

# Sunken Dream, the adventure game

Javier Blanco Martinez

**Resumen**— Sunken Dream es un proyecto que trata de la creación de un videojuego de aventuras en 2 dimensiones usando la tecnología flash, lo que le permite ser ejecutado en diferentes dispositivos desde móviles hasta navegadores de PC convencionales. Este videojuego consiste en una historia dividida en diferentes niveles, el cual conforman entre todos un gran rompecabezas que el jugador tendrá que resolver mediante el uso de diferentes objetos o interactuando con eventos del mapa para averiguar cuál es el misterio que esconde. Este juego va dirigido principalmente a casi todos los públicos preferentemente mayores de 13 años por su complejidad, no a la hora de la jugabilidad ya que la interfaz de la que dispone es muy intuitiva si no a la hora de resolver los enigmas que esconde. Para su realización se ha utilizado la metodología de desarrollo del software SUM, la cual se explicará en este documento.

**Palabras clave**—juego, juego de aventuras, juego de plataformas, flash, Action Script 3, Starling, Citrus Engine

**Abstract**— Sunken Dream is a project that of an adventure game in 2D using the flash technology, which allows it to be executed on different devices from mobile browsers to conventional PC. This game is divided into different levels, which form together a large puzzle that the player must solve by using different objects or interacting with events of the map to figure out the mystery behind the history. This game is mainly aimed at almost every public preferably greater than 13 years old because of its complexity, not for the game play since the available interface is very intuitive but for resolving the mystery behind it. For its realization the software development methodology used was SUM, which will be explained in this document.

**Index Terms**—game, adventure game, platform game, flash, Action Script 3, Starling, Citrus Engine



## 1 INTRODUCCIÓN

EN esta sección encontrará una breve descripción de la situación actual, la estructura general de este documento y el objetivo principal de este proyecto.

### 1.1 La situación propuesta

La evolución de las tecnologías para el entretenimiento hoy en día es algo esencial para cubrir las necesidades humanas, dentro de esta rama cabe destacar los videojuegos, en este campo podemos encontrar inmensas variedades de géneros desde aventura[1] hasta plataformas [2] pasando por acción. Hoy en día también la necesidad de que estos videojuegos puedan ser multiplataforma, hace esta característica obligatoria para casi todos los juegos.

#### 1.1.1 Áreas de interés del proyecto

Este proyecto se centra básicamente en personas que les apasione los juegos de aventura, principalmente para personas de entre los 13 años hasta los 20 años. Sunken Dream pretende ser un juego algo complejo de resolver pero a la vez interesante, por esa razón no puede ir dirigido a personas que se cansen o se den por vencidas fácilmente.

#### 1.1.2 Requerimientos de aplicabilidad

El video juego consta de siguientes requerimientos:

- Planteamiento sencillo: la historia tiene que ser

muy simple, una mera excusa para el desarrollo del juego pero lo suficientemente explícita para que el Jugador sienta que tiene un objetivo.

- Objetivo: El principal objetivo tiene que ser resolver los diferentes rompecabezas propuestos en cada mapa y abatir enemigos para conseguir avanzar.
- Dinamismo: al contrario que algunos juegos de aventura, Sunken Dream debe ser dinámico y provocar una sensación de tensión en el Jugador .
- Ampliación: Sunken Dream debe ser ampliable con nuevos niveles y enemigos de forma sencilla. El motor será todo lo independiente posible del contenido. De esta forma los artistas podrán generar nuevos niveles, habilidades o enemigos.
- Multiplataforma: Tiene que ser un videojuego capaz de funcionar en diferentes plataformas [2].

#### 1.1.3 Límites del proyecto

Este proyecto está limitado principalmente por las limitaciones técnicas de los hardwares, ya que al ir dirigido a muchos dispositivos distintos se ha de tener en cuenta todos y cada uno de ellos. Además de las restricciones técnicas también se trata de un proyecto muy limitado en cuanto al tiempo y al personal asignado, ya que solo es desarrollado por una persona.

### 1.2 Esquema del documento

El artículo contiene la parte de introducción donde se explica las limitaciones y los requisitos para el desarrollo

- E-mail de contacto: javicuba11@gmail.com
- Mención realizada: Ingeniería del Software.
- Trabajo tutorado por: Dr. Antonio López (departamento)
- Curos 2013/14

del proyecto, así como mostrar cuales son las áreas relacionadas con la temática del proyecto y también a los posibles Stakeholder al cual va dirigido dicho proyecto. Posteriormente se expone las metodologías, la planificación inicial del proyecto y su elaboración y después toda herramienta usada en la creación del videojuego, es decir, herramientas de diseño de programación, de control de versiones. También se detalla toda la parte del diseño de la interfaz, como el menú principal, los diferentes niveles, el diseño del protagonista y de los personajes, diseño de objetos y armas, los movimientos físicos y los efectos sonoros incorporados. Se define los cambios realizados respecto a la planificación inicial y toda la planificación final junto con todas las tareas que se han acabado de definir y su duración. Se expone el estado actual del proyecto, en el que se muestra todas las tareas acabadas y las tareas que se desean terminar en el futuro. Finalmente se concluye todo aquello que se ha extraído del proyecto.

## 2 OBJETIVOS DEL PROYECTO

En esta sección se explicará la situación actual del mercado centrándonos en los juegos más famosos del género o campo relacionado con este proyecto. Además se explicará las características que diferencian este proyecto de los demás juegos del género actuales.

### 2.1 Estado del arte

Existen muchísimos juegos relacionados con el género de aventuras [1] y plataformas [2], entre ellos se encuentran dos de los juegos más famosos de cada género y que aunque antiguos todavía se siguen liderando los top de su respectivo género, estos juegos son Super Mario [3] y Monkey island [4].

Super Mario [3] se ha convertido en uno de los clásicos del género de plataformas [2] y es que desde el tramo que va desde que pulsamos Start hasta que llegamos a la primera tubería ya sabemos que va a haber algo que nos va a enganchar a la pantalla durante horas.

Monkey island [4] es uno de los juegos más característicos del género de aventuras y no por su jugabilidad, ya que existen muchos juegos iguales que este, sí por su historia intrigante e interesante.

Este proyecto se ha basado en seguir los diseños y características de estos dos videojuegos.

### 2.2 Mejoras sobre el estado del arte

Aunque no se trata de un proyecto que mejora las aplicaciones o juegos existentes, por ser precisamente un videojuego, sí que hay una cierta mejora al unir estos dos tipos de géneros en uno solo juego. Normalmente los juegos de plataformas [2] suelen ser dinámicos, es decir el personaje solo va hacia delante superando obstáculos para llegar al final del nivel, pero para conseguir ese objetivo pueden haber diferentes maneras dentro de un mismo nivel. En cambio los juegos de aventura [1], normalmente son un poco más estáticos ya que, como siguen un guión de una historia, el jugador solo puede conseguir los objetivos del juego de una manera concreta.

El videojuego que se quiere conseguir con este proyecto tiene elementos de esos dos tipos de juegos, por lo tanto al jugador se le hace más interactivo ya que todo el videojuego se trataría de un nivel entero en el cual el jugador puede avanzar o retroceder a su gusto para resolver enigmas de la forma que el desee. Además a diferencia de los videojuegos convencionales, este juego dispone de actualizaciones constantes con nuevos niveles o trozos de la historia principal, por lo tanto cuando el jugador obtiene el juego no lo obtiene completo si no que solamente obtiene los niveles lanzados hasta la fecha. Esto permite cambiar el videojuego sobre la marcha e incluso alargar la historia si es necesario.

Finalmente y no por ser lo menos importante, crear un juego multiplataforma es muy importante para poder competir en el mercado actual por eso la mayoría de videojuegos ya lo son, por lo tanto este proyecto tendrá eso como prioridad a la hora de elegir el lenguaje de programación.

### 2.3 Objetivos cubiertos por la mejora del estado del arte

De momento con este proyecto, se ha cubierto la mejora de unir los dos géneros en uno solo, ya que el videojuego funciona como uno de plataformas [2] pero a su vez tiene elementos de los juegos de aventura, como el inventario.

En cuanto a la mejora de un videojuego interactivo el cual el jugador puede retroceder o avanzar a su gusto por los niveles, no se ha conseguido cumplir, básicamente por las razones explicadas en el apartado 5.1.

Finalmente el objetivo de hacerlo multiplataforma sí que se ha podido cumplir y por lo tanto el videojuego puede funcionar en diferentes dispositivos sin necesidad de adaptarlo.

## 3 METODOLOGÍAS Y HERRAMIENTAS

En esta sección se expondrá la metodología del software que se ha utilizado para conseguir llevar a cabo este proyecto y además las herramientas, tanto de diseño como de programación que se han utilizado.

### 3.1 Metodología seleccionada

Para este proyecto se ha decidido utilizar la metodología SUM [5] para videojuegos, que tiene como objetivo desarrollar juegos de calidad en tiempo y coste, así como la mejora continua del proceso para incrementar su eficiencia y eficacia. Aunque SUM [5] fuera concebido para que se adapte a equipos pequeños de dos a siete personas, también puede adaptarse para solo una persona.

Utilizando esta metodología, este proyecto se compondrá de las siguientes fases: planificación, elaboración o implementación, beta y finalmente cierre. A su vez estas fases se subdividen en varios pasos mostrados en el work

flow de la figura 1.

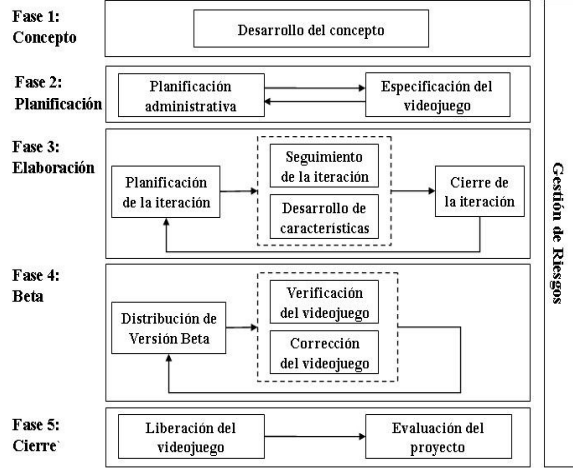


Figura 1

3.1.1 Concepto

En esta fase principalmente se han elegido las características principales del videojuego, como son: el género, el tipo de diseño, la ambientación, el género de la historia, las plataformas en las que puede estar disponible... Además se ha de investigar las diferentes herramientas y plataformas disponibles en el mercado y elegir las que más se adecuen a las necesidades de este proyecto.

3.1.2 Planificación inicial

Para la fase de planificación inicial ha sido necesario crear un cronograma del proyecto apéndice 1, del cual hablaremos en este mismo apartado y donde se indica el tiempo que se le dedicará a cada tarea, además de un diagrama de Gantt que se encuentra en el apéndice 2. Para recopilar tanto requisitos funcionales como requisitos no funcionales se ha realizado una encuesta online de selección múltiple además de una investigación sobre las características de los videojuegos del mercado. Esta encuesta ha sido realizada por más de 20 individuos de entre 15 y 23 años. Las respuestas de cada uno por cada pregunta han sido bastante similares, por lo que hace pensar que todos tienen un mismo modelo de juego en mente. Estas respuestas obtenidas han servido para recopilar las diferentes características que pueden ayudar a mejorar este videojuego. Un ejemplo de la encuesta realizada con las respuestas obtenidas sería el mostrado en la figura 2

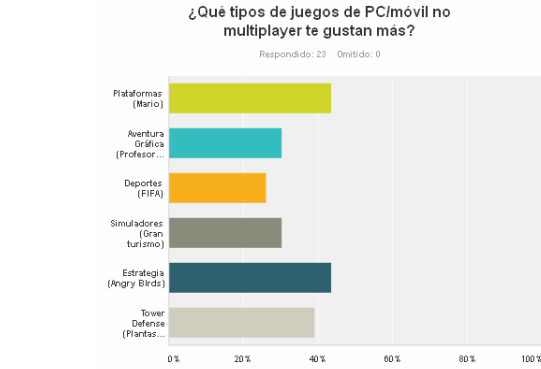


Figura 2

El cronograma inicial del proyecto se compone de las siguientes tareas:

- Formación previa, del (16/09/13) hasta el (30/09/13).
- Documentación del proyecto, del (16/09/13) hasta el (17/01/14).
- Prototipo inicial, del (01/10/13) hasta el (25/10/13).
- Diseño de personajes, del (21/10/13) hasta el (26/11/13).
- Primer nivel, del (26/11/13) hasta el (17/12/13)
- Segundo nivel, del (18/12/13) hasta el (01/01/14).
- Guardar y Cargar partidas, del (02/01/14) hasta el (10/01/14).
- Menú principal, del (01/11/13) hasta el (05/11/13).
- Pruebas finales, del (13/01/14) hasta el (17/01/14).

Desde esta planificación inicial han sucedido varios cambios hasta llegar a la planificación final explicada en el apartado 5.2 de este mismo documento.

Para finalizar la etapa de planificación se han realizado varios diagramas para facilitar el entendimiento del videojuego, como son un diagrama de casos de uso mostrado en el apéndice 2 y un diagrama UML con todas las clases necesarias para crear el juego. Además de un Game Desing Document donde se explica con detalle todas las características del juego como: enemigos, mapas, el flujo del juego etc.

3.1.3 Elaboración

En esta fase se ha dedicado exclusivamente implementar el videojuego. Para ello se ha de trabajar de forma iterativa e incremental para lograr una versión ejecutable en cada iteración. Cada una de las interacciones se compone de tres etapas: en la primera se planifican los objetivos a conseguir en la iteración, en la segunda se desarrollan las características y objetivos planificados y en la tercera etapa se realiza una evaluación de la iteración para hacer cambios, si es necesario, en el plan actual del proyecto. De esta manera se podía saber sobre la marcha si se podrían realmente cumplir los plazos planificados inicialmente.

3.1.4 Beta

En esta etapa se tiene como objetivo evaluar y ajustar distintos aspectos del juego como: jugabilidad, interfaz... Para ello se han realizado varias versiones "Beta" y

“Alpha” a lo largo del proyecto, de manera que un grupo de usuarios pueda hacer un “Exploratory Testing” o lo que es lo mismo un testeo manual y retornar un feedback de posibles cambios o mejoras que necesite el videojuego. Todo esto mediante una comunicación iterativa entre el desarrollador del proyecto y el usuario.

Durante el proyecto se han realizado varias de las pruebas comentadas anteriormente con usuarios de distintos perfiles, mediante esas pruebas se han obtenido resultados y información muy valiosa para la mejora del videojuego.

### 3.1.5 Cierre

Para la finalización del proyecto se ha de producir un versión ejecutable final que cumpla con todos los objetivos establecidos, además de hacer una evaluación del proyecto, estudiando todos los problemas ocurridos, los éxitos conseguidos y las soluciones halladas para mejorar los proyectos futuros.

Este proyecto, Sunken Dream, todavía no ha finalizado ya que faltan muchos objetivos por cumplir, por lo tanto no se tiene una versión final ejecutable del videojuego que cumpla con todos los objetivos establecidos.

## 3.2 Herramientas utilizadas

Para crear un proyecto de estas características o más en concreto un juego, es necesario el uso de muchas y diferentes herramientas, ya sea para el diseño de la parte artística del videojuego o para la programación del código necesario para el correcto funcionamiento. Por lo tanto en esta sección se explicarán las diferentes herramientas utilizadas para la realización de este proyecto así como el porqué se ha decidido utilizarlas y si hay algún tipo de limitación como la licencia de la misma.

### 3.2.1 Herramientas de diseño

Para el desarrollo del diseño de un videojuego es necesario distintas herramientas, ya sea para dibujar los personajes o decorado como para diseñar los niveles etc.

Entre toda la cantidad de herramientas de diseño que existe ya sea de pago como open source, las seleccionadas para este proyecto son: El programa de diseño de gráficos vectoriales de Adobe, Adobe Flash Profesional [6], Adobe Photoshop [7] para la edición de las imágenes, Spriter [8] para el montaje de los personajes, Tiled [9] para el montaje del mapa y finalmente Texture Packer [10] para el empaquetado de las texturas.

- Adobe Flash Profesional [6]: se trata de uno de los programas más famosos de gráficos vectoriales y animación. Aunque sea un software de pago se ha decidido utilizar en este proyecto porque facilita mucho el diseño de personajes a partir de un sketch en papel.
- Photoshop [7]: es uno de los software más famosos de Adobe para la edición de imágenes. Aunque sea de pago igual que el anterior, se ha decidido utilizar por la gran información que hay sobre su uso por la red.
- Spriter [8]: se trata de un software que permite la animación modular [11]. Aunque sea un software

muy nuevo se ha ganado un puesto muy importante entre los software de animación de personajes 2D. Esa razón y que sea open source han sido las dos razones que han hecho que me decante por este programa para el proyecto y no por otro.

- Tiled [9]: Se ha seleccionado este software por su facilidad de uso y su potencia a la hora de manejar tiles [12];
- Texture Packer [10]: Es uno de los software más utilizados para el empaquetado de texturas, por esa razón se ha seleccionado este entre otros, aunque no se tratase de un software totalmente Free.

### 3.2.2 Herramientas de programación

Las herramientas de desarrollo junto con las de diseño son la parte más importante en el desarrollo de un juego, sin unas buenas herramientas de programación o de diseño puedes perder grandes cantidades de tiempo que se podrían aprovechar para llevar a cabo otras tareas. Este es uno de los problemas que se ha tenido con este proyecto y que lo ha retrasado.

Inicialmente las herramientas de programación utilizadas eran:

- Adobe Flash Buldier [13]: Una herramienta para programar en el lenguaje AS3 [14] de flash, su limitación es que es de pago.
- Citrus Engine [15]: Se trata de uno de los motores de juego para flash que ha ido ganando fama últimamente, gracias a que es gratuito y dispone de mucha información para ayudar a los desarrolladores.
- Starling [16]: es el framework más famoso para desarrollar el núcleo de los videojuegos orientados a AS3 [14] o IOS, es utilizado en grandes videojuegos como Angry Birds [17] y además es 100% gratuito.

Por problemas de compatibilidad entre el framework y Adobe Flash Buldier [13] se perdió mucho tiempo intentando arreglar ese problema, cosa que obligo a hacer cambios en la planificación que serán explicados con más detalle en el apartado 5.1. Finalmente para solucionar estos problemas se cambió la herramienta Flash Buldier [13] a otra herramienta completamente gratuita llamada FDT [18].

Para finalizar, como hemos comentado anteriormente en este apartado el lenguaje de programación seleccionado para la realización del proyecto ha sido AS3 [14], ya que es uno de los más utilizados para crear videojuego multiplataforma, y precisamente por eso es del que se dispone más información.

### 3.2.3 Herramientas de Control de versiones

Para el desarrollo de un software una de las cosas que quizás sea la más importante es tener un buen control de versiones, ya que es la única manera en la que las empresas puedan proteger su producto final durante todas las

fases de dentro de su proceso productivo. Es muy importante mantener un control sobre los cambios realizados durante el proyecto. Para ello se utilizan las llamadas herramientas de control de versiones. De estas herramientas existen muchas, desde un poco más antiguas como SVN [19], pasando por CVS [20], hasta incluso el famoso GIT [21]. Este último es el que se ha utilizado para este proyecto, en concreto GITHUB. Para ello se han hecho diferentes versiones y se han ido subiendo al repositorio para tener un buen control de las versiones del videojuego. Finalmente se ha obtenido la red de versiones mostrada en la figura 3.

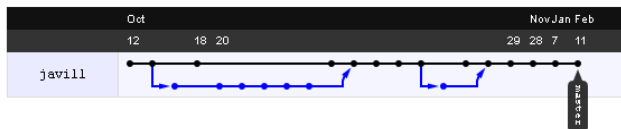


Figura 3

### 3.2.4 Otras herramientas

A parte de todas las herramientas de programación, diseño y control de versiones, para la realización de un proyecto se necesitan herramientas auxiliares pero no por ello menos importantes. Estas herramientas serían: herramientas de conversión, de procesadores de texto, herramientas para ayudar en la planificación del proyecto...

En este proyecto estas herramientas auxiliares utilizadas han sido:

- PNG2ATF: Se trata de un convertidor de imágenes PNG al formato ATF [22], para este proyecto se ha utilizado este formato para representar las texturas por las razones que se explicarán en el apartado 5.1.
- GanttProyect: programa utilizado para crear los diagramas de Gantt para planificar las tareas.
- Creately: programa online utilizado para crear los diagramas necesarios para el proyecto.

## 4 DISEÑO DEL VIDEOJUEGO

En esta sección encontrará los diferentes aspectos del diseño del videojuego, así como diseño de las vistas de la interfaz, personajes, mapa... como los diferentes paradigmas o modelos de programación y diseño que se han utilizado para la creación del juego.

### 4.1 Diseño de la interfaz

En esta sección se especificará con detalle cada una de las pantallas que componen Sunken Dream. Además, se indicarán las transiciones entre ellas así como la utilidad de cada elemento de la GUI (Graphical User Interface).

#### 4.1.1 Menú principal

El menú principal es muy intuitivo ya que dispone solamente de unas cuantas opciones como son: la de silenciar el sonido, comenzar una nueva partida, continuar partida y finalmente opciones. Todo ello acompañado de un fondo y una banda sonora acorde para introducir al jugador en el juego.

#### 4.1.2 En el nivel

La interfaz del nivel o lo que es lo mismo cuando estas dentro del juego, consiste básicamente en tres cosas diferentes: la vida actual del jugador situada arriba a la izquierda y el botón del inventario que se representa como una mochila y está situado arriba centrado, tal y como muestra la figura 4



Figura 4

Por otro lado tenemos el inventario abierto que muestra todos los objetos de los que el jugador dispone, tal y como se muestra en la figura 5.



Figura 5

### 4.2 Diseño de los personajes

En esta sección procedemos a enumerar y describir todos los personajes de Sunken Dream así como una breve descripción de sus habilidades y comportamiento.

#### 4.2.1 Protagonista

El protagonista de esta historia se trata de un muchacho de 14 años llamado Demix, nacido en una ficticia isla del Caribe llamada Tryland. Es un chico muy solitario y con una imaginación fuera de lo común, además de valiente y aventurero.

El diseño del personaje, tanto el dibujo como las animaciones, se han basado en esta descripción. Y por lo tanto como muestra la figura 6 se trata de un chico de piel mo-

rena y pelo castaño acorde con el aspecto de la mayoría de personas que viven por esa zona del mundo.



Figura 6

Este personaje empieza con un máximo de 100 de vida y con el arma inicial "ataques básicos o puñetazos".

En cuanto a las habilidades el jugador puede usar las mostradas en la Tabla 1.

Tabla 1: Habilidades Personaje

Habilidad	Descripción
Largo Alcance	Armas de largo alcance como cañones.
Corto Alcance	Armas de corto alcance como el puñetazo
Pociones	Se pueden usar para obtener habilidades especiales como son: saltar más alto...

#### 4.2.2 IA Pasiva

La IA Pasiva son los enemigos sin ninguna habilidad especial, solo atacan cuerpo a cuerpo y si chocas con ellos. El diseño de estos enemigos está orientado a hacerlos parecer lentos y más resistentes pero a la vez con un menor poder de ataque. La vida con la que normalmente disponen es de 80 puntos y el daño de ataque varía entre los 5 y 15 puntos. Un ejemplo de este tipo de enemigos serían los esqueletos simples, como el de la figura 7.

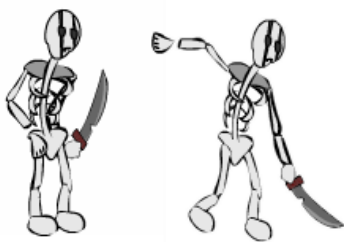


Figura 7

#### 4.2.3 IA Activa

La IA Activa, se trata de enemigos agresivos y con habilidades especiales. Su diseño está orientado a hacerlos parecer veloces y activos o despiertos. Estos enemigos pueden atacar tanto cuerpo a cuerpo como a distancia y no hace falta que los toques para que te ataquen ya que si te detectan cerca de ellos te persiguen durante un cierto tiempo. En cuanto a sus características, no suelen tener mucha vida pero si algo más de daño que la IA Pasiva. Un ejemplo de este tipo de enemigos serían los fantasmas, como el de la figura 8.

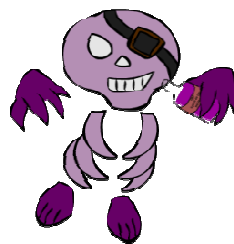


Figura 8

#### 4.2.4 NPC

Los NPC son personajes que están en el juego para ayudar al jugador, ya sea dándole pistas, objetos e incluso activando diferentes eventos que le permitirá al jugador avanzar en el nivel. Su diseño es algo más agradable que los enemigos ya que tienen que el jugador tiene que sentir que no representan ninguna amenaza para él. Estos personajes no pueden ser vencidos ni atacados por lo que siempre están disponibles.

Un ejemplo de estos personajes serían los ciudadanos de la isla como el de la figura 9.



Figura 9

#### 4.2.5 Monturas

Las monturas son animales o incluso objetos que se encuentran en los niveles y que el jugador puede montar para obtener habilidades especiales. Estas monturas basan su diseño en que sean lo suficientemente intuitivas para que el jugador sepa que se puede hacer con ellas. Un ejemplo de ello sería el gorila mostrado en la figura 10.



Figura 10

### 4.3 Diseño del mapa

Existen diferentes tipos de mapa para los juegos en dos dimensiones, mapas isométricos, ortogonales, isométrico escalonado, entre otros. Entre estos mencionados anteriormente el más usado para este tipo de videojuegos es el ortogonal, ya que los mapas suelen ser lineales y unidireccionales y eso proporciona al jugador un objetivo claro. Por estas razones se ha decidido usar ese tipo de mapa



aunque con ciertas diferencias ya que en este videojuego al jugador se le permite ir en las dos direcciones del mapa, hacia adelante y hacia atrás.

En cuanto al tipo de representación del mapa se ha seleccionado la representación mediante tiles [12] por las razones que se explicarán en el apartado 5.1 de este mismo documento.

Como hemos mencionado anteriormente en el apartado 3.2.1 la herramienta utilizada para el diseño del mapa ha sido Tiled [9], por lo que para su uso primero se ha tenido que desmontar el mapa como si de un rompecabezas se tratase, quedando como ejemplo la figura 11.



Figura 11

#### 4.4 Diseño de los objetos y Armas

En cuanto al diseño de los objetos y armas la parte más importante de este es que sea fácilmente interpretable por el jugador y desde el primer vistazo sepa para que sirve y que es. Para ello se ha inspirado en objetos reales que sean fácilmente relacionables por la mente de una persona como por ejemplo la llave y el candado mostrados en la figura 12

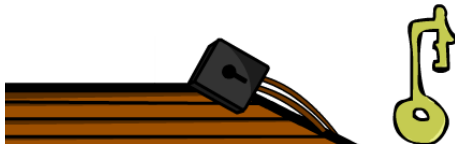


Figura 12

#### 4.5 Movimiento y físicas

Sunken Dream se desarrolla sobre un plano y tanto los enemigos como el personaje pueden desplazarse por él con cierta gravedad. Este plano presenta ciertos obstáculos como paredes o mobiliario que no podrán ser atravesados por ninguna entidad. Para el diseño de estas físicas se ha utilizado el motor de físicas Box2D [23] explicado anteriormente en el apartado 3.2.2

#### 4.6 Audio

Como todo videojuego la parte sonora es un pilar muy importante en el diseño y ambientación de los escenarios, para obtener una buena experiencia con el audio siempre es necesario guardar y entregar el proyecto con el fichero de audio en el formato que use el software con el que se produce o el software que lo usa, por eso la música y efectos sonoros que se usarán en este proyecto se conver-

tirán a .mp3.

##### 4.6.1 Música

Para este videojuego se han usado diferentes “bandas sonoras” acordes a según qué mapa o menú estés. Entre ellas se encuentran las siguientes más importantes:

- Menú principal: Una música de aventura y tensión aunque más relajada que la correspondiente a los niveles.
- Juego o niveles: Una música animada e intensa que debe provocar en el Jugador sensación de tensión, también debe estar relacionada con la escenografía del nivel.
- Victoria: se trata de una música breve que sonará cuando completemos un nivel. Debe ser alegre y hacer que el Jugador se sienta recompensado.
- Game Over: pieza muy breve de derrota con un tinte cómico, para restarle gravedad a la partida (que el Jugador piense que haber perdido una vez no es tan terrible).

##### 4.6.2 Efectos sonoros

Igual que se utiliza “banda sonoras” o música para la ambientación, estos también tienen que venir acompañados de unos buenos efectos sonoros que ayuden al jugador a situarse en el juego. Unos cuantos ejemplos de estos efectos sonoros son:

- Unir objetos: Un sonido corto pero alegre.
- Seleccionar opción: Sonido de tic al pasar el mouse por las opciones del menú.
- Recibir daño: Sonido de grito.
- Atacar: Sonido acorde con el arma usada.
- Enemigo: Sonido depende del enemigo que sea
- Caminar: Sonido dependiendo de lo que pise
- Hablar: Sonido de balbuceo.

## 5 RESULTADOS

En esta sección se repasará la planificación inicial y se explicará la planificación final pasando por los cambios realizados entre estos dos puntos y el por qué se han tenido que realizar, todo esto explicando cada tarea planificada y el tiempo de duración de la misma. Además de la planificación, en este apartado podemos encontrar el estado actual del proyecto.

### 5.1 Cambios en la planificación

Como en todo proyecto de estas características, puede haber problemas inesperados que obligan a hacer cambios en la planificación temporal de las tareas. Este subapartado muestra todos los cambios surgidos y el porqué han sido necesarios para la correcta continuación del proyecto, además de la nueva planificación de las tareas afectadas.

#### 5.1.1 Cambios en el software

Principalmente ha habido dos cambios importantes en cuanto al software, el primero y quizás el más importante es del no uso de un framework o motor de juego como

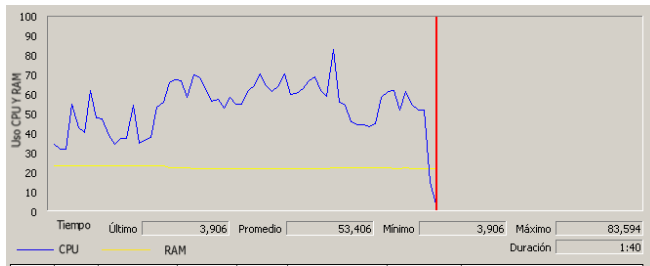


Figura 14

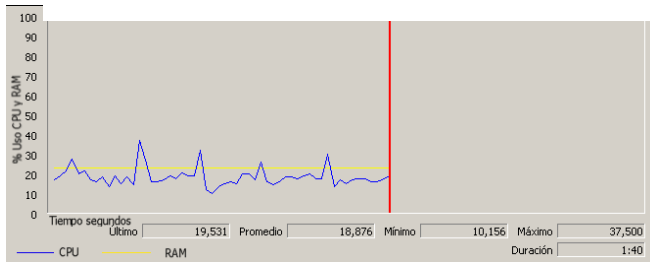


Figura 13

Citrus engine [15] a su uso. Antes de que se probara este motor de juego, el primer prototipo se desarrolló de 0 sin ayuda de ningún framework. Como se puede observar en la figura 13 el uso de la CPU y RAM eran disparatados sin el uso del framework desde picos de 83% de uso frente a los resultados obtenidos con su uso de un máximo de 37% como muestra la figura 14. Esto por supuesto, supuso una mejora de incluso más del doble de rendimiento. Esta mejora es debida a que el framework utiliza la potencia de las GPU de la tarjeta gráfica, que están hechas precisamente para renderizar las imágenes, y solo la potencia de la CPU.

El segundo cambio en cuanto al software de programación, es el comentado en el apartado 3.2.2. Por razones de compatibilidad entre el software de programación Flash Buldier [13] y el motor Citrus Engine [15], este primero tuvo que ser substituido por otro. Después de investigar las distintas alternativas posibles se ha seleccionado el programa FDT [18] ya que al contrario de muchas de las demás alternativas es un software gratuito y muy intuitivo de usar.

### 5.1.2 Cambios en el diseño

De cambios en el diseño también ha habido dos, que han retrasado aún más la planificación de las tareas iniciales. El primero y más importante es el cambio en la distribución o la forma de mostrar el mapa. Principalmente el mapa se mostraba mediante una imagen de fondo, pero a medida que este crecía la imagen también, hasta que llegó al límite permitido para mostrar imágenes según el framework Starling [16] que es 2048x 2048 píxeles. Esta restricción es necesaria para un correcto funcionamiento en dispositivos móviles. Finalmente se tuvo que cambiar radicalmente la manera de mostrar el mapa, a mostrarlo mediante tiles [12] o lo que es lo mismo, pedazos de la imagen de fondo repetidos en cuadrados de un tamaño fijo. Mediante esta forma el framework es capaz de crear mapas inmensos por trozos ahorrando así mucha canti-

dad de memoria.

El segundo cambio y menos importante de todos, ya que no retrasó mucho el proyecto fue la conversión de las imágenes de PNG a ATF [22] que se trata de un formato de adobe especialmente diseñado para correr en tarjetas gráficas. Aunque la compresión de este formato no es tan alta, la mejora de rendimiento y calidad es notablemente visible, ya que antes de ser usada, al contrario de los demás formatos, este no tiene que ser descomprimido por Stage 3D [24].

### 5.1.3 Cambios en la planificación de las tareas

Finalmente a causa de estos cambios inesperados en el diseño y el software, las tareas han tenido que ser retrasadas hasta el punto que algunas incluso han sido eliminadas del proyecto por falta de tiempo. Por lo que las tareas que han sido alteradas son: El diseño de personajes que ahora acaba el 28/01/14, la fecha final de la creación del menú principal ahora es el 15/01/14, la implementación del primer nivel se ha alargado hasta el 16/01/14. Las tareas que se han eliminado son: la creación del segundo nivel, guardar y cargar partidas y pruebas finales. Y finalmente se ha añadido la tarea de Test Beta y Alpha.

### 5.2 Planificación final

Finalmente se ha realizado un cronograma con todas las tareas finales del proyecto mostradas en el apéndice 4 y un diagrama de Gantt reflejado en el apéndice 5.

El cronograma final del proyecto se compone de las siguientes tareas:

- Formación previa, del (16/09/13) hasta el (30/09/13).
- Documentación del proyecto, del (16/09/13) hasta el (17/01/14).
- Prototipo inicial, del (01/10/13) hasta el (25/10/13).
- Diseño de personajes, del (21/10/13) hasta el (28/01/14).
- Primer nivel, del (26/11/13) hasta el (17/01/14)
- Menú principal, del (18/12/13) hasta el (15/01/14).
- Beta y Alpha test, del (26/11/13) hasta el (18/12/13).

### 5.3 Estado actual del proyecto

Hasta la fecha de este documento 11/02/14 se han realizado varios ejecutables del videojuego, el último ejecutable realizado cumple con las tareas mencionadas en el apartado 5.2 y que por lo tanto se trata de un videojuego jugable que tiene ya sus objetivos principales y es capaz de interactuar con el jugador. La versión actual esta titulada como 1.0 Alpha, ya que todavía no es una versión muy estable, aunque los "bugs" o errores existentes están debidamente detectados y documentados.

Finalmente el proyecto será alargado temporalmente para poder cumplir las tareas eliminadas de la planificación inicial y acabar de reparar todos los errores que tenga para poder obtener una versión más estable. Además de testear el juego en diferentes dispositivos móviles para comprobar la compatibilidad.



## 6 CONCLUSIÓN

En el presente proyecto se ha desarrollado un videojuego de plataformas [2] y aventuras sobre la tecnología AS3 [14] de flash. En concreto pretende ser un videojuego que una esos dos géneros en uno solo y que por lo tanto el jugador disfrute de las mejores ventajas de cada uno de ellos, además de disfrutar de una historia totalmente distinta a las que se encontraría en este tipo de juegos.

El videojuego implementado puede ejecutarse en muchos dispositivos, ya sea un dispositivo Android como en iOS pasando por un PC convencional incluido de manera online por el navegador web. Esto es posible gracias a que para su implementación se ha usado además de la tecnología AS3 [14], el framework Starling [16]. Aunque solo se ha comprobado en todo tipo de navegadores y en Windows y Linux. Sunken Dream está diseñado para un mínimo consumo de los recursos del dispositivo, haciéndolo así un juego bastante ligero.

Para realizar el proyecto de este videojuego se ha utilizado la metodología de diseño del software SUM [5] lo que ha permitido un rápido avance durante todo el proyecto y cumplir muchos de los objetivos principales.

## AGRADECIMIENTOS

El autor del proyecto quiere agradecer principalmente al Dr. Antonio López por guiar el proceso de documentación del diseño del software. Finalmente este proyecto fue apoyado y guiado por los chicos del foro de Citrus Engine

## BIBLIOGRAFÍA

- [1] Género Aventura Gráfica: Antonio Orrán Guerrero, Aventuras gráficas, el legado de Lucas Arts, 05/04/2013  
[WWW.TEKNOCONSOLAS.ES/ARTICULOS/ARTICULOS\\_AVENTURAS-GR-FICAS-EL-LEGADODE-LUCAS-ARTS-PRIMERA-PARTE/72600](http://WWW.TEKNOCONSOLAS.ES/ARTICULOS/ARTICULOS_AVENTURAS-GR-FICAS-EL-LEGADODE-LUCAS-ARTS-PRIMERA-PARTE/72600)
- [2] Género Plataforma: Juanma García, el género no está muerto, 11/10/2013  
[HTTP://GAMIKIA.COM/2013/10/11/EL-GENERO-ESTA-MUERTO/](http://GAMIKIA.COM/2013/10/11/EL-GENERO-ESTA-MUERTO/)
- [3] Super Mario BROS: Rus McLaughlin, The history of Super Mario Bros, 13/09/2010,  
[HTTP://WWW.IGN.COM/ARTICLES/2010/09/14/IGN-PRESENTS-THE-HISTORY-OF-SUPER-MARIO-BROS](http://WWW.IGN.COM/ARTICLES/2010/09/14/IGN-PRESENTS-THE-HISTORY-OF-SUPER-MARIO-BROS)
- [4] The Secret of Monkey Island: Alex Hudson, remembering the Secret of Monkey Island, 5/04/2013  
[HTTP://WWW.BBC.CO.UK/NEWS/MAGAZINE-22028682](http://WWW.BBC.CO.UK/NEWS/MAGAZINE-22028682)
- [5] SUM: una metodología para desarrollo de videojuegos  
[HTTP://WWW.GEMSERK.COM/SUM/](http://WWW.GEMSERK.COM/SUM/)
- [6] Flash: Adobe Flash Pro  
[HTTP://WWW.ADOBE.COM/ES/PRODUCTS/FLASH.HTML](http://WWW.ADOBE.COM/ES/PRODUCTS/FLASH.HTML)
- [7] Photoshop: Image Editor.  
[HTTP://WWW.ADOBE.COM/ES/PRODUCTS/PHOTOSHOP.HTML](http://WWW.ADOBE.COM/ES/PRODUCTS/PHOTOSHOP.HTML)
- [8] Spriter: Spriter editor.  
[HTTP://WWW.BRASHMONKEY.COM/SPRITER.HTM](http://WWW.BRASHMONKEY.COM/SPRITER.HTM)
- [9] Tiled: Map editor.  
[HTTP://WWW.MAPEDITOR.ORG/](http://WWW.MAPEDITOR.ORG/)

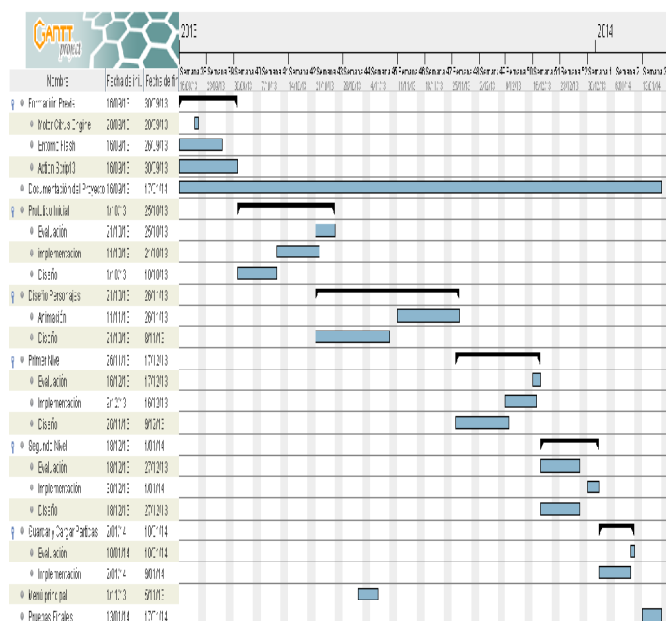
- [10] Texture Packer: Code and web.  
[HTTP://WWW.CODEANDWEB.COM/TEXTUREPACKER](http://WWW.CODEANDWEB.COM/TEXTUREPACKER)
- [11] Animación modular: En Dragon Bones.  
[HTTP://DRAGONBONES.GITHUB.IO/](http://DRAGONBONES.GITHUB.IO/)
- [12] Tiles: Qué son? 24/11/2013.  
[HTTP://ES.WIKIPEDIA.ORG/WIKI/TILE](http://ES.WIKIPEDIA.ORG/WIKI/TILE)
- [13] Adobe Flash Buldier:  
[HTTP://WWW.ADOBE.COM/ES/PRODUCTS/FLASH-BUILDER.HTML](http://WWW.ADOBE.COM/ES/PRODUCTS/FLASH-BUILDER.HTML)
- [14] Action Script 3: Lee Brimelow, six reasons to use Action Script 3, 18 Agosto 2008  
[HTTP://WWW.ADOBE.COM/DEVNET/ACTIONSRIPT/ARTICLES/SIX\\_REASONS\\_AS3.HTML](http://WWW.ADOBE.COM/DEVNET/ACTIONSRIPT/ARTICLES/SIX_REASONS_AS3.HTML)
- [15] CitrusEngin2D.  
[HTTP://CITRUSENGINE.COM/FEATURES/](http://CITRUSENGINE.COM/FEATURES/)
- [16] Starling: Daniel & Holger, 21/02/2012  
[.HTTP://GAMUA.COM/STARLING/](http://GAMUA.COM/STARLING/)
- [17] Angry Birds: Luis Castro.  
[HTTP://APRENDERINTERNET.ABOUT.COM/OD/JUEGOS/F/DE-QUE-SE-TRATA-EL-JUEGO-ANGRY-BIRDS.HTM](http://APRENDERINTERNET.ABOUT.COM/OD/JUEGOS/F/DE-QUE-SE-TRATA-EL-JUEGO-ANGRY-BIRDS.HTM)
- [18] FDT: Flash Developer Tool.  
[HTTP://FDT.POWERFLASHER.COM/](http://FDT.POWERFLASHER.COM/)
- [19] SVN: Subversion.  
[HTTP://ES.WIKIPEDIA.ORG/WIKI/SUBVERSION\\_%28SOFTWARE%29](http://ES.WIKIPEDIA.ORG/WIKI/SUBVERSION_%28SOFTWARE%29)
- [20] CVS: Repository.  
[HTTP://WWW.PUSHOK.COM/SOFTWARE/SVN-VSCVS.HTML](http://WWW.PUSHOK.COM/SOFTWARE/SVN-VSCVS.HTML)
- [21] GIT: Lars Vogel, 08/02/2014  
[HTTP://WWW.VOGELLA.COM/TUTORIALS/GIT/ARTICLE.HTML](http://WWW.VOGELLA.COM/TUTORIALS/GIT/ARTICLE.HTML)
- [22] Atf: Starling, ATF textures.  
[HTTP://WIKI.STARLING-FRAMEWORK.ORG/MANUAL/ATF\\_TEXTURES](http://WIKI.STARLING-FRAMEWORK.ORG/MANUAL/ATF_TEXTURES)
- [23] Box 2D: Physics engine.  
[HTTP://BOX2D.ORG/](http://BOX2D.ORG/)
- [24] Stage 3D: hardware-accelerated architecture.  
[HTTP://WWW.ADOBE.COM/DEVNET/FLASHPLAYER/STAGE3D.HTML](http://WWW.ADOBE.COM/DEVNET/FLASHPLAYER/STAGE3D.HTML)

## APÉNDICE

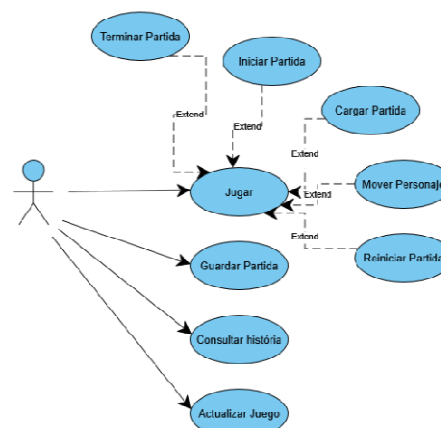
### A1. PLANIFICACIÓN DE TAREAS INICIAL

Nombre	Fecha de ini...	Fecha de fin...
♀ • Formación Previa	16/09/13	30/09/13
• Motor Citrus Engine	20/09/13	20/09/13
• Entorno Flash	16/09/13	26/09/13
• Action Script 3	16/09/13	30/09/13
• Documentación del Proyecto	16/09/13	17/01/14
♀ • Prototipo Inicial	1/10/13	25/10/13
• Evaluación	21/10/13	25/10/13
• Implementación	11/10/13	21/10/13
• Diseño	1/10/13	10/10/13
♀ • Diseño Personajes	21/10/13	28/01/14
• Animación	11/11/13	28/01/14
• Diseño	21/10/13	8/11/13
♀ • Primer Nivel	26/11/13	17/01/14
• Evaluación	16/12/13	17/01/14
• Implementación	9/12/13	16/01/14
• Diseño	26/11/13	9/12/13
♀ • Segundo Nivel	18/12/13	1/01/14
• Evaluación	18/01/14	27/01/14
• Implementación	30/12/13	1/01/14
• Diseño	18/12/13	27/01/14
♀ • Guardar y Cargar Partidas	2/01/14	10/01/14
• Evaluación	10/01/14	10/01/14
• Implementación	2/01/14	9/01/14
• Menú principal	1/11/13	5/11/13
• Pruebas Finales	13/01/14	17/01/14

### A2. DIAGRAMA DE GANTT INICIAL



### A3. DIAGRAMA DE CASOS DE USO



### A4. PLANIFICACIÓN DE TAREAS FINAL

Nombre	Fecha d...	Fecha d...
♀ • Formación Previa	16/09/13	30/09/13
• Motor Citrus En...	20/09/13	20/09/13
• Entorno Flash	16/09/13	26/09/13
• Action Script 3	16/09/13	30/09/13
• Documentación de...	16/09/13	17/01/14
♀ • Prototipo Inicial	1/10/13	25/10/13
• Evaluación	21/10/13	25/10/13
• implementacion	11/10/13	21/10/13
• Diseño	1/10/13	10/10/13
♀ • Diseño Personajes	21/10/13	28/01/14
• Animación	11/11/13	28/01/14
• Diseño	21/10/13	28/01/14
♀ • Primer Nivel	26/11/13	17/01/14
• Evaluación	16/12/13	17/01/14
• Implementación	9/12/13	16/01/14
• Diseño	26/11/13	9/12/13
• Menú principal	18/12/13	15/01/14
• Beta y Alpha test	26/11/13	18/12/13

A5. DIAGRAMA DE GANTT FINAL

