

Fatbondiga Arcade

Roger Juncà López

Master en Desenvolupament de Videojocs
Treball Final de Màster

Javier Luis Cánovas Izquierdo

Helio Tejedor Navarro y Jordi Duch Gavalda

30/06/19



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivad a [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Fatbondiga Arcade</i>
Nombre del autor:	<i>Roger Juncà López</i>
Nombre del consultor/a:	<i>Javier Luis Cánovas Izquierdo</i>
Nombre del PRA:	<i>Helio Tejedor Navarro y Jordi Duch Gavalda</i>
Fecha de entrega (mm/aaaa):	30/06/2019
Titulación::	<i>Master en desarrollo de videojuegos</i>
Área del Trabajo Final:	<i>Trabajo Final de Máster</i>
Idioma del trabajo:	Castellano
Palabras clave	<i>Videojuego, 3D y Arcade</i>

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

El título, *Fatbondiga Arcade*, nos dice de manera sintetizada de lo que se trata este trabajo final de máster. És una fusión entre dos palabras de diferentes idiomas, el inglés y el castellano. La primera parte del título es en inglés , “fat”, significa en castellano gordo, grueso... La segunda, “bondiga” es el conjunto de letras restantes de la eliminación de la primera sílaba de albóndiga y suprimiendo el acento. De este modo, de la unión de estas dos palabras, nace *Fatbondiga Arcade* un videojuego en el que encarnas el papel de una albóndiga que tiene que comer y crecer para escapar de las fauces de sus enemigos. No obstante, aparece la palabra arcade en el título, ja que define el género del videojuego en el que tendrás que auto superarse y mejorar para batir la máxima puntuación.

Método: La producción de este videojuego ha estado hecha en el lenguaje C#, dentro del motor gráfico Unity y usando otros programas como Maya y Zbrush para modelar y texturizar. También he usado Photoshop para el diseño de la GUI. El método usado ha sido programación orientada a objetos.

El **resultado** obtenido ha sido un videojuego para dispositivos android de genero Arcade con una historia de índole humorista.

Concluyendo, en este trabajo final de máster, me he dado cuenta del potencial adquirido tanto en el máster como en mis estudios anteriores. El potencial de poder hacer un videojuego divertido y creativo con mis propias manos y de manera individual, en poco tiempo.

Abstract (in English, 250 words or less):

The title, *Fatbondiga Arcade*, tells us in a synthesized way what this final master's project is about. It is a fusion between two words from different languages, English and Spanish. "Fat" is the first part of the title and is in English. The second part is "bondiga" is the set of remaining letters of the elimination of the first syllable of meatball and suppressing the accent. Since in Spanish the meatball is "albóndiga". In this way, from the union of these two words, *Fatbondiga Arcade* is born a video game in which you play the role of a meatball who has to eat and grow to escape from his enemies. However, the word arcade appears in the title, which defines the genre of the game in which you will have to overcome and improve to beat the maximum score.

Method: The production of this video game has been done in the C # language, inside the Unity graphics engine and using other programs such as Maya and Zbrush to model and texturize. I have also used Photoshop to design the GUI. The method used has been object-oriented programming.

The **result** obtained has been a videogame for android devices of genre Arcade with a humorous story.

Concluding, in this final master's degree project, I have realized the potential acquired both in the master's degree and in my previous studies. The potential to be able to make a fun and creative video game with my own hands and individually, in a short time.

Índice

1. Introducción	5-11
1.1 Contexto y justificación del Trabajo	5
1.2 Objetivos del Trabajo	6
1.3 Enfoque y método seguido	6-7
1.4 Planificación del Trabajo	7-9
1.5 Breve resumen de productos obtenidos	10
1.6 Breve descripción de los otros capítulos de la memoria	10-11
2. Estado del arte	11-12
3. Definición del Juego	13-20
3.1. Idea del juego	13-14
3.2. Conceptualización	14-17
3.3. Desarrollo y roadmap	18-19
3.4. Audiencia	19
3.5. Análisis de las Mecánicas	19-20
4. Diseño Técnico	20-26
5. Diseño del Nivel	26-28
6. Manual de Usuario	28-29
6. Conclusiones	30-31
7. Glosario	32
8. Bibliografía	33

1. Introducción

1.1 Contexto y justificación del Trabajo

Mi trabajo de final de máster es la culminación de todos los conceptos adquiridos, aplicados en dicho. Pero no solo de lo aprendido en el máster de desarrollo de videojuegos, sino de toda mi carrera.

En este apartado voy a exponer y justificar, de manera simple la necesidad del producto que he elaborado, un videojuego casual de genero Arcade. para ello voy a explicar, antes, lo que un simple juego hace con las personas, su utilidad, que tiene muchas. Me voy a centrar en la de relajarse y desconectar del mundo por unos instantes. Es sumamente importante dejar descansar la mente, y que mejor para dejar los problemas y el estrés del trabajo y los estudios que poder jugar a un videojuego casual sin necesidad de tener conexión a internet y de un modo portátil con nuestro dispositivo móvil. Después de todo esto cabe recalcar que desconectar y dejar descansar la mente un rato se es más productivo a continuación.

Yo personalmente he podido apreciar dicha necesidad, trabajo de cara al público y hablo con muchas personas a diario, gente con trabajos axfisiantes, con poco tiempo etc. Algunas personas incluso me han comentado dicha necesidad de un juego parecido al que he creado que no requiere mucho tiempo para aprender a manejarlo y sin necesidad de estar pendiente de él como en muchos otros juegos de android que están todo el dia reclamandote con notificaciones. Aparte, cabe destacar que el ambiente de fantasía creado hace que la desconexión de lo que nos preocupa o nos estresa sea más fuerte, y por lo tanto, realice mejor su función, mi videojuego. Ya que un juego debe proporcionar dicha desconexión del mundo.

En el ámbito de los videojuegos hay muchos juegos parecidos que realizan las funciones comentadas previamente, pero la gente se aburre de que las compañías exploten las ideas que funcionan hasta la saciedad, por eso, yo he creado algo distinto y creativo. Muchos de los juegos de ahora, exagerando, son lo mismo y simplemente cambia el nombre. Obviamente a nivel comercial vende más ja que es lo que el consumidor promedio juega, y a mi me gustaria cambiar eso. Este producto que he creado quiero que cambie un poco la manera de pensar de muchos y que abran la mente para probar juegos nuevos.

1.2 Objetivos del Trabajo

Los objetivos principales de mi trabajo son crear un videojuego para android y en un futuro pasarlo a nintendo switch, cuando haya conseguido el *kit* de desarrolladores para unity de dicha consola. uno de mi grandes sueños sería poder jugar y que los demás jueguen a mi juego en una nintendo. Por otra parte, también busco divertir, entretener y mostrar algo nuevo y diferente al mundo.

El último de mis objetivos era crear una base de juego dentro de unity lo suficientemente sencilla para seguir produciendo niveles después del primero.

1.3 Enfoque y método seguido

En el momento que decidí hacer un juego arcade, se me abrieron un abanico de posibilidades para llevar a cabo dicha tarea. Había dos caminos a elegir que se ramifican en otras elecciones para hacer el videojuego.

una de ellas era crear el motor de videojuegos, programarlo, y dentro de mi motor, específico para android, poder hacer mi producto. Pero esta idea la descarté rápidamente, ya que a pesar de ser una gran idea y poder optimizar mucho mejor el juego para la plataforma esmentada anteriormente. Tiene contras que no me agradaban, en ese momento me retrasaría demasiado en la producción. Uno de los contras era el tiempo, al tener que programar un motor y después continuación programar el videojuego hacía que tuviese que simplificarlo todo si quería hacerlo de ese modo. El otro contra, es complicado pasar ese videojuego a otras plataformas ya que ha estado programado en un motor específico para android, entonces no cabía en mis planes hacerlo de este modo si en un futuro quería pasarlo a nintendo switch. Por lo tanto, el camino o idea de hacer, yo mismo, el motor quedaba descartada. Entonces, el motor ya debe estar creado y yo programar dentro de él.

La idea de coger un motor comercial se ramifican en muchas posibilidades, hay muchos motores para programar videojuegos: Unity, Unreal Engine, Cryengine, Lumberyard etc. Pero había dos en concreto que me llamaban la atención por su reputación y por lo que me ofrecían, Unity y Unreal Engine. Los

dos son excelentísimos motores gráficos, los cuales ya había trabajado con ellos, no me eran para nada desconocidos. Unreal Engine tiene el gran punto a favor de poder llevar los gráficos al límite y también puedes crear un juego sin la necesidad de picar código con los blueprints. Pero los blueprints no son una forma correcta de optimizar bien un juego, al ser prefabricados usamos más código del que a lo mejor necesitamos y de esta manera cargamos la escena, además, yo no quería hacer un juego con unos gráficos realistas y que el producto pesara más de lo necesario en cuestión de memoria de disco, estos dos hechos me tiraron para atrás y decidí escoger el motor Unity que me brindaba la posibilidad de poder crear algo totalmente nuevo y diferente sin los blueprints, sé que unreal también tiene el lenguaje de programación C++, pero me desenvuelvo mucho mejor en C#. De esta manera, al escoger Unity podía programar lo necesario y optimizarlo al límite para traerlo a Android y que la respuesta del dispositivo fuera buena. Por otra parte hice búsqueda en la red de que motores comerciales se usaban más para crear aplicaciones móvil y corroboré lo que yo ya había calculado previamente, que el motor de Unity se usa mucho para dispositivos Android, al contrario que Unreal Engine.

Una vez escogido el motor tenía que definir mi *workflow*, que programas usaría, qué recursos etc.

Decidí usar Maya, Zbrush, Premiere y photoshop para crear el arte del videojuego y descargar de webs de contenido abierto el audio y algún modelo, decidí hacer las animaciones dentro del mismo Unity, por la practicidad y la sencillez del mismo.

En definitiva, usé unity y los programas esmentados aparte de recursos libres de internet para hacer mi producto de la manera más comercial posible, de este modo me sirviera en futuros proyectos para usar el mismo *workflow*.

1.4 Planificación del Trabajo

Recursos necesarios para el trabajo:

-Maya: Es un programa de 3D de modelado, texturizado, rigging, animación, render... Lo he usado para modelar y hacer las UVs de mis modelos, aparte, también para texturar y diseñar el nivel.

-Zbrush: es un programa de 3D muy parecido al maya pero que es más famoso por sus herramientas para esculpir. Lo he usado para esculpir a nuestra protagonista la albóndiga.

-Photoshop: programa principalmente de edición de la imagen y para dibujo 2D. Lo he usado para crear paletas de colores y la elaboración de sprites para la GUI.

-Premiere: programa principalmente de edición de video y para maquetado audiovisual. Lo he usado para hacer la introducción al videojuego y la pantalla de título.

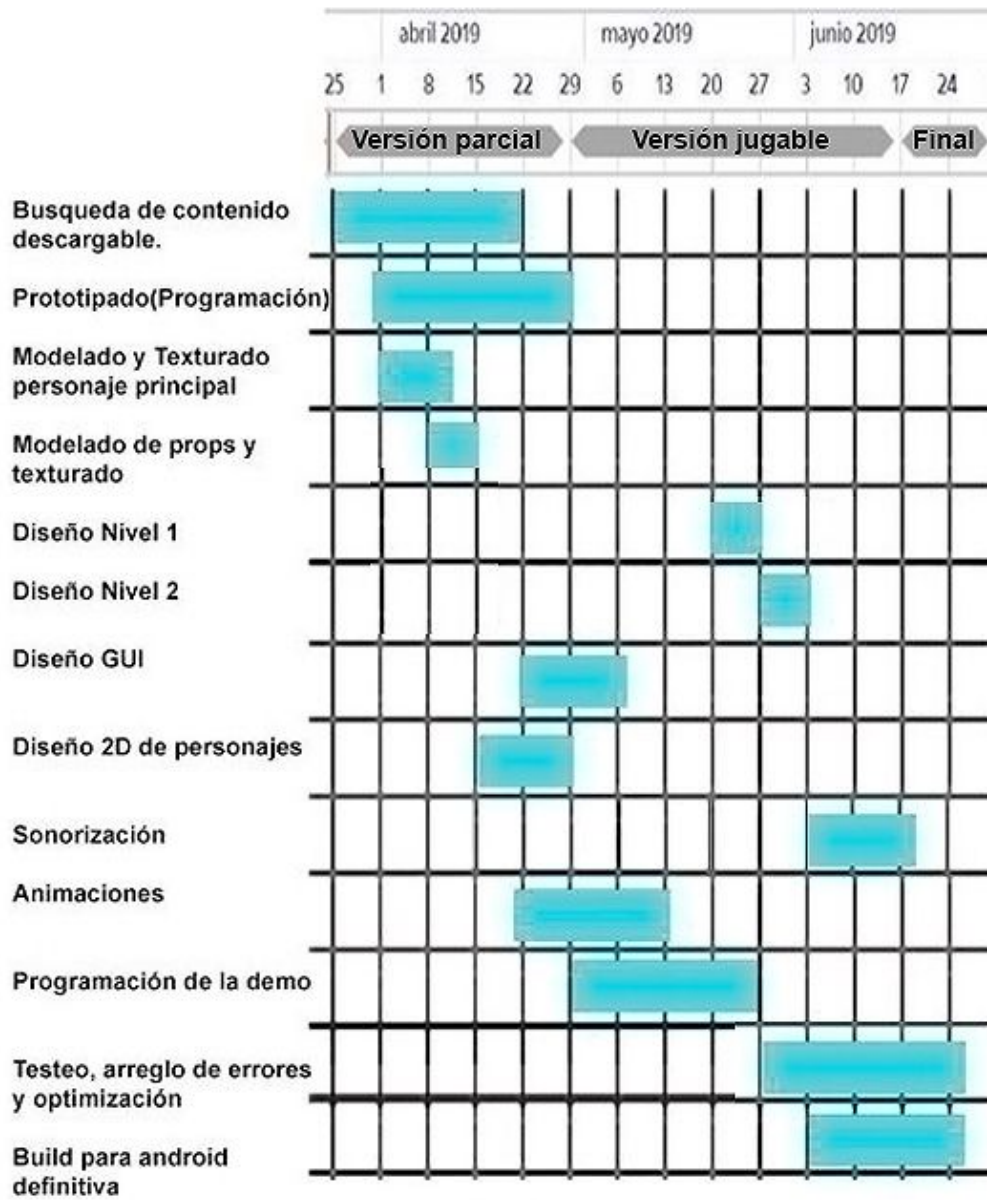
-Unity: Motor gráfico 2D y 3D principalmente usado para la producción de videojuegos o *demoscenes*. Lo he usado para programar, diseñar los niveles y animar.

-Webs de contenido gratuito: He usado internet y principalmente estas dos webs para acelerar el proceso de producción de mi videojuego:

-OpenGameArt: web dedicada a desarrolladores de videojuegos que abarca mucho contenido, tanto audios como modelos como texturas etc. Yo lo he usado principalmente para descargar audio gratuitamente.

-Sketchfab: Web con una comunidad de millones de creadores de contenido 3D, principalmente modelos 3D texturizados. En esta comunidad podemos encontrar recursos gratuitos que nos ofrecen los creadores y en este caso he aprovechado para descargar algunos modelos gratuitos y sin ánimo de lucro para mi juego.

A continuación os voy a mostrar mi diagrama de producción de este videojuego, he tenido que modificar algunas cosas como los niveles siguientes para centrarme principalmente en los estados del primer nivel y en definitiva en el primer nivel. Por razones de tiempo.



**diagrama de gantt y hitos de las PEC*

1.5 Breve resumen de productos obtenidos

Lo que he obtenido es un proyecto elaborado en unity, un videojuego para Android, género Arcade, nombrado Fatbondiga Arcade.

-Material artístico:

-Cocina: la cocina forma parte del primer y segundo nivel muchos de los modelos han estado hechos por mi en maya pero algunos han estado descargados en la red. Como pueden ser las sillas, fruta, nevera y el microondas.

-Personaje principal, albóndiga: ha estado creado en Zbrush, texturado y retopología en maya.

-GUI: la he hecho con photoshop y alguna parte con assets de Unity.

-Audio: todo el audio ha sido descargado de OpenGameArt.

-Efectos: Cuchillo, "dash", muerte, muerte por caída, conseguir punto y herida.

-Música: Un tema musical para el juego y otro para el final del juego, la parte de créditos.

1.6 Breve descripción de los otros capítulos de la memoria

En este apartado voy a hacer una introducción o breve resumen a los apartados que siguen a este.

-Apartado 2, estado del arte, expongo una revisión del género arcade y qué plataformas se usan para estos tipos de videojuegos. Obviamente la respuesta es sencilla los juegos arcade están encarados a ser jugados en recreativas, aunque en España, al menos donde yo vivo no hay dichos salones recreativos y en resumidas cuentas, por este motivo, he hecho un juego arcade para Android. La gente prefiere jugar donde y cuando quiera.

-Apartado 3, definición del juego: en este apartado narro todo lo referente a Fatbondiga arcade, su producción, su arte, su género, el motor usado, el gameplay, historia, personajes etc. En definitiva, un juego arcade y humorista aparte de fantasioso con una historia familiar y cotidiana, un juego casual.

-Apartado 4, Diseño técnico: En este apartado relato todo lo referente a los programas usados como Unity, los que he mencionado en anterioridad en el que incluyo un inventario de los recursos utilizados y un esquema de la arquitectura del proyecto en Unity, además de la explicación de la inteligencia artificial que en el caso de mi juego y el primer nivel es una anciana que nos quiere comer.

-Apartado 5, Diseño de niveles: En este punto detallo todo lo referente al diseño de mi nivel con una imagen del mapa con especificaciones y justificaciones de porque el nivel ha estado montado de esta manera y porque motivos tienen dichas limitaciones o no. También explicó la ruta a seguir que es bastante obvia y lineal pero con sus dificultades al no contar al jugador hacia donde debe dirigirse, esto también es un reto para él mismo.

-Apartado 6, Manual de usuario: muestro todos los requerimientos técnicos para poder jugar al videojuego de forma cómoda y también los controles a usar para poder interactuar de forma adecuada con la aplicación.

-Apartado 7, conclusiones: En dichas concluyo con el resultado y todo lo que me ha aportado y he aprendido con este proyecto y el master aparte de concluir todos los puntos de la memoria.

2. Estado del Arte

Fatbondiga Arcade es un juego de habilidad *arcade* de esquivar proyectiles lanzados por un enemigo. Los juegos arcade son de los primeros videojuegos de la historia y aparecieron en las recreativas.

El género *arcade* Consiste en la evolución directa de los *pinballs* y las tragamonedas, pero estos no se basan en el azar sino en la habilidad del

jugador. Las máquinas arcade contienen juegos de muchos géneros como deportes simuladores, carreras etc. Pero no me he centrado en el significado de arcade como a la maquina o a los salones recreativos sino en la evolución de dicha palabra hacia los videojuegos, determinando de este modo un juego el significado de arcade como un videojuego clásico o que recuerdan a las máquinas del mismo nombre.

Suelen ser juegos relativamente fáciles de jugar o que no responden fielmente a las físicas como la gravedad o otras fuerzas. Con unos controles sencillos y diseño sencillo, aparte contienen niveles cortos. También tienen una interrupción mínima entre niveles.

En esta época, muchos juegos, principalmente en los móviles y los del navegador, se diseñan para volver a la simpleza y rapidez de aquellas máquinas antiguas y son conocidos como videojuegos estilo *Arcade* o simplemente juegos *Arcade*. Así pues, podemos apreciar la evolución de la palabra *Arcade* en el tiempo y como ha acabado siendo a día de hoy.

Las plataformas a las que va dirigidas, como ya hemos explicado anteriormente, son ahora dispositivos móviles y el mismo navegador. Al evolucionar la tecnología y poder tener todo lo que queramos en un bolsillo no hace falta tener sitios específicos con recreativas para jugar juegos del género *arcade*. Aparte, debido a esta conclusión o a este hecho he tenido que investigar cuales eran los mejores programas de desarrollo o motores gráficos comerciales para producir este tipo de videojuegos para Android. Y he llegado a la conclusión que Unity es uno de los mejores por su gran facilidad de utilización y todas las herramientas que ofrece para optimizar de manera correcta una aplicación para móvil. En la *play store*, aplicación para android para descargar aplicaciones y otro tipo de contenido, hay juegos arcade como por ejemplo:[3] Arkanoid vs Space Invaders (remasterizados), Minesweeper Genius, Pinball Deluxe: Reloaded, Sky Force Reloaded etc. Son grandes juegos arcade para dispositivos móviles.[3]

En definitiva, hay mercado en dispositivos móviles de este género porque hay consumidores que buscan partidas cortas y sencillas y no buscan juegos largos.

3. Definición del juego

3.1. Idea del juego

3.1.1. Descripción del juego

Fatbondiga Arcade es un juego de habilidad en el cual tendrás que ser muy rápido y calcular muy bien las distancias y el *timing* para no ser aplastado por algún objeto, cortado o incluso quemando. Deveras ir recogiendo trozos de carne picada para conseguir la máxima puntuación y pasar a los siguientes subniveles del juego.

El juego está estructurado en 5 niveles distintos con subniveles dentro de cada nivel. Hasta que no se hayan pasado todos los subniveles no se va a desbloquear el siguiente nivel, si se te acaba la barra de vida y mueres pasarás al primer subnivel y vas a tener que recorrer todo el camino de nuevo.

A medida que vayas recogiendo carne vas a crecer en volumen y tú peso va a aumentar, tu volumen juega en tu contra ya que va a ser más complicado pasar por ciertos lugares del nivel però vas a necesitar ese peso extra para poder mover ciertos objetos del nivel y avanzar.

La Fatbondiga tendrá la habilidad de hacer un *dash* (movimiento rápido hacia una dirección) para poder esquivar más rápido los objetos que te van lanzando, aparte este dash te hace perder parte de la barra de vida y comprime tu volumen por un instante. Este dash tiene un *cooldown* de unos segundos, así pues tendrás que calcular el momento justo para hacerlo y también el momento justo para perder un porcentaje de vida.

3.1.2. Subgénero y referencias a videojuegos existentes

Un juego en 3D con cámara isométrica de habilidad però también es un juego arcade, por otro lado, tienes que usar la lógica para solucionar ciertas situaciones con el peso de nuestra protagonista.

Me he inspirado en ciertos aspectos del juego de disparos Gungeon de Doner Rock y Devolver Digital. Aunque en mi juego no se dispare he recogido, de dicho, la manera de esquivar y lo de no guardar tu progreso cuando mueres a no ser que hayas avanzado hasta cierto nivel.

En la parte de recoger pedazos de carne picada para crecer más i con tu peso desbloquear puertas o mover diferentes objetos que te obstruyen, me he inspirado en Katamari Damacy de Namco.

3.1.3. Tipo de interacción juego-jugador

El jugador se convierte en esta aventura en una albóndiga con los dilemas y problemas de dicha. Las mecánicas que tiene la albóndiga serán las de moverse rodando por el escenario y recogiendo pedazos de sí misma o de otras albóndigas para hacerse más grande y pesada. También podremos activar la habilidad de *dash* lo cual nos hace perder la parte de la vida y nos haremos más pequeños por unos instantes. Cabe recalcar que recoger la carne picada que hay en el escenario te va a devolver parte de la vida.

Todo esto implica que el jugador tenga una cantidad de dinámicas muy grandes al tener que combinar las mecánicas i el factor escenario. Tendremos que calcular muy bien si nos movemos hacia una dirección observando el escenario y con la complicación que te están intentando matar. Aparte uno de los problemas de una albóndiga es que está indefensa y muy expuesta.

3.1.4. Plataforma destino

A las plataformas para las que va a ir destinado el juego son dispositivos móviles y más adelante nintendo switch.

Han sido escogidas estas dos plataformas ya que disponen de una portabilidad y movilidad muy elevada y como este juego va a ser un casual podríamos afirmar que estos dos dispositivos son muy adecuadas para el juego.

3.2. Conceptualización

3.2.1. Historia, ambientación y/o trama

Todo empieza en un vecindario pequeño dentro de una casita muy acogedora donde vive una pobre anciana, Carmen, con su gatito. Ese día, el día de la catástrofe, así lo vamos a llamar, vino su nieto a comer y trajo a su perro, como cada día después del cole. El niño, Max, dejó la maleta en la entrada y se fue a la habitación donde tenía sus juegos.

La pobre anciana estaba ya cansada de cocinar todos los días y recordó que dentro de la nevera tenía una fiamblera con albóndigas que su amable vecina le había dado el día anterior. Carmen se extrañó mucho cuando su vecina le dio comida, pero pensó que como abuelas y vecinas que són tenían la necesidad de ser altruistas entre ellas y un regalo como aquel era muy gratificante.

Carmen se dispuso a calentar las albóndigas en el microondas, cabe recalcar que la anciana no veía muy bien, puso las albóndigas, el tiempo y la temperatura, cerró la puerta y al cabo de unos segundos una gran explosión hizo retumbar la casa tirándolo todo por el suelo. Max no se enteró de absolutamente nada, ya que de por sí el cuarto de los juguetes lo tenía siempre desordenado y llevaba los cascos puestos en ese momento.

¡Aquí en este preciso instante lleno de caos, empieza nuestra aventura, las albóndigas han cobrado vida y tú como jugador eres una de ellas. A medida que avances en la partida se van revelando pequeños fragmentos de la historia del pasado mientras avanzas.

Descubriremos que la vecina es bruja, que el gato de Carmen no deja dormir por las noches a la vecina, el perro de su nieto defeca en su jardín, el nieto con la pelota le ha roto una ventana y un jarrón y también que la plaga de cucarachas que tiene la abuelita se ha extendido a la casa de su vecina. Por lo tanto nos damos cuenta que nuestra creadora es una bruja y que está buscando venganza. Y nos vengamos.

3.2.2. Definición de personajes/elementos

-Carmen: la pobre anciana la cual ha perdido un poco la vista y la cordura, abuela de Max. Suele ser muy simpática y agradable, como todas las abuelitas. Su familia la tiene muy abandonada y vive sola, tiene la suerte de tener un nieto al que cuidar.

-Max: El nieto de Carmen el cual tiene 13 años y es un pasota. Le encanta jugar a pelota y a videojuegos. Tiene un perro llamado Tormenta. Es un niño delgadito con el pelo corto y oscuro. Sus padres trabajan mucho y apenas pasa rato con ellos. Su abuelita es la única persona que le presta atención y cuidado a Max.

-Tormenta: Un perro muy alocado de color marrón, una mezcla de razas, lo que llamaríamos un perro faldero. De un tamaño pequeño que tiene la manía de hacer sus necesidades en el patio de la vecina.

-Pelusilla: El gatito de Carmen es de color blanco con manchas negras y por las noches arma un buen jaleo y no deja dormir a la vecina.

-Fatbóndiga: Es la protagonista de esta aventura, una de las albóndigas supervivientes de la explosión.

En el juego aparecen otros elementos como cucarachas y todo lo que pueda haber en casa de una abuelita como sillas, libros, juguetes (del nieto), pelusas (del gato), cristales rotos, piezas de microondas, todo esto provocado por la explosión.

Cabe destacar que aparecen partes de otras albóndigas que han estallado por el camino debido a la inestabilidad generada por el microondas.

3.2.3. Interacción entre los actores del juego y/o elementos

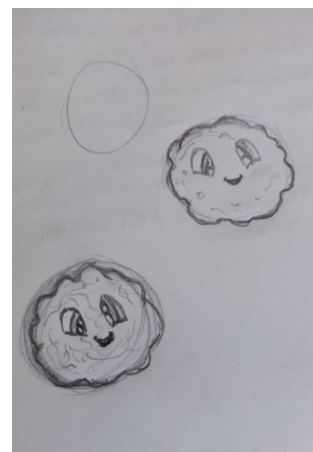
- Interactuamos con partes de otras albóndigas de manera que cojemos mas carne picada i crecemos.
- La abuela quiere comernos, atraparnos o acabar con nuestra existencia como sea.
- El perro quiere comernos y el gato atraparnos para jugar con nosotros.
- El nieto al vernos se asusta y nos tira la consola lo que crea un espacio virtual dentro del cuarto de juguetes.
- La interacción con el escenario, el peso y el volumen de la albóndiga dependen de cada nivel en el que estemos.
- La abuela tambien interactua con los elementos de manera inconsciente, como por ejemplo rompiendo un trozo de cristal con una maza y los cristales salir propulsados hacia todas partes lo que hace de un obstáculo estático lo transforma en móvil por un instante.

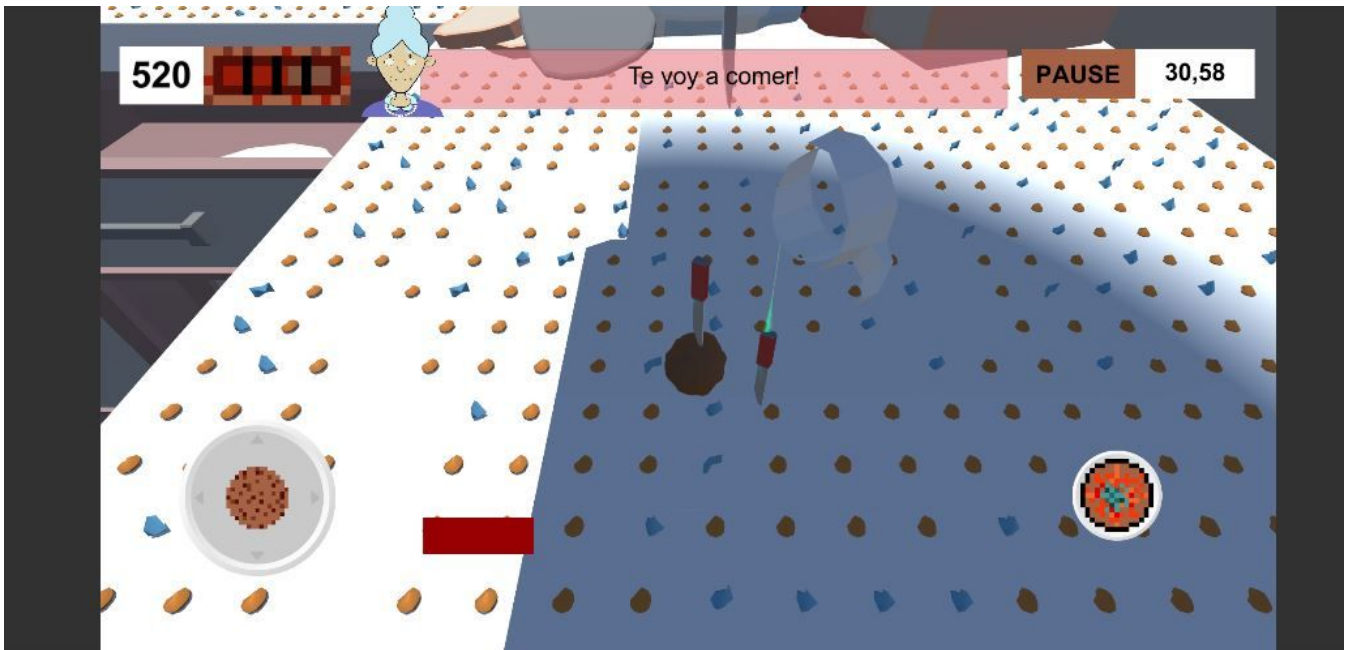
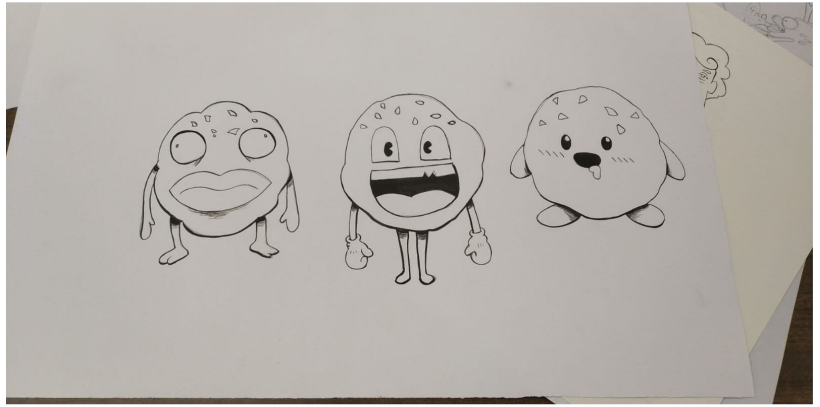
3.2.4. Objetivos planteados al jugador

De objetivos hay de dos tipos los objetivos a corto plazo o inmediatos como correr, esquivar, recoger pedazos de carne y también más adelante se van a rebelar los objetivos a largo plazo como la venganza de la bruja hacia la anciana, el perro, el gato, las cucarachas y el nieto.

Tu objetivo final es vengarse del nieto que acaba de romper el jarrón preferido de la vecina, provocando una indigestión.

3.2.5. Concept Art y screenshots





3.3.Desarrollo y Roadmap

3.3.1. Evaluación de engines y kits de desarrollo

Para este proyecto voy a usar un motor de videojuegos llamado unity ya que me proporciona una facilidad para hacer de mi juego un juego multiplataforma, como ya he expuesto anteriormente este juego va enfocado a plataformas portátiles, previamente para móvil y posteriormente su distribución a la nueva consola switch.

Hay otros motores como por ejemplo unreal que me podrían servir ya que también tienen salida para móvil y switch pero me desenvuelvo mejor con la plataforma unity y C#. A parte, unity tiene una gran cantidad de documentación disponible ya sean manuales y tutoriales que se pueden encontrar en su página oficial como en otras páginas. Unity tiene una comunidad muy activa.

Otra de las ventajas de unity es la rapidez con la que se puede empezar a trabajar y funciona muy bien para el prototipado.

Junto con todos estos pros tenemos la gran variedad de contenidos de terceros, el cual usaré en algún momento para agilizar el proceso de creación del videojuego ya sean *props*, personajes, música ...

A mi parecer el uso de *prefabs* me optimiza mucho el trabajo y en unreal no existen, sino que lo más parecido son blueprints representadas de manera gráfica pero esto te obliga a repetir código y desoptimizar el proyecto.

Los kits de desarrollador que voy a usar son los JDK para construir la APK del juego y en un futuro cuando el juego esté terminado enviar un correo al soporte de desarrolladores de nintendo para conseguir los kits de nintendo switch.

Les pregunté a ellos qué debía hacer para conseguirlo y me respondieron esto: "We are currently looking for developers who want to bring content to the Switch. In order to be considered for approval, please send a pitch to accountmanagement_publisher@nintendo.de if you are in the EU region, or thirdpartypublisher@noa.nintendo.com if you are in the NA region. Please also ensure that support@noa.com is in the CC. For your pitch, please include information about your development experience, your company (if applicable), and what you plan to bring to the Switch platform. Screenshots and/or videos help!

Andy Harpel"

Solo tienen desbloqueados los kits de desarrollo para wii U y new 3ds.

3.3.2. Recursos por objetivo

En el diseño de niveles usare contenido descargable de internet de *opengameart*, ejemplos de sketchfab y Maya para recolocar los objetos y modelar algún prop.

Las animaciones las haré simples y desde Unity, también usare *sprite sheets* para animar ciertos detalles, estos, los bajaré de internet o los creare desde photoshop.

En el apartado de sonorización, tengo un compañero que es músico y me va ha hacer la música del juego pero toda la parte de efectos la sacaré de *opengameart*

3.4. Audiencia

3.4.1. Audiencia

El juego claramente va destinado a un público casual. Este juego es para la gente que quiera relajarse y pasarlo bien resolviendo los diferentes niveles y avanzando en la trama aunque el juego tenga niveles y una curva de dificultad progresiva no tiene modo difícil para los jugadores Hardcore ya que el juego en si esta pensado para un jugador casual. Puede ser jugada más bien en una franja de edad de 12-20 años mayoritariamente, aunque no se debe descartar que el tono irónico del juego sea atractivo para gente de más edad.

El videojuego va a gustar a los dos géneros tanto al masculino como al femenino ya que no está lleno de violencia, es un juego neutral en este sentido.

3.4.2.Código PEGI

El código PEGI es de +7, al no aparecer escenas explícitas ni insultos, ni violencia. El juego es apto para todos los públicos.

3.5. Análisis de las mecánicas

La mecánica principal del juego es nuestra propia destreza y precisión para mover a la albóndiga por la zona estratégica, que son los puntos donde no hay obstáculos, y no caer en los agujeros o al vacío que nos alejan de poder pasar el nivel. En este caso la zona segura no aparecería o sería el enlace de subnivel a el siguiente nivel. Nos moveremos por la cocina para llegar a él "goal point".

Seguidamente, otra de las grandes mecánicas para resolver los niveles, la mecánica de configuración, es la del peso de la albóndiga cuantos más trozos de carne picada recolecta, nuestra protagonista, pesará más y podrá mover objetos para seguir adelante en el subnivel, esta segunda como mecánica de gestión y recolección. Otra de las mecánicas relacionada con la de recoger pedacitos de carne picada es la de hacerse más grande.

Nosotros como jugadores encarnamos el papel de la albóndiga en un control Focus Loci. Muchas de las mecánicas son autorreferenciales y con una lógica propia interna ya que no partimos de ninguna referencia en cuanto las habilidades de una albóndiga. La vecina hace la función de protagonista y nos da pistas para avanzar.

4. Diseño Técnico

El **entorno escogido** como ya he comentado en anterioridad es Unity un motor gráfico muy práctico. Unity es uno de los mejores motores comerciales del mercado actualmente compitiendo con Unreal Engine.

Empezaremos este apartado con la definición de Unity y lo que nos ofrece. Unity es un motor de videojuegos multiplataforma creado por la empresa Unity Technologies. Unity está disponible como plataforma de desarrollo para Microsoft Windows, OS X(macOS) y Linux.

Unity se puede usar juntamente con otros programas que lo complementan como podría ser Blender, 3ds Max, Maya etc. Los cambios que podamos hacer nosotros en dichos objetos creados con estos programas se actualizan automáticamente a todas las instancias de este objeto durante todo el proyecto sin necesidad de la reimportación manual.

Unity utiliza OpenGL a Windows, Mac y Linux, pero Direct3D només a Windows, OpenGL en el nostre cas per programar en Android. Dentro del programa podremos utilizar dos tipos de lenguaje para programar con una

aplicación externa pero totalmente compatible como puede ser visual studio o usando el propio MonoDevelop para la depuración de scripts que nos ofrece Unity. Usaremos los lenguajes C# o JavaScript para programar, yo me centro en el lenguaje C# ya que estoy más habituado a él.

Finalmente, dentro de unity podremos montar escenas, animar, sonorizar y todo lo referente para acabar haciendo un videojuego. Lo que de momento no se puede hacer es modelar sin usar una herramienta externa.

Los requerimientos técnicos para poder usar Unity son los siguientes.

[1]**Citación.*

- Sistema Operativo: Windows 7 SP1+, 8, 10, solo versiones de 64-bit; macOS 10.12 o superior

- CPU: Soporte para el conjunto de instrucciones SSE2.

- GPU: Tarjeta de video con capacidad para DX10(shader modelo 4.0).

Aparte yo devo usar más requerimientos técnicos ja que voy a desarrollar para android:

- Android: Kit de desarrollo Android SDK y Java(SDK); el IL2CPP scripting backend requiere Android NDK.

[1]

Las herramientas que he usado en la producción de mi videojuego han sido Maya, Zbrush, Photoshop, Premiere y un par de webs de contenido gratuito para videojuegos.

- Maya:[2] software de animación, modelado, simulación y renderización en 3D, proporciona un potente conjunto integrado de herramientas. Se usa para animación, entornos, gráficos de movimiento, realidad virtual y creación de personajes[2]. Los desarrolladores de este software son Autodesk y Alias Systems Corporation. Surgió de la evolución de otro programa llamado Power Animator. Además, Maya es el único software de 3D acreditado con un Oscar gracias al enorme impacto que ha tenido en la industria cinematográfica con la creación de efectos visuales.

Maya trabaja con cualquier tipo de superficie NURBS(son figuras creadas a base de curvas), polígonos y subdivisión de superficies, e incluye la posibilidad de convertir entre todos los tipos de geometría.

El espacio tridimensional que maya nos ofrece es un espacio matemático virtual creado por el programa de diseño 3D. Está definido por un sistema cartesiano de tres ejes: X, Y e Z. Y en dicho espacio virtual creamos nuestras figuras o efectos visuales. Lo he usado principalmente para modelar y texturizar el escenario.

-Zbrush: software de modelado 3d, escultura y pintura digital encarado más a la parte artística del mismo y no a la parte técnica. Los creadores de este programa son pixologic. Esta herramienta puede ser usada para hacer modelos de alta resolución es utilizado en películas, juegos y animaciones. Zbrush usa niveles dinámicos de resolución para permitir a los modeladores trabajar de un modo global o local.

En Zbrush hay diferentes pinceles customizables que nos ofrecen atributos únicos para nuestros trabajos. Aparte de todo esto tiene varias herramientas como el DynaMesh que nos permite generar de manera rápida un modelo con una distribución uniforme de los polígonos. Con este programa he creado a la albóndiga.

-Photoshop: Adobe Photoshop pertenece a la Adobe Creative Suite, es un editor de gráficos rasterizados desarrollado por Adobe Systems, su creador es Thomas Knoll. Utiliza principalmente el retoque de fotografías y gráficos, su nombre en español significa literalmente “taller de fotos”. Photoshop trabaja principalmente en formatos 2D y se pueden exportar con el mismo programa a muchos otros formatos como pueden ser: JPEG, PNG, BMP, TIFF, PDF etc. Dentro del photoshop podemos trabajar con máscaras y con capas, de este modo conseguiremos un trabajo más limpio.

Photoshop contiene en sí muchas herramientas y filtros. Yo en mi proyecto he usado los más básicos como pueden ser los pinceles, con ellos he creado la “GUI” de mi videojuego y la paleta de colores.

-Premiere: Adobe Premiere también pertenece a la Adobe Creative Suite, es un programa destinado a la edición de video en tiempo real. Este editor de video profesional es bastante intuitivo. El desarrollador es Adobe. Cuenta con una interfaz personalizable, por defecto organizada en espacios de trabajo que coinciden con las etapas de la edición: ensamblaje, edición, color, efectos, audio y títulos. Permite el trabajo con múltiples secuencias, que además pueden ser ordenadas, unas detrás de otras. A parte de medios, las líneas de tiempo admiten una serie de

elementos especiales como capas de ajuste, como las capas de Photoshop.

Usando este programa puedes coger imágenes y crear secuencias de las mismas como he hecho yo en mi proyecto, también añadirles audio de fondo. Lo he usado para hacer la pantalla de título y la introducción a mi videojuego.

-Webs de contenido gratuito: En mi trabajo he descargado contenido de la red de webs libres sin ánimo de lucro para poder agilizar el proceso de producción de Fatbondiga Arcade.

-OpenGameArt.org: es un contenedor para usar gratuitamente con recursos abiertos de software para videojuegos, proyectos que nos ofrece un abanico casi ilimitado de contenido.

Nos permite a los desarrolladores de contenido descargar recursos de arte tanto 2D como 3D y audios.

De esta web he descargado los efectos necesarios de sonido para hacer mi videojuego

-Sketchfab: es un sitio web utilizado para visualizar y compartir contenido 3D online. La compañía se fundó en Francia y hoy se encuentra localizada en París y Nueva York. Sketchfab proporciona un visualizador de modelos 3D basado en tecnología WebGL que permite reproducir modelos 3D tanto en páginas web para móviles como de escritorio.

Sketchfab también es usado para descargar contenido gratuito de creadores. Yo lo he usado para coger referencias de arte para modelar mi proyecto y he descargado algún modelo gratuito para completar el escenario.

Los **recursos usados** y descargados de webs libres son los siguientes, mostraré un inventario de los objetos 3D y los audios bajados de las dos webs usadas explicadas anteriormente.

-OpenGameArt: En esta web principalmente use los efectos de sonido y la música.

-Efectos de sonido: “dash”, cuchillos, recogida de coins, muerte , muerte por caída y herida o daño

-Música: El tema Principal del juego y otro tema para la pantalla de créditos, la que yo nombro “continuará”.

-Sketchfab: En esta web he descargado partes de proyectos gratuitos para montar mi escena, he obtenido modelos 3D: fruta , verduras, sillas , mesa, nevera, microondas y cafetera.

Esquema de la arquitectura de mi videojuego explicaré los principales controladores que hacen que el juego evolucione y que avance.

Antes que nada definiremos las escenas y su orden, de la escena 1 pasamos a la 2 y de la 2 a la 3. Una vez en la 3 podemos escoger pasar a la 4 y la cuatro nos enviará a la escena 5 Nivel uno. Dependiendo de si nos pasamos el juego o morimos nos llevará a la escena 7 o 6 respectivamente.

1-Intro

2-Pantalla de Título

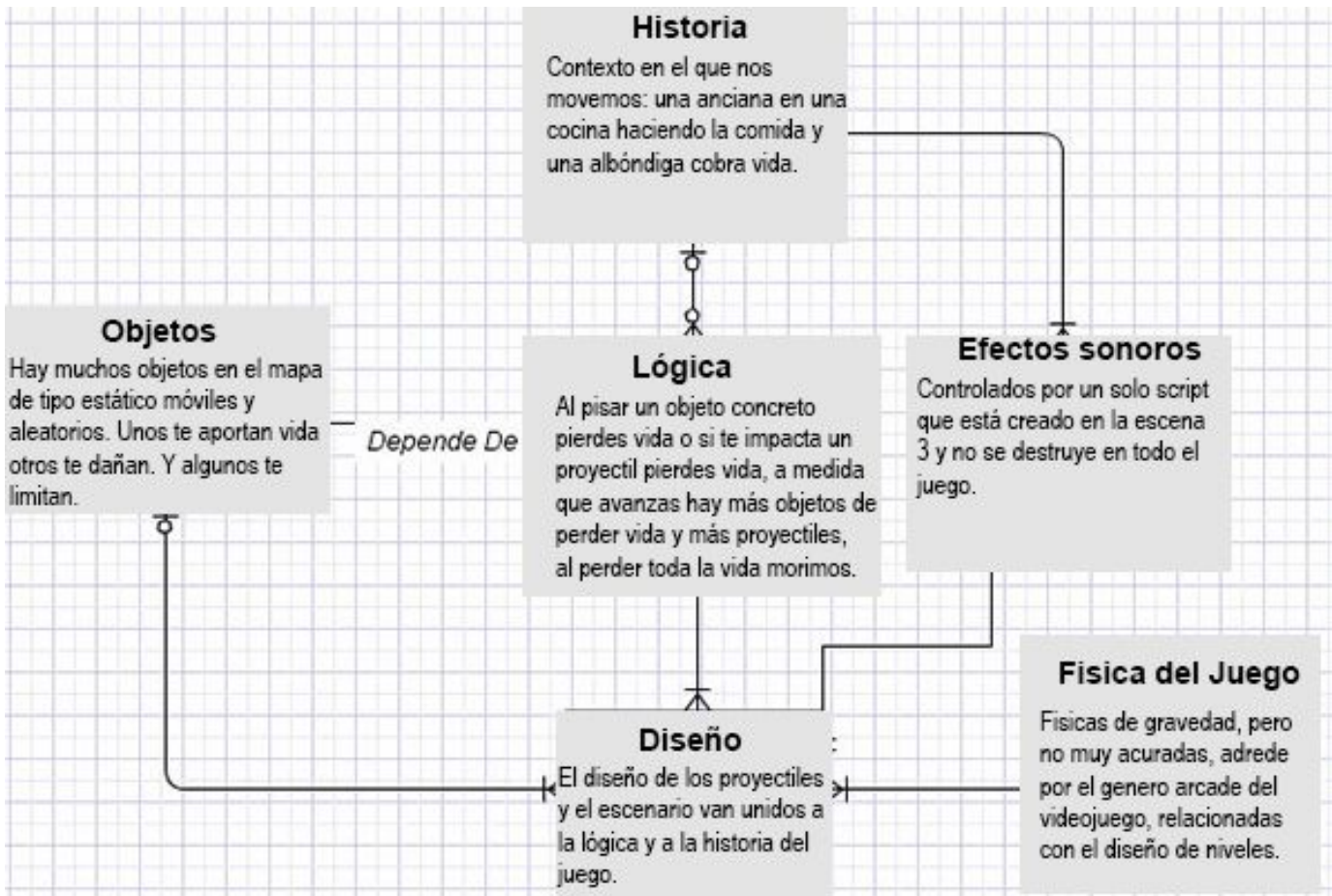
3-Menu

4-Loading

5-Nivel uno

6-Game Over

7-Continuará



En este esquema podemos apreciar la arquitectura de mi videojuego y como está todo entrelazado usando la breve explicación de las escenas de arriba.

La inteligencia artificial de mi videojuego es algo simple. Tenemos una barra en la parte inferior de la pantalla que a medida que nuestra protagonista va ganando volumen y massa, dicha barra va creciendo y vamos cambiando de estado del juego. Nuestra IA tiene 5 estados aquí los enumero y los describo:

1-Estado 0: En este estado la anciana lanza cuchillos a ciegas en un rango cercano a la albóndiga.

2-Estado ojo de águila: Aparte de sumar lo que pasa en el Estado 0, tirará un cuchillo en la última posición de la albóndiga, así si se queda parada le va a impactar el proyectil.

3-Estado cuchillos fantasma: Lanza un cuchillo que cuando impacta salen, a su alrededor, cuchillos más pequeños, pero no caen de arriba hacia abajo, como todos los otros, sino que van en diagonal paralelos al suelo y si unimos sus puntas formarán una circunferencia donde el radio se hace más grande y al final desaparecen los cuchillos. El cuchillo madre se instancia de manera aleatoria alrededor de la albóndiga.

4-Estado milhojas: Este estado instancia un cuadrado lleno de cuchillos, el centro de esta instancia es nuestra protagonista.

5-Estado dentadura viviente: Es una dentadura animada que nos quiere comer se instancia delante de la albóndiga y se mueve hacia delante en el eje Z por lo tanto si seguimos hacia delante colisionamos con este proyectil.

Estos estados se van sumando por lo que en el estado dentadura viviente contendrá todos los demás estados, y los anteriores respectivamente en el orden comentado anteriormente.

5. Diseño del Nivel

El diseño de los niveles me supuso una gran complicación, ya que tenía que adaptar correctamente el nivel al problema del crecimiento de la albóndiga, para que pudiera pasar correctamente por todos los niveles, o también aprovechar y hacer lugares estrechos pero controlados para no frustrar al jugador pero si complicarse a medida que avanza la partida.

Seguidamente muestro el esquema del nivel 1, los números con el recuadro gris que van del uno al cuatro son los subniveles de dicho nivel.

La cruz roja nos indica donde vamos a empezar nuestra aventura.

Los objetos marcados en amarillo son los objetos móviles que deberemos atravesar para acceder al siguiente subnivel.

-Estáticos: Son los decorativos del escenario, te bloquean el paso pero para que no accedas a ciertos lugares, son las limitaciones marcadas por mi. Estos no permiten nunca el paso.

Dentro de estos subniveles aparecerá la carne picada que nos hará recuperar vida y crecer. También se genera aleatoriamente.

6. Manual del Usuario

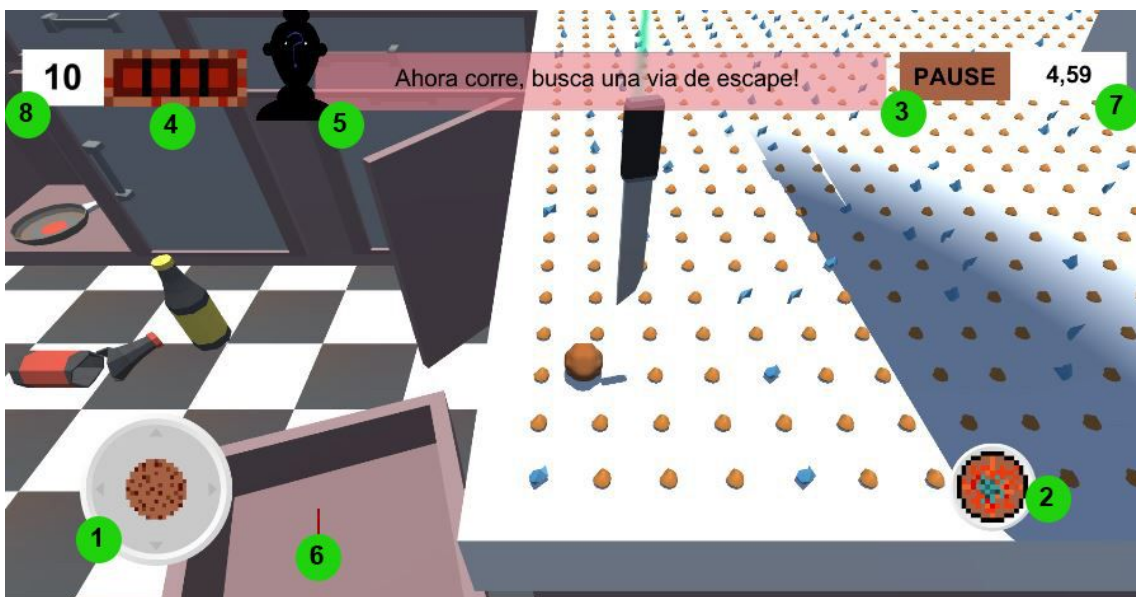
[1]**Citación.*

Los requisitos para ejecutar este videojuego son tener un dispositivo Android con un sistema operativo superior al 4.1; ARMv7 CPU con soporte NEON o CPU Atom; OpenGL ES 2.0 o posterior.

[1]

Lo he estado testeando en varios dispositivos Android y me ha respuesto correctamente tanto en móviles de gama alta como de gama baja.

Los controles del videojuego son sencillos para que puedan ser usados incluso para un niño, los controles son táctiles y eso los hace más fáciles al uso.



-1. Joystick: Con el joystick moveremos hacia donde queramos a nuestra protagonista hacia la dirección del puntero o handler y el handler o puntero lo moveremos con nuestro dedo, preferiblemente y para más comodidad el pulgar izquierdo. Con el handler también marcaremos la dirección del “dash”: arriba abajo derecha o izquierda. Respectivamente en el eje de las “Y” y las “X”.

-2. Botón “Dash”: este botón se pulsa para hacer un “dash” hacia las 4 direcciones arriba abajo derecha o izquierda marcado por el “handler”. Este botón tiene un enfriamiento de unos segundos antes de poder usarlo de nuevo.

-3. “Pause”: EL botón de pause sirve para abrir el menú ingame y pausar el juego. Dentro de este menú encontraremos varios botones:

- Seguir jugando: Desactiva el menú “ingame” y quita la pausa del juego para continuar.

- Opciones: Despliega el menú de opciones que contiene el volumen de la música y los efectos de sonido.

- Menú Principal: Nos lleva al menú principal del juego que contiene los siguientes botones.

 - Jugar: Inicia el juego pero antes pasa por la pantalla de carga.

 - Opciones: Despliega las mismas opciones que en el menú ingame

 - Salir: Salimos de la aplicación.

- Salir: Al pulsarlo se cerrará el juego.

-4. Barra de vida: Esta barra nos indica la vida que tenemos, cuando esta barra llega a cero la protagonista morirá

-5. Cuadro de diálogo: Aquí al pasar de estado en estado nos aparecerá el diálogo de los diferentes personajes.

-6. Barra de “rampage”: es una barra en la que se podrá apreciar como va aumentando la dificultad y nos guiará en el estado del enemigo.

-7. Tiempo: en este recuadro nos aparecerá el tiempo que llevamos jugado en segundos. Al reiniciar la partida o al volver a jugar se reinicia el contador.

-8. Puntuación: en este otro recuadro tenemos la puntuación que estamos consiguiendo en la partida actual.

Al morir iremos a parar a la pantalla de “Game Over” donde nos aparecerá en letras amarillas la máxima puntuación y abajo la puntuación que hemos conseguido en la partida actual. Podremos volverlo a intentar o salir del juego.

7. Conclusiones

En este punto voy a hacer las conclusiones y la reflexión del arduo trabajo realizado en el trabajo final de máster, del máster en desarrollo de videojuegos, nombrado Fatbondiga Arcade. Un videojuego arcade para móviles ha sido el resultado final. Este trabajo es la culminación de todo el proceso de aprendizaje, no solo de este máster, sino de toda mi carrera profesional en el mundo de los videojuegos. Cabe recalcar que el tiempo para elaborar este proyecto no ha sido el óptimo para mi ya que era individual, pero estoy muy contento del resultado obtenido. Lo he dejado probar, a más de cinco personas, y todas se han sorprendido al jugar a un juego echo por mi y aparte de eso, les ha parecido de una alta calidad.

En parte se lo agradezco a los profesores pero también a Unity como una excelente plataforma para que tus ideas se conviertan en videojuego.

En este apartado voy a hacer la conclusión de cada punto de este trabajo.

En el punto estado del arte, he hablado de la revisión del género y he acabado concluyendo que todo lo que he hecho se está haciendo ya a día de hoy Algunos juegos arcade son programados y producidos para dispositivos móviles. No obstante, antes en el pasado, se programaban dentro de motores gráficos propios para acabar siendo producidos y vendidos para máquinas recreativas. He intentado innovar en este aspecto pero al hacer búsqueda de los juegos arcade para móvil he encontrado varios pero ninguno parecido al mi, así pues, en definitiva he creado un juego arcade diferente al resto de juegos arcade para móviles del mercado.

En el segundo punto, definición del juego, relato todo lo referente al videojuego tanto la historia como la jugabilidad, las mecánicas, el arte etc. Es el GDD (documento de diseño del videojuego). Es el documento que marca la idea del mi videojuego antes de ponerme a producirlo para tener una idea global de lo que quiero que acabe siendo mi videojuego. Podríamos decir que es mi guía de producción, la cual me costó más de un mes elaborar al tener la dificultad de crear algo nuevo y ser creativo. Al no querer copiar una arquitectura de juego o un juego que ya funcionará.

En el siguiente punto, diseño técnico, narro todo lo referente al entorno usado, en mi caso, Unity y los requerimientos técnicos de dicho motor, aparte también

el inventario de todo el material que he reaprovechado de la red y el que he creado yo. Las herramientas utilizadas y expongo el esquema de la arquitectura que me costó bastante a la hora de hacer el proyecto de estructurarlo y finalmente hablo de mi inteligencia artificial y sus 5 estados: estado cero, vision de aguila, cuchillos fantasma , milhojas y dentadura viviente. Fue todo un reto imaginar y crear dichos elementos y que concordarán con el nivel a parte de aportar humor al juego.

En penúltimo lugar tenemos el diseño de niveles, que en definitiva, es un nivel dividido en cuatro subniveles, los cuales están delimitados hasta conseguir una cierta masa para poder moverlos y pasar de subnivel a subnivel. El diseño del nivel me llevo complicaciones al principio, ya que nuestra protagonista crecía a medida que avanzabamos en el nivel. Entonces tuve que ajustar muchos parámetros y mover y rediseñar la forma del nivel para aprovechar dicho problemas, convertí el problema en una dificultad a superar y me lo llevé a mi terreno.

En el último lugar tenemos el manual del usuario que en resumidas cuentas son los controles y los requisitos para poder jugar a Fatbondiga arcade. Los rasgos más importantes de los requisitos es tener un dispositivo Android. Los controles, los he hecho muy intuitivos y fáciles de manejar hasta para los más pequeños un Joystick y un botón de “dash”.

En definitiva, ha sido una experiencia muy gratificante hacer este trabajo y sentirme orgulloso de mi mismo viendo el potencial que he adquirido al poder hacer un videojuego de este calibre con mis propias manos. yo me defino como un “ 3D generalist artist”, pero ahora también soy un desarrollador de videojuegos. Cuando hablo de mi videojuego se me pone la piel de gallina y los sentimientos fluyen, es como mi creación. Os agradezco a todos que me habéis brindado esta oportunidad para demostrarme a mí mismo y a los demás de lo que soy capaz si me esfuerzo lo.

Este trabajo me ha hecho ver muchas cosas pero sobretodo que puedo mejorar en muchos aspectos y ahora conozco mis debilidades y poder trabajarlas para ser mejor desarrollador de videojuegos, todo esto lo agradezco a todos los profesores que he tenido ya sean los del máster en desarrollo de videojuegos como de estudios anteriores. Muchas gracias.

8. Glosario

-“Dash”: desplazamiento repentino hacia la dirección estipulada.

-“Rampage”: en español furia, este término se usa en los videojuegos para cargar la barra de poder de los personajes.

-“Ingame”: en español en el juego, este término se usa para definir lo que pasa dentro del juego.

-“GamePlay”: este término significa todo lo referente a los videojuegos cuando los estas jugando.

-“GUI”: es la guía para el usuario, le aporta información y también se puede interactuar con ella. Suelen ser gráficos simples.

5. Bibliografía

1-Unity3d:<https://unity3d.com/es/unity/system-requirements> 25/06/19

2-Autodesk:<https://www.autodesk.es/products/maya/overview> 26/06/19

3-Androidsis:<https://www.androidsis.com/los-mejores-juegos-de-arcade-para-an-droid/> 26/06/19