

Escuela Superior de Informática

Universidad de Castilla-La Mancha

CURSO DE EXPERTO EN DESARROLLO

DE VIDEOJUEGOS

TRABAJO FIN DE CURSO

MAKEMAKE

JULIO 2016



Escuela Superior de Informática

Universidad de Castilla-La Mancha

CURSO DE EXPERTO EN DESARROLLO

DE VIDEOJUEGOS

TRABAJO FIN DE CURSO

MAKEMAKE

Calzado Naranjo, Jesús Pedro

Muñoz Fraile, Pamela

JULIO 2016

CURSO DE EXPERTO EN DESARROLLO DE VIDEOJUEGOS

**Calificación Trabajo Fin de Curso**

CONVOCATORIA: Julio 2016

TÍTULO DEL PROYECTO: MAKEMAKE

AUTORES (ORDEN ALFABÉTICO):

CALZADO NARANJO, JESÚS PEDRO

MUÑOZ FRAILE, PAMELA

TRIBUNAL:

Presidente: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Vocal: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Secretario: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

FECHA DE DEFENSA: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| PRESIDENTE | VOCAL | SECRETARIO |
| Fdo: | Fdo: | Fdo: |

**Índice**

[1. Resumen 9](#_Toc455327602)

[2. Introducción 11](#_Toc455327603)

[3. Objetivos 13](#_Toc455327604)

[4. Arquitectura de la solución 15](#_Toc455327605)

[5. Análisis de costes 17](#_Toc455327606)

[6. Manual de usuario 19](#_Toc455327607)

[7. Conclusiones y trabajo futuro 21](#_Toc455327608)

[1. Conclusiones 21](#_Toc455327609)

[8. Bibliografía 25](#_Toc455327610)

# Resumen

*MakeMake* es un juego de plataformas tridimensional en el que un personaje, manejado por el jugador mediante un mando de control con “stick” analógico, debe completar numerosos niveles.

El juego se compone de varios mundos y cada mundo de varios niveles. Cada nivel consta de varios puzles para conseguir el pase al siguiente nivel, logrado al alcanzar la única meta por nivel que existe. De este modo, el jugador irá desbloqueando todos los niveles que componen el juego.

Para completar cada nivel, el personaje principal puede interactuar con distintos elementos que aparecerán en pantalla y que le permitirán avanzar.

El jugador puede, mediante el mando de juego, controlar al personaje y además, controlar la cámara de visión. Esto le será muy útil a la hora de descubrir secretos en cada nivel y calcular distancias de objetos. Así pues, el jugador puede rotar la cámara alrededor del nivel y conseguir una visión más clara del puzle que se le propone.

Además de todo esto, el personaje principal cuenta con habilidades como el salto y el arrastre de objetos de tamaño similar a él.

Con este proyecto final se persigue la puesta en escena de todas las habilidades obtenidas durante la duración del curso, para conseguir desarrollar de forma satisfactoria un videojuego completo, entretenido, vistoso y con posibilidad de extensión de contenidos. Además se ha pretendido poner hincapié en la parte técnica, tanto en la arquitectura como en el desarrollo de soluciones para determinadas características.

Por tanto, para el desarrollo de este juego se ha optado por el motor de render Ogre3D, de libre uso y programado en C++. Además se han usado otras librerías para el manejo de sonidos, interfaces de entrada/salida y cálculos físicos así como herramientas de software libre para diseño de elementos 2D y 3D.

# Introducción

El presente documento refleja y describe el proceso de elaboración y desarrollo del juego MakeMake. En este se explicarán todas las fases y retos encontrados durante su tiempo de creación. A continuación se puede encontrar una lista indicando qué encontrará a lo largo de los apartados siguientes.

Objetivos: Antes de comenzar un proyecto es necesario fijar una meta. Un lugar o estado a donde llegar, y que al conseguirlo se dé por concluido el trabajo. Lejos de ser una cosa nimia es vital.

Arquitectura de la solución:

Metodologia: nombra y explica la metodología seguida durante la fase de desarrollo del proyecto.

Diagramas de clase: Es el diagrama de clase creado para el desarrollo

Patrones: descripción de los patrones implementados.

Herramientas: enumeracion de las aplicaciones utilizadas durante el desarrollo del presente videojuego.

Lenguajes de programación: lenguajes principales usados en el desarrollo.

Bibliotecas: para la consecución del juego ha sido necesario el uso de bibliotecas externas que sirven de gran ayuda a la hora de implementar ciertas características contenidas en el juego.

Proceso de desarrollo: descripción del proceso de desarrollo del presente proyecto. Dentro de este proceso podemos encontrar las siguientes fases:

* Diseño: desarrollo de la idea, diseño y planificación del videojuego.
  + Bocetos. Se crean una serie de diseños a mano alzada sobre papel y se comenta en equipo.
  + Game concept. Se desarrolla la idea inicial dando como resultado un concepto. Una mecánica de juego, que intetará resultar divertida y entretenida de jugar y que suponga un reto para el jugador.
  + Diseño de solución software. Esqueleto del juego.
  + Creación de modelos de primera etapa. Se diseñan los modelos que se cargaran durante el desarrollo del juego. Estos modelos son creados simplemente para el desarrollo del juego y en fases finales se sustituirán por los definitivos.
* Implementación
  + Creación del esqueleto del juego basado en estados. En esta parte se crea las clases GameState y GameManager, piedra angular del videojuego.
  + Desarrollo de mundo de test e inclusión en el esqueleto inicial. Llegados a este punto se puede desarrollar y hacer pruebas de forma simultánea.
  + Desarrollo de personaje principal. En esta parte se desarrolla el manejo del personaje principal del juego, movimiento y salto.
  + Desarrollo de objetos interactivos. Se crean los diferentes objetos con los que se va a interactuar durante el juego y se desarrolla su comportamiento.
  + Implementación de colisiones. Se desarrollan los comportamientos que tendrán determinados elementos del juego cuando colisionan entre si.
  + Implementación de control por mando de Xbox.
  + Desarrollo de linea argumental y pantallas. Se crea un sistema de carga de escenarios a través de archivos de texto. Esto hace que el juego pueda crecer de forma sencilla.
  + Desarrollo de la camara del juego. Comportamiento de la camara muy especifico. Se mueve entre 4 puntos. Se anima la transición para dar mejor sensación de juego.
  + Desarrollo de pantalla para conocer el avance del jugador. Mediante esta pantalla accedemos a un mapa donde nos muestra de forma clara y sencilla por donde va el jugador en ese momento.
  + Creación de modelos definitivos de objetos interactivos.
  + Creación de animación para personaje principal. Es necesario que el personaje muestre movimiento al caminar.
  + Creación de 4 mundos ambientados de forma diferente. Se toma especial cuidado en determinados efectos visuales e iluminación de la escena acorde a lo que se presenta.
  + Elaboración de texturas. Se crean las texturas finales del juego. En algunos diseños se ha puesto interés en hacer una simulación de iluminación para estar en consonancia con la escena.
  + Creación de sonidos. Se crean los sonidos que empleará el juego. Los sonidos han sido creados con herramientas gratuitas.
  + Inclusión de músicas. Se incluyen finalmente las músicas del juego. Esto hace del juego una experiencia mas inmersiva.
  + Creación de diseños de pantallas de menú. Se añade el diseño definitivo de pantalla de “Press Start” y el mapa. Además se añade un diseño para la pantalla final, cuando el jugador concluya el juego.
* Análisis de costes: en esta parte se muestra el esfuerzo en horas desglosado por secciones de desarrollo de los integrantes del grupo.
* Manual de usuario: sección en la que se explica la historia detrás del juego y como jugar.
* Conclusiones y trabajo futuro. En esta sección se exponen una serie de conclusiones que se han sacado tras la elaboración del presente proyecto y una serie de mejoras que podrán ser incluidas en un futuro.
* Bibliografia: listado de fuentes consultadas durante la elaboración de este proyecto.

SALTO SECCION DEBAJO DE LA LINEA

# Objetivos

Objetivos funcionales.

Creación de sistema de juego con un personaje situado sobre plataformas. Es la base del juego. Es necesario disponer de un personaje que se pueda mover con soltura y jugabilidad por un mundo cargado en pantalla.

Desarrollar objetos con los que interactuar para conseguir la meta. Para que el juego empiece a complicarse se desarrollan estos elementos. La mayor parte de las veces sirven de ayuda aunque bien es cierto que otras tantas estorban.

Creación de mundos diferentes, que den impresion de estar bien integrados haciendo uso de iluminación y efectos, empleando luces y texturas. Así pues se crean efectos, como la lava del mundo castillo haciendo uso de texturas procedurales, partículas y efectos de luz

Desarrollo de sistema de carga y guardado de partida.

Implementación de control por mando de Xbox.

Desarrollo de mapa para control de avance del jugador.

Desarrollo de puzles simples.

Creación de varias pantallas que componen el juego, en las cuales se incluyen los puzles creados anteriormente.

Editor de niveles. Aunque no se ha implementado, es por donde debería seguir el proceso de desarrollo. De este modo, la creación de nuevos niveles se agiliza tanto a la hora de elaborarlos como de probarlos y modificarlos.

Objetivos no funcionales

Jugabilidad a la altura. No es necesario leer un manual para manejar a un personaje en vista subjetiva. El manejo se ha buscado acercar a cualquier juego de plataformas 3D en el mercado. Sencillo y con soltura.

Curva de dificultad en lugar de tutoriales. En lugar de crear un tutorial para cada objeto hemos optado por la solución que mas nos gusta. Jugar y probar. Cada elemento que se presenta en escena ha pasado por una pantalla inicial donde se le ha propuesto al jugador un puzle muy sencillo, el cual para completarlo debe hacer un uso sencillo del objeto presentado. Tras eso, la cosa se ira complicando poco a poco.

Variedad en ambientes. Se ha peusto especial incapié en crear mundos diferentes y que gusten visualmente. De este modo, el jugador ira variando entre mundos y la sensación de estar jugando una aventura es mayor.

Musica y sonidos adaptados. Se ha tratado de completar el juego con un conjunto de sonidos y música que puedan enlazar bien entre ellos.

Puzles y diversión. Creemos fehacientemente que el enfrentarse a determinados puzles puede ser satisfactorio. Si bien algunos son sencillos se ha tratado de buscar dificultad en otros, de tal modo que cuando el jugador lo supere consiga una gratificación por el tiempo dedicado.

SALTO DE SECCION DEBAJO DE LA LINEA

# Arquitectura de la solución

Metodología

Para el desarrollo del presente proyecto se ha optado por emplear scrum, debido a la necesidad de hacer entregas periódicas y se adapta bastante bien para el desarrollo en equipo.

El hecho de tener que realizar entregas cada cierto tiempo y tras ellas realizar modificaciones ha hecho que la naturaleza iterativa e incremental de scrum se ajuste a la perfección.

El rol de coordinador del equipo ha sido asumido por Jesus Pedro, teniendo una visión general del producto final a conseguir y organizando las diferentes tareas a abordar y concluir.

Se han realizado reuniones con periodicidad entre los miembros del equipo. De este modo, las tareas definidas han sido puestas sobre la mesa y se ha ido avanzando en ellas. En esas reuniones se han tratado las tareas concluidas y las que quedan por concluir asi como las que están en marcha. Además se ha puesto especial cuidado en solucionar los posibles conflictos entre desarrollos realizados por cada miembro del grupo.

Diagramas de clase

En fases tempranas, se desarrolló un diagrama de clases que fue variando con el tiempo. En primera instancia es difícil acertar completamente con todo lo que se va a necesitar en el desarrollo. Además, los cambios que puedan surgir por funcionalidad añadida o eliminada hacen que el diagrama inicial vaya variando.

El diagrama final de clases se encuentra en el Anexo 1.

Patrones

Para este proyecto se ha buscado la inclusión de patrones de diseño, si bien es cierto, que la exigencia en los plazos y la presión ha hecho que muchas veces se opte por una solución rápida para conseguir una funcionalidad y llegar a tiempo, dejando para una fase posterior un posible rediseño del código que lo haga mas amigable para modificaciones y mas sencillo de ampliar por parte de miembros futuros en el grupo.

Así pues los patrones que se han usado son los siguientes:

Singleton

Este patrón ha sido empleado en el uso del motor gráfico. La implementación que ofrece este es muy completa y hace que el desarrollo se acelere bastante.

La justificación de su uso es conseguir que para todos los elementos del juego exista una única instancia y que todos puedan manejar los mismos recursos.

Patrón State

El patrón estado ha sido implementado en los diferentes estados del juego. Existe una clase superior llamada GameState de la cual parten para su deficion las diferentes clases que definen los posibles estados del juego, que son Juego, Pausa, Intro, Replay, End.

Herramientas

Durante la elaboración del presente proyecto han sido varias las herramientas que se han usado, para varios ámbitos como son el diseño, la programación, retoque grafico, desarrollo de modelos 3D, documentación, etc.

A continuación se listan y detallan todas ellas.

Visual Studio C++ 2012

Visual Studio es un IDE de desarrollo software para diferentes lenguajes de programación. Dado que el motor de render Ogre3D usado esta escrito en C++ se ha optado por este mismo lenguaje de implementación.

Además de las diferentes ayudas en linea a la hora de desarrollar, opciones de autocompletado y detección de errores durante la edición, tiene un depurador muy potente. Todas estas herramientas hacen que el proceso de desarrollo se acelere mucho y se cometan menos errores además de que sea más fácil detectarlos y corregirlos.

Notepad++

Editor de texto muy útil a la hora de modificar los diferentes archivos de configuración del motor de render y además, muy usado para crear y modificar archivos clásicos de gráficos de Ogre, como partículas, materiales, overlays, etc

Blender

Programa de modelado, iluminación y renderizado animación y creación de graficos tridimensionales. Además de ser muy potente es software libre.

Se ha empleado en el diseño y desarrollo de modelos empleados en el juego, como la plataforma base, el personaje principal con su animación o los objetos interactivos.

InkScape

Inkscape es un editor grafico de vectores muy potente. Gratuito y de código libre se ha usado para la creación de todos los elementos graficos 2D del juego. Texturas de modelos, y overlays usan diseños creados a través de esta herramienta.

GIMP

Es un software de edición de imágenes digitales en formato mapa de bits. En este caso se ha usado para ciertos retoques en graficos 2D.

Lenguajes de programación

C++

El lenguaje en que se basa el desarrollo completo del juego es C++ dado el hecho de que es el lenguaje que usa el motor de renderizado. En un lenguaje muy utilizado y esto hace que haya una comunidad muy potente y activa alrededor. De tal forma, cuando encuentras un problema, no tardas en solucionarlo buscando en la web. Ademas son numerosas las bibliotecas que facilitan el trabajo y aceleran el desarrollo.

Ademas de todo ello, este lenguaje es el mas usado, con diferencia, en el mundo del desarrollo de videojuegos.

Bibliotecas

Ogre3D

Del acrónimo ingles Object-Oriented Graphics Rendering Engine es un motor de renderizado 3D orientado a escenas. Esta escrito en lenguaje C++ y es software libre. Facilita el acceso a capas inferiores de librerías graficas como pueden ser Direct3D y OpenGL. Ademas es bastante extensible por medio de bibliotecas externas.

SDL

Simple DirectMedia Layer es una biblioteca para manejo de sonidos en C++. Ha sido usada para dotar al juego de efectos de sonido y música ambiental.

OIS

Librería para control de dispositivos de entrada. Es muy útil a la hora de incluir el control por parte de raton, teclado o mandos de juego. Esta biblioteca viene incluida en Ogre de partida.

STL

Completa librería incluida en C++. Muy útiles resultan las diferentes implementaciones de estructuras de datos, como las listas, usadas en el desarrollo del presente videojuego.

SALTO DE SECCION DEBAJO DE LA LINEA

# Análisis de costes

En la siguiente tabla se muestra el coste en tiempo del desarrollo del presente proyecto. En ella puede verse desglosado el coste por persona y tarea.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | TAREA | Pedro | Pamela | H/Tarea |
| 1 | Brainstorming y division de tareas | 4 | 4 | 8 |
| 2 | Diseño de arquitectura y solucion software | 16 | 16 | 32 |
| 3 | Reuniones | 10 | 10 | 20 |
| 4 | Desarrollo de esqueleto del juego. Estado de juego y transiciones. | 16 | 10 | 26 |
| 5 | Creacion de modelos de mundo y tiles | 0 | 16 | 16 |
| 6 | Carga de mundo de test | 5 | 3 | 8 |
| 7 | Funcionalidad de makecamera | 25 | 10 | 35 |
| 8 | creacion de modelo de personaje principal | 3 | 16 | 19 |
| 9 | desarrollo comportamiento de personaje principal | 20 | 10 | 30 |
| 10 | implementacion de control de juego con xbox controller | 5 | 20 | 25 |
| 11 | implementacion de comportamiento de objetos interactivos. | 30 | 25 | 55 |
| 12 | implementacion de sonidos | 16 | 5 | 21 |
| 13 | generador de escenarios | 15 | 30 | 45 |
| 14 | carga de escenarios con archivos externos | 15 | 40 | 55 |
| 15 | carga y guardad de avance | 5 | 3 | 8 |
| 16 | diseño y desarrollo de pantallas de dialogo | 30 | 15 | 45 |
| 17 | documentacion | 56 | 56 | 112 |
| 18 | pruebas y validacion | 60 | 60 | 120 |
| 19 | desarrollo de niveles | 40 | 40 | 80 |
| 20 | sistema de particulas para el juego | 35 | 10 | 45 |
| 21 | programacion de colisiones | 40 | 15 | 55 |
| 22 | desarrollo de diseños 2d | 30 | 20 | 50 |
| 23 | creacion de modelos de objetos interactivos | 10 | 50 | 60 |
|  | Totales | 486 | 484 | 970 |

SALTO SECCION DEBAJO

# Manual de usuario

MakeMake es un juego sencillo en su manejo, aunque no tanto en la consecucion de los puzzles. De hecho es la principal premisa. Se ha puesto especial interes en hacer lo mas sencillo posible el acceso al juego.

HISTORIA - SACAR DEL ANTEPROYECTO.

Nori, habitante del pequeño planeta MakeMake, es el único ser de su especie que queda, como consecuencia de una invasión imprevista. Todos los Makelianos, excepto él, fueron capturados en naves Bistecca-Grasso a las órdenes del General Obvious Salame.

Las pretensiones de Salame son claras: vender a granel a todos los habitantes de MakeMake y usar sus tierras para producir el famoso combustible barato Aspro. A bordo de su nave Cetriolo, Salame tuvo la misión casi completa cuando detectó un punto de vida en el planeta. Nori.

Con este panorama, solo hay una posibilidad. Reunir las 4 piezas que completan Lattuga, un objeto milenario que neutraliza las naves modelo Grasso además de bloquear el acceso de habitantes externos al planeta, haciéndolo habitable únicamente por Makelianos.

Nori debe completar 4 mundos, al final de cada cual obtendrá una de las 4 piezas de Lattuga. Una vez completados los mundos, la invasión será neutralizada y todos los habitantes de MakeMake podrán ser rescatados de la nave prisión Carcere Grissini modelo Bistecca-Barattolo.

El juego

Escenario.

Un escenario consta de una base cubica donde ocurre toda la accion. se encuentra flotando en un mundo con un entorno claramente definido y reconocible. Ademas el escenario se compone de bloques de aspecto similar al bloque base, con los que se construye toda la escena. Nuestro personaje vive en esa plataforma y no debe salir de ella si no es por el teletransportador que en ella existe. Solo hay uno y puede estar a la vista o no. ¡Explora!

Nori.

El personaje pricipal es Nori. Nori es un bloque de sushi, de la especie MakeMake que habita el planeta hominimo. Un exceso de alga ha probocado a la especie tener patas que facilitan su desplazamiento. Ademas, las patas le posibilitan saltar para llegar a zonas mas altas que el.

Objetos interactivos

Para conseguir completar los puzzles, nori puede hacer uso de varios objetos que aparecen en la escena. En algunos escenarios es practicamente obligado usar todos los que salen, pero... ¡cuidado! Puede que ese bloque que mueves, no sea necesario y este ahi solo para despistar.

Los diferentes bloque son:

Piedra.

es un elemento muy util para alcanzar determinados objetivos. Unicamente puedes moverla, pero interactua con practicamente todos los elementos. Si la arrastras sobre el suelo, puedes transportarla, pero si la situas encima de una piedra fragil, esta se romperá. Puede que antes de hacerlo tengas unos segundos para usarla con el fin de llegar algo mas alto. Ademas, la piedra es util a la hora de usar los interruptores. situala encima y activaras completamente el interruptor.

Piedra fragil.

Este elemento permite permanecer sobre el un tiempo, transcurrido el cual se derrumba y desaparece para siempre. úsalo para llegar a sitios altos y recuerda, que pueden ocultar cosas. Quizá algo que no veas, este tras un muro derrumbable. Ten en cuenta tambien que algun bloque que derrumbas sin porqué puede ser que lo necesites mas adelante para conseguir completar el presente nivel.

Interruptor y bloques transparentes.

En el mondo de MakeMake existen unos bloques muy especiales. son transparentes y no son interactuables. No puedes pisar sobre ellos ni bloquean nada. Pero mediante el uso de un interruptor la cosa cambia. Con el interruptor activado, los bloques transparentes se materializan y se transforman en solidos. En ese momento, el personaje, o cualquier cosa que haya en el escenario puede hacer uso de los bloques para caminar sobre ellos o bloquear el paso de cierto elemento.

Meta

Es el destino final. La razon de Nori. Debe llegar ahi para seguir su aventura. una vez en la plataforma, Nori emprende un viaje al siguiente escenario y se muestra ante el jugador un nuevo reto.

Cámara

La camara del juego cobra un protagonismo obligado. La escena se presenta en un punto de la camara que podras mover. El hecho de poder moverla es necesario, ya que para entender la profundidad de ciertos objetos es vital. si bien es cierto que las sombras ayudan al jugador a calcular distancias entre objetos, habra mundos en los que la posicion de la luz haga necesario cambiar la camara para tener claro donde esta cada elemento. ademas puede que a simple vista no quede claro donde se encuentra la plataforma de meta y haya que explorar visualmente un poco hasta encontrarla.

la camara tiene 4 puntos que corresponden a las cuatro caras laterales del mundo cubico. ¡Úsala y verás!

ideas de pantallas.

Control

El control del personaje es muy sencillo. Corresponde con el uso normal de un personaje en un juego de plataformas clásico. Puedes moverte y saltar. Para ello puedes hacer uso de dos tipos de controles:

- Teclado:

teclas de direccion: mueven el personaje

tecla espacio: salto

teclas q y w: movimiento de la camara de izquierda y derecha respectivamente.

- Mando de control

se ha implementado el control mendiante mando de la consola xBox o compatible. los controles son los siguientes:

analogico. control del personaje.

boton 1 salto

boton 4 restart

boton 2 pausa

botones L y R giro de la camara

asi mismo para el paso de pantallas en los menus bastara con pulsar el boton 1.

Juego completado

una vez el jugador haya completado todos los puzzles propuestos, aparecerá una imagen que felicita del logro y lo anima a rejugarlo, volviendose al primer nivel y empezando nuevamente la aventura.

teletransportador oculto en bloques de mundo y una tapa encima rompible

miedra arrastrada a rompible y acercarla para ponerla encima y antes de romperse, saltar para llegar al teleport

cambiar las teclas de juego. poner flechas para mover personaje, espacio para salto y q y w para mover la camara.

SALTO SECCION DEBAJO

# Conclusiones y trabajo futuro

## Conclusiones

El desarrollo del presente proyecto ha sido una experiencia muy enriquecedora. Tanto es así que nos hemos quedado con ganas de más, a pesar de la cantidad de horas empleadas y el esfuerzo personal. Aquí no acaba MakeMake.

Durante el desarrollo muchos han sido los problemas que hemos ido encontrando y siempre, aun costando, hemos dado con una solucion satisfactoria. Es cierto que al final, con la presion de los plazos, el código no es todo lo bonito que deseamos, y que considero podemos hacer, pero la velocidad con que hemos tenido que desarrollarlo ha hecho que ciertas cosas se quedasen de un modo "solución rápida". Sabemos que esto no es lo ideal, pero hemos buscado que el juego este completo al nivel que inicialmente acordamos, y queda por realizar un trabajo de rediseño general. Con él, el código quedara más legible y usable por parte de cualquiera que entre a formar parte del equipo.

Hemos conocido durante este año varias herramientas, que confluyen todas en el motor de renderizado Ogre3D. Es cierto que al principio es un golpe duro y sin avisar. No es facil hacerse con el manejo de este motor. Incluso echarlo a andar es todo un reto. Pero una vez que te has familiarizado con el, es muy generoso. Se puede decir que empezamos sufriendolo y hemos terminado disfrutandolo.

Makemake nos gusta. Estamos contentos con el resultado final. Esto no quiere decir que no queden cosas por hacer. Si bien es cierto que el juego se acerca mucho a lo que en un principio imaginamos, hemos quedado con ganas de más, de mejorar e incluir muchas funcionalidades que harán al juego más entretenido y jugable, y puede que con ellas, sea el jugador sea el que quiera más.

A continuación describimos varias de las características que más importacia tienen a la hora de abordar nuevos trabajos en este proyecto:

Editor de niveles

Una de las cosas que echaremos de menos al jugar es que podamos crear nuestros propios niveles para compartirlos con algún amigo. En el actual desarrollo se pueden crear nuevos niveles, pero no es una forma amigable. Actualmente para crear nuevos niveles se puede crear o modificar un archivo de texto plano con el formato conocido y añadir el número de matrices que componen un escenario (como se dijo anteriormente un escenario esta compuesto por una matriz tridimensional de 15x6x15 posiciones).

Lo ideal es crear un editor de niveles que permita varias cosas:

- colocacion de elementos en el espacio tridimensional.

- colocacion de elementos de atrezzo del escenario. bloque sin funcionalidad.

- colocación de personaje principal y objeto meta.

- carga de niveles disponibles.

- guardado de modificaciones.

- prueba del nivel durante la edición.

Enemigos. ia y modelos.

En el concepto inicial del juego aparecían personajes que te hacían mas dificil conseguir pasar de nivel. Estos enemigos deambulan por el escenario y si consiguen tocarte, reinicias el nivel. Además, pueden estar provistos de armas, lo cual puede hacer aun mas complicado superar el reto.

Los enemigos deben estar provistos de IA, para que sea mas laborioso escapar de ellos.

Mayor número de niveles.

Un punto del juego que nos ha quedado corto es el número de niveles a conseguir. El desarrollo de estos ha sido manual. Debido al elevado coste de tiempo que ha supuesto la creación del juego y los plazos, el número de niveles, y la dificultad de los mismos se ha quedado corta. No obstante, teniendo ya los elementos para construccion de niveles, se pueden ir añadiendo sin problema. Hay que tener en cuenta que la máxima en la creación de este juego ha sido tener completo la mecanica del juego separando la edición de niveles. De este modo se puede subcontratar la creación de ellos o incluso que los jugadores puedan subirlos a un servidor del que puedan actualizarse el resto de usuarios.

Detección de caminos.

Cuando se crea un escenario, o cuando estas jugando en el, sería interesante que el juego te indicase si es posible llegar a la meta, con la disposicion actual de elementos en el escenario y dependiendo de la posicion de nori y la meta. De este modo, el escenario se reiniciaria y empezarias en la posicion inicial. Esta caracteristica, si bien supone un reto interesante en lo que programacion se refiere, Pensamos que es algo menos "romantico". Quizá es mas interesante que el propio jugador, tras deambular por el escenario, termine dandose cuenta de que no tiene solución. Esto hace que pueda estudiar mejor cómo se consigue descifrar el puzzle al que se enfrenta.

Nuevos objetos. Saltadores, Puertas de única dirección, plataformas moviles, palancas, etc

En este aspecto, el juego admite mucho margen de ampliación. Para esta ocasión y teniendo en cuenta el plazo, se han implementado un número limitado de elementos interactivos. Ahora bien, en el concepto inicial del juego se tuvieron en cuenta un número mayor de dichos elementos del juego. A la piedra movil, la piedra fragil, el interruptor con activación de objetos se le unirán los siguientes elementos:

- Saltador. objeto que tras caer en el permite ejecutar un salto el doble de altura que con el salto normal de nori. De este modo el personaje puede llegar a sitios a los que seria imposible llegar.

- Puertas de única dirección. Este elemento puede ser útil a la hora de proponer retos al jugador. así pues se puede añadir en un lugar del escenario del que no se pueda volver y hacer que el jugador haya tenido que "hacer cosas" en un lado, antes de pasar al otro, como por ejemplo lanzar una piedra o romper algun elemento.

Powers up para el personaje.

Otro de los elementos que no se han podido añadir ha sido los powerups. mediante estos items, el personaje puede hacer uso de habilidades que normalmente no tiene, como por ejemplo un salto mayor, o la posibilidad de traspasar algun muro. Ademas, la idea es que estos elementos vayan a parar a una mochila que lleva el jugador y pueda guardarlos para futuros escenarios, si no los necesita en el actual. así pues, puede que en un escenario futuro necesites haber hecho ciertas cosas en otro anterior. Incluso quiza debas rejugar niveles.

Selección de niveles.

En el desarrollo actual, el destino esta guiado y escrito. Es decir, la ruta es lineal y va desde el nivel 1 hasta el último. Este modo esta bien, pero lo ideal es poder rejugar los niveles que consideremos mas entretenidos, e incluso cargarlo para que alguien a quien se lo recomiendes pueda jugarlo. Para un futuro se espera tener una pantalla con un mapa mas grande por el que se pueda navegar y entrar en cada mundo. Una vez dentro, la idea se cargar todas las pantallas que pertenecen a ese mundo.

lista de advievements.

Se pretende añadir un conjunto de logros al juego. Esto completa la experiencia de juego y da mas vida al mismo. Los logros pueden ir desde completar el conocimiento de manejo del jugador a completar determinado puzzle en algun escenario o la recoleccion de determinados elementos. Ademas, estos logros pueden ser conectados con un servidor para consultar que logos ha conseguido cada jugador y poder cargar los tuyos en cualquier lugar.

varias cuentas de juego.

En esta entrega existe un unico jugador. Se comienza el reto desde el principio y se va avanzando, pero no existe la posibilidad de arrancar un juego desde el pricipio. Para mejorar queda dotar al juego de varias cuentas de juego, y poder navegar entre ellas. Ademas, estas cuentas pueden hacer uso de servicios en la nube de los cuales obtener determinada información y además guardar avance y logros.

Nuevos mundos.

MakeMake en su primera edición tiene 4 mundos que ya se han comentado. Como trabajo futuro queda completar un numero importante de ellos. Además, en estos mundos queda por añadir diferentes modificaciónes como:

- nuevos modelos 3d para mundos mas complejos.

- efectos sobre las condiciones de juego. Cosas como un mundo helado andando sobre hielo escurridizo, viento que actue sobre el personaje y elementos del mundo, movimientos de luz, terrenos poco estables, estanques, efectos climaticos, inclinaciones y demas efectos podrían completar el desarrollo.

Efectos de transicion entre pantallas.

En el presente desarrollo se ha puesto especial detalle en hacer una experiencia amigable y vistosa. El menu es muy sencillo, ya que solo te lleva a jugar y hay poco donde fallar. En este apartado, la mejora pasa por hacer un efecto fade-out/fade-in en la transicion entre diferentes pantallas de menu y juego y componer los elementos en pantalla con animaciones. Si bien es un cambio menor, puede que de mayor empaque al conjunto.

SALTO SECCION DEBAJO

# Bibliografía

SECCION FINAL