirección + BTI.
- Golpe de serpiente: Dirección + BTI + BTI.
- ...

### Movimientos de defensa

- Protegerse: Botón derecho del ratón (BTD).

## 2.4. Arte

El estio gráfico del juego estará inspirado por el mundo de los cuentos de niños, *cartoon*, pero adaptado a un público mayor con una dimensión *gore* y sangriento. Este estilo permite exagerar el diseño de los objetos y del entorno en general. Para dar aspecto *cartoon*, se aplicará la técnica del *cell shading* [14] (figura 2.4).

Para reproducir un ambiente lúgubre, la iluminación estará marcada por la Luna y los faros que habrá repartidos por el bosque. El ambiente será nocturno y siniestro. La cabaña será de colores más vivos y con más luz. Finalmente, la cueva volverá a ser un lugar con poca luz y muy tenebrosa. Sólo estará iluminado por pequeñas luces repartidas por el escenario.

### 2.4.1. Caperucita

El personaje de Caperucita está inspirado por la chica que protagoniza la película *Hard Candy* y de los personajes del juego *Fairytales Fights*. En el juego, no veremos su cara de forma limpia, sólo sus ojos blancos o rojos que brillan. El resto del rostro estará oculto a causa de la sombra que genera la capucha de su capa (figura 2.5).



Figura 2.5: Idea de la cara de Caperucita



Figura 2.6: Idea del Lobo

El personaje será pequeño pero sus armas serán más grandes para exagerar y amplificar el lado *gore* del juego.

La indumentaria de Caperucita se basa en el típico vestido con capa roja del cuento. Esta podría moverse con el viento o con los movimientos que el jugador realice siempre.

#### 2.4.2. El Lobo

El Lobo (figura 2.6) será un enemigo muy grande con una gran cantidad de pelo y muy musculado.

#### 2.4.3. Enemigos

Los enemigos serán feroz, con partes desgarradas y deformadas (figura 2.7).



*Figura 2.7: Idea de los enemigos*

#### 2.4.4. Armas

Las armas (figura 2.8) estarán hechas de madera y hierro y serán más grandes que la protagonista.

#### 2.4.5. El bosque

Es un bosque muerto (figura 2.9) en la época de invierno después de una nevada, iluminado por la luza de la luna y unos candelabros que van marcando el camino que debe recorrer Caperucita para llegar a la cabaña de la abuela. Este escenario tendrá árboles, rocas y matojos donde las hojas de los árboles se revuelven muy inquietas, expectantes, donde apenas hay ruido y donde se visualiza también una crecida luna, tan grande como luminosa. En este escenario se modelarán varios árboles con ramas largas y sin hojas para apoyar este estilo lúgubre.

Aún así, el diseño del entorno será bastante sencillo mientras que la escena será bastante larga para permitir al jugador moverse con cierta libertad para posibilitar el escape de los ataques de olas de enemigos. Habrá una luz fija emitida por la luna, y antorchas o faros que iluminaran el camino a seguir por el jugador.



Figura 2.8: Idea de armas



Figura 2.9: Idea del bosque



*Figura 2.10: Idea de la cabaña*



*Figura 2.11: Idea de la cueva*

#### 2.4.6. La cabaña

Estará situada en medio del camino del bosque y estará construida de piedra y madera (figura 2.10). Tendrá una chimenea en un lateral donde siempre saldrá humo indicando que en su interior se respira tranquilidad y amor. El jugador tendrá que entrar dentro de ella. El entorno exterior de la cabaña estará limitado por la mismas reglas del bosque. Tendrá más iluminación que el bosque o la cueva, para que sea más visible.

#### 2.4.7. La cueva

Será bastante oscuro aunque existirán puntos de luz que iluminarán el camino hacia la madriguera del Lobo (figura 2.11). Los puntos de luz pueden ser faros o antorchas con fuego. En la sala donde se combate con el Lobo, será una sala grande con más luz que el resto de la cueva para que se pueda disfrutar de la lucha. En la sala principal habrá un poco de agua.

## 2.5. Historia

Esta es la historia de Caperucita Roja, como nadie lo había contado antes...

*«Nunca vayas al bosque sola, mi nieta, ya que están habitadas por el lobo y aunque nos han hecho retroceder las hordas y hordas de enemigos, nadie se puede comparar con la ferocidad del Lobo.»*

La Caperucita Roja era una niña seria y disciplinada que vivía una vida tranquila en el pueblo de una tierra lejana. Después de muchas guerras sangrientas contra los invasores que venían de todas las partes del Reino, la gente estaba acostumbrada a defenderse y se habían convertido en brillantes combatientes. Los secretos de este estilo de lucha estaban en manos de una persona temida y respetada por todos, la Abuela de Caperucita Roja. Ella había enseñado mientras varias generaciones técnicas de combate mortal, convirtiendo se en formidable arma de guerra. Los rumores corrían acerca de su pasado, que poca gente sabía verdaderamente, que ella era la descendiente de un gran guerrero que había diezmado a ejércitos enteros de enemigos equipado con un arma legendaria, la Hoz de la Serpiente. Pero no se sabía si estas historias estaban dentro de la leyenda o la realidad. El pueblo era construido en el borde de un inmenso bosque. Todos los aldeanos tenían miedo del bosque porque allí vivía el Lobo, un animal maléfico que la gente pensaba llegado del Infierno. La ferocidad del Lobo no se comparaba con ningún enemigo del Reino. Poca gente había visto el Lobo en verdad porque el monstruo casi nunca se deja el bosque.

La abuela de Caperucita Roja tuvo solo su nieta en el mundo. Ella la amaba el amor de una madre y se había notado en su nieta una increíble potencial de combate. Caperucita Roja era ágil como Mantis Religiosa, fuerte como las garras del Tigre. La Caperucita Roja seguido todos los días entrenamiento intensivo que exigía disciplina y su abuela estaba orgulloso del progreso notable de su nieta, día tras día.

Pero un día, varios residentes de la aldea desaparecieron sin dejar rastro. La gente entró en pánico y la abuela decidió que tenía aclarar la situación. Ella comenzó a buscar pistas. Al examinar una de las casas al norte del pueblo, fue aterrorizada cuando se dio cuenta de las señales de garras impresionantes, 2 de 3 metros, penetrando profundamente en el suelo. De repente, ella sabía y sentía el suelo ceder bajo sus pies: era el Lobo. Ella lanzó un grito de rabia y corrió con todas sus fuerzas hacia el bosque para hacer frente a sus peores temores, su pesadilla de la más formidable, el Lobo, que ella quería ser una mera leyenda.

Pasaron varios días sin que nadie sepa el paradero de la abuela. Caperucita Roja estaba temblando de miedo y de rabia por no ser lo suficientemente fuerte para luchar contra el terrible Lobo, al lado de su Abuela.

Fue sólo después del cuarto día que el pueblo recuperó la abuela agonizando a las puertas del Pueblo, la cara ensangrentada y el cuerpo salpicado

de rasguños profundos. La llevaron en su cama y, equipada con las últimas fuerzas que le quedaban ordenó que se hace de Caperucita Roja llegan a su lado. Caperucita Roja entró en la habitación de su Abuela moribunda. Ella se molestó hasta su alma cuando vio a su Abuela en esta condición, ella que era de una fuerza y agilidad sin igual. La Abuela no dijo nada, se volvió y agarró algo que ella entregó a su hija. Bajo la luz de las velas, el objeto de una extraordinaria luz brilló. Caperucita Roja no podía creer lo que veía: su Abuela le dio la Hoz de la Serpiente Legendaria. Su Abuela le dijo que era un arma con una letalidad sin precedentes y que había tomado un sin-número de vidas. Ella había escondido la Hoz todos estos años porque temía el poder de la Hoz de la Serpiente Legendaria y decidió que se pondría de manifiesto sólo en contra de un enemigo llegado directamente del Infierno. En su último aliento, le dijo a la Caperucita Roja que llevar esta arma era una gran responsabilidad y que ahora se había convertido en una guerrera verdadera.

*«Te ha dicho que nunca vayas al bosque sola, mi nieta, dijo la Abuela, ya que están habitadas por el lobo y aunque nos han hecho retroceder las hordas y hordas de enemigos, nadie se puede comparar con la ferocidad del Lobo. Pero equipada con la Hoz Legendaria del Serpiente, nadie se puede quedar en tu camino. Ve, mi nieta, no tiene miedo, ve a proteger el pueblo, y abraza tu destino...»*

Y así la Abuela se murió.

Caperucita Roja se puso de pie, temblando de rabia y de la tristeza de su pérdida. Nunca se había sentido tan dispuesto. Fue a las puertas de la aldea, bajo la mirada de la gente asustada. Se aventuró en el bosque bajo la lluvia, las gotas chorreando en su rostro sereno y determinado. Al entrar en un bosquecillo, ella llegó de repente frente al Lobo. El Lobo fue impresionante. Era enorme, tenía los dientes tan largos que se rascaban el suelo y las garras brillando en la luz de la luna. Fue a lo largo de al menos 4 pies, su piel era impresionante y reflejo tonos púrpura. Caperucita Roja pensó en las palabras de su abuela, que era un enemigo vino directamente del Infierno. Se percibe en la boca del Lobo como una sonrisa y se dio cuenta de que el estaba esperando por ella. En este momento, la Caperucita Roja era tan enfadada y tan intimidante que la leyenda se recuerda de ella cómo la Caperucita Loca. Y la lucha empezó.

## 2.6. Tecnología

El desarrollo del juego se centra esencialmente sobre las animaciones de los personajes. Para obtener un juego visualmente fluido y no sobrecargar el motor, se ha elegido una escena sencilla con poco elementos (el bosque en una noche de invierno), lo que permite dar más peso a los personajes y sus movimientos.

El control será una parte clave del *gameplay* junto con la inteligència artificial.

Se diseñará un motor 3D que soporte las últimas tecnologías del mercado. Utilizará técnicas avanzadas de renderizado como *normal mapping* [13].

## 2.7. Sonido

Los sonidos y la música tienen que estar acorde con las características del escenario. En líneas generales la música será terrorífica con sonidos que generen sustos y suspense. Respecto a los sonido de las acciones, ítems y enemigos, tenderán a ser sonidos con aspecto sanguinario y chillidos de horror.

La música estará compuesta exclusivamente para el juego, y los sonidos se buscarán en bibliotecas gratuitas *online*.

# Capítulo 3

## Desarrollo

El desarrollo del juego se puede separar en dos grandes bloques diferentes entre sí: Arte y programación.

### 3.1. Arte

Todo el arte del juego se ha hecho mediante los programas Autodesk 3ds Max para el modelado y Photoshop para la creación de las texturas.

#### 3.1.1. Escenario

El escenario del juego es un bosque nevado durante el atardecer de una noche de invierno.

Inicialmente el escenario del juego estaba diseñado en tres niveles (figura 3.1). El bosque donde el jugador aprendía a usar las armas, así como el resto de mecánicas. La cabaña formaría parte del bosque. Después una zona llena de columnas que sostentaban el bosque. En esta zona el jugador podría esconderse de los enemigos, y los enemigos de Caperucita. Finalmente, el último escenario sería la cueva del lobo donde habría la batalla final terminando el juego con la muerte del Lobo.

A medida que se iba avanzando en el proyecto, se eliminaron los dos últimos escenarios, la zona de columnas y la cueva, por falta de tiempo. Así también, permitía centrar los esfuerzos en un solo escenario haciendo que este, luzca más y mejor. Con este cambio de escenario la historia no cambia demasiado así que la decisión fue acertada.

#### Terreno

En el desarrollo del terreno se probaron varios tamaños. En primer lugar se hizo un escenario de  $1km^2$ , pero se desestimó cuando se vió que era demasiado grande para el juego a desarrollar. El tamaño final del escenario es de  $400m^2$ , siendo un tamaño de escenario grande permitiendo dar sensación de

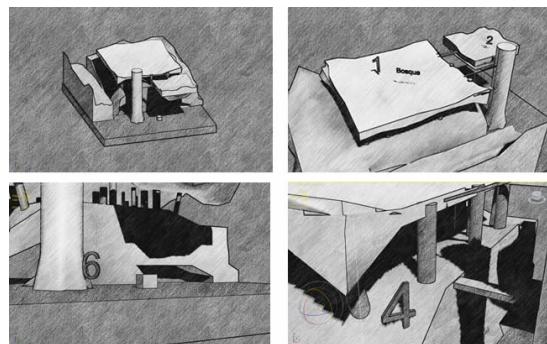


Figura 3.1: Bocetos sobre el escenario inicial

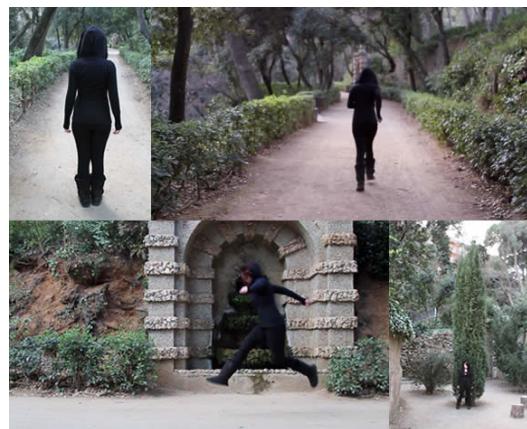


Figura 3.2: Investigación del escenario y movimientos en el Laberint d'Horta

libertad, y al mismo tiempo, conseguir que no sea un escenario vacío debido al tamaño.

Después de obtener los bocetos del terreno y un recorrido coherente para el jugador, se realizó una pequeña investigación de campo (figura 3.2). En esta investigación se hicieron fotografías a una mujer con una estatura similar a la que se quería con Caperucita y con una ropa similar. Con este trabajo se vió el tiempo promedio el tiempo a recorrer una cierta distancia, la altura a saltar, entre otros valores. Eso permitió poder obtener escalas relacionadas con el entorno. La investigación se realizó en el *Laberint d'Horta* de Barcelona.

Después de tener claras las medidas en el mundo real y en el papel tener las proporciones adecuadas, se realizó un diseño del escenario mediante cajas. Una vez se tiene este diseño inicial, se empieza a dar forma al escenario modificando la forma de las cajas, así como los agujeros para el río y del

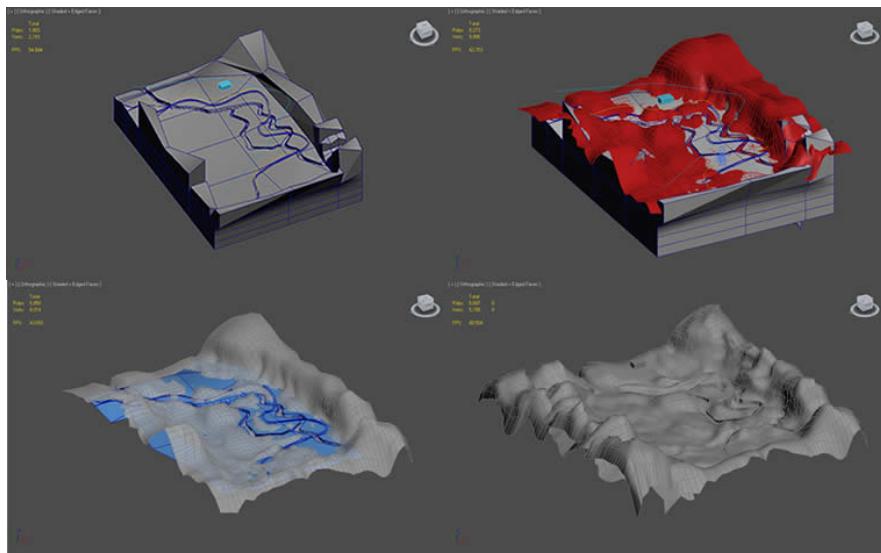


Figura 3.3: Proceso de creación del escenario (de izquierda a derecha y de arriba a abajo)

camino. Seguidamente se empieza el modelado de las montañas teniendo en cuenta las medidas iniciales. Las dos partes, las montañas y el terreno llano, se modelan separadamente. Finalmente, y una vez obtenido un buen resultado para las montañas y el terreno, se unieron para formar un solo escenario en una sola pieza (figura 3.3).

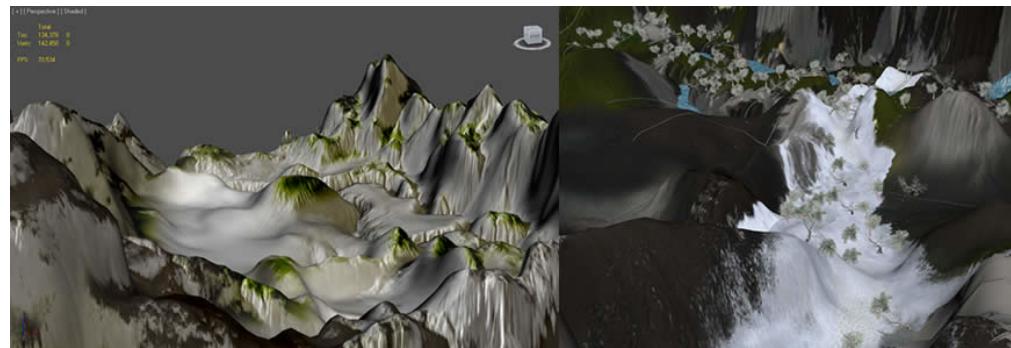
### Texturización

Una vez modelado el escenario y probado en el motor del juego con una textura común, se pasó a la realización de la textura.

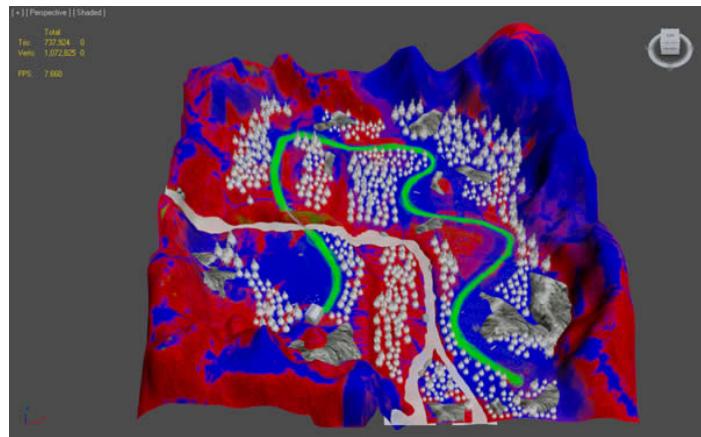
La primera prueba fue una sola textura de 4096x4096 para todo el terreno (figura 3.4). A simple vista el resultado era bueno, pero había sitios, sobretodo donde el terreno era más irregular, donde se notaban los fallos de textura. También se apreció que el terreno no era lo suficiente nevado, tal y como se tenía la idea original.

La siguiente prueba fue dividir el terreno en diversas secciones, teniendo una textura por cada sección. La idea parecía buena dado que así se podría tener más resolución de textura en el escenario. El trabajo se complicó en este punto. Una vez en el motor del juego, las juntas que dividían las secciones se veían y no era un trabajo trivial hacerlas desaparecer.

Finalmente, la solución fue utilizar la técnica de texturizado *splat mapping*. Esta técnica permite tener una textura que define el terreno, donde cada canal de esta defina la cantidad de textura a aplicar de una textu-



*Figura 3.4: Textura de 4096x4096 aplicada como primera prueba)*



*Figura 3.5: Escenario en 3ds Max con la textura de definición del terreno)*

ra concreta. Así pues, con una textura donde se define el texturizado del terreno y cuatro texturas, se logró texturizar todo el terreno (figura 3.5).

### Iluminación

En el documento de diseño se comentó que la luz sería de noche, pero rápidamente se cambió a atardecer, ya que podríamos jugar con el Sol y algunos efectos que este provoca.

Se buscaron referencias de atardeceres y se vió que hay atardeceres con diferentes tonalidades. Unos atardeceres son más amarillentos y anaranjados, y otros más rojizos. Así que se probaron en el motor estas diferentes tonalidades de iluminación. Finalmente, la balanza se decantó más por una tonalidad rojiza con un poco de amarillo (figura 3.6).

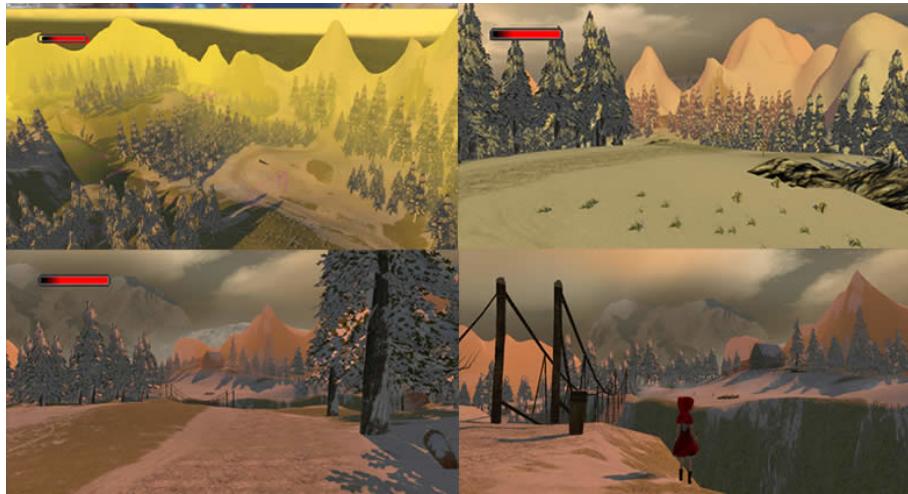


Figura 3.6: Iluminación del escenario. Arriba izquierda: tonalidad amarillenta. Arriba derecha: Amarillenta con rojizo. Debajo: Tonalidad final rojiza)



Figura 3.7: Textura del skydome final

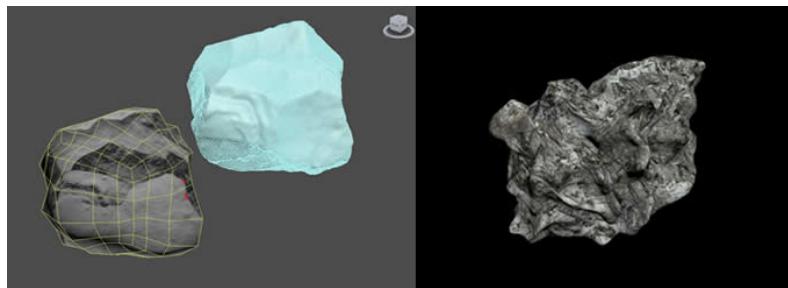


Figura 3.8: roca en *high poly* y *low poly*

### Skydome

Para dar sensación de estar en el mundo real, se ha aplicado un *skydome*. Primeramente se creó una textura sin montañas, pero no daba sensación de inmersión, así que se añadieron montañas y se aplicó una corrección de color para ir en consonancia con la iluminación escogida (figura 3.7).

### Props

Para el escenario se han diseñado diferentes elementos que conforman el paisaje, como árboles, rocas, plantas. También se ha creado un puente de madera y cuerda por donde el jugador accede a la zona final del juego.

Los elementos fueron creados en *high poly* primero para después, pasarlo a *low poly* y así poder poner más elementos en el escenario 3.8.

#### 3.1.2. Personajes

El diseño de los personajes (figura 3.9) fue sacado de varios universos, pero sobre todo se ha intentado que los personajes tengan un estilo realista dentro de la ambientación. Se ha buscado siempre que creen una ambientación visual coherente y armónica.

### Caperucita

Ficha técnica:

- Nombre: Caperucita
- Tipo: Personaje jugable / Protagonista
- Altura: 1.50m
- Polígonos: 6185 (7800 con las armas)



*Figura 3.9: Todos los personajes del juego*

- Huesos: 36 + 4 huesos independientes
- Animaciones: 12

Caperucita Roja es el personaje principal del juego, la protagonista del cuento de Charles Perrault. Es el personaje más complicado que se ha creado ya que es el modelo animado que lleva el jugador y, por lo tanto, el que más se ve. Antes de empezar la creación de Caperucita se han buscado muchas referencias para inspirar el arte (figura 3.10).

El diseño de Caperucita (figura 3.11) requería mucha reflexión porque se le añadió una falda, un trabajo complicado en el momento de animar el personaje. La falda tiene cuatro huesos independientes asociados, que permiten la animación de la ropa. Estos huesos se han unido al *mesh* del personaje con un sistema de resorte (*Spring System*), una opción de 3ds Max que permite a huesos independientes seguir el movimiento de un bípedo y rebotar en función de la fuerza de impacto de las animaciones del personaje. Con esta técnica aplicada, se ha conseguido que la falda siga al cuerpo con un movimiento harmónico.

Las armas que lleva el jugador están unidas a la misma malla de Caperucita por motivos de rendimiento. Así pues, la textura de las armas también está en el mismo fichero que la textura del personaje.

El personaje tiene textura de normales y oclusión.

Las animaciones creadas para Caperucita son:

- Idle. Posición estática del personaje respirando.
- Run. Movimiento de correr.



Figura 3.10: Referencias de arte sobre Caperucita

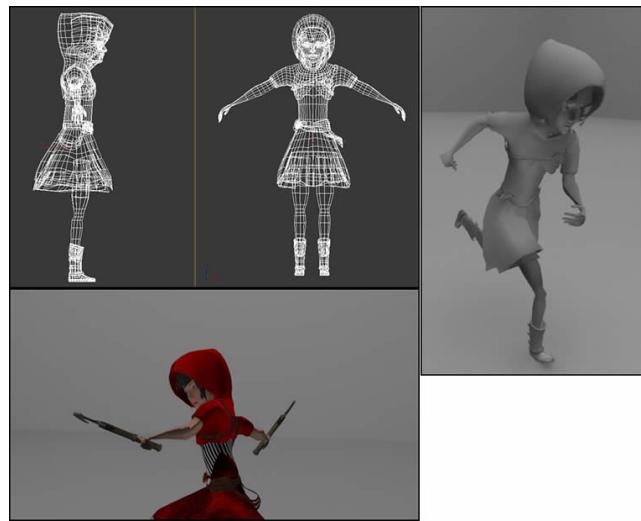


Figura 3.11: Proceso de creación de Caperucita

- Jump. Salto hacia adelante.
- Hit. Cuando se recibe un golpe.
- Defense. Postura de defensa.
- Combo de ataque 1. Secuencia de tres golpes con las armas. Se termina con un golpe potente horizontal.
- Combo de ataque 2. Secuencia de tres golpes. Los golpes son más potentes. El combo finaliza con un ataque circular devastador.

### El Lobo

Ficha técnica:

- Nombre: The Wolf
- Tipo: Enemigo / Enemigo final
- Altura: 3.25m
- Polígonos: 2226
- Huesos: 28
- Animaciones: 9

El Lobo fue el primer enemigo que se creó para el juego. Se ha exagerado el tamaño respecto al tamaño de Caperucita para que parezca gigantesco y dar sensación de miedo y ansiedad en el combate final.

Este enemigo fue diseñado con pocos polígonos y huesos ya que al ser el primer enemigo, sirvió también como prueba técnica para el motor (figura 3.12).

Su diseño está inspirado en dibujos y fotografías de criaturas de hombres lobo (figura 3.13). Se han hecho varios esbozos para concluir en un resultado final, un enemigo bípedo y corpulento.

El personaje tiene textura de normales y oclusión.

Las animaciones creadas para el Lobo son:

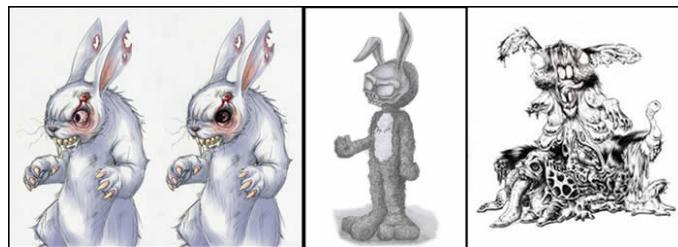
- Idle. Posición estática del personaje respirando pesadamente.
- Walk. El personaje camina.
- Run. El enemigo corre.
- Hit. Recibe un golpe.
- Die. El Lobo se cae y muere.



Figura 3.12: Proceso de creación del Lobo



Figura 3.13: Referencias para el Lobo



*Figura 3.14: Referencias para el conejo*

- Defense. Postura de defensa.
- Howl. Aulla para llamar a más enemigos.
- Attack / Combo. Ataque del personaje.
- Attack / Run. Ataque corriendo.

## Conejo

Ficha técnica:

- Nombre: Bad Rabbit
- Tipo: Enemigo ligero
- Altura: 1.25m
- Polígonos: 2672
- Huesos: 34 + 4 huesos independientes
- Animaciones: 10

El conejo es un enemigo ligero y rápido, con ataques ágiles pero no muy fuertes. Se han sacado referencias de varios universos para crear el personaje (figura 3.14). La idea del personaje es darle un aspecto siniestro y demencial, un conejo diabólico. Se caracteriza por su postura curva y rota. Con este aspecto de locura y con la agilidad que tiene, se intenta transmitir una sensación de agobio al jugador.

Está diseñado con más polígonos y huesos que el Lobo (figure 3.15). Se añadieron cuatro huesos independientes a las orejas para conseguir más realismo en los movimientos de la cabeza. Estos huesos son necesarios para crear el *overlapping* en las animaciones.

El personaje tiene textura de normales y oclusión.

Las animaciones del conejo son:

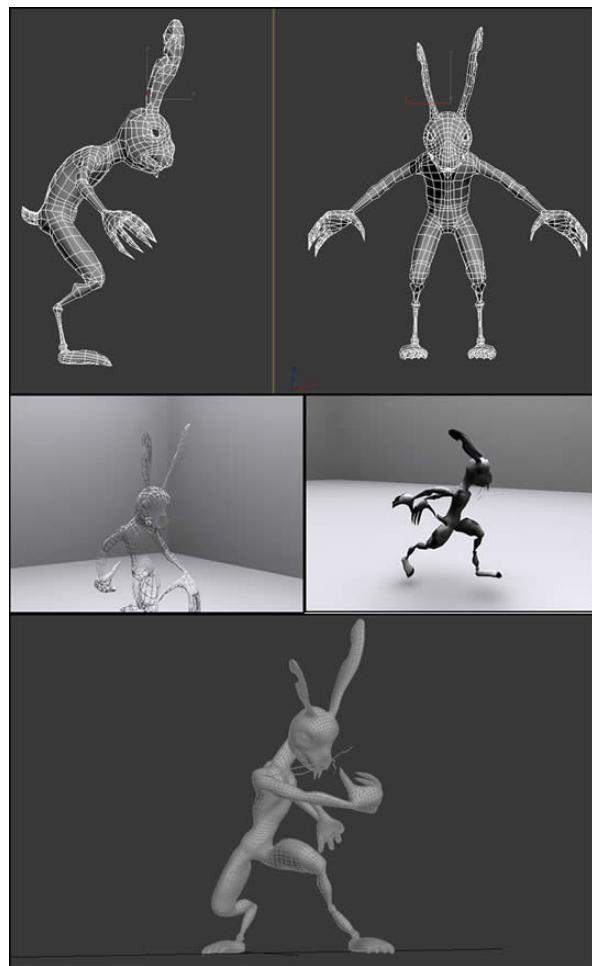


Figura 3.15: Proceso de creación del conejo

- Idle. Posición estática del personaje respirando.
- Walk. El personaje camina.
- Run. El enemigo corre.
- Pasos laterales. El modelo animado se desplaza lateralmente.
- Jump. Un pequeño salto.
- Hit. Recibe un golpe.
- Die. Se cae y muere.
- Defense. Postura de defensa.
- Attack 1. El conejo hace un golpe directo con las uñas
- Attack 2. El golpe es circular con las uñas.

## Ciervo

Ficha técnica:

- Nombre: Bad Deer
- Tipo: Enemigo pesado
- Altura: 2m
- Polígonos: 2701
- Huesos: 33
- Animaciones: 8

Este personaje ha sido más complejo de diseñar dado que no existen muchas referencias de enemigos similares (figura 3.16). Se hizo una mezcla de varias referencias para obtener el resultado final. La idea general en el diseño del ciervo era la de tener un enemigo más grande que el conejo, más pesado pero a la vez, mucho más destructivo. Aunque tenga una gran altura, su delgada corpulencia le permite moverse con velocidad. Igual que el conejo, se le quería dar un aspecto demencial, extraño y deformado, como por ejemplo su tamaño distinto para sus ojos.

El personaje tiene textura de normales y oclusión. Para dar efecto de pelo, se mezcló dos colores en la textura para así intentar imitar la piel de los animales y para dar la sensación de locura. No se le ha añadido ningún hueso independiente tal y como ha pasado con otros personajes (3.17).

Las animaciones del ciervo son:



Figura 3.16: Referencias para el ciervo

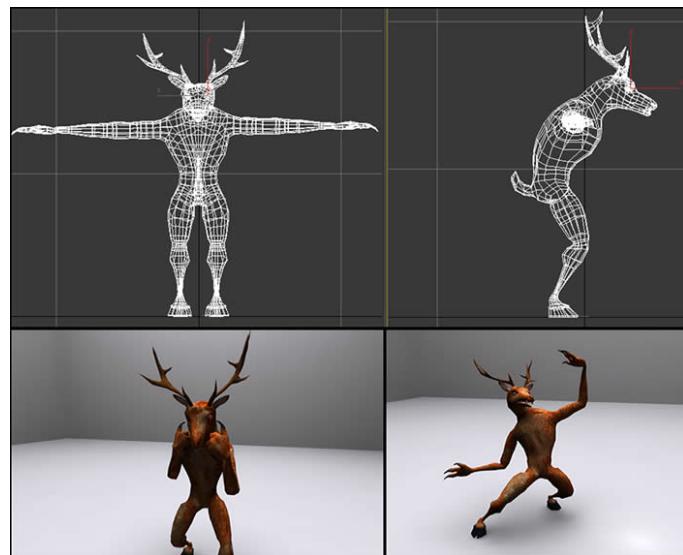


Figura 3.17: Proceso de creación del ciervo

- Idle. Posición estática del personaje respirando pesadamente.
- Walk. El personaje camina.
- Run. El enemigo corre.
- Hit. Recibe un golpe.
- Die. El ciervo muere.
- Defense. Postura de defensa.
- Attack Punch. Dos golpes con los brazos.
- Attack Run. El personaje baja el torso y carga su presa con sus cuernos.

## 3.2. Programación

*Boc Engine* es el motor de videojuegos desarrollado en C++ para el máster y el juego. Este motor ha sido creado tanto con tecnología propia como de terceros, que nos ha facilitado tener una base sólida para poder construir todas las partes que componen el videojuego.

*Boc Engine* ha sido dividido en diferentes proyectos, cada uno especializado en un ámbito distinto (gráficos, sonido, GUI,...) pero interconectados entre sí.

La tecnología de terceros utilizada en el motor es la siguiente:

- Boost. Utilizado para fines varios.
- Cal3D. Para las animaciones de los modelos animados.
- Lua/Luabind. Utilizado como lenguaje de *scripting* en el juego.
- PhysX. Utilizado para el motor de físicas del juego.
- Theora-Ogg / OpenAL. Para la reproducción de ficheros de video.
- Wwise. Para la reproducción de sonido.
- XML. Para guardar toda la información del juego.

Toda la estructura del motor, y del juego en sí, está basada en ficheros XML. El lenguaje XML permite externalizar los parámetros del motor, así como también guardar toda la información referente al juego.

Como entorno de desarrollo utilizado ha sido Microsoft Visual Studio 2010 para todo el motor y Notepad++ para editar ficheros XML y Lua.



Figura 3.18: Escena final con las sombras

### 3.2.1. Gráficos

Los aspectos donde se ha hecho más enfasi durante la programación de los gráficos a sido con la iluminación, las sombras, las partículas y los efectos de post procesado.

#### Iluminación y sombras

La iluminación del juego se ha hecho con *forward shading*. Se ha implementado la luz con esta técnica porque en el juego hay una gran cantidad de elementos con transparencia y sólo una luz direccional que representa el Sol. Esta técnica es más eficiente con transparencias y pocas luces.

Las sombras (figura 3.18) se han hecho mediante la combinación de dos técnicas, *Variance Shadow Mapping* [17] y *Cascade Shadow Mapping* [18].

La técnica del *Variance Shadow Mapping* ha sido utilizada para crear las sombras. Para conseguir sombras detalladas en todo el escenario, se ha hecho con *Cascaded Shadow Mapping*. Esta técnica crea mapas de sombras según la distancia del jugador. Así, las sombras más próximas al jugador tienen más detalle que las más alejadas, pero al tener gran distancia no se percibe el cambio de calidad. Para el juego usamos tres mapas de sombras a diferentes distancias (figura 3.19 y 3.20).

#### Partículas

El sistema de partículas diseñado se conforma de cuatro bloques principales: *Settings*, *systems*, *emitters* y *shader*.

- *Settings*. Define el comportamiento de unas partículas. Tiene la información específica de como tiene que funcionar un sistema de partículas.



Figura 3.19: Mapas de sombras creados con Cascaded Shadow Mapping

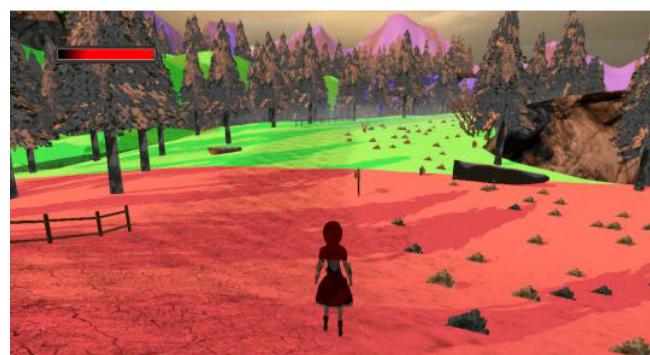


Figura 3.20: Diferentes mapas de sombras aplicados a la escena. El rojo es el mapa con más calidad y el azul el que tiene menos calidad.

- *Systems*. Es el responsable de mantener un orden sobre qué partículas se tienen que renderizar y cuáles no. También informa al *shader* sobre que textura utilizar y cuantas partículas dibujar.
- *Emitter*. Se asocia con un sistema de partículas y define en que posición del mundo deben ejecutarse las partículas.
- *Shader*. Se encarga del renderizado principal de las partículas, así como también de los cálculos asociados a las mismas, como por ejemplo la velocidad o la posición de una partícula concreta, entre otros parámetros.

### Post procesado

Los efectos de post procesados son aquellos que se aplican una vez se ha renderizado toda la escena con la iluminación. Se han aplicado tres efectos de post procesado.

- *Motion Blur* [19]. El objetivo de este efecto es hacer un difuminado en los objetos cuando la mirada del jugador se mueve rápidamente. La técnica transmite el jugador sensación de velocidad y hace una semejanza con el mundo real.
- *Z-Blur*. El efecto provoca un enfoque de los objetos cercanos al jugador, i difumina los objetos más alejados.
- *Lens Flare* (figura 3.21). Al mirar el cielo en dirección al Sol con una cámara, estas tienen un reflejo. Esta técnica imita este efecto provocados por las lentes de las cámaras.

### 3.2.2. Física

En el apartado de física se ha utilizado la librería PhysX [16] de Nvidia. Con esta librería se ha desarrollado toda la parte de colisiones y interacciones con los diferentes elementos que conforman el juego, tanto el escenario como los personajes animados.

Para ello se ha creado un *wrapper* para encapsular toda la parte de la librería útil para el proyecto, adaptándolo a nuestras necesidades. Con la encapsulación, nos ha permitido desarrollar todos los movimientos físicos y colisiones entre personajes y los diferentes elementos estáticos o interactivos del escenario.

Con la librería de física también se ha hecho uso de los *triggers* físicos que incorpora, aunque el *trigger* lógico, donde hay todo el código de *gameplay*, está hecho mediante Lua.



Figura 3.21: Efecto de lens flare en el juego

Se ha utilizado los rayos que permite PhysX para la gestión y el control de la cámara principal del juego. Así, mediante testeo de rayos, podemos saber si hay objetos que obstaculizan la visión de la cámara y, tomar las decisiones necesarias para que siempre se vea el jugador y no quede oculto por diferentes objetos.

Las áreas de colisión también han sido utilizadas para el comportamiento de los enemigos. Las áreas de colisión nos permiten saber si hay enemigos cerca de otro enemigo, y así el personaje no jugador poder tomar las decisiones oportunas en su comportamiento.

### Escenario de pruebas

Debido a la gestión por procesos de toda la aplicación, se ha creado una pequeña escena de test (figura 3.22) para realizar el correcto desarrollo y testeo de todas las físicas del juego.

En este escenario de pruebas se han podido probar situaciones como:

- Muros de colisiones.
- Controladores (*controllers* en PhysX) para los personajes animados.
- Escaleras con diferentes alturas de peldaños.
- Elementos dinámicos como, por ejemplo, pelotas.
- Objetos fijos al escenario y objetos dinámicos.
- Colisiones entre diferentes grupos de colisión.

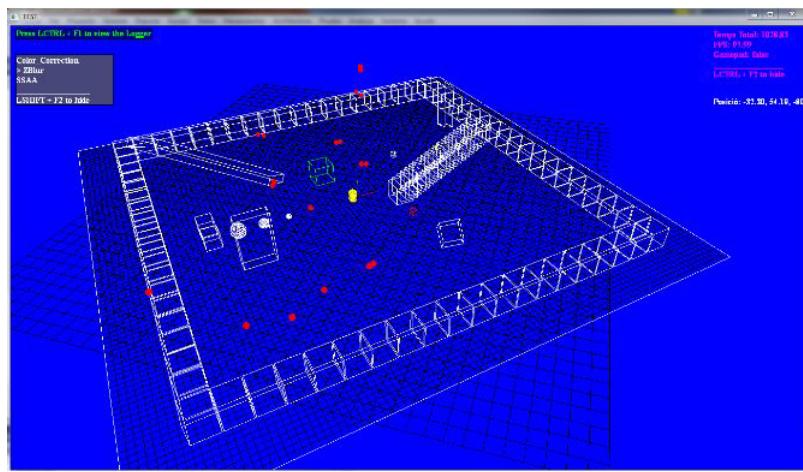


Figura 3.22: Escena de test de físicas

- Puente dinámico.

Una de las pruebas más exhaustivas a hacer ha sido la reacción del jugador delante de todas las situaciones físicas. De esta manera, se ha podido configurar posteriormente para el juego final.

Otra de las pruebas más grandes ha sido crear un puente dinámico, es decir, que reaccione al paso del jugador por encima de él. Finalmente, el puente no se ha incluido en el juego.

### 3.2.3. Inteligencia Artificial

Para el juego de *hack 'n' slash* inicialmente se planteó una sencilla inteligencia artificial. Durante el desarrollo del juego, cada vez se le fue dando más importancia a otorgar a los enemigos un comportamiento lógico mostrando que se comportan de una forma inteligente.

Para generar esos comportamientos inteligentes se han utilizado las técnicas de máquinas de estados y los *steering behaviors*.

#### Máquinas de estados

Cada personaje está gestionado por dos máquinas de estados diferentes. Por una parte, una máquina de estados lógica donde se gestionan los diferentes estados relacionados con el comportamiento de los personajes. Por otro lado, una máquina de estados gráfica que permite la gestión de todos los estados animados, animaciones, de forma cómoda y sencilla.

El motivo de tener dos máquinas de estados diferentes es por facilitar el desarrollo, permitiendo, además, poder gestionar con un estado lógico varios

estados gráficos. Por ejemplo, por un estado de reposo del personaje, poder tener diferentes animaciones de *idle* a ejecutar.

El uso de la máquina de estados también nos permite commutar entre el estado actual de un personaje con sus estados anteriores, facilitando así las commutaciones entre ellos y poder tener un registro de lo que ha hecho y qué hará.

### Steering Behaviors

En los juegos, el objetivo principal de los enemigos es alcanzar al jugador. Por este motivo y para permitir crear los movimientos de los enemigos, se ha implementado *steering behaviors*.

Los *steering behaviors* son un conjunto de algoritmos de inteligencia artificial que simulan el comportamiento de movimiento de agentes para que parezcan autónomos y permiten tomar decisiones según la interacción con los demás elementos del entorno. Estos comportamientos dan cierto grado de flexibilidad y facilidad para permitir mover los diferentes enemigos por el escenario evitando las obstrucciones, sean móviles o no.

Los comportamientos implementados son básicos y permiten ir a una posición determinada, perseguir a un objetivo, evitar colisiones, mantener cierta separación con el resto de enemigos,...

La flexibilidad de estos comportamientos recae en la posibilidad de añadir o quitar comportamientos con mucha facilidad. Los comportamientos pueden combinarse a la vez, así por ejemplo, poder perseguir al jugador y al mismo tiempo mantener cierta separación con el resto de enemigos para así, poder realizar ataques fácilmente.

La filosofía de los *steering behaviors* que se han diseñado es de obtener una fuerza, un vector resultante de ese comportamiento, y finalmente sumarlos todos evitando sobrepasar siempre una fuerza resultante máxima. Los pesos permiten indicar qué cantidad de esa fuerza se aplica. Finalmente se aplican unos límites para que los personajes no se muevan de forma brusca o llegando a sobrepasar el escenario.

La gran desventaja de estos comportamientos es la combinación de ellos, los cuales pueden producir movimientos no muy adecuados para realizar algún ataque o simplemente dirigirse contra el jugador. Para solucionar el problema, se ha utilizado pesos para los comportamientos para determinar el grado de importancia y así permitir priorizar unos comportamientos u otros.

Otro de los problemas que hay estos algoritmos es ajustar todos los parámetros que estos procesan de manera que el movimiento de los agentes sean claros y correctos.

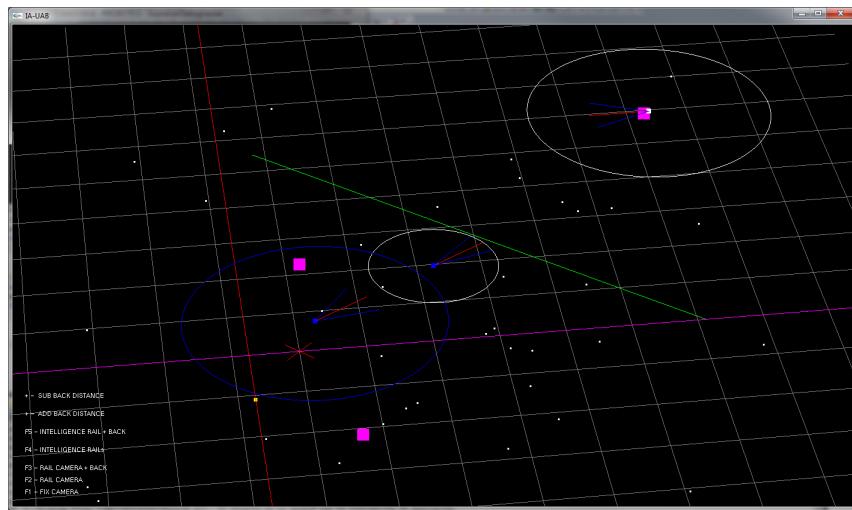


Figura 3.23: Escena de test de de inteligencia artificial

### Escenario de pruebas

Para hacer pruebas de inteligencia artificial se ha desarrollado un pequeño escenario de pruebas (figura 3.23). Para agilizar la programación del escenario se ha programado con OpenGL.

En el escenario se probó diferentes comportamientos para los enemigos, como por ejemplo perseguir, escuchar o proximidad entre enemigos. También se hicieron pruebas de cámaras inteligentes funcionando con raíles, para ver como podría funcionar en el juego.

#### 3.2.4. Scripting

El lenguaje de *scripting* utilizado para el juego ha sido Lua [20]. Ya que la gran desventaja del lenguaje es no trabajar con clases, se ha utilizado como Luabind [21] como solución a la encapsulación de clases en Lua.

Con este lenguaje de *scripting* se ha implementado parte del *gameplay* del juego, como por ejemplo los *triggers*.

#### 3.2.5. Audio

Para la gestión de sonido y música se ha utilizado la librería Wwise [15] de Audiokinetic. La librería es de pago menos para Windows que es gratuita si el producto final es de uso no comercial. Wwise se utiliza en multitud de juegos actuales, entre ellos *Mass Effect 3* [5], *Assassin's Creed II* [6], *inFamous 2* [7] y *Batman: Arkham City* [8], entre otros.

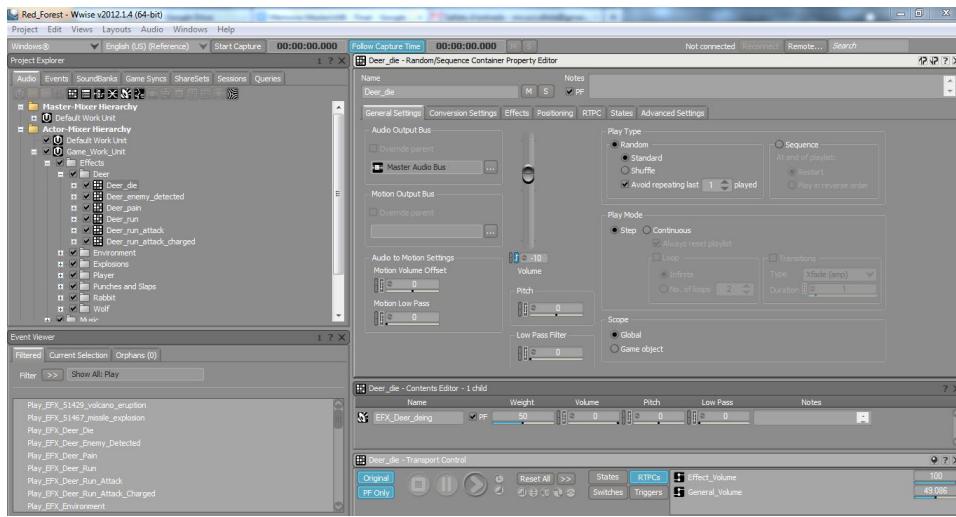


Figura 3.24: Editor de Wwise con el proyecto del juego

Wwise funciona mediante eventos y bancos de sonido. En los bancos de sonido es donde residen todas las pistas de audio y los eventos son instrucciones sobre las pistas de audio. La creación de estos dos elementos se hace mediante el editor que lleva la propia librería (figura 3.24).

También se ha hecho uso de los RTPCs (*Real-Time Parameter Controls*). Los RTPCs son curvas asociadas a un audio que, a partir de un parámetro, se puede modificar el sonido en tiempo real. Se han utilizado para poder modificar el volumen del sonido y de la música en cualquier momento.

### 3.2.6. Tools

Se han hecho diferentes herramientas para el fácil y ágil desarrollo de algunas partes del juego. Así pues, se han hecho la herramientas: Particle Editor, GUI Editor y 3ds Max Exporter.

#### Particle Editor

Durante el desarrollo del juego, se ha llevado a cabo un simple editor de partículas (figura 3.25) para poder generar las partículas de una forma más ágil, rápida y visual. Está hecho con MFC (*Microsoft Foundation Class*).

El editor permite modificar todos los parámetros de un sistema de partículas en tiempo real. Al mismo tiempo, se puede ejecutar el sistema de partículas creado en diferentes tipos de emisores (tipo punto, caja, esfera, anillo, línea).

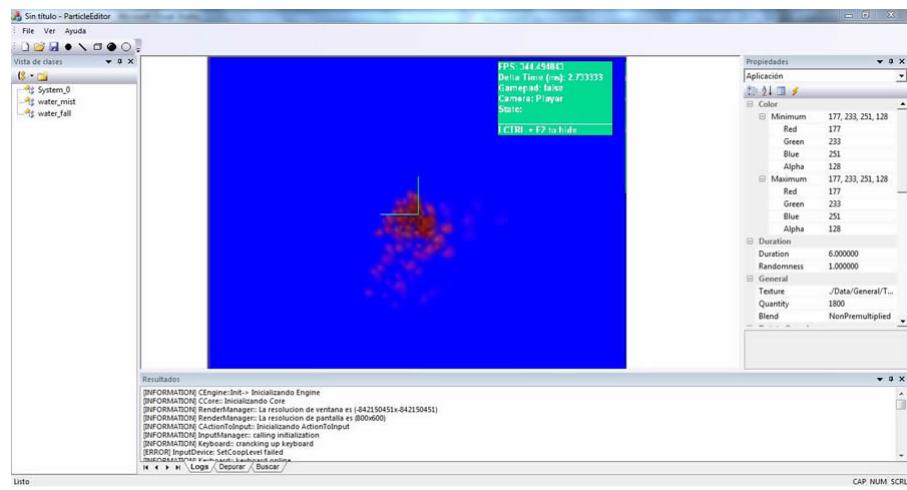


Figura 3.25: Particle Editor

La herramienta permite generar sistemas de partículas distintos y guardarlos en un fichero XML preparado para ser leído directamente por el motor del juego.

### GUI Editor

Se ha desarrollado un editor (figura 3.26) para poder hacer los menús del juego de una forma sencilla y rápida. Como el *Particle Editor* (sección 3.2.6), está hecho con MFC.

El editor permite poner todos los controles de los que dispone (botones, imágenes,...) mediante *drag and drop*. Además, se puede modificar las propiedades de todos los controles que conforman la ventana, pudiendo personalizar totalmente la ventana de una forma visual.

La herramienta permite generar ventanas para trabajar con toda la GUI del juego a la vez y, también, guardar toda la información en ficheros XML que son leídos por el motor.

### 3ds Max Exporter

Como Autodesk 3ds Max se ha utilizado como editor de niveles ha sido conveniente hacer una herramienta que pudiera exportar toda la información referente al escenario. Así pues, esta herramienta tiene las siguientes funcionalidades:

- Exportar toda una escena, o una parte de ella.
- Definir los tipos de elementos que conforman la escena.



Figura 3.26: GUI Editor

- Definir y exportar toda la información sobre las luces del escenario.
- definir y exportar raíles para cámaras.
- Definir la física de los objetos y exportarla.
- Definir y exportar las texturas del terreno.

### Viewer

Se ha desarrollado un visor de elementos. Este visor permite cargar modelos, tanto estáticos como animados, y así poder apreciar como se ve dentro del motor. Esta herramienta está especialmente diseñada para los artistas para que pudiesen probar los modelos generados al momento. Así se agiliza el trabajo.



## **Capítulo 4**

# **Resultados**



## Capítulo 5

# Conclusiones

- Se ha diseñado e implementado un juego *hack 'n' slash* sobre el cuento de Caperucita Roja, creando una reproducción del cuento distinta en una ambientación oscura y *gore*.
- Se ha creado una inteligencia artificial pensada y orientada al combate cuerpo a cuerpo, teniendo en cuenta que cada personaje con inteligencia no está solo en el escenario sino con más entidades com él.
- Se ha diseñado un escenario grande para el combate, con sensación de estar en un bosque nevado abierto, pero al mismo tiempo estar limitado a deambular por zonas concretas, limitando así todo el acceso al bosque.
- Se ha implementado un motor de juego genérico con tecnologías que se utilizan actualmente en el desarrollo de videojuegos, como por ejemplo Direct, Wwise o PhysX.
- Se han encontrado problemas en el momento de desarrollar con Lua / Luabind. Al crear una clase de Lua en C++ que esta heredaba de código de C++, en algún momento aleatorio de la ejecución, Lua perdía esas referencias haciendo que el juego dejara de funcionar. Se solucionó escribiendo todo el código de nuevo en C++.
- Se han encontrado problemas en la reserva de memoria de Lua. Siempre se iba reservado más y más memoria hasta agotar toda la memoria del ordenador en que se ejecutaba el juego. Se solucionó pasando el código de Lua a C++.
- Se han encontrado bastantes problemas con el exportador de Cal3D y 3ds Max 2012. El exportador muchas veces o, dejaba el programa de modelado inutilizado, o generaba ficheros corruptos, modificando así toda la malla que se exportaba.

- El escenario se ha tenido que rehacer más de una vez por motivos de diseño, ya que las versiones anteriores no cumplían los objetivos de jugabilidad.
- Al no tener un editor de niveles completo, la ubicación de los árboles se hizo manualmente, lo cual resultó ser una tarea compleja y lenta, gastando bastante tiempo.
- Animar un personaje con ropa es mucho más complejo de lo esperado, y tomó más tiempo e inclusive modificar el diseño original del personaje principal.

# Bibliografía

- [1] <<http://www.ea.com/alice>>, Alice Madness Returns: American McGee's Alice 2 - EA Games. Página web del juego Alice Madness Returns.
- [2] <<http://www.playlogicgames.com/game/fairytale-fights>>, Playlogic Games - Fairytale Fights. Página web del desarrollador del juego Fairytale Fights.
- [3] <[www.godofwar.com](http://www.godofwar.com)>, God of War. Página web del juego God of War.
- [4] <<http://www.darksiders.com/>>, Darksiders II. Página web del juego Darksiders.
- [5] <[www.masseeffect.com](http://www.masseeffect.com)> Mass Effect. Página web oficial del juego Mass Effect.
- [6] <<http://assassinscreed.ubi.com>> Assassin's Creed. Página web oficial del juego Assassin's Creed.
- [7] <[www.infamousthegame.com](http://www.infamousthegame.com)> inFAMOUS 2. Página web del juego inFamous 2.
- [8] <<http://community.batmanarkhamcity.com>> Batman: Arkham City. Página web oficial del juego Batman: Arkham City.
- [9] <<http://windows.microsoft.com/es-ES/windows/home>>, Microsoft Windows. Página web oficial del sistema operativo Windows de Microsoft.
- [10] <<http://es.playstation.com/ps3/>>, PlayStation 3. Página web oficial de la consola PlayStation 3 desarrollada por Sony.
- [11] <<http://www.xbox.com>>, Xbox.com. Página web oficial de la consola Xbox 360 desarrollada por Microsoft.



# Apéndice A

## Manual de usuario

**Resum**

**Resumen**

**Abstract**