

Arcade Last project: Memoria del TFM

Nombre Estudiante: Beatriz Zallo Hernández

Plan de Estudios del Estudiante:

Master en Diseño y desarrollo de videojuegos

Área del trabajo final

Nombre Consultor/a: Helio Tejedor Navarro, Jordi Duch Gavalrà

Nombre Profesor/a responsable de la asignatura: Joan Arnedo Moreno

Fecha Entrega: 16 de junio de 2019



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

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Arcade Last project</i>
Nombre del autor:	<i>Beatriz Zallo Hernández</i>
Nombre del consultor/a:	<i>Helio Tejedor Navarro, Jordi Duch Gavaldà</i>
Nombre del PRA:	<i>Javier Luis Cánovas Izquierdo</i>
Fecha de entrega (mm/aaaa):	06/2019
Titulación:	<i>Postgrado Diseño y desarrollo de videojuegos</i>
Área del Trabajo Final:	<i>Trabajo final de master</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Videojuego, Gestión, Recreativas</i>

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

En este proyecto se ha planteado un videojuego de micro gestión donde el jugador tiene una tienda donde poder hacer resurgir las salas de máquinas recreativas. Donde tenemos la posibilidad de comprar más maquinas con características diferentes, y mejorar nuestro establecimiento y gestionar nuestra tienda para no llegar a la bancarrota.

Se ha seguido una metodología Scrum para su desarrollo, y se ha utilizado el motor de Unity para un desarrollo ágil. La finalidad es poder practicar, ver de primera mano los desafíos que implica desarrollar un juego de gestión y sus diferentes vertientes a la hora del desarrollo.

Se ha desarrollado la funcionalidad core del juego, con su sistema correspondiente de clases y distintos tipos de eventos. Además de un diseño de interfaz, logo y nombre, propios para el juego. Para el desarrollo del mismo se han utilizado distintos tipos de herramientas, desde SmartGit, hasta Rider como IDE.

Como conclusión, este proyecto ha servido a plantear estructuras de datos generadas de forma procedural con interacciones diferentes unas con otras y ha supuesto un reto con el uso del sistema de navegación e IA del motor. Y como resultado un producto base para poder seguir trabajando en ello, y en un futuro poder publicarlo.

Abstract (in English, 250 words or less):

This project is micromanagement videogame where the player has a store and he has the change to revive the arcade machine time. We have the possibility to buy more machines with different characteristics, and improve our establishment and manage our store to avoid bankrupt.

For this development we follow the scrum methodology, and I have been using the unity engine v.2018.3.11. The purpose is to practice, see first-hand the challenges involved in developing a management game and its different aspects of the genre.

I have developed the core functionality of the game, with it is corresponding class system and every kind of event available. Moreover, a logo and User Interface design has been made, according to the game genre. As for the development of the game, several software tools have been used, as SmartGit, or Rider as an Integrated Development Environment.

To Sum up, this project has been useful to the end of generating procedurally generated data with different interaction between each other and it's been a challenge with the navigation system and the engine's AI. As a result, a product to keep developing has been made, ready for the continuous work on it and a possible publish the videogame in the future.

Índice

Trabajo Final de Máster	iii
1. Introducción	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo	2
1.3 Enfoque y método seguido.....	3
1.4 Planificación del Trabajo	3
1.5 Breve resumen de productos obtenidos	4
1.6 Breve descripción de los otros capítulos de la memoria	4
2. Metodología de trabajo.....	5
3. Presupuesto.....	5
4. Estado del arte	7
4.1. Género de videojuego	8
4.2. Subgéneros de Construcción y Gestión	8
4.3. Historia del Género	9
4.3.1. Hamurabi	9
4.3.2. Utopia	10
4.3.3. SimCity	10
4.3.4. Theme Park	11
4.3.5. Anno 1602	12
4.3.6. Saga Trópico	13
4.3.7. Cities: Skylines	13
4.3.8. Game Dev Tycoon	13
4.4. Motor de juego	14
4.4.1. Unreal	14
4.4.2. CryEngine	15
4.4.2. Unity	15
4.5. Tecnologías complementarias.....	16
4.5.1. Photoshop	16
4.5.2. Blender (2.8)	16
5. Definición del juego	17
5.1. Descripción del juego	17
5.2. Subgénero y referencias	17
5.3. Tipo de interacción juego-jugador	17
5.4. Plataforma de destino	18
5.5. Ambientación	18
5.6. Elementos e Interacción	19
5.7. Objetivos planteados al jugador	20
6. Ruta de desarrollo.....	21
6.1. Pre producción	21
6.2. Producción	21
7. Diseño técnico	22
7.1. Entorno y requerimientos	22
7.2. Herramientas empleadas y assets	23
7.2.1. Herramientas	23
7.2.2. Assets	24

7.3. Esquema de la arquitectura.....	27
7.3.1. IA	30
8. Diseño de niveles	32
9. Manual de usuario.....	33
9.1. Requerimientos Técnicos	33
9.2. Manual de usuario.....	33
10. Conclusiones	34
11. Glosario.....	35
12. Bibliografía	36

Lista de figuras

Figura 1 – Grafica de los juegos más vendidos en pc	1
Figura 2 – Ejemplo de editor de los sims 4.....	3
Figura 3 – Distribución de las tarjetas de trello.....	3
Figura 4 – Grafico sobre las ventas de Videojuegos	7
Figura 5 – Ciudad SimCity	11
Figura 6 – Portada del Themepark.....	11
Figura 7 – Anno 1602.....	12
Figura 8 – Game Dev Tycoon	13
Figura 9 – Trabajo de capas en photoshop	16
Figura 10 – Ejemplo de recreativas.....	19
Figura 11 – IDE de desarrollo	22
Figura 12 – Acceso directo al debugger de Rider.....	22
Figura 13 – Representación de gitflow	23
Figura 14 – Tablero trello del trabajo.....	24
Figura 15 – Pagina de Kenney.....	24
Figura 16 – Modelos creados.....	25
Figura 17 – Interfaz dentro del juego.....	25
Figura 18 – Interfaz de compra	26
Figura 19 – Autor de la música que se ha usado	26
Figura 20 – Imagen del paquete de recreativas de Asset Store	26
Figura 21 – Diagrama de clases.....	27
Figura 22 – Flujo del programa	28
Figura 23 – Estructura de las carpetas Unity.....	28
Figura 24 – Estructura de los componentes de la escena principal	29
Figura 25 – Diagrama de flujo del cliente	30
Figura 26 – Escenario dentro del Motor	32
Figura 27 – Escenario dentro del juego.....	32

1. Introducción

1.1 Contexto y justificación del Trabajo

Al tratarse de un género de gestión y de un tema tan concreto como lo son las recreativas estamos enfocando el juego a un nicho de mercado concreto. La necesidad a cubrir es la diversión del público al que va dirigido en este caso es gente ya adulta que pasó su infancia en las recreativas o tienen recuerdo de ello. En nuestro País (España) este fenómeno pasó al olvido, pero en otros países como América o Japón siguen estando a la orden del día. Por esto mismo y ya que no hay un juego actualmente de este enfoque se ha planteado debido al auge de los juegos de simulación y gestión.

Lo que se espera es que la gente que se divierte con los juegos de gestión y quiere recordar su tiempo en las recreativas tenga una oportunidad con este juego. Como dato a incorporar dentro de los juegos más vendidos en pc tenemos la siguiente grafica que nos muestra como la estrategia y la simulación son los géneros por excelencia en el pc y queda justificado la plataforma elegida.

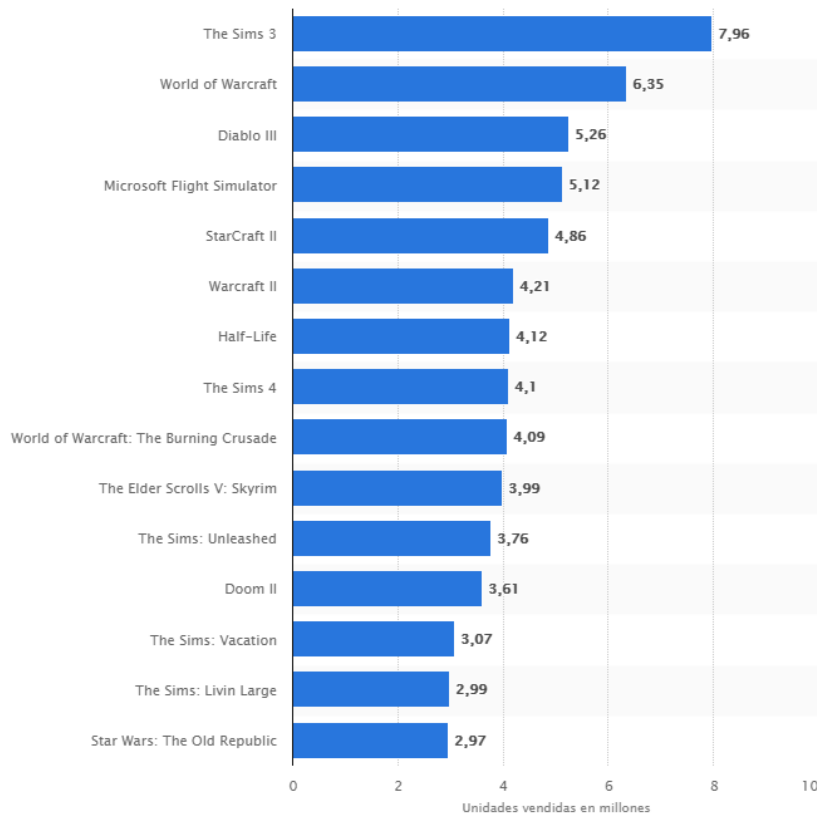


Figura 1 – Grafica de los juegos más vendidos en pc

1.2 Objetivos del Trabajo

Los objetivos del trabajo a desarrollar dentro del juego, son los siguientes puntos marcados:

- Compra de objetos, el jugador deberá de ser capaz de comprar elementos para su negocio ya sean decorativos como máquinas que le permitan, conseguir beneficios para el negocio.
- Venta de objetos, el jugador deberá de ser capaz de deshacerse de los objetos que quiera vendiéndolos, pero no por el precio que les costó, tiene que haber una penalización para que se piense dos veces el jugador antes de comprarlo.
- Sistema de economía simple, ganar dinero al día restar en función del alquiler, objetos y distribuidoras.
- Sistema de contratos con las distribuidoras (simple), aceptar misiones, rechazarlas y finalizarlas. Ya sea porque lo ha terminado en el periodo de tiempo, o porque no lo ha cumplido. Solo se plantea un tipo de misión para simplificar el desarrollo.
- Sistema de calendario, base simple solo para representar los días y el cambio de día y noche.
- Sistema de uso de las Maquinas, lógica de uso y captación de clientes.
- Sistema de IA de los clientes, con sus necesidades, navegación y obstáculos dinámicos.
- Diseño de entorno, con su respectiva representación de la ciudad.
- Diseño completo de interfaz, con términos de usabilidad y sencillez dejando la vista a lo que pasa en la tienda.

Otro claro objetivo dentro del trabajo de fin de master era hacer un desarrollo escalable con las herramientas nuevas que nos proporciona el motor en la versión que se está utilizado 2018.3.

1.3 Enfoque y método seguido

Al tratarse de un género ya trabajado por muchas compañías, ciertas mecánicas han sido basadas en otros juegos, pero teniendo una base temática que no hay en ningún otro. Mecánicas como un sistema simplificado de compra, y resolución de una IA con necesidades son partes necesarias de este género que son comunes y no puedes evitar tener que adaptarlas al tuyo. Como se ve por ejemplo en los sims su forma de edición y compra de objetos.



Figura 2 – Ejemplo de editor de los sims 4

De cara a cómo avanzar para realizar el desarrollo he basado en lo mostrado en clase, e ir incorporando mecánicas base poco a poco. Haciendo modelos de whitebox para hacer pruebas de funcionalidad.

1.4 Planificación del Trabajo

La planificación seguida durante todo el proyecto ha sido prácticamente la mencionada en las anteriores prácticas, una fase de preproducción donde se realizaban los componentes gráficos, modelajes del proyecto y diseño de interfaz.

Después de tener los componentes se ha ido desarrollado cada parte de forma organizada, en la fase de producción del juego donde ha entrado como recurso necesario Unity tanto como motor como librería de consulta a la hora de necesitar información de cómo utilizar ciertas partes del motor, se ha utilizado trello para revisar tareas y seguir una lista de tareas pendientes.

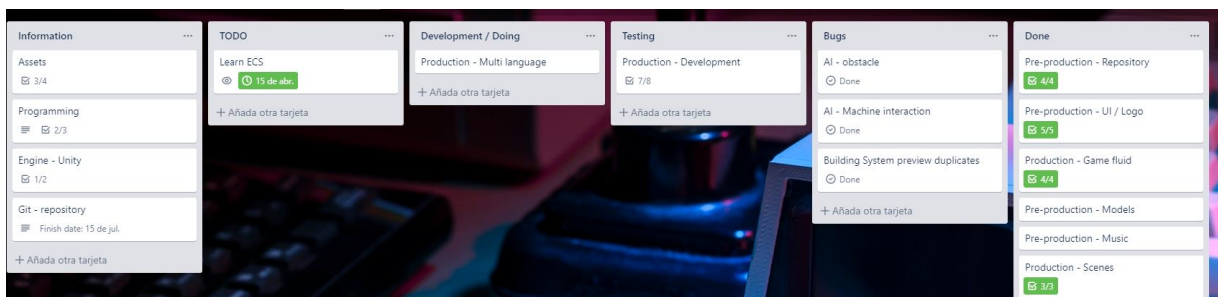


Figura 3 – Distribución de las tarjetas de trello

Dentro del desarrollo se han separado las funcionalidades, estando como la interacción con la UI y el flujo con el juego y las escenas. Por otro lado, la representación de la ciudad del juego, y los recursos que puede usar el jugador.

Después se realizará un sistema de eventos para la interacción con el calendario ya que es la base del sistema del tiempo del juego. La tarea siguiente la IA de los clientes que gastan su dinero en la tienda, y el sistema de compra y venta de objetos para el jugador. Por ultimo como final añadido un sistema de misiones con distribuidoras si diese tiempo después del resto de desarrollo.

Para poder seguir un desarrollo continuado y con fluidez se han utilizado las siguientes herramientas de trabajo:

- Git: integrado con SmartGit para mejor uso de gitflow y su respectivo versionado.
- Trello: Como mostramos el desarrollo de y su proceso por medio de las tarjetas.
- Unity: como motor que nos ayudara en el desarrollo del juego, abstrayéndonos de lo menos importante.

1.5 Breve resumen de productos obtenidos

Para este desarrollo lo que obtendremos es una alfa técnica del juego que se ha planteado ya que se ha reducido la funcionalidad al core del juego. Con su respectiva marca y nombre propio; Arcade Project es un juego en el que comienzas con unos pequeños ahorros que te permite tener un local alquilado y unas pocas máquinas de segunda mano que te encuentras en una tienda vieja del antiguo dueño.

1.6 Breve descripción de los otros capítulos de la memoria

En los siguientes capítulos de la documentación se va a narrar los diferentes aspectos a la hora de desarrollar un juego, así como el apartado técnico además también la investigación de cara a reconocer lo que hay en el mercado y saber cómo visualizarlo y poder aplicar mejoras.

Hablando sobre la tecnología y el género en cuestión sus ventajas e inconvenientes de hacer un juego de este tipo. Definición concreta de lo que es el videojuego que estamos explicando, y en el apartado técnico que se ha realizado y como ha sido su distribución. Por último, pero no menos importante el diseño de niveles y un breve manual de usuario.

Cabe destacar que se ha desarrollado de forma que se ha planteado como si fuese un juego a publicar con su respectivo presupuesto y salario de cara a la publicación a inversiones o a posibles ayudas con la investigación del mercado actual.

2. Metodología de trabajo

A la hora de realizar un trabajo ya sea para una empresa o para uno es conveniente realizar ciertas pautas o metodologías que nos faciliten una fluidez constante y capacidad de retro alimentarnos respecto a lo que vamos haciendo y nos damos cuenta de ciertas decisiones.

Por esto mismo una de las mejores metodologías de desarrollo y que está en boca de todos en la actualidad es la metodología ágil en este caso la que se denomina Scrum. Como experiencia personal en mi entorno laboral, ha sido la forma más cómoda y fructífera de cara al cliente ya que nos permite seguir un flujo de desarrollo constante y estable.

Los puntos críticos han sido las entregas parciales, que nos permiten ver a groso modo los avances y los cambios en las estimaciones y objetivos planteados. Internamente cada vez que se desarrollaba una nueva mecánica se volvían a analizar las demás de cara un posible cambio para adecuar su balance y/o problemas ocasionados. El testeo integro después de terminar una funcionalidad, es importante para verificar que lo anterior no falle por la integración de las nuevas funcionales, y esto es ve complementado con un control de versiones y la metodología ágil.

3. Presupuesto

En este apartado se ha querido suponer lo que sería partir de un desarrollo completo de un juego hasta su publicación con un supuesto, revisando precios del entorno, salarios y licencias necesarias para poder publicar sin ningún problema ni contratiempos.

Para la parte de desarrollo del juego tendríamos por una parte el software que consta de:

- La licencia de Unity para los estudios que son 125 € al mes.
- En principio para temas de interfaz y diseño 2D tendríamos el photoshop que serían unos 29.99 € al mes.
- Sistema operativo del ordenador W10, ya que la herramienta de trabajo cuenta para el coste de presupuesto, que sería 135€ la licencia.
- Y para tema de documentos, office que son 150€ la licencia completa.

Por otro lado, es necesario de una herramienta de trabajo y en nuestro caso es el ordenador. Al tratarse de desarrollo de videojuegos y necesitar correr un motor a la vez que un juego requiere de un gasto mayor de recursos y por lo tanto un ordenador de gama alta por lo que rondaría el gasto entre los 900 – 1200 €.

Además de un gasto en infraestructura lo que sería coste de oficina, internet, y sueldo del trabajador. Todo esto dentro del supuesto de una publicación y un desarrollo de una nueva empresa enfocada en el producto que está desarrollando. Los costes de las líneas de internet esta alrededor de unos

60 – 100 €, sumándole una necesidad de una línea de teléfono, y el alquiler de una oficina pequeña ronda sobre los 400€ o menos si queremos abaratar el coste y se realiza desde el hogar.

Por último, el sueldo del desarrollador, estaría entorno a los 1.600€ (por un total de 2420 horas estimadas) reduciendo lo posible para poder reducir gastos, esto nos quedaría con una tabla reducida de supuestos gastos totales como se puede ver en la siguiente tabla. Al gato total del proyecto se ha sumado tiempo invertido en campañas de crowdfunding¹, publicidad y tiempo de trato con usuarios y feedback² por su parte.

Gastos de personal	19.200 €
Gastos generales	450 € al mes.
Gastos de infraestructura	1100 + 400€ al mes
Gasto TOTAL del proyecto:	45000 €

¹ Es el término inglés para referirse a financiación colectiva, es decir, a un grupo de personas apoyando económicamente a un proyecto.

² Termino en ingles usado para referirse a la comunicación entre dos puntos en este caso el desarrollador con los usuarios.

4. Estado del arte

El videojuego hoy en día se ha dejado de ver como mero entretenimiento para un grupo minoritario, a ser considerado una forma de arte, y uno de los sectores que más beneficios aporta.

La industria factura millones de dólares en todo el mundo, en 2018 facturó \$134.9 billones, un aumento de 10,9% respecto del año anterior. Además, teniendo en cuenta que Europa es de las regiones más importantes en lo que refiere al consumo de videojuegos a nivel mundial.

Más concretamente los ingresos por plataforma, como podemos ver en el gráfico inferior como el campo de PC tiene la mayoría de los ingresos 25%, por detrás las consolas con 28%. Las grandes empresas, como PlayStation, Microsoft... además están generando ayudas para la creación de videojuegos “indies”, juegos de empresas que no tienen capacidad financiera, y de los campos mostrados en el gráfico el que ofrece más facilidad para salir al mercado, es en el sector de PC.

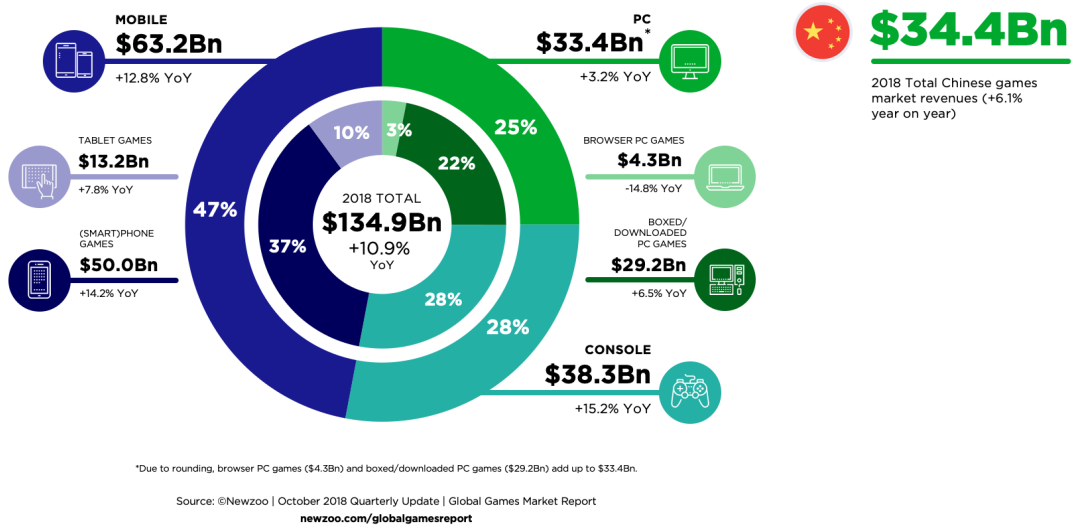


Figura 4 – Gráfico sobre las ventas de Videojuegos

La industria del videojuego es un sector que crece a un ritmo constante, siendo el pc el mejor sector para empezar como desarrollador indie. La justificación de realizar un videojuego en ordenador a parte de por la facilidad de publicación que hay con los sistemas que se mencionan después, es la dificultad y el ingreso monetario inicial que hay para realizarlo en otras plataformas como es PlayStation o Microsoft. Dentro de la plataforma de PC tenemos sistemas como Steam o ahora en la que está en boca de todos Epic Games por ofrecer mejores precios a las empresas que quieren empezar, y plataformas como KickStarter para la financiación y ayudas.

4.1. Género de videojuego

En el género de gestión y construcción dentro de los videojuegos, los jugadores deben hacer uso de una serie de recursos limitados a su disposición para construir o expandir cierto tipo de comunidad, institución o imperio mientras se hace cargo de las dificultades internas (tales como el crimen en “SimCity” o los empleados que te roban dinero en “Los Sims 3”.) y externas (como los desastres naturales en “SimCity” o “Cities Skylines”) mientras se compite contra otros jugadores o máquinas dentro de la partida.

Los juegos de un solo jugador de estas categorías tienen como objetivo ver cómo la institución que controla la persona crece y va cambiando a lo largo del tiempo, mientras que los juegos de múltiples jugadores suelen tener el objetivo de controlar al resto para terminar la partida.

Visto esto, las características que definirían un juego de gestión y construcción serían las siguientes:

- Gestión de recursos limitados.
- Solución a problemas encontrados a lo largo de la partida.
- Capacidad para evolucionar y expandir tu institución.

4.2. Subgéneros de Construcción y Gestión

Dentro del género de construcción y gestión existen varios tipos de subgéneros entre los que se componen el mercado del mismo, son los siguientes:

- **Juegos de creación de ciudades:** En la que el jugador interactúa como gobernante de la ciudad y mediante una vista aérea de la misma, el mismo controla el crecimiento y desarrollo de la metrópolis. Los jugadores suelen atenerse a realizar labores de gestión tales como modificar salarios, construir edificios, mientras que los ciudadanos que son PNJ (Personajes no jugables) realizan el trabajo de construcción. En este subgénero entrarían: Utopia, SimCity, Caesar, Anno 1602, Cities Skylines.
- **Juegos de creación de parques temáticos:** Este subgénero tal y como su nombre indica consiste en la creación y mantenimiento de diversos parques temáticos, tal y como veremos más adelante, resultó en un completo éxito y dio al género un aumento de número de usuarios en sus inicios. Los juegos más emblemáticos de este subgénero serían: Theme park.
- **Juegos de simulación de negocios:** El subgénero de simulación de negocios se centra en gestionar y simular distintos tipos de empresas o economías dentro del juego, normalmente se centran más en la gestión que en la construcción, a diferencia de otros subgéneros. Esto hace que el usuario tenga decisiones más complejas que tomar que le recompensará / castigará de forma significativa. Entre los juegos del subgénero podemos

contar: Theme park, Transport Tycoon, Capitalism, Monopoly Tycoon, Tycoon city: New York, IndustryPlayer

- **Juegos de simulación de gobiernos:** En el que el jugador representa una institución política o gobierno de un país, en este subgénero los jugadores tratarán de tomar las decisiones burocráticas que crean correctas para una parte de la nación o la nación al completo. Estos juegos podrían incluir situaciones geopolíticas o simulación de campañas políticas. En el subgénero de gobiernos nos encontraremos los siguientes títulos: Hamurabi, SimCity, Caesar, Monopoly Tycoon, trópico.
- **Juegos de gestión de deportes:** En este subgénero de gestión los jugadores tomarán parte de la junta directiva o el control de una institución deportiva o un equipo de un deporte en concreto. La diferencia de los videojuegos dentro de este subgénero será en sí el videojuego en sí se centra en jugar a ese deporte, o simplemente la toma de decisión y gestión del mismo. Así mismo, este subgénero ha sido uno de los más populares en la historia, por la propia competitividad que genera, al igual que los deportes que suelen representar.

El videojuego que se plantea en este proyecto, dentro de este género, ocupa el subgénero de simulación de negocios, principalmente. Como añadido a futuro en la construcción de ciudades, viendo lo planteado a posteriori.

4.3. Historia del Género

Para esta documentación técnica se ha hecho un análisis de los juegos de gestión que ha habido y cuales han sido los más importantes a lo largo de todo este tiempo.

4.3.1. Hamurabi

El primer juego conocido con mecánicas del género de gestión fue “The Summer Game”, o “Hamurabi” tal y como se llamó más adelante. Es un juego basado en texto desarrollado por Doug Dyment en 1968. El juego comenzó como un software para niños de gestión de recursos en lenguaje FOCAL y más adelante se comercializó su uso en una versión programada en BASIC bajo el nombre Hamurabi.

En el juego el jugador encarna al rey babilónico Hammurabi, y es llamado a tratar un problema de gestión de recursos durante 10 años de mandato, durante los cuales tendrá que gestionar las tierras, las plantaciones y los trabajadores que las llevan para evitar que el pueblo muera de hambre y seguir reinando.

Tal y como hemos mencionado anteriormente, el juego se considera el primer videojuego de gestión de la historia, y uno de los primeros juegos que existió. Ha dejado un legado de juegos que se basan en este, tales como “Kingdom” (1974), M.U.L.E (1983) o Anacreon (1987).

4.3.2. Utopia

Nos tenemos que ir hasta la década de los 80 no solo para ver los siguientes videojuegos de gestión, si no para ver el “boom” de los videojuegos en sí, en 1981 es cuando sale a la luz el videojuego Utopía, el primero videojuego oficialmente bajo el género de simulación, creado por Don Daglow para Intellivision.

Utopía es un videojuego de dos jugadores en el que cada uno controla una isla y va gestionando por turnos y gastando el oro disponible ya sea construyendo edificios en su isla o apoyando la actividad rebelde en la isla del contrincante. Este juego no disponía de una IA, por lo que el modo de un solo jugador no tenía gran desafío.

Este videojuego fue el primero en incluir mecánicas de turnos en él, ayudando a la fundación de los juegos de estrategia en tiempo real, pero además fue el primero en incluir mecánicas de gestión de ciudadanía y cámara isométrica (god games).

4.3.3. SimCity

En 1989 salió el primero de una larga saga de juegos de gestión de ciudades, SimCity, fue el juego que revolucionó el sector de la gestión simulada, fue creada en principio para la Commodore 64 en 1985, pero no fue publicada por la recién fundada Maxis hasta 4 años después.

En SimCity el jugador comenzará una partida con un solar vacío y tendrá que construir una ciudad y gestionarla desde 0, en este juego se introducirán también las primeras consecuencias “Externas” como tornados o inundaciones. El jugador comenzará con unos recursos limitados y tendrás que obtener más por medio de impuestos, sus acciones tendrán consecuencias en la propia ciudad y los habitantes que viven en ella.

SimCity es una referencia para el sector y la industria en sí, introdujo mecánicas de gestión de ciudades que se siguen utilizando hoy en día y revolucionó los juegos existentes, ya que añadió una visión de juego propia, en la que el jugador no tiene un objetivo fijo, y el jugador no dispone de unos requisitos para ganar o perder, ya que no existe una partida ganada o perdida como tal. Esta visión hizo que las mayores empresas de la industria le dieran la espalda durante 4 años, hasta que Maxis obtuvo el juego.



Figura 5 – Ciudad SimCity

Más adelante, la saga SimCity publicaría los juegos de SimCity 2000 (1994), SimCity 3000 (1999), SimCity 4 (2003) y SimCity 5 (2013). Durante estas entregas, los creadores obtuvieron un éxito en el género incluyendo nuevas mecánicas como el tratado de residuos, gráficos 3D, implementación de cambios climáticos e implementación de funcionalidades multijugador, entre otros.

4.3.4 Theme Park

Theme Park es un videojuego de gestión y construcción publicado por Electronic Arts en 1994. Es uno de los primeros videojuegos centrados en la gestión y desarrollo de un negocio, dado que en el juego el jugador toma el papel del dueño de un parque de atracciones y mediante sus decisiones tendrá que hacer que éste prospere. El juego contiene micro gestión basada en los recursos disponibles (el dinero que los visitantes le van dando al parque) y un método de calificación de las acciones que toma el jugador basado en las valoraciones que le dan los visitantes del parque.



Figura 6 – Portada del Themepark

El juego, tal y como vimos en SimCity y mayormente la rama de gestión y construcción, no disponía de un objetivo final, ya que el objetivo principal del usuario era construir parques de atracción a lo largo del mundo y hacerlos rentables para generar más ingresos.

Las mecánicas del juego proponen al usuario tres vías de generar dinero. Obtener dinero mediante entradas a las atracciones, vendiendo merchandising del parque, o haciendo que los visitantes gasten dinero en los puestos de comida del mismo. Este dinero tendría que reinvertir en el salario de los trabajadores y los gastos del propio parque, así como para su futura expansión.

Los sucesores de Theme Park fueron Theme Hospital, que salió en 1997 y abordaba la temática de gestión de hospitales y Theme Park Inc. en 2001.

4.3.5. Anno 1602

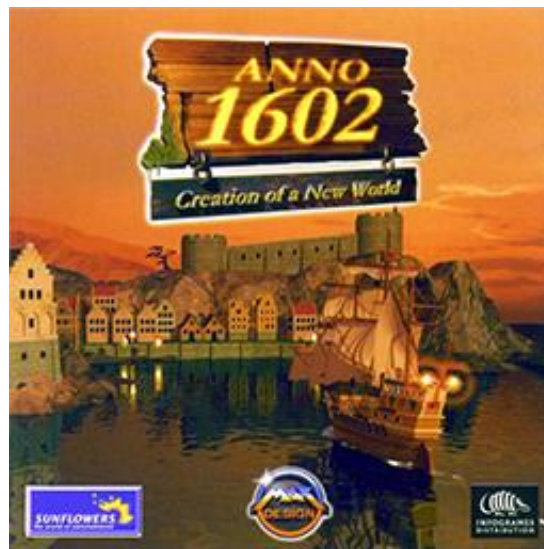


Figura 7 – Anno 1602

Anno 1602 es un videojuego de construcción de ciudades que se publicó en 1998 por Sunflowers entertainment. En él, el jugador tenía que ponerse al mando de un grupo de comerciantes europeos que buscaba establecer su colonia en el nuevo mundo.

Dado el ambiente en el que se centra el juego, los desarrolladores decidieron dotar a este simulador de un sistema de comercio en el que se basaría toda la experiencia de juego del usuario. De forma que la diplomacia, exploración e incluso gestión de la colonia, se verá alterada por las decisiones que el usuario tome en el comercio establecido.

En Anno 1602, se incluye una IA renovada, en la que esta actuaba conforme las acciones del jugador y como las tomase. Además, en este juego se añade la función de multiplayer local de 4 personas.

La saga de Anno comenzó con este juego, pero más adelante sacó títulos de estrategia como Anno 1503 (2003), Anno 1701 (2006), Anno 1404

(2009), Anno 2070 (2011), Anno 2205, en 2015 y recientemente el último título de la saga, Anno 1800 en 16 de abril de 2019.

4.3.6. Saga Trópico

Trópico fue publicado por Gathering of developers en 2001 para PC y Mac, en el que el jugador encarnaba a “El presidente”, el mandatario de una isla caribeña durante la guerra fría.

El título fue un éxito debido a las mecánicas nuevas que añadía, al ser una sátira de las repúblicas bananeras existentes en los años 50 en adelante, el jugador disfrutaba de una gestión de una ciudad basada en el totalitarismo. La corrupción y el fraude ayudaban al jugador a hacerse cargo de la isla mientras se enfrentaba tanto a peligros dentro de la isla (rebeldes que podrían hacer incursiones a la ciudad) como fuera.

Dada la aceptación recibida por el público en su salida, trópico tuvo una expansión, “Paradise Island”, además de sucesores de la saga: Trópico 2 (2003), Trópico 3 (2009), Trópico 4 (2011), Trópico 5 (2014) y su sexta entrega de este 29 marzo de 2019.

4.3.7. Cities: Skylines

Cities Skylines es un videojuego de creación de ciudades publicado por Paradox interactive en 2015. Aunque tuvo poca expectación debido a la dominancia del mercado por parte de SimCity desde su primera entrega, Cities Skylines fue bien acogido, en parte al fracaso de la quinta entrega de SimCity y a la idea principal del juego presente en todos sus ámbitos, una ciudad viva.

4.3.8. Game Dev Tycoon

Un juego nacido en 29 agosto de 2013, concentrado en la micro gestión de una empresa de videojuegos, en su desarrollo y su publicación así como la gestión de las opiniones de los medios de las mejores formas de realizar uno.

Al tener un buen sistema de micro gestión te permite estar concentrado y entretenido a lo largo del tiempo. Y al final del transcurso de la partida te va añadiendo nuevos retos e investigaciones para poder mantener un flujo constante en el jugador. Y estas y otros detalles fueron clave de su éxito.



Figura 8 – Game Dev Tycoon

4.4. Motor de juego

El mercado de los videojuegos está repleto de juegos tanto de pago como gratuitos, y lo que tienen todos en común es que están siendo ejecutados por un motor de juego. El motor de juego influye muchas veces en el resultado final del propio juego, ya que de él depende el formato y la estructura de los gráficos utilizados en el juego y actúa como un pivote alrededor del cual el flujo de trabajo funciona.

Para este proyecto se han analizado los motores que hay y se ha hecho una pequeña síntesis de cada una de ellas y se ha elegido como se ha dicho con anterioridad Unity como motor para este desarrollo.

4.4.1. Unreal

Unreal engine fue lanzado al mercado en 1998 por la compañía “Epic Games”, este motor de juego, programado en C++, cada día crece con más comunidad a su favor. Aunque su primera aparición fuese para el shooter “Unreal” en la misma fecha de su lanzamiento, se ha utilizado en multitud de videojuegos de distintos géneros a lo largo de los años, haciendo de este, un motor versátil.

Aunque Unreal siempre ha sido un motor con una curva de aprendizaje compleja para el usuario novato, la creación de la abstracción por bloques o “prototipado” ha favorecido que gran parte de los diseñadores puedan manejarse con el programa sin necesidad de tocar código. Además, unreal es el motor por defecto si lo que deseas es optimizar tu proyecto al máximo.

Por otro lado, para poder utilizarlo necesitarás una copia con licencia, y en caso de publicar un juego, tendrás que pagar un royalty del 5% de tus ingresos. Además, este motor de juego está pensado para proyectos grandes con equipos de personas especializadas, haciendo que los pequeños desarrolladores eviten este motor.

4.4.2. CryEngine

CryEngine fue publicado por la empresa CryTek en 2002 como demostración para la empresa de Nvidia. Al demostrar el gran potencial que tenía este motor, decidió implementarse por primera vez en Far Cry. La versión actual de CryEngine es la 5 y se dio a conocer en 2016.

El punto fuerte de CryEngine es la potencia del motor, los gráficos y las físicas que puedes desarrollar con este motor eclipsan completamente a las de sus competidores, favoreciendo así el uso de este software cuando se utiliza para proyectos de potencia gráfica o que requieran de unas físicas complejas. Además, desde 2016 se introdujo un nuevo modelo de pago, dejando disponible el código fuente para el que quiera utilizarlo bajo un modelo de licencia "Pay what you want".

El contrapunto de CryEngine es su poca documentación. La mayoría de la comunidad de desarrolladores optan por otros motores a la hora de desarrollar sus videojuegos, haciendo un hueco en esa tan necesaria comunidad que necesita apoyarse en otros para subsistir, haciendo que la curva de aprendizaje de este motor sea altamente desaconsejable para cualquier estudio pequeño.

4.4.2. Unity

Por último, Unity fue lanzada al mercado en 2005 por la desarrolladora Unity Technologies. Aunque empezó siendo un motor de juego diseñado para Apple ha terminado siendo uno de los motores, si no el mas, utilizados en el mundo. La última versión del mismo se lanzó en abril de 2019.

Lo que caracteriza a Unity es su asequibilidad, la capacidad de todos los desarrolladores puedan hacer sus proyectos 2D o 3D en esta plataforma independientemente de sus necesidades. Una curva de aprendizaje sencilla y unas licencias de usuario asequibles hacen que la gran mayoría de los pequeños desarrolladores se decantan por esta opción, si esto le añadimos que al tener muchos usuarios dispone de una de las documentaciones y foros de soporte más extensas de la red, hacen este motor de juego muy atractivo.

Si bien las características positivas de Unity lo hacen destacar sobre los demás, cabe decir que este motor no está pensado para grandes compañías o proyectos. La carencia de potencia del motor se hace notar cuando realmente necesitas unos gráficos espectaculares o unas físicas elaboradas. Es por eso que los "triple A" descartan realizar sus proyectos en esta plataforma, y tener sus propios motores.

Para finalizar, Unity es un motor de juego ideal para estudios pequeños y es por ello que lo hace el motor de juego idóneo para el proyecto y ya que es el que hemos adquirido conocimiento a lo largo de estas asignaturas.

4.5. Tecnologías complementarias

Cuando se desarrollan videojuegos requieres de distintos componentes aparte de la programación de este, como pueden ser la interfaz, modelos y música. Por esto mismo, de cara al software que se ha usado para realizar distintos assets del videojuego se van a listar a continuación.

4.5.1. Photoshop

Para el diseño de la interfaz, se ha utilizado un sistema en capas y con formas vectoriales para tener adaptación a cada resolución. Así si por lo que sea hay que hacer algún ajuste se puede realizar de forma muy sencilla.

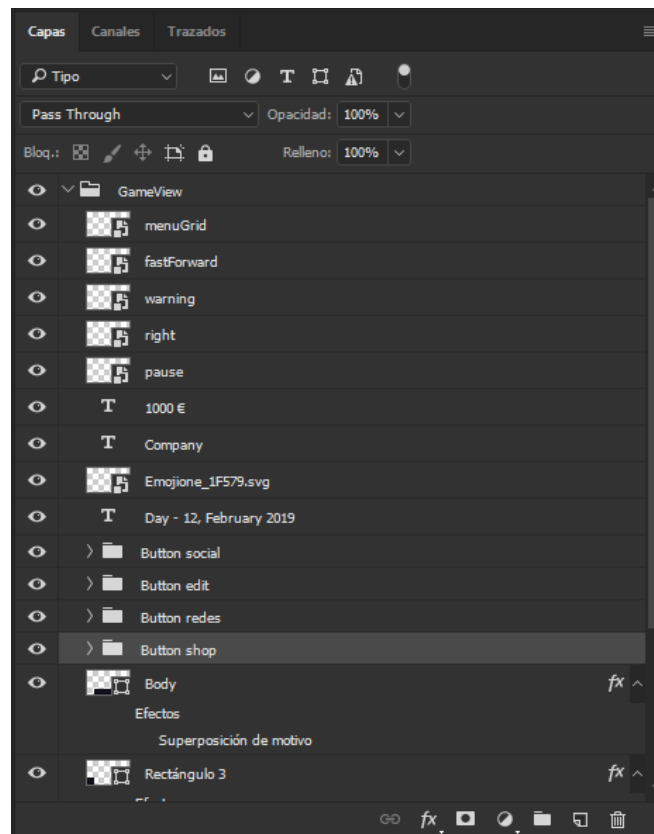


Figura 9 – Trabajo de capas en photoshop

4.5.2. Blender (2.8)

En esta parte se ha usado para crear objetos sencillos como son suelos y materiales, ya que se ha requerido de una previa adaptación al motor ya que no se había utilizado con anterioridad.

Como anotación a este software se ha usado por una de las mayores características que es libre y tiene una gran comunidad detrás, a pesar que en el curso se ha trabajado con la versión de estudiante de maya. Pero de cara a que se está planteando como un proyecto que a futuro se va a vender, la licencia de maya es excesivamente cara para un producto indie por lo que se hace insostenible.

5. Definición del juego

En esta sección se va a narrar de forma más extensa la conceptualización del videojuego que se ha planteado. Esta descripción es mayor que lo que permite el tiempo que se tiene para el desarrollo del trabajo de fin de master por lo que luego se especificara en esta versión.

5.1. Descripción del juego

El juego que se quiere crear se trata de un juego en el que se tiene de gestionar un negocio de máquinas recreativas. Desde ser un local minúsculo teniendo una inversión mínima de un pequeño banco a tener diferentes secciones dentro del lugar ampliando el negocio, y nuevas formas de conseguir dinero. La finalidad del juego es sacar a flote esa tienda que muchos han dicho que está abocada al fracaso, el hecho es pasar por todas las eras que han sufrido estas máquinas y conseguir adaptarse al paso del tiempo.

“Arcade last project”, es un juego en el que comienzas con una pequeña inversión que te permite alquilar un local y unas pocas máquinas de segunda mano que te encuentras en una tienda vieja. Y de ahí se va avanzando, por distintas fases, desde las siguientes máquinas recreativas hasta las nuevas que se conocen hoy en día de realidad virtual. Ampliando el negocio una vez tenido éxito, como añadir un lugar donde coman y descansen los clientes para luego querer invertir más dinero en nuestras máquinas y así ganar más beneficio.

5.2. Subgénero y referencias

El género/subgénero de este juego es el de gestión, teniendo parte de simulación ya que se trata de llevar un proyecto a cabo siguiendo unas estrategias con recursos limitados. Hay muchos competidores dentro de este género (también los catalogan como juegos tycoon³), pero ninguno con la temática que se está enfocando con este.

El juego más cercano al que se le podría asemejar es al juego “Game dev tycoon” que se trata de crear una empresa de videojuegos desde la nada. Pero este tiene ligeras diferencias con el que se plantea crear, por un lado, no tiene gestión de recursos en cuanto a la construcción del local. Que manejas solo un producto,

5.3. Tipo de interacción juego-jugador

La capacidad que tiene el jugador dentro de este juego es la gestión de los recursos en este caso el dinero. Este es un bien que al jugador le sirve para poder interactuar con cualquier mecánica dentro del juego, como comprar muebles o nuevas máquinas para el local y atraer a gente.

³ Tycoon: Término que se utiliza para designar videojuegos que requieran de dirigir de una empresa u organización que conviertan en un magnate de un negocio.

El juego está basado en días con un sistema de calendario, lo que nos permite tener distintos tipos de afluencia de jugadores, pudiendo focalizar tareas de ofertas o vales, en ciertas fechas (como un sistema de estaciones). Aunque esto último sería en una parte más avanzada ya que el jugador tiene que aprender cómo moverse por un negocio.

En un plano medio largo plazo a lo largo de la partida, el jugador podrá invertir dinero en mejorar el establecimiento comprando distintas parcelas donde podrá especializarlas en otros negocios que complementen la zona de recreativas, como puede ser: restaurante, o salón de eventos.

En resumen, el jugador es capaz de comprar: muebles, para aportar puntos de entorno (afectan a la satisfacción de los clientes, que esto hace que quieran gastar más), máquinas para que la gente gaste dinero en ellas y obtener beneficios. El jugador tiene una lista de contactos con distribuidores que le permiten hacer contratos de patrocinio como, por ejemplo: Coca-Cola, te da un patrocinio para que vendas su producto en tu local, dándole parte de los beneficios a ellos. Y no solo con comida si no con marcas de recreativas o empresas de consolas.

5.4. Plataforma de destino

Como se ha mencionado con anterioridad el juego que se plantea al tratarse de un género de simulación / gestión, se maneja mejor en el pc. Además, al tratarse de un primer juego el coste asociado a la publicación de un juego en una plataforma tiene un peso importante, y por ello la mejor tanto en coste como en el género del juego la mejor opción es PC.

Dentro de esta nos encontramos varias plataformas de distribución digital como son las más conocidas: Steam, Uplay, Origin, y GOG. Al tratarse de un juego independiente y de no participar con publisher como Ubisoft u Origin, las mejores opciones son Steam y GOG. Dentro de estas últimas, la que más jugadores abarca y más comunidad tiene en este género es steam y nos permite llegar a más público por ello es la plataforma de distribución elegida.

Cabe destacar la última plataforma de distribución que está siendo epic games, pero al estar en el inicio y no estar calando bien en la comunidad de jugadores no se contempla, a pesar que el gasto económico que requiere es inferior en esta que en Steam.

5.5. Ambientación

Al tratarse de un juego de gestión, la trama no es lo más importante, pero a lo referente a la ambientación, el jugador se encuentra que su tienda favorita donde disfrutaba de un buen rato jugando a las recreativas con los amigos va a cerrar. Decide que no quiere que los tiempos buenos desaparezcan y con una inversión mínima consigue salvar parte de esas máquinas y crea su pequeño negocio. Poco a poco con pequeñas inversiones y ayudas de distribuidoras conseguirá sacar a flote su pequeño sueño o por el contrario no puede hacer nada más que dejar pasar el tiempo.

5.6. Elementos e Interacción

El jugador como tal no tiene representación dentro del juego, ya que es el gestor de la compañía, por ello es el elemento más importante. Al ser la tienda / compañía el elemento más importante, hace que lo que genere también lo sea, el dinero ya que con ello podemos comprar todo lo que necesitemos para crecer. Otro elemento importante es la recreativa como concepto porque hay de varios tipos, por un lado, tenemos las maquinas simples que solo tienen botones y joysticks.



Figura 10 – Ejemplo de recreativas

Luego tenemos recreativas que pueden jugar cooperativo con dos jugadores, como pueden ser las de disparos cooperar para pasarse un nivel, o de competir entre unos con otros como las de coches, motos, o incluso las de baile.

Otro tipo de recreativas son las enfocadas a una actividad física como son las de golpear con un martillo a una plataforma, los discos flotantes de un jugador contra otro (obligatorio 2 jugadores), lanzamiento de balón de baloncesto, dardos.

En un futuro avanzado dentro del juego empezarán a salir nuevas máquinas de entretenimiento que tienen que ver con la realidad virtual y realidad aumentada, que están empezándose a poner de moda sobre todo en Estados Unidos.

Otros elementos importantes dentro del juego son distribuidores, que nos permitirán tener ingresos menores, pero al inicio serán importantes para poder manejarnos en el negocio. Luego dándonos la oportunidad a aceptar mejores contratos con condiciones más fructíferas económicamente hablando.

5.7. Objetivos planteados al jugador

Los objetivos del jugador es llegar a ser una empresa que pueda competir con el resto de empresas dedicadas al mundo del entretenimiento, ganar una cantidad de dinero mínima al día para que sea considerado una victoria.

Para esta primera versión del juego el objetivo del jugador es llegar a ser solvente dentro de la empresa, que le permita vivir de ello. Con las mecánicas core, como son la construcción y la economía básica del juego, esto se explicará de forma más detallada en la siguiente sección del documento.

6. Ruta de desarrollo

En este apartado explicare la distribución, los recursos y las características desarrolladas, ya que el juego conceptualizado requiere de un tiempo de desarrollo muy largo y por esto se desarrolla las bases del mismo.

En primer lugar, el desarrollo se ha dividido en dos fases, por un lado, pre producción y la siguiente producción.

6.1. Pre producción

En esta primera fase del desarrollo nos hemos dedicado a la presentación de los componentes gráficos del juego, es decir, la interfaz de usuario, modelos para el videojuego (recolectando muchos de ellos de los assets que se nombran en los siguientes apartados de forma más específica), e investigación de música, nombres y posibles logos.

6.2. Producción

La fase de producción que se ha dado en las semanas finales del proyecto ha sido el paso de la obtención de los assets, a integrarlos dentro del motor. Donde se han realizado los siguientes objetivos expuestos a continuación.

- Creación de las escenas del juego y el flujo entre ellas. El juego presenta el escenario principal con su respectiva escena de inicio, con un menú y el acceso a las opciones, y la segunda escena es la del juego en sí.
- Diseño de nivel: creación de la ciudad y entorno de la tienda del juego, así como de su estructura.
- Modelos (clases lógicas) de los clientes, maquinas, tienda, y jugador.
- Lógica de los clientes de la tienda, un sistema básico de decisión con las bases para poder ampliarlo.
- Lógica de las Maquinas, que clientes lo están usando el tiempo invertido en ella.
- Lógica de gestión del calendario, con un sistema de propagación de eventos fácilmente escalable.
- Lógica de sencilla de una “banda sonora”.
- Lógica de un sistema de tienda que nos permita pre visualizar el objeto que vamos a colocar y si queremos rotarlo o no.
- Sistema de opciones básico, sin ser demasiado complejo de manera funcional.

7. Diseño técnico

7.1. Entorno y requerimientos

En lo que respecta al entorno de desarrollo del proyecto, se ha usado el motor de Unity la versión de 2018 como he mencionado en secciones anteriores y Rider como IDE de desarrollo. Este IDE es de software privado, pero se está usando una licencia de estudiante. La gran razón de uso de este software es porque he trabajado con él en las versiones de desarrollo web dentro de un entorno laboral y la integración completa que tiene con el desarrollo de software no tiene comparación con otros.



Figura 11 – IDE de desarrollo

Voy a exponer de forma resumida la comodidad de este software a porta, ya que para el desarrollo de software es esencial un uso productivo que nos ahorre tiempos muertos.

- Facilidad de uso, comandos fáciles para hacer búsquedas por los ficheros.
- Guardado automático, es una funcionalidad de comodidad.
- Integración directa con git, el editor reconoce cuando un proyecto tiene que tener asociado su repositorio de git y te permite hacer commits de forma muy simple. Aunque para este desarrollo se ha usado SmartGit.
- Sistema pre configurado de debugger, es la función que nos permite parar el tiempo de ejecución el programa, y poder saber el estado que tienen ciertos valores a la hora de interactuar los scripts en la escena.

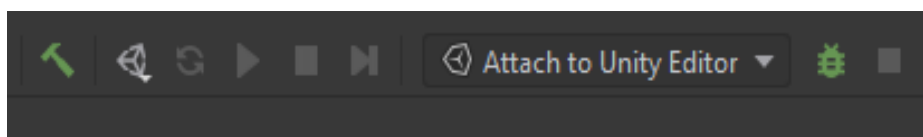


Figura 12 – Acceso directo al debugger de Rider

Por otro lado, al trabajar con Unity nos obliga en cierta manera a trabajar con C#, aunque este permita el trabajo con JavaScript. Se ha elegido en

primera instancia C# porque es un lenguaje más sólido y más fuertemente tipado, a parte de la gran comunidad que tiene por detrás.

7.2. Herramientas empleadas y assets

Para el desarrollo de este proyecto se han usado diferentes softwares para distintas tareas, las cuales vamos a nombrar en el siguiente apartado.

7.2.1. Herramientas

Por un lado, tenemos las herramientas de desarrollo, en la parte de diseño se ha usado Blender que es software libre, lo cual no es un gasto y tiene un gran respaldo de la comunidad.

Para el apartado de la interfaz se ha usado el photoshop con la licencia de prueba gratuita, y de cara a un desarrollo extendido para llegar a un producto final se compraría dicha licencia durante los meses de uso.

Dentro de las herramientas usadas también se ha utilizado un software de terceros para la manejabilidad de git que ha sido SmartGit, ya que nos permite una integración muy bien llevada de lo que se denomina “GitFlow” que no es más que una forma estandarizada de trabajar con git, como podemos ver en la siguiente imagen se representa la manera de trabajar o su forma idónea.

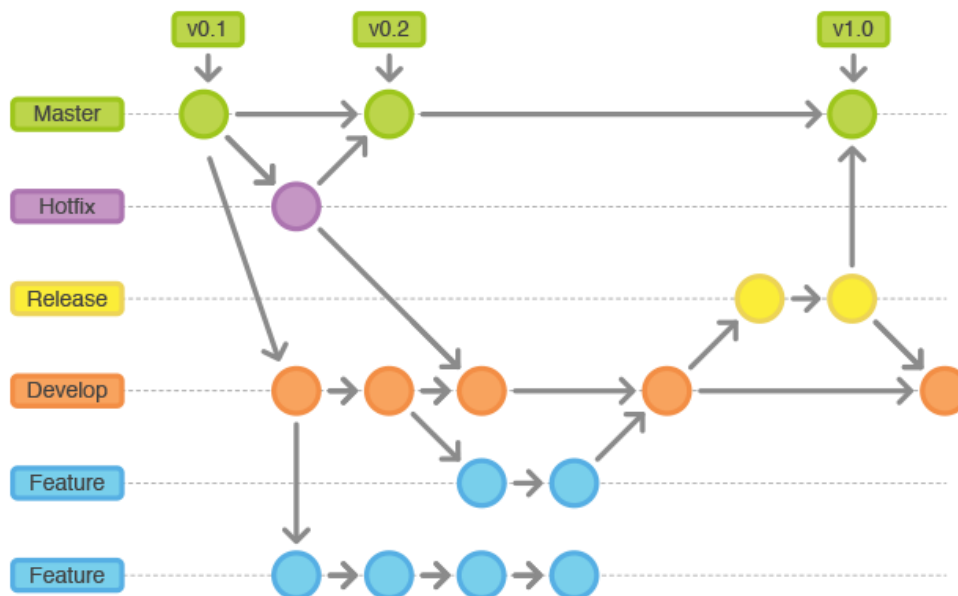


Figura 13 – Representación de gitflow

Otra herramienta de seguimiento que se ha utilizado ha sido trello que viene de la mano con la metodología Scrum ya que nos permite (si hubiera más componentes en el equipo) de manera instantánea comunicar cambios en el desarrollo y pasar por las fases de manera fluida.

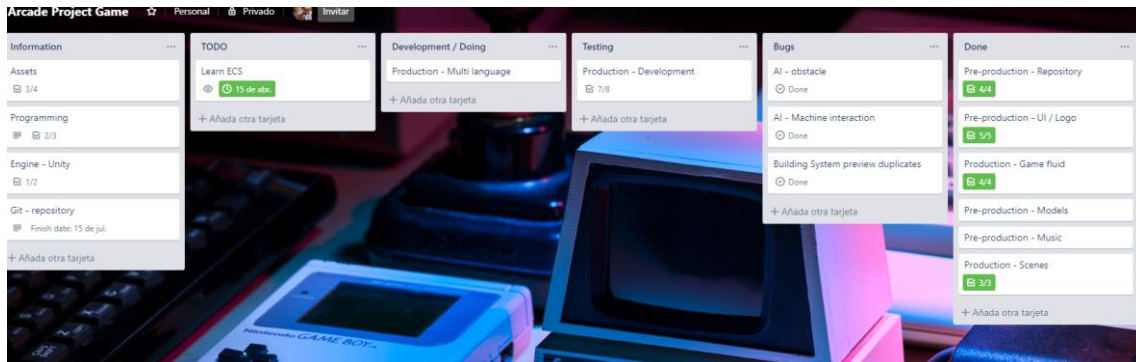


Figura 14 – Tablero trello del trabajo

7.2.2. Assets

Como este juego ha resultado tener una carga lógica muy grande, y aprender en el manejo del sistema de IA de Unity, se han usado varios assets en lo que respecta al modelaje en sí.

Los Assets de Kenney lo bueno que tienen es que son de libre uso, lo que ha permitido poder integrarlos sin ningún tipo de problema dentro del juego.

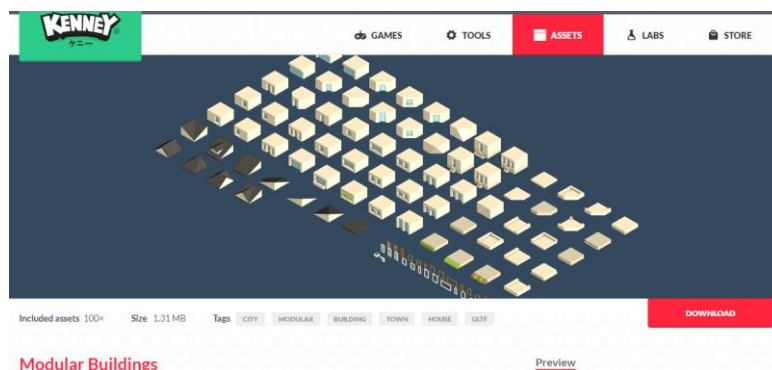


Figura 15 – Pagina de Kenney

Cabe destacar es que no son modelos completos y que tienen un proceso creativo ya que nos obliga a crearlos por nosotros mismos como podemos ver a continuación. Dando así autenticidad creativa sin excedernos en tiempos por una complejidad de modelaje.

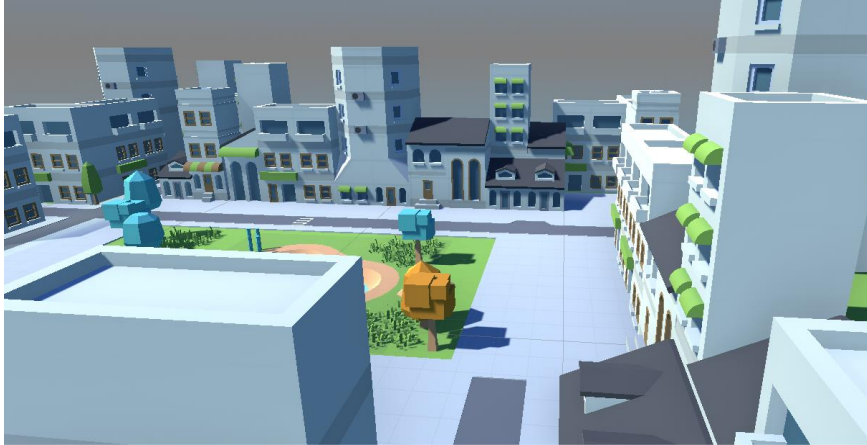


Figura 16 – Modelos creados

Por otro lado, la interfaz ha sido diseñada íntegramente por el alumno, analizando juegos de su género y aplicando una usabilidad acorde que el usuario pueda entender. Manteniendo también una interfaz homogénea que no choque pasar entre un menú y otro.

Como se puede ver en la imagen siguiente, se tiene de forma muy reducida la información que el usuario necesita para que pueda ver el resto de la escena.

En un simple cuadro se puede ver mucha información, por un lado, tenemos el calendario que nos muestra los días en los que estamos. El nombre de la empresa del usuario que en un futuro se podrá editar (ya que esta parametrizada), el dinero que tiene el usuario, los botones del tiempo (4 tipos: parado, normal, rápido, rápido x2) para pasar más rápido cuando no tienes mucho que hacer. Y por último el botón de rejilla que despliega los iconos que nos permite interactuar con el resto de acciones.



Figura 17 – Interfaz dentro del juego

La interfaz de compra está basada en un sistema de rejilla dinámico para que se adapte a ciertas resoluciones. Representa de forma sencilla que maquina vas a comprar y el precio de esta, por otro lado, como se puede interactuar con otro menú de compra o salir de este.

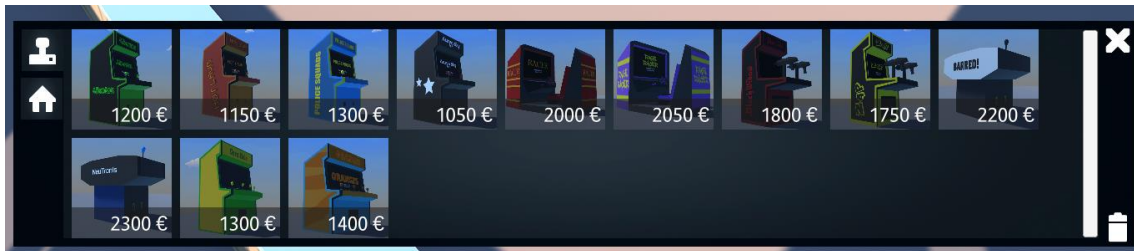


Figura 18 – Interfaz de compra

En lo que respecta a la música, he utilizado ciertas piezas de un paquete de assets que tenía en el Unity Asset Store, como se puede ver en la siguiente imagen.

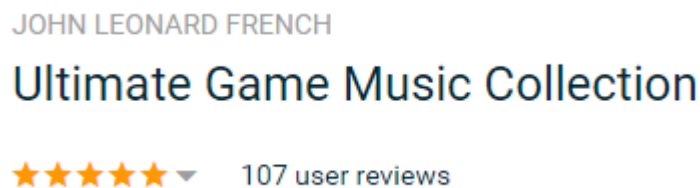


Figura 19 – Autor de la música que se ha usado

Por último, los modelos de las máquinas recreativas en un inicio se iban a desarrollar por parte del alumno para aprender el software de Blender, pero dado que requiera de un grado alto de tiempo en realizar cada modelo, se optó por unos assets gratuitos del siguiente autor. Se tardó alrededor de varios días en realizar un modelo ni con la mitad de calidad de este pack así que se descartó por falta de tiempo en el modelaje.

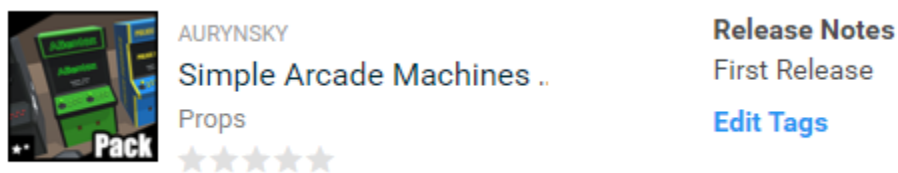


Figura 20 – Imagen del paquete de recreativas de Asset Store

7.3. Esquema de la arquitectura

En lo que respecta a la arquitectura del juego se planteó un diagrama de clases de la manera que se ve en la imagen siguiente. Fue el diagrama inicial sobre papel ya que se planteó un inicio realizar misiones que las pudiera aceptar el jugador y luego estas le daban unas recompensas. Por falta de tiempo en el desarrollo, se ha optado por hacer el esquema sin ellos, estas clases están reflejadas en el juego desarrollado, y se hacen uso dentro del manager del juego (el script general de la partida).

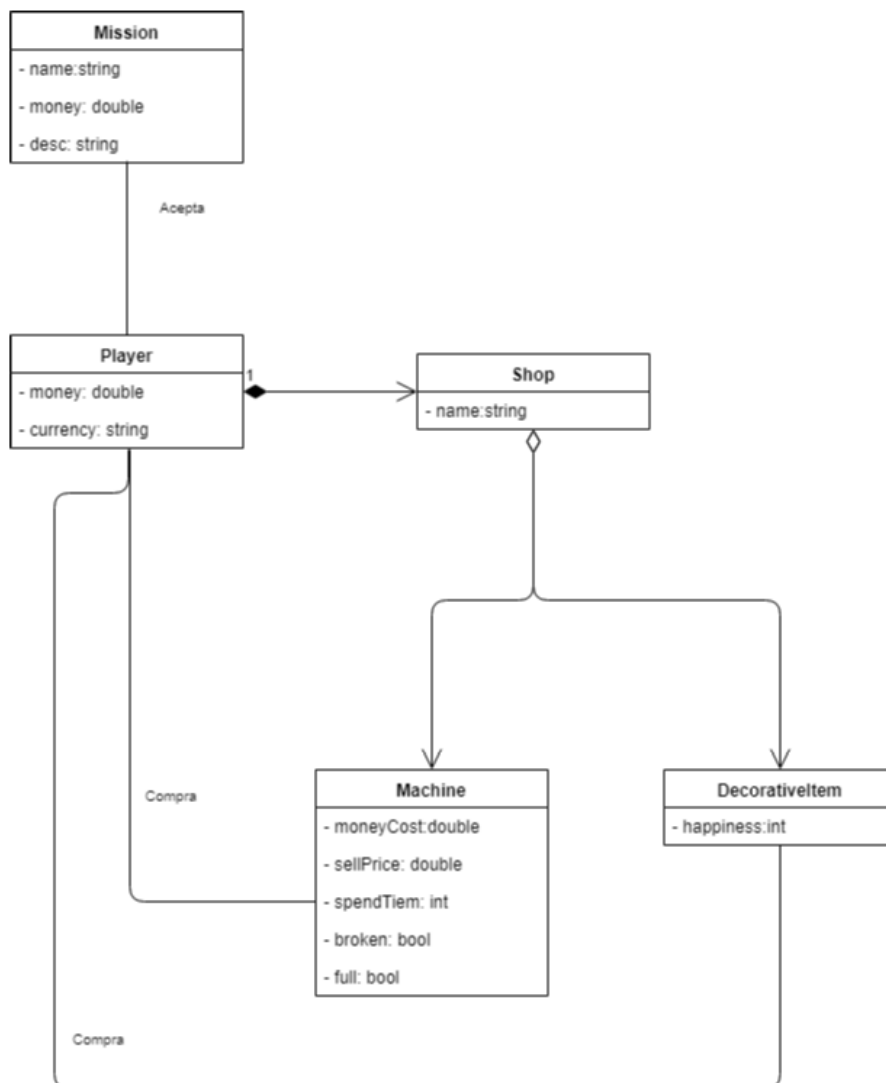


Figura 21 – Diagrama de clases

En el siguiente grafico se puede ver el flujo que sigue el programa, es un grafo simple para la demo que se presenta, pero de cara a futuros desarrollos en este juego se puede ver modificado y se suele actualizar a medida que se desarrollan nuevas pantallas.

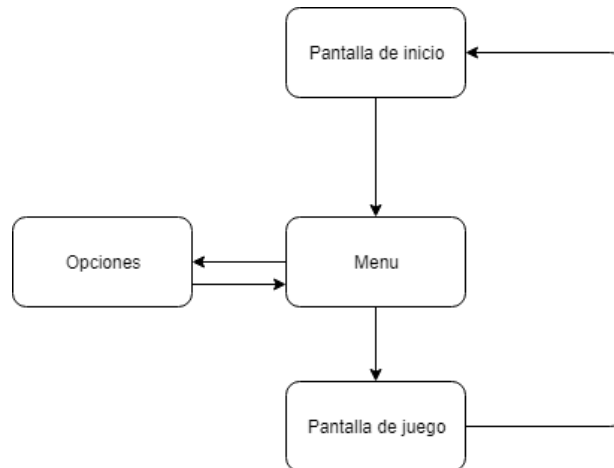


Figura 22 – Flujo del programa

Para la distribución dentro de Unity no hay una estandarización como tal, cada equipo de desarrollo se estructura de una forma concreta pero la que se ha usado para este proyecto me ha parecido la más acorde.

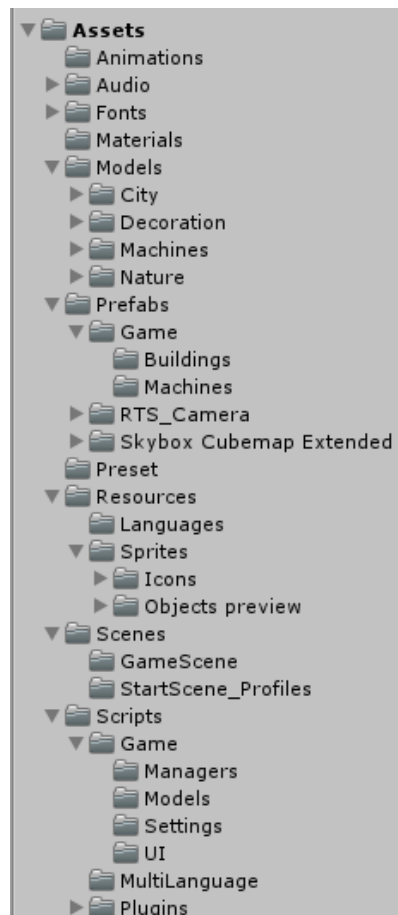


Figura 23 – Estructura de las carpetas Unity

Cada tipo de asset va en su respectiva carpeta y luego cada una de ellas tiene una subdivisión como pasa en los modelos o en los prefabs ya que de cara a una ampliación del proyecto puede resultar muy lioso. Y la carpeta de los scripts también se ha subdividido entre los que son de utilidad y los que son de mecánicas de juego ya que así nos resulta más fácil identificarlos.

También se ha seguido una estructura de trabajo uniforme en las escenas ya que como es bien sabido en Unity puede ser algo caótico si no está bien distribuidos los gameObject.

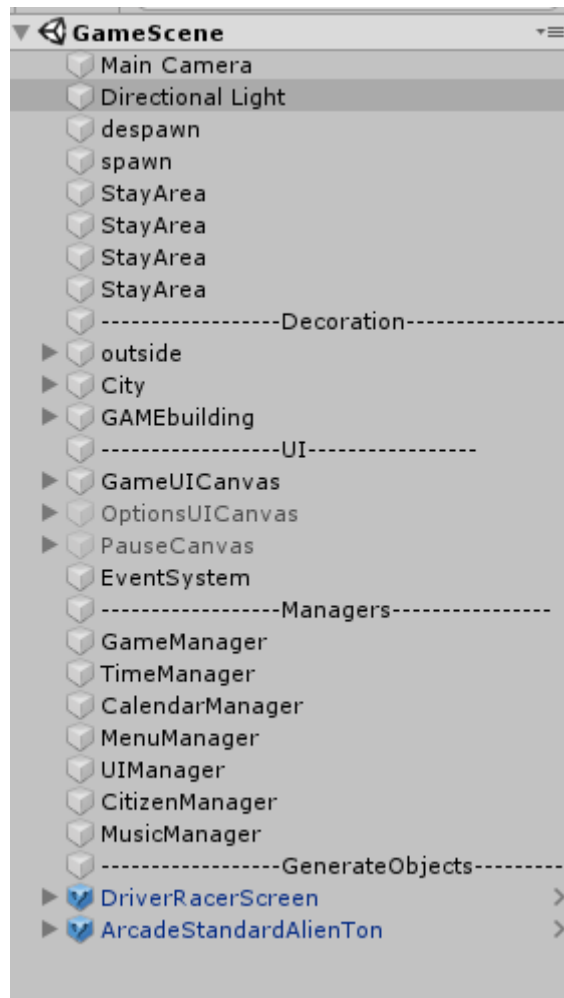


Figura 24 – Estructura de los componentes de la escena principal

Como se puede observar en la imagen se ha separado de tal forma que se pueden ver para que sirve y de que está compuesto cada GameObject. Los objetos de la decoración no aportan nada, y por eso están almacenados en otros mayores para no molestar al trabajar con la escena. Por otro lado, los managers al estar cada uno independiente con su nombre se sabe cuál hay que modificar cuando toca sin tener que ir uno por uno mirando en cual lo dejamos, lo que lo hace muy útil.

Dentro de lo que es la lógica del juego, el sistema de calendarios tiene implementado un sistema de eventos que lo que hacen es propagar la información cuando ha pasado el mes, en este caso es para cobrar el alquiler. Así el script encargado de gestionar el dinero sabe cuándo tiene que quitar el dinero al jugador sin tener una comunicación directa, si afectar en exceso de llamadas entre GameObjects.

7.3.1. IA

Dado que el juego que se ha planteado tiene una carga lógica muy grande y no hay excesiva interacción del jugador con la escena (en esta versión core) sí que hay trabajo por detrás que son la lógica de los clientes y la máquina.

En primer lugar, se puede observar el diagrama de la toma de decisión del cliente que llega a la tienda, llega a la tienda y mira si tiene dinero y si lo tiene busca una maquina libre. Se pone a jugar y comienza de nuevo ese ciclo a no ser que este se quede sin dinero, sino “se va a casa”. También se ha contemplado como añadido un sistema de necesidades, estilo “sims” sin ser tan rebuscado ni tan complejo pero que si no comen o van al baño estos se tengan que ir.

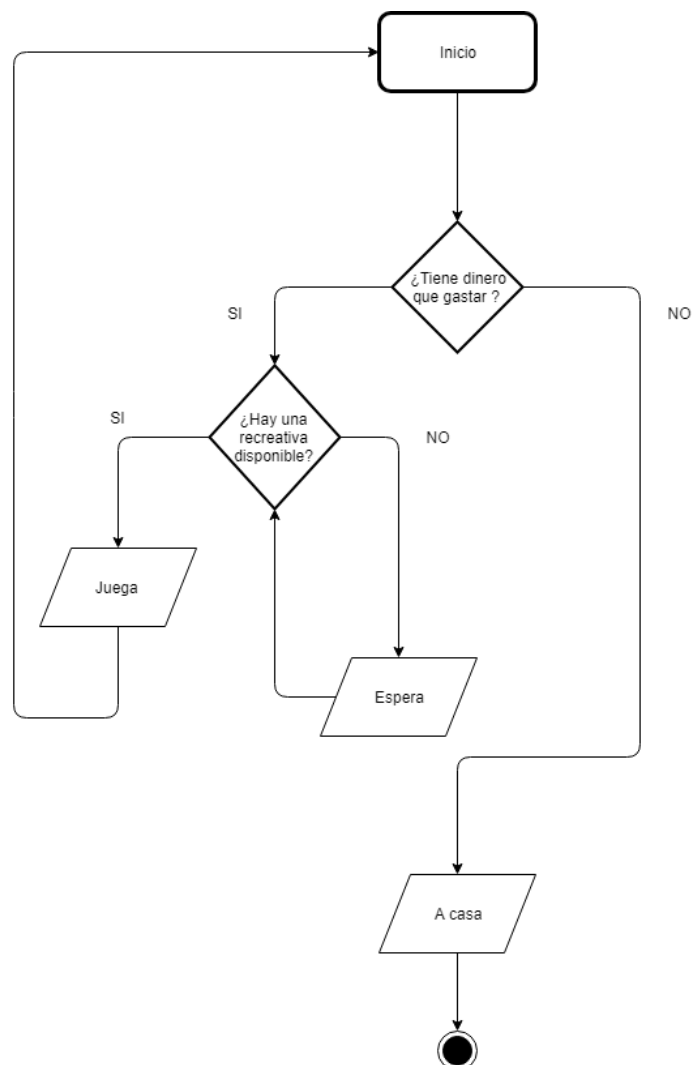


Figura 25 – Diagrama de flujo del cliente

Para poder aplicar todo este sistema de IA se ha investigado y utilizado el sistema de navegación. Uno de los problemas que había era generar obstáculos dinámicos para la inteligencia artificial ya que se podía construir maquinas en sitios distintos y hacia que estas se atascaran. Se implementó con los obstáculos de Unity para generar un radio de obstáculo dinámico en función del tamaño de este.

Como añadido, tenemos el sistema de la máquina, que no es una IA como tal, pero tiene lógica que hace acaparar el tiempo del cliente. Una vez que entra en contacto el cliente con la maquina pregunta si está ocupada ya que puede haber llegado una persona antes y si no se queda “haciendo cola”, hasta que pueda interactuar con la máquina cuando esta se libere. Se asemeja a un sistema de semáforos, con la diferencia que la forma de decir que no se puede pasar es diciendo que ya hay otros clientes en ella.

8. Diseño de niveles

En este juego al ser la partida en el escenario principal de la ciudad no tenemos más escenas que comentar, podríamos destacar la puesta en escena de la pantalla principal que son las maquinas puestas de tal forma que parezca que está cerrado.

En la escena del juego como podemos ver la escena dentro del motor tenemos la ciudad del lado que la cámara puede visualizar y en la otra parte no ya que así ahorramos en recursos.



Figura 26 – Escenario dentro del Motor

Ya enfocándonos en la cámara del juego y en su visualización tenemos la escena y la cámara al máximo de distancia y lo que podemos ver es que la tienda está en la plaza central de la ciudad con un parque en medio y la variedad de los edificios.

Aun así, el foco principal es el establecimiento y no deja ver más allá que una pequeña parte de esta.



Figura 27 – Escenario dentro del juego

9. Manual de usuario

9.1. Requerimientos Técnicos

Siguiendo una misma estructura que hace Steam para representar los valores del juego voy a listar a continuación lo que serían sus requisitos para poder jugarlo en máximo, haciendo una estimación a lo alto del juego final.

Requiere un procesador y un sistema operativo de 64 bits

- SO: Microsoft Windows
- Procesador: Intel Core 2 Duo, 3.0GHz or AMD Athlon 64 X2 6400+, 3.2GHz
- Memoria: 4 GB de RAM
- Gráficos: nVIDIA GeForce GTX 260, 512 MB or ATI Radeon HD 5670, 512 MB
- Almacenamiento: 0.4 GB de espacio disponible

9.2. Manual de usuario

El juego al tratarse de un género de estrategia/gestión tiene mucho juego el teclado y el ratón por lo que los controles son con estos mismos. Se van a enumerar los siguientes controles naturales, ya que siguen el canon del genero para que cuando se incorpore un jugador que nunca haya jugado a este, este familiarizado al menos con los controles.

- Mover la cámara, en todos los lados con A, W, S, D.
- Rueda del ratón arriba/abajo, subir y bajar la cámara.
- Modo construcción: Click izquierdo del ratón construir, click derecho del ratón rotar el objeto a construir.
- Control el tiempo: 1, 2, 3, y 4 de las teclas numéricas superiores del teclado, pausa, normal, rápido, rápido x2 respectivamente.

10. Conclusiones

Terminado el proyecto se pueden sacar ciertas conclusiones, del trabajo realizado. Se va a mencionar el tiempo como gran reto a salvar, manteniendo un mismo planteamiento a lo largo de este. Parte de la planificación se vio afectada por intentar generar modelos en el tiempo que resta, y retraso parte del resto de funciones.

Aun así, se ha podido cumplir con los objetivos de mecánicas a desarrollar, a falta de la parte de las distribuidoras por no entregar una mecánica con bugs se ha dejado de implementar en la escena final. De cara al desarrollo de la lógica y la implementación de la inteligencia artificial ha supuesto una investigación muy fructífera de sistemas de decisión.

Otra de las mecánicas que ha llevado a improvisar fue la del sistema de construcción y la forma de medir cuando se puede o no construir, fue un reto que me ha hecho tener más conocimiento técnico del sistema de capas de Unity.

En lo que respecta a las entregas se ha podido finalizar de forma acorde a lo planteado en la asignatura, y se ha gestionado de forma correcta para entregar en el tiempo estimado.

Como conclusiones finales, hay varias lecciones aprendidas por un lado no excederse con los objetivos cuando tienes que tener un proceso formativo antes de poder hacerlas, esto ha sucedido con la parte de modelaje en 3d de las máquinas. Por otro lado, la estimación y distribución de tareas, ha sido una labor compleja de visualizar ya que parece que podemos abordar ciertas cosas sin tener conocimiento de estas.

Más haya de cómo distribuirse, el poder gestionar un juego tan complejo lógicamente hablando, ha servido para visualizar las cosas desde otra perspectiva y analizar a papel antes de poder ponerse a programar. Aprender diferentes softwares de trabajo como ha sido Blender, ha sido una experiencia satisfactoria y ha merecido la pena para sentar las bases para futuros proyectos con él, con más tiempo de por medio. Como añadido, el haber diseñado una interfaz simple y eficaz pasando por distintas fases de procesos creativos ha hecho desarrollar una faceta más creativa para tener otro punto de vista dentro del perfil de programadora que tengo.

Como planteamiento final, me gustaría plantear de forma breve las ideas a futuro que tengo para el proyecto. Por un lado, tendríamos un sistema con más variedad de objetos en la tienda, y un sistema de investigación que nos permita desbloquear ciertas máquinas o bonus para el personaje (como x % de dinero extra con las ventas en máquinas cooperativas). Un apartado de diseño propio del juego sin tener que basarse en los assets de Kenney (Gracias por su gran calidad). Además de la funcionalidad de misiones planteada, enriquecida para atraer más al jugador, con un sistema también de ampliación de negocio como menciono en anteriores apartados de forma más extensa.

11. Glosario

- **Feedback:** comunicación de resultado.
- **Debugger (depurador):** es un programa o plugin (complemento) usado para probar y depurar (eliminar) los errores del programa objetivo.
- **Sprints:** periodos de tiempo que suelen ser de entre 2 semanas a 3 meses.
- **Royalty:** Comisión que hay que pagar a una empresa para utilizar su producto en beneficio propio.
- **GameObject:** Objeto utilizado dentro de la escena del juego.
- **Prefab:** Plantilla de GameObject a partir de la cual se pueden crear nuevas instancias de objeto.
- **Tycoon:** Término que se utiliza para designar videojuegos que requieran de dirigir de una empresa u organización que conviertan en un magnate de un negocio.
- **Endgame:** se llama al diseño de mecánicas una vez avanzado un largo tiempo en una partida.
- **Assets:** son los recursos o las herramientas que usa un videojuego y que van dentro de él. Como son imágenes, iconos, música, modelos...
- **GameObject:** Es el concepto más importante en el Editor de Unity. Cada objeto en su juego es un GameObject, desde personajes y objetos coleccionables hasta luces, cámaras y efectos especiales. ...
- **Script:** Es un componente que se le añade a un gameObject para que adquiera una funcionalidad programada, como puede ser mover por la escena o encender una luz...

12. Bibliografía

- Statista. (1 de Febrero de 2019). *Statista*. Recuperado el 2 de Junio de 2019, de Statista: <https://es.statista.com/estadisticas/635816/juegos-de-pc-mas-vendidos-de-todos-los-tiempos-a-nivel-mundial/>
- Unity Technologies. (1 de Agosto de 2017). *Docs Unity*. Recuperado el 20 de Junio de 2019, de Unity: <https://docs.unity3d.com/es/current/Manual/GameObjects.html>
- Warman, P. (2 de Noviembre de 2018). *New zoo*. Recuperado el 5 de Junio de 2019, de New zoo articles: <https://newzoo.com/insights/articles/newzoo-cuts-global-games-forecast-for-2018-to-134-9-billion/>