Atila, Tierra quemada

Xavier Pallicera Malivern
Master en Diseño y desarrollo de videojuegos
Juego para plataformas móviles con Unity

Jordi Duch Gavaldà y Helio Tejedor Navarro
Javier Luis Cánovas Izquierdo

30 de junio de 2019

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

Copyright © 2019 Xavier Pallicera

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© Xavier Pallicera
Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Atila, Tierra quemada*
Nombre del autor:	*Xavier Pallicera Malivern*
Nombre del consultor/a:	*Jordi Duch Gavaldà*
Nombre del PRA:	*Javier Luis Cánovas Izquierdo*
Fecha de entrega:	06/2019
Titulación::	*Master en Diseño y desarrollo de videojuegos*
Área del Trabajo Final:	B2.504
Idioma del trabajo:	*Castellano*
Palabras clave	*Unity, Estrategia, Android*

Resumen del Trabajo:

Con el objetivo de profundizar en el aprendizaje de un juego para dispositivos móviles se ha diseñado y desarrollado un juego de estrategia isométrico con el propósito de aprender la metodología del diseño completo de una aplicación móvil en Unity.

Se ha profundizado en la interacción de los procesos constructivos de una aplicación en todas sus vertientes para ver como los intereses poliédricos de diseño, jugabilidad, usabilidad, rentabilidad económica interactuaban para la concepción final del proyecto y encontrar el equilibrio para en un contexto móvil, hacer un juego interesante, rentable y divertido.

Sin duda, una excelente oportunidad para, a través de un proyecto real ver todas las dificultades técnicas, de diseño, de jugabilidad y analizar a través del proceso de testeo cuales eran las mejores decisiones y los errores generados en el proceso y como solventarlos.

Abstract:

In order to improve the learning of a game for mobile devices, an isometric strategy game has been designed and developed with the purpose of learning the methodology of the complete design of a mobile application in Unity.

It has been deepened in the interaction of the constructive processes of an application in all its aspects to see how the polyhedral interests of design, playability, usability, economic profitability interacted for the final conception of the project and to find the balance for in a mobile context, to make a game interesting, profitable and fun.

Undoubtedly, an excellent opportunity to, through a real project see all the technical difficulties, design, gameplay and analyse through the testing process which were the best decisions and errors generated in the process and how to solve them.

Índice

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 4
    1.5 Breve sumario de productos obtenidos 6
2. Estado del arte 7
3. Definición del juego 10
    3.1 Trasfondo histórico de un juego de estrategia 10
    3.2 Subgénero y referencias 11
    3.3 Interacción 14
    3.4 Ambientación y trama 15
    3.5 Personajes e interacción 16
    3.6 Objetivos del juego 17
    3.7 Concept art 18
4. Diseño técnico 19
    4.1 Requerimientos técnicos 20
    4.2 Herramientas de desarrollo 21
    4.3 Assets y recursos del juego 22
    4.4 Los gráficos y elementos visuales 26
    4.5 Arquitectura del juego 27
    4.6 El juego como elemento empresarial 44
5. Diseño de niveles 47
    5.1 Editor de niveles 47
    5.2 Contexto histórico 48
    5.3 Análisis de un nivel 48
6. Manual de usuario 51
    6.1 Instrucciones del juego 52
    6.2 Enlaces 53
7. Conclusiones 54
8. Glosario 55
9. Bibliografía 56

Lista de figuras

Figura 1. *Solución de Euler*
Figura 2. *Atila, rey de los hunos*
Figura 3. *Dinámica de trabajo*
Figura 4. *Diagrama de Gantt (1)*
Figura 5. *Diagrama de Gantt (2)*
Figura 6. *Referencia en Google Play Store*
Figura 7. *Análisis del mercado móvil en Alemania (Mayo, 2019)*
Figura 8. *Requerimientos para aplicaciones Android 2019*
Figura 9. *Atila, Tierra quemada en su entorno de desarrollo Unity 2018.3*
Figura 10. *Definición del movimiento del caballo en el juego de Ajedrez*
Figura 11. *Incursiones de Atila en el imperio romano*
Figura 12. *Capturas de Knight Ride y Chess Horse*
Figura 13. *Captura del juego Chess Knight*
Figura 14. *Captura de los juegos Knight Chess y Knight's move*
Figura 15. *Movimiento del caballo sobre un tablero*
Figura 16. *Recursos necesarios del ejército de Atila*
Figura 17. *Movimiento del ejército, objetivos y casilla final*
Figura 18. *Objetivos y casillas finales*
Figura 19. *Tilesets isométricos (propuesta 1)*
Figura 20. *Icono y logotipo del juego*
Figura 21. *Flujo de juego*
Figura 22. *Isometric builder*
Figura 23. *I2 Localization*
Figura 24. *Easy Mobile*
Figura 25. *2D Isometric Starter*
Figura 26. *Freepik*
Figura 27. *Iconmstr*
Figura 28. *Ejemplo de composición gráfica*
Figura 29. *Flujo de juego*
Figura 30. *Menú principal y opciones*
Figura 31. *Selector de etapas*
Figura 32. *Movimiento de sobre el tablero*
Figura 33. *Posibles movimientos del jugador*
Figura 34. *Resultado de una batalla*
Figura 35. *Obtención de recursos*

Figura 36. *Atila conquista Roma*
Figura 37. *Diseño del tablero isométrico*
Figura 38. *Creación dinámica del tablero de juego*
Figura 39. *Asociación de clases, sprites, selector y banderas.*
Figura 40. *Flujo del juego a través de las clases*
Figura 41. *Derrota por movimiento*
Figura 42. *Derrota en el campo de batalla*
Figura 43. *Disposición de las tropas antes del ataque*
Figura 44. *Derrota por falta de recursos*
Figura 45. *Cálculo de saltos disponibles*
Figura 46. *Tutorial (1)*
Figura 47. *Tutorial (2)*
Figura 48. *Usabilidad sobre la pantalla de juego*
Figura 49. *La interfaz de usuario*
Figura 50. *Retención del usuario de aplicaciones móviles*
Figura 51. *Recompensas virtuales a cambio de monetizar el juego*
Figura 52. *Sistema de logros de Google Play Games*
Figura 53. *Editor de niveles*
Figura 54. *Principales vías romanas*
Figura 55. *Nivel 2 del juego: Mursa*
Figura 56. *Resolución del nivel 2*
Figura 57. *Compra de suministros al final del nivel*
Figura 58. *Requisitos en Google Play*
Figura 59. *Como jugar a Atila, Tierra quemada*

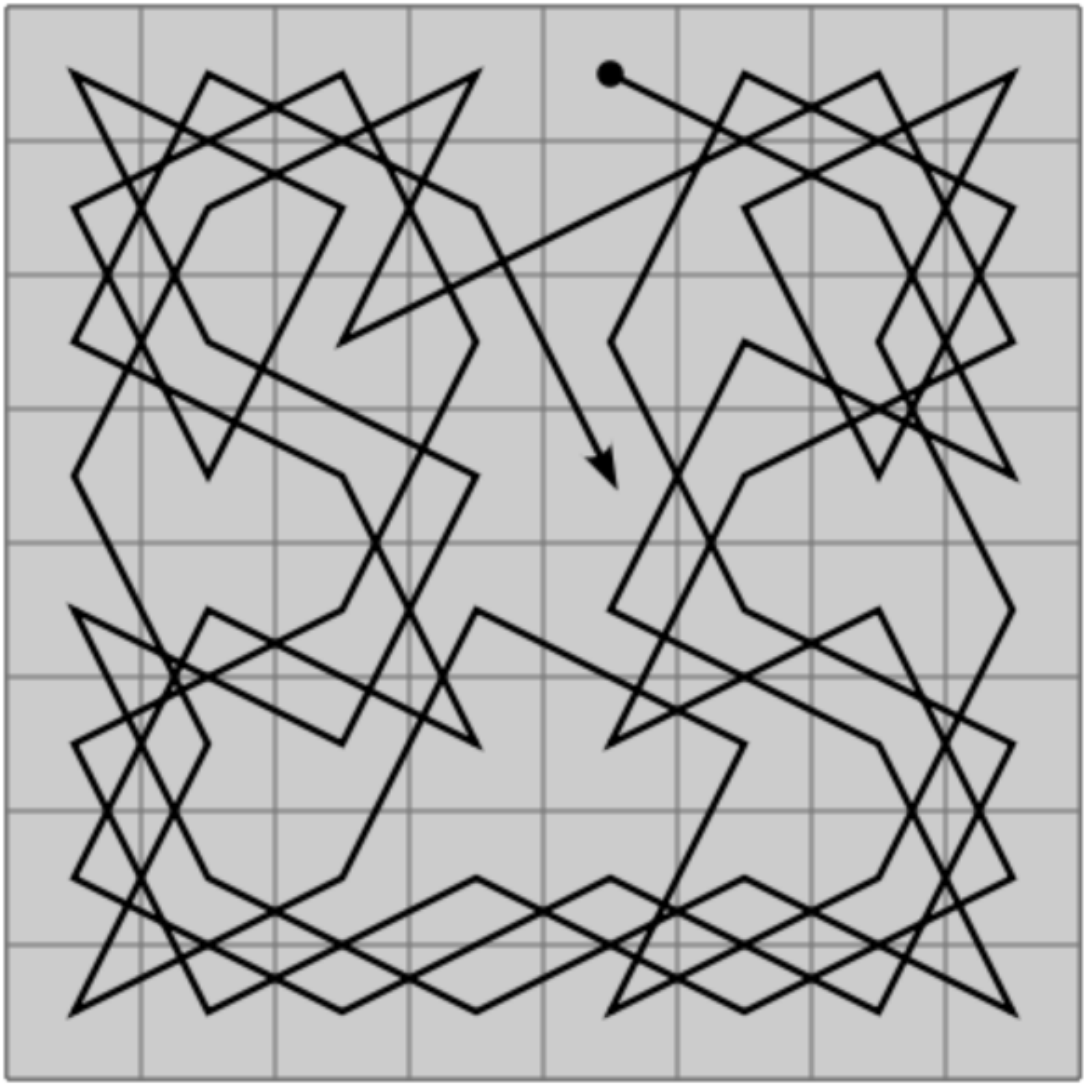
1. Introducción

1.1 Contexto y justificación del Trabajo

El **problema del caballo de ajedrez** ha despertado el interés de muchos matemáticos a lo largo de la historia. Uno de los más famosos que halló solución al enigma fue el mejor matemático del siglo XVIII, el suizo *Leonhart Euler*.

Este problema matemático consiste en recorrer un tablero de ajedrez recorriendo todas las casillas utilizando un caballo de ajedrez sin poder repetir ninguna. Se puede empezar por la casilla que queramos, pero se debe pasar por todas las casillas del tablero.

Figura 1. *Solución de Euler*



Solución de *Euler* al problema del caballo de ajedrez.

Este concepto, de per se, ya es un objetivo interesante ya que, aún hoy, es un juego recurrente que muchas personas juegan sin darse cuenta con un papel cuadriculado como un juego causal, entretenido y motivador en cualquier momento y lugar.

Figura 2. *Atila, rey de los hunos*

Para dar mayor profundidad al contexto del juego se quería dar un trasfondo histórico al reto al introducir un elemento clásico al ambientarlo como un juego de estrategia. En este caso una ambientación en la época romana tardía.

Atila, rey de los hunos, fue un gran contrincante del imperio romano, poniendo en jaque lo que hoy conocemos como los cimientos de la cultura occidental. A través de su ejército a caballo y la movilidad que le proporcionaba estuvo a punto de conquistar la misma Roma después de saquear la Galia y el Imperio Romano de Oriente. Estos hechos históricos son una oportunidad para ser el entorno perfecto para crear un juego de estrategia a través de unos de los problemas matemáticos más divertidos como el que propone el caballo de ajedrez.

1.2 Objetivos del trabajo

El Trabajo permite trabajar varios aspectos relevantes de las etapas del desarrollo de un videojuego desde la idea conceptual hasta la fase de pruebas antes del lanzamiento del producto al mercado pasando por el desarrollo del arte y las mecánicas del mismo.

Otro objetivo importante del proyecto es que se puede trabajar cuestiones que de otro modo no sería posible que fueran trabajados. En un desarrollo real, sería muy difícil que una misma persona tuviese todos los roles que aquí se han trabajado por lo tanto difícil de como la interacción de las diferentes etapas del desarrollo de un videojuego interactúan entre ellas y como las decisiones afectan a las diferentes etapas del desarrollo.

Además este trabajo tiene como objeto el aprendizaje de crear un juego para dispositivos móviles, con sus limitaciones y especificaciones, que sea divertido en el contexto de movilidad de los jugadores. Además se buscaba el aprendizaje de uno de los modeles gráficos que en su momento aproximaron los juegos 3D en entornos 2D como es el sistema isométrico para dar un efecto 3D al juego que tiene sus peculiaridades y limitaciones en el entorno de trabajo.

Otro objetivo importante es el tratamiento de la usabilidad y las mecánicas de juego así como el diseño de la interfaz de usuario. Las múltiples medidas de pantalla de teléfonos inteligentes, tabletas y otros dispositivos compatibles hacen necesario un trabajo para adecuar la interfaz dentro de un espectro muy diferente sin perder aspectos de la jugabilidad que son importantes para el éxito o fracaso de un juego.

1.3 Enfoque y método seguido

El método de trabajo puede diferenciarse en tres etapas diferenciadas que el propio TFM ha propiciado durante su concepción y desarrollo. En una primera etapa se ha creado **un diseño del juego** basado en las siguientes premisas de análisis:

1. Análisis del subgénero del juego y sus referencias
2. Relación entre el juego y jugador
3. Historia, ambientación y trama
4. Personajes e interacción
5. Objetivos del juego
6. Concept art
7. Tiempos de desarrollo y recursos necesarios

Una vez terminada esta etapa, empieza la etapa de **programación y diseño** donde se buscará soluciones prácticas y reales a los problemas y retos planteados en la etapa de concepción. En esta etapa, y respetando los tiempos de desarrollo, se trabajaran aspectos como:

1. Secuencia de juego
2. Editor de niveles
3. Tablero de juego en entornos isométricos
4. Mecánica de juego
5. Condiciones de victoria y derrota
6. Usabilidad
7. Interfaz de usuario
8. Arte y estética

En la tercera parte del desarrollo se **analiza la interacción** de los diferentes elementos diseñados en función de la estructura lúdica.

1. El perfil del jugador
2. Anhelos y necesidades del jugador
3. Limitaciones del entorno móvil
4. Recompensas
5. Progresión en el juego
6. Economía en el juego
7. Monetización
8. Marketing y posicionamiento

De un modo u otro las tres fases de desarrollo no son elementos estancos ya que decisiones en un punto u otro obligan a hacer cambios en algún aspecto del diseño.

Todas las etapas tienen mecanismos validación que se fundamentan en el análisis de la respuesta de los jugadores o potenciales jugadores a entradas ofrecidas en función de saber si la respuesta obtenida es la que como diseñadores, programadores o artistas se busca en cada momento de la producción.

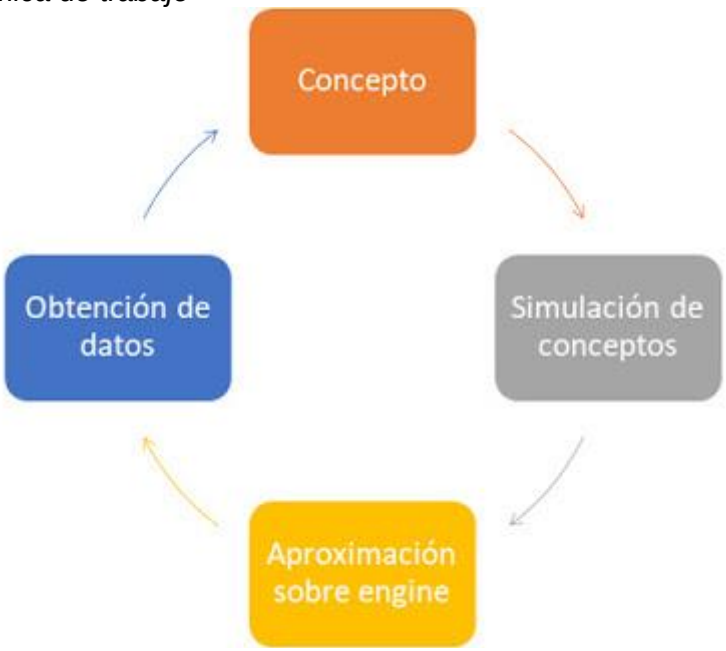
1.4 Planificación del Trabajo

El proyecto ha tenido una duración estimada de cuatro meses de trabajo hasta la confección de esta memoria de trabajo.

El primer mes de trabajo se dedicó a la concepción del proyecto que tal y como se ha referido configurando la primera etapa de trabajo consistiendo en el diseño y concepción del juego (**PEC1**).

La segunda etapa de trabajo, la más extensa, se ha dedicado en la implementación del diseño creado, establecer las mecánicas del juego, diseñar el editor de niveles, la interfaz gráfica y trabajar en la usabilidad. Para ello se definieron un conjunto de pruebas para simular los conceptos diseñados e ir modelando el diseño a la implementación final.

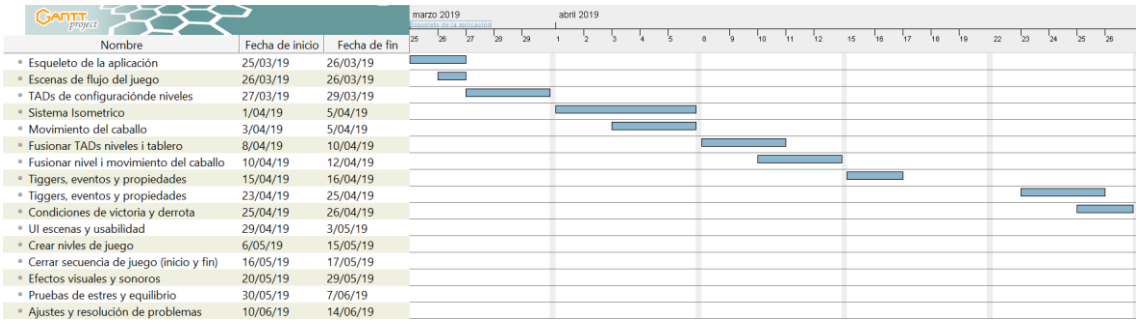
Figura 3. *Dinámica de trabajo*



PEC2: Entre el 25 de Marzo i el 28 de Abril

Durante el primer periodo de trabajo tengo previsto hacer el *core* del juego para poder probar las mecánicas del juego y poder recibir las valoraciones del tutor para hacer las correcciones oportunas.

Figura 4. *Diagrama de Gantt (1)*



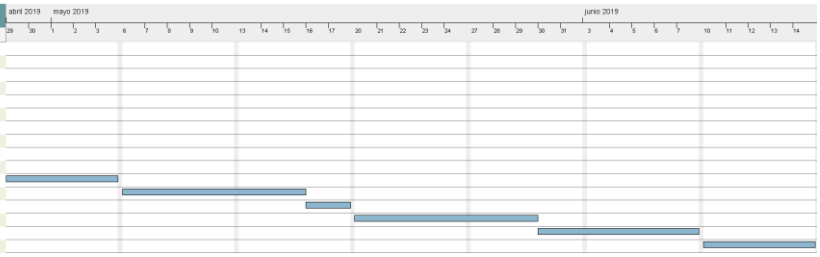
Nombre	Fecha de inicio	Fecha de fin
Esqueleto de la aplicación	25/03/19	26/03/19
Escenas de flujo del juego	26/03/19	26/03/19
TADs de configuraciónde niveles	27/03/19	29/03/19
Sistema Isometrico	1/04/19	5/04/19
Movimiento del caballo	3/04/19	5/04/19
Fusionar TADs niveles i tablero	8/04/19	10/04/19
Fusionar nivel i movimiento del caballo	10/04/19	12/04/19
Tiggers, eventos y propiedades	15/04/19	16/04/19
Tiggers, eventos y propiedades	23/04/19	25/04/19
Condiciones de victoria y derrota	25/04/19	26/04/19
UI escenas y usabilidad	29/04/19	3/05/19
Crear nivles de juego	6/05/19	15/05/19
Cerrar secuencia de juego (inicio y fin)	16/05/19	17/05/19
Efectos visuales y sonoros	20/05/19	29/05/19
Pruebas de estres y equilibrio	30/05/19	7/06/19
Ajustes y resolución de problemas	10/06/19	14/06/19

PEC3: Entre el 29 de Abril i el 16 de Junio

Se reservó para la segunda parte la creación de niveles, cerrar las mecánicas del juego, la interfaz gráfica, usabilidad y los efectos visuales y de sonido. Realizar las pruebas para que la partida esté compensada y las pruebas finales entre lo que sería el final de la segunda etapa del proyecto e inicio de la tercera etapa referida en el punto anterior.

Figura 5. *Diagrama de Gantt (2)*

Nombre	Fecha de inicio	Fecha de fin
Esqueleto de la aplicación	25/03/19	26/03/19
Escenas de flujo del juego	26/03/19	26/03/19
TADs de configuraciónde niveles	27/03/19	29/03/19
Sistema Isometrico	1/04/19	5/04/19
Movimiento del caballo	3/04/19	5/04/19
Fusionar TADs niveles i tablero	8/04/19	10/04/19
Fusionar nivel i movimiento del caballo	10/04/19	12/04/19
Tiggers, eventos y propiedades	15/04/19	16/04/19
Tiggers, eventos y propiedades	23/04/19	25/04/19
Condiciones de victoria y derrota	25/04/19	26/04/19
UI escenas y usabilidad	29/04/19	3/05/19
Crear nivles de juego	6/05/19	15/05/19
Cerrar secuencia de juego (inicio y fin)	16/05/19	17/05/19
Efectos visuales y sonoros	20/05/19	29/05/19
Pruebas de estres y equilibrio	30/05/19	7/06/19
Ajustes y resolución de problemas	10/06/19	14/06/19



En estas fechas, mientras se escribe esta memoria se esté en fase de pruebas con varios *beta testers* para evaluar los últimos aspectos de diseño y obtener una respuesta global al proyecto, analizar sus valoraciones y ajustar parámetros de juego, dificultad, niveles y mecánicas para mejorar la experiencia de usuario.

Una vez diseñado el proyecto se ha trabajado en varias fases para garantizar la focalización del esfuerzo en etapas diferentes del proyecto.

- El primer objetivo era hacer un esqueleto de aplicación para móviles que contenga los elementos básicos de una aplicación profesional. Configuración del motor para dicho propósito, generar las rutinas de ficheros para gestionar valores de configuración y del juego en formato fichero. Crear el flujo básico de la aplicación.

- El segundo objetivo era hacer una estructura interna para configurar los niveles: capas, elementos, objetivos y que el *engine* sea capaz de sobreponer la información sobre el nivel generado a través de *tilesets* isométricos para tener estructuras internas que reflejen la realidad del escenario.

- En tercer lugar poner la figura del caballo y dotarle de movilidad por encima del nivel teniendo en cuenta las peculiaridades del movimiento del caballo en el ajedrez.

- En cuarto lugar, ser capaz de relacionar el movimiento del caballo con la información almacenada del nivel en las estructuras internas. Crear y gestionar los *triggers* que generen los movimientos en relación al escenario. Destrucción de información dinámica.

- Seguidamente deberíamos gestionar esos *triggers* entre el caballo y el escenario en función a los eventos asociados y cómo influyen en las propiedades del ejército del personaje definidas como condiciones de victoria y derrota del nivel.

- El sexto objetivo es dotar al escenario de elementos de usabilidad de usuario, para el movimiento a través del mapa e interfaz de usuario de las diferentes escenas del proyecto.

- En séptimo lugar es necesario crear al menos dos etapas de juego para definir la interacción entre ellas y definir el progreso en el juego y mostrar en ellas todas las mecánicas previstas (elementos, eventos, condiciones de victoria/derrota…).

- En octavo lugar saber definir correctamente una coherencia visual entre todos elementos del juego para tener un producto cerrado y completo. Pantallas de inicio i fin, menús, elementos inter etapas.

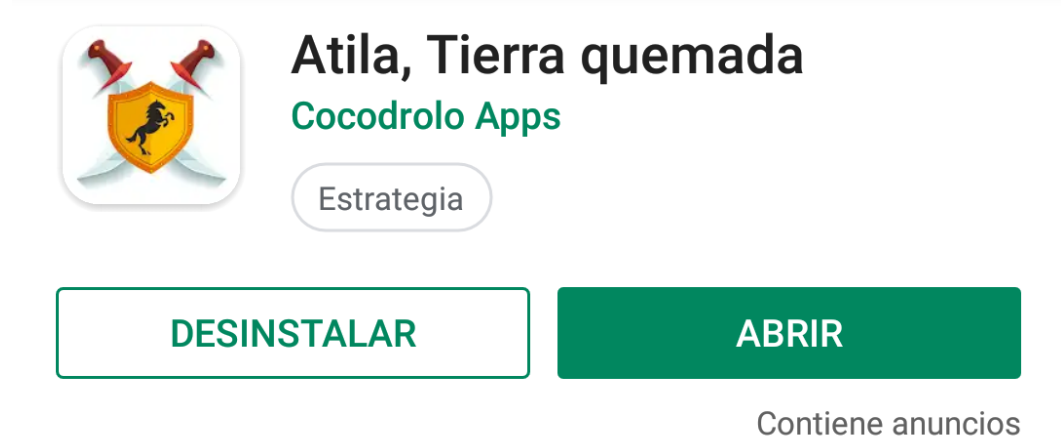
- En noveno lugar colocar los elementos visuales y sonoros, efectos, partículas, *shaders* que dotan al juego de un envoltorio estético uniforme y funcional.

- En decimo y último lugar hacer todas la pruebas, análisis del equilibrio y dificultad entre niveles para determinar que el juego es coherente y divertido.

1.5 Breve sumario de productos obtenidos

Después de realizar el TFM hemos obtenido un juego de estrategia / puzle para dispositivos Android disponible en la tienda de aplicaciones Google Play para su despliegue en una fecha futura próxima.

Figura 6. *Referencia en Google Play Store*



2. Estado del arte

Atila, Tierra quemada es fundamentalmente una mezcla entre un juego de estrategia y un juego de resolución de puzles. Esta propuesta se basa en un juego de lógica base como es el juego del movimiento del caballo de ajedrez sobre un tablero de 8x8 casillas como máximo sobre un trasfondo histórico que al ser lejano en el tiempo y parcialmente documentado permite a través de una cierta flexibilidad histórica recrear un periodo poco explotado en juegos móviles.

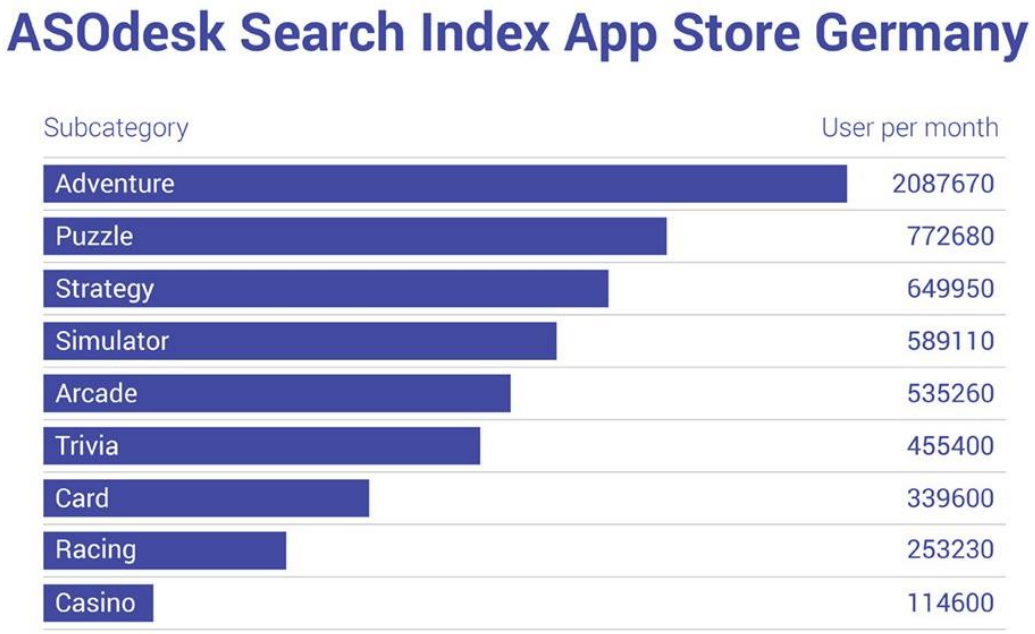
Para ello se ha intentado mezclar componentes que tal y como describe *Durgan A. Nallar* en su obra **Diseño de videojuegos en América Latina** que favorecen que un juego cumpla con el propósito que todo juego debe buscar, **ser divertido**. El autor define como elementos a tener en cuenta para que un juego sea divertido.

- **Ganar**
- **Resolver problemas**
- **Explorar**
- Trabajo en equipo
- **Reconocimiento personal**
- Coleccionismo
- **Sorpresa**
- Imaginación
- Compartir
- **Jugar con un rol**
- Personalización
- Perder el tiempo

En este caso los elementos en remarcados son elementos buscados en el diseño del juego para favorecer la diversión del jugador, probablemente un perfil de jugador limitado por la temática i el tipo de dispositivo, pero que encaje en el objetivo de ofrecer un juego divertido.

En concreto los juegos de estrategia y puzles son de los tipos de juegos más demandados en dispositivos móviles tal y como indica los análisis de mercado que diferentes empresas ASO analizan para sus clientes.

Figura 7. *Análisis del mercado móvil en Alemania (Mayo, 2019)*



Las aplicaciones móviles de temática estratégica y puzles son de las más demandadas en las búsquedas que hacen los usuarios en las tiendas de aplicaciones para dispositivos móviles.

Aunque hacer un juego divertido es una premisa básica de todo juego, hacer un juego demandado por el mercado no lo es. Así mismo al tener en cuenta elementos de monetización en el proyecto creo que es relevante tener en cuenta el análisis del mercado para enfocar un proyecto como simulación de un proyecto empresarial más allá de que se trate de un proyecto de carácter académico.

Así mismo se tienen en cuenta otros elementos además de los descritos para el diseño y desarrollo del juego. Aspectos como el arte y la estética también son importantes como transmisores de la experiencia y las emociones. Hasta ahora, en un análisis exhaustivo del mercado sobre el género, no hay títulos con calidad artística relevante o un enfoque creativo a un juego de tablero como es el movimiento del caballo de ajedrez. En proyecto también es esto, y el enfoque propuesto también forma parte de la innovación gráfica de este tipo de juegos.

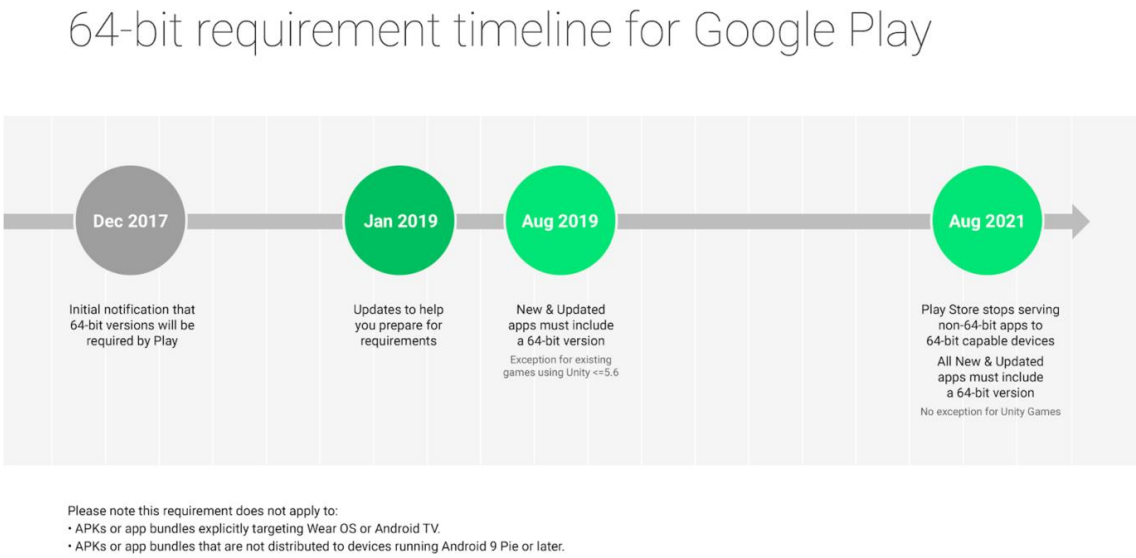
Todo ello conforma el fondo de trabajo para el TFM. En este caso la plataforma de desarrollo escogida para la implementación del proyecto es **Unity**, ya que dispone de todas la herramientas actuales para la programación de juegos para dispositivos móviles y cumple con los estándares completos que debe cumplir para satisfacer las exigencias de las plataformas de aplicaciones que cambian a menudo y que otras herramientas a veces no pueden satisfacer.

En concreto, el objetivo de Google es que las aplicaciones descargadas por los usuarios estén optimizadas para las versiones más recientes de Android.

Véanse algunos ejemplos:

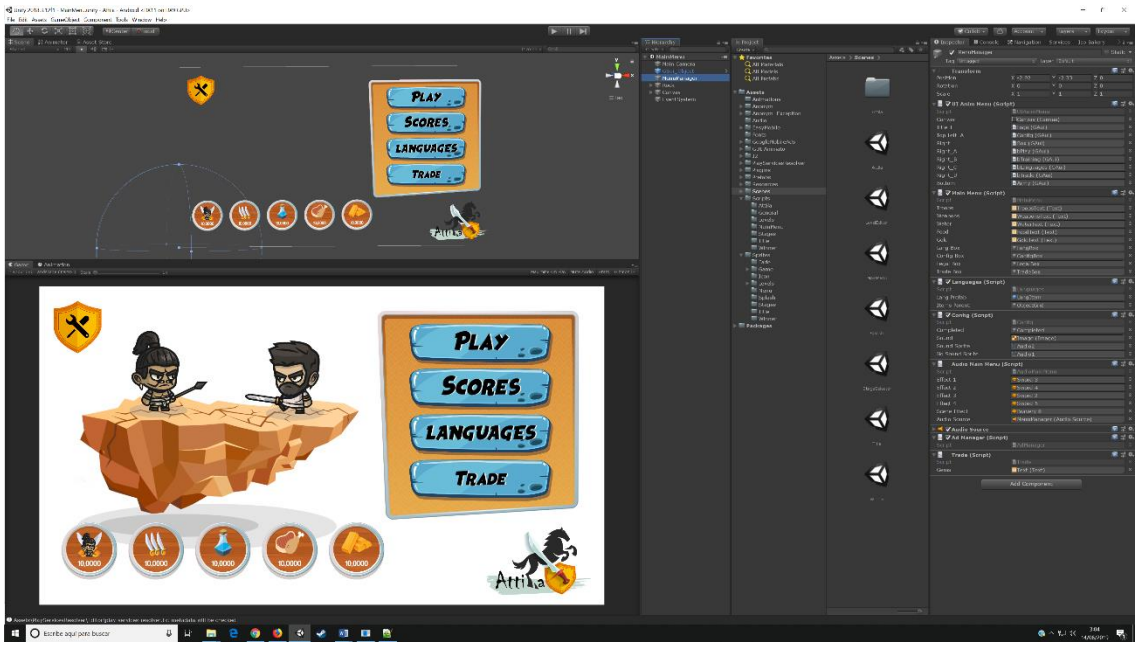
- Agosto de 2019: las nuevas aplicaciones tendrán que estar optimizadas a Android 9 (API 28) o superior.
- Noviembre de 2019: las actualizaciones de las aplicaciones existentes tendrán que estar optimizadas a Android 9 (API 28) o superior

Figura 8. *Requerimientos para aplicaciones Android 2019*



Aunque para ello se podría utilizar otras herramientas optimizadas, Unity está especializada en la creación de juegos para diferentes plataformas, con lo que si en un futuro se quisiese portar el juego a otras plataformas como iOS, Nintendo Switch, PS4 u otras, esto sería posible.

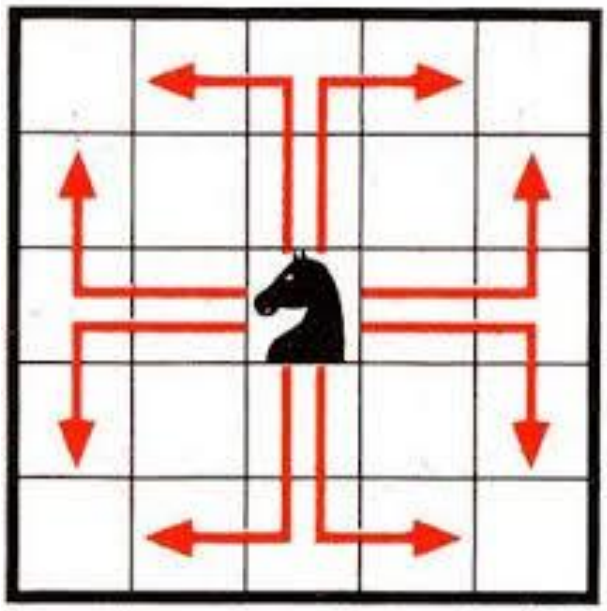
Figura 9. *Atila, Tierra quemada en su entorno de desarrollo Unity 2018.3*



3. Definición del juego

Originalmente el juego del caballo de ajedrez se juega con una hoja cuadriculada y un bolígrafo. A partir de una cuadricula de 8x8 y se llenan los cuadritos interiores con números haciendo el salto del caballo del ajedrez.

Figura 10. *Definición del movimiento del caballo en el juego de Ajedrez*



A partir de este problema se quiere desarrollar un juego de estrategia por niveles donde a partir de un punto inicial debemos recorrer las máximas casillas posibles antes de quedarnos atascados o llenar todo el tablero. Cada avance nos reportará puntos que iremos acumulando e iremos nutriendo las propiedades del caballo que simboliza nuestro ejército.

A partir de un nivel se irán encontrando elementos/casillas con propiedades particulares que modificaran de alguna manera la mecánica del juego (ciudades, ríos, montañas, suministros, etc.) permitiendo ganar más puntos en el camino hacia las casillas objetivo o final de cada etapa. Además no todos los tableros serian de 8x8 y rectangulares. Se puede encontrar tableros con otras formas y tamaños que nos permiten ajustar la dificultad del juego en función de la cada etapa.

A través de un conjunto de niveles con dificultad progresiva iremos avanzando por la historia de conquista de los ejércitos de Atila, has llegar a Roma si conseguimos terminar el juego con un nivel final que sería el desafío de Euler con la opción de recrear el reto original del reto del caballo.

3.1 Trasfondo histórico de un juego de estrategia

Aunque la mecánica básica queda claramente definida por el problema de Euler, era necesario buscar un trasfondo histórico para convertir el juego en un juego de estrategia.

En este caso el ejemplo histórico más interesante son las tres incursiones de Atila, rey de los hunos, en el Imperio Romano. Los hunos eran conocidos como unos excelentes

jinetes, desde donde atacaban con la espada, lanza o arco, con sus rápidos movimientos estratégicos, acabar con sus enemigos. Esta característica histórica es un excelente hilo argumental donde el caballo simbolizando el ejército huno se desplaza por diversas etapas que representan los avances por el oriente y occidente romano.

Figura 11. *Incursiones de Atila en el imperio romano*

En concreto, Atila, entre los años 443 y 452 de nuestra era, hizo tres incursiones en territorio romano, que en juego serán represadas por **3 grandes bloques con 10 etapas cada una**. La primera incursión (443) la sufrió el imperio romano de oriente, hasta las puertas de Constantinopla. La segunda incursión fue en su intento de conquistar la Galia (451) hasta ser derrotado en la batalla de los Campos Cataláunicos y obligado a retroceder. Finalmente en 452 atacó la península itálica hasta las puertas de Roma donde la leyenda cuenta que sólo la intervención del papa León I detuvo las tropas de los hunos y se retiraron a sus territorios.

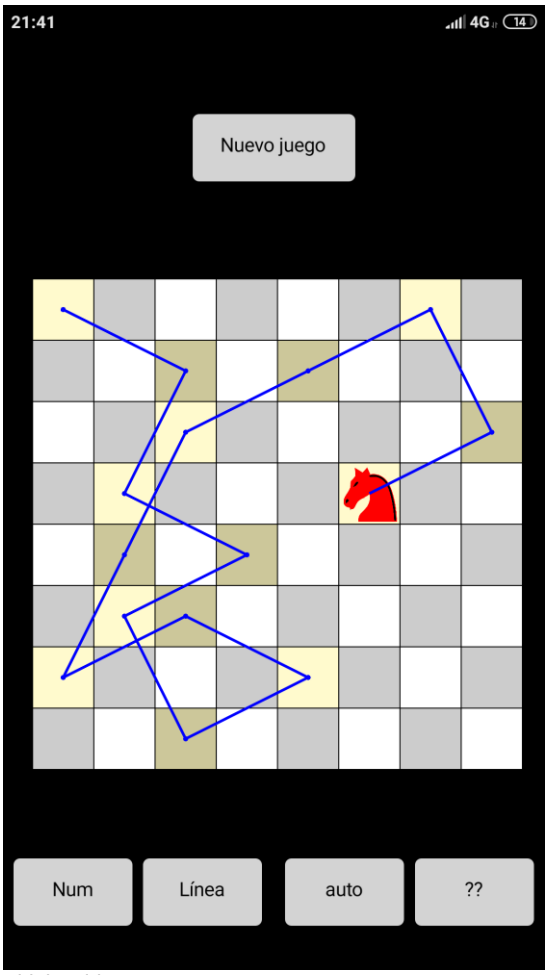
3.2 Subgénero y referencias

El juego es una mezcla de juego de estrategia y solución de problemas lógicos. Por un lado tendríamos la mecánica base de movernos a través de casillas con el movimiento del caballo del ajedrez y por el otro lado la gestión del ejército de Atila que debe recorrer el mapa para conseguir su objetivo.

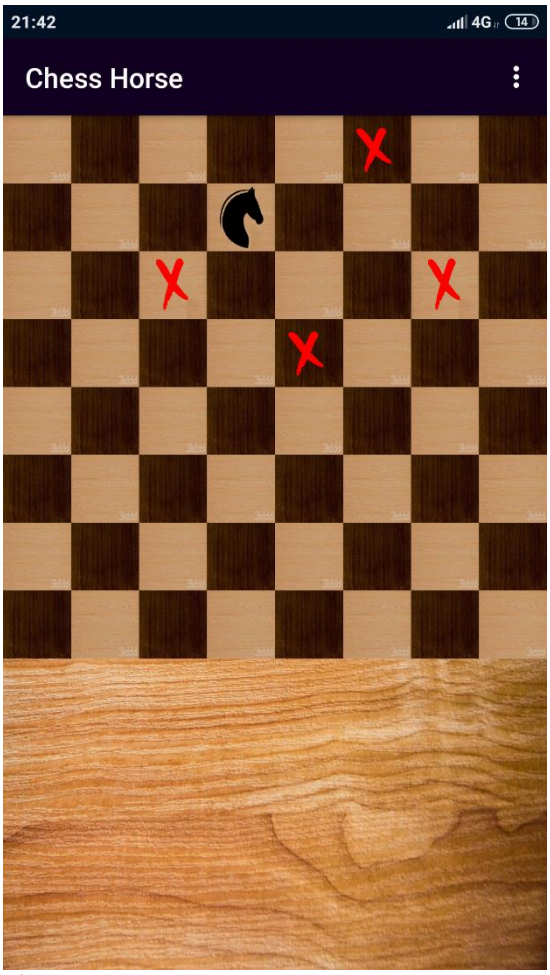
Para diseñar el juego se han buscado referencias en el mundo de las aplicaciones móviles ya que este sería el formato del juego. Después de hacer un análisis se pueden agrupar en 3 enfoques diferentes:

Juegos del caballo clásicos, donde el objetivo es rellenar un tablero sin repetir casillas. Algunos ejemplos pueden ser *Khight Ride* o *Chess Horse*, además de algunos juegos para plataformas web que buscan el mismo enfoque.

Figura 12. *Capturas de Knight Ride y Chess Horse*



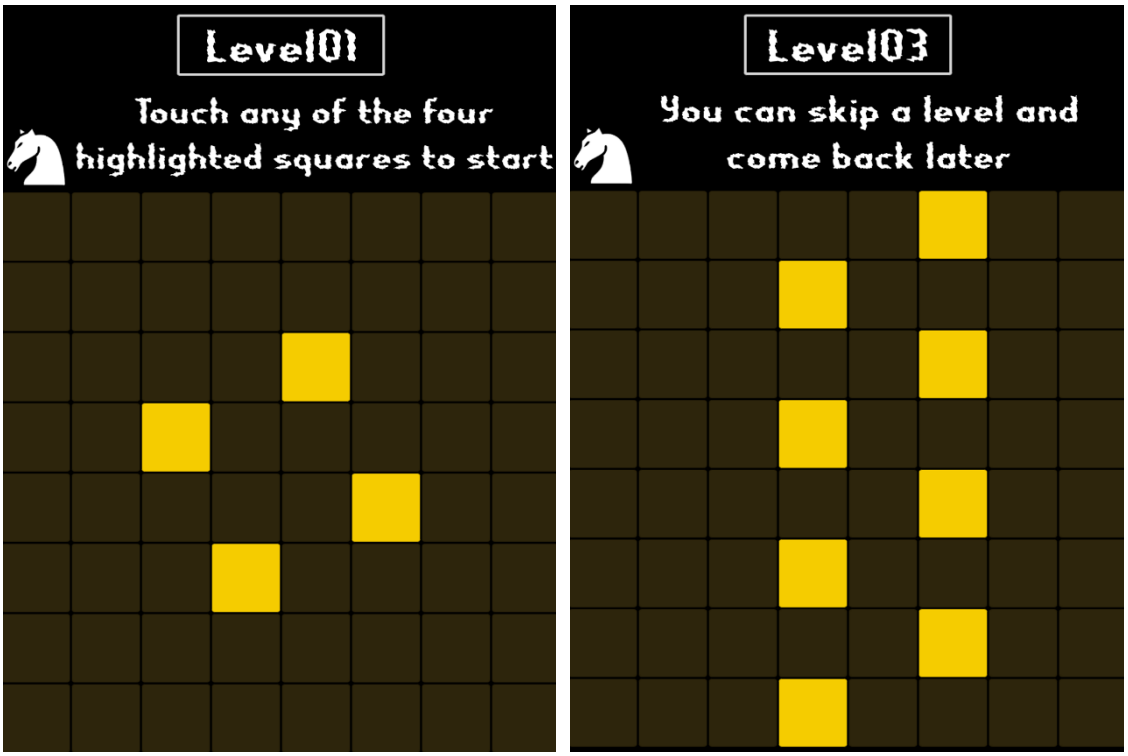
Khight Ride



Chess Horse

Por otro lado, también están juegos con un caballo de ajedrez como protagonista donde la mecánica básica consiste a partir de una cuadrícula y unos puntos clave, tener que pasar por todos ellos sin cortarse el camino y no poder pasar por alguno. Estos juegos se dividen en niveles donde la dificultad va aumentando progresivamente en el transcurso de la partida. Un ejemplo de este tipo de juegos podría ser el juego para Android *Chess Knight*.

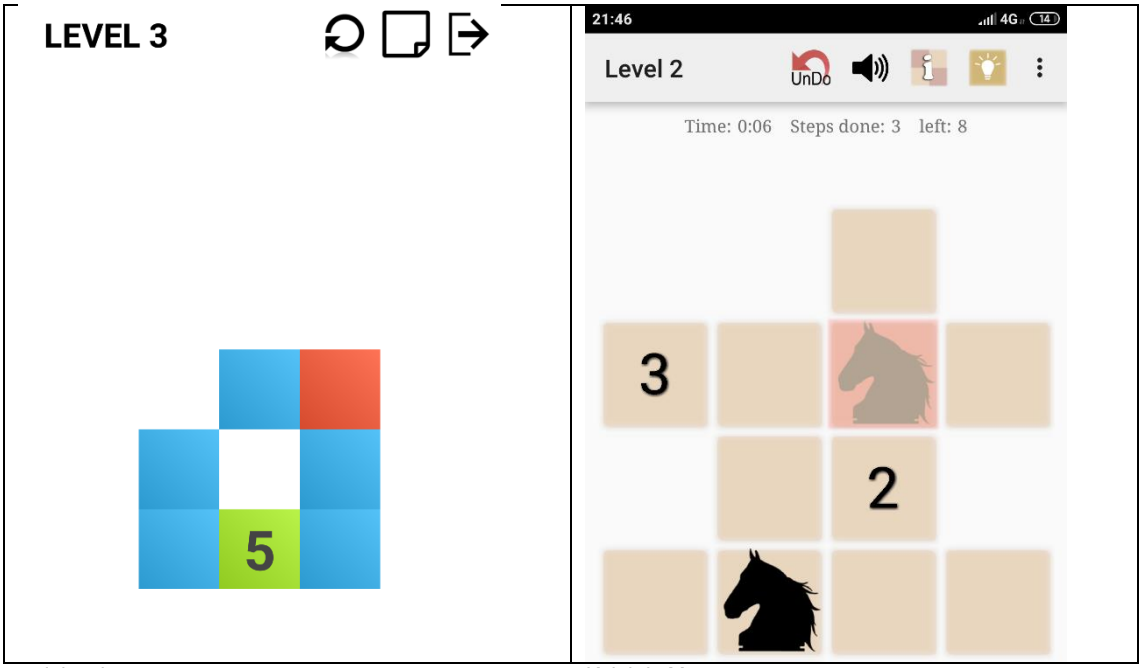
Figura 13. *Captura del juego Chess Knight*



Chess Knight

El tercer enfoque consiste en tener que eliminar todas las casillas (recorriéndolas) con tableros de formas irregulares parar graduar la dificultad. Ejemplos de esta mecánica podemos encontrar juegos como *Knight's Move* o *Knight Chess*.

Figura 14. *Captura de los juegos Knight Chess y Knight's move*



Knight Chess Knight's Move

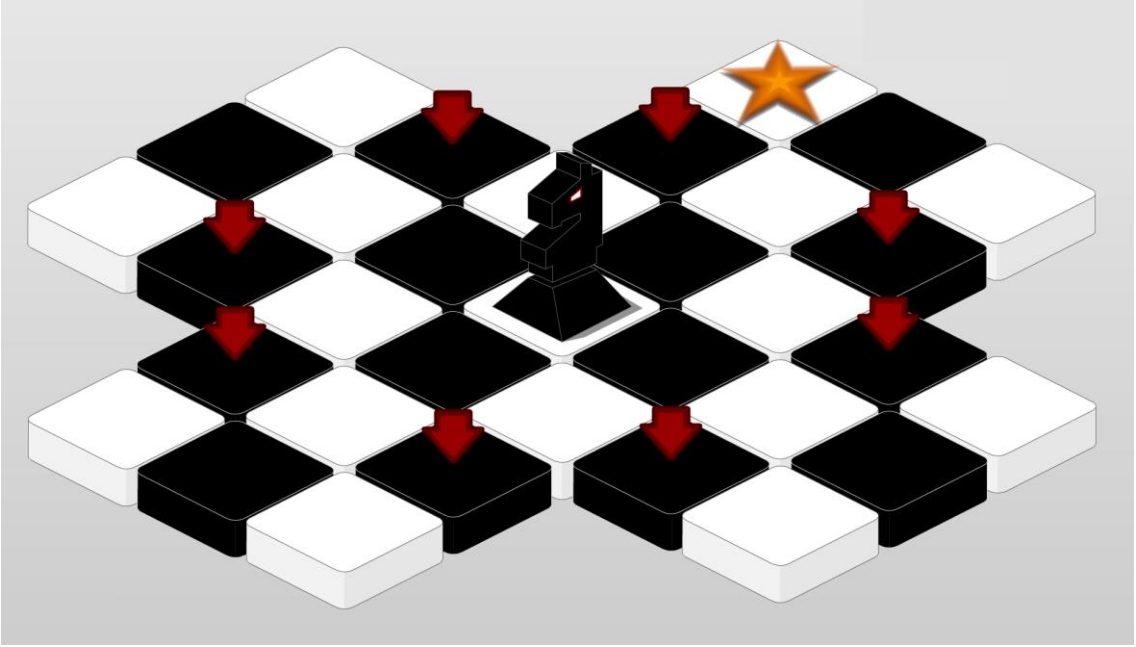
Por último el último referente que se ha encontrado son aquellos que utilizan el juego del caballo como excusa para hacer un juego de tipo puzle donde debemos recoger monedas para completar un nivel. En estos casos podemos repetir casillas un número *n* de veces y algunas casillas tienen propiedades especiales que pueden beneficiar o perjudicar al jugador en su objetivo. Además podemos usar recursos especiales que van más allá del movimiento del caballo para transmutar al personaje o al entorno de juego.

3.3 Interacción

La propuesta del juego consiste, al ser un juego para dispositivos móviles, un avatar con el que jugador interactúa que se mueve por un tablero irregular a través del movimiento del caballo. Como la plataforma de destino será un dispositivo móvil podrá utilizar mecánicas de *click and go* para mover el caballo.

Otro elemento a tener en cuenta en el desarrollo de aplicaciones móviles es intentar hacer lo más complejo lo más simple posible. Esto sugiere que debemos dar pistas al jugador sobre que casillas tiene disponibles para el siguiente salto. Dominar la técnica del movimiento de ajedrez no debe ser condición indispensable para jugar al juego, ya que la dificultad del mismo está en el recorrido y no en el propio movimiento.

Figura 15. *Movimiento del caballo sobre un tablero*



Así mismo, el jugador deberá llegar a su destino obteniendo el máximo de recursos posibles en el camino y salvando los obstáculos que se interponga siempre sin repetir nunca la misma casilla.

3.4 Ambientación y trama

Al pensar en el juego y al buscar una frase que pueda resumir la esencia del juego se podría decir este juego de palabras:

Atila, tierra quemada
Por donde pasa su caballo
No quedará nada.

Atila, rey los hunos, llevo su ejército de jinetes desde la estepa rusa hasta las puertas de Roma y Constantinopla. El juego quiere representar el movimiento del ejército huno a través de Europa proponiéndole al jugador objetivos en cada etapa que condicionen el juego en etapas posteriores bajo la premisa que **nunca podrá pasar dos veces por la misma casilla**. Al pasar por una casilla esta se destruye y no podemos pasar más por ella.

Aprovechando que parte de la historia de Atila es desconocida por los historiadores ya que sólo han llegado a la actualidad básicamente textos romanos de la época, con lo cual hay mucha libertad creativa para construir la historia deseada.

El juego busca por un lado el contraste ente una mecánica muy simple para un juego de estrategia (**el caballo del ajedrez y la no repetición de casillas**) y un trasfondo ficcional de la construcción de una historia a través de objetivos en cada nivel que el jugador debe cumplir para seguir con la aventura.

Además tiene una parte de **gestión estratégica de recursos** ya que para seguir avanzando no se puede descuidar el estado del ejército y sus necesidades. Si el jugador se queda sin recursos el caballo no podrá avanzar y perderemos la partida.

- Tropas
- Armas
- Agua
- Comida
- Oro

Figura 16. *Recursos necesarios del ejército de Atila*

Así mismo, estos dos factores permiten un conjunto amplio de estrategias para el jugador, ya que no sólo debe conseguir los objetivos marcados en cada etapa, sino que además obliga tener en cuenta que el **camino recorrido también sea importante** para alimentar, pagar y suplir de recursos en todo momento a los jinetes hunos.

3.5 Personajes e interacción

Básicamente hay un solo personaje activo en el juego, el caballo que representa el ejército de los hunos. Este ejército tiene unas propiedades que reflejan las necesidades básicas de cualquier ejército (agua, comida, efectivos, armas, oro).

El resto de elementos son elementos sobre el tablero que interactúan con el personaje cuando el caballo pasa por una casilla con otro elemento interactuarle y pueden alterar las propiedades del caballo de Atila.

En el juego podremos encontrar **elementos pasivos** que modelan nuestro camino como elementos naturales o **elementos activos** que modificaran las propiedades del ejército y que forman parte del objetivo de la etapa actual del juego.

Algunos ejemplos:

Ciudades. Podemos saquearlas para obtener recursos si nuestro ejército es superior a sus defensas o pagar tributo para cruzar sus fronteras o ser derrotados si perdemos la batalla.
Casillas objetivo. Casillas que debemos pasar obligatoriamente para terminar la etapa.
Ejercito enemigo. La batalla condicionará cómo evoluciona el juego.
Campos. Podemos obtener comida y agua.
Ríos y lagos. Podemos obtener agua.
Minas y canteras. Podemos obtener oro.

Figura 17. *Movimiento del ejército, objetivos y casilla final*

Mecánicas de etapas:

1. El jugador empieza en una posición del tablero
2. Hay casillas objetivos que deben ser eliminadas
3. Ataca la casilla final
4. Avanza hasta la siguiente etapa

Mecánicas básicas:

1. El jugador mueve el caballo
2. Recoge los recursos de la casilla destino
3. Interactúa con la casilla si es necesario y puede modelar su ejercito
4. Modifica las propiedades del ejército en función de la interacción
5. La casilla de origen se destruye.

3.6 Objetivos del juego

El juego sólo tiene un objetivo: **conquistar Roma**. Si el jugador supera todas las 30 etapas y su ejército es suficiente fuerte para llegar a la ciudad, gana. Al ser un juego para dispositivos móviles está dividido en etapas para segmentar el tiempo de juego y poder re-jugarlas en caso de dejar la partida a medias, morir o quedar bloqueado en alguna de ellas.

Figura 18. *Objetivos y casillas finales*

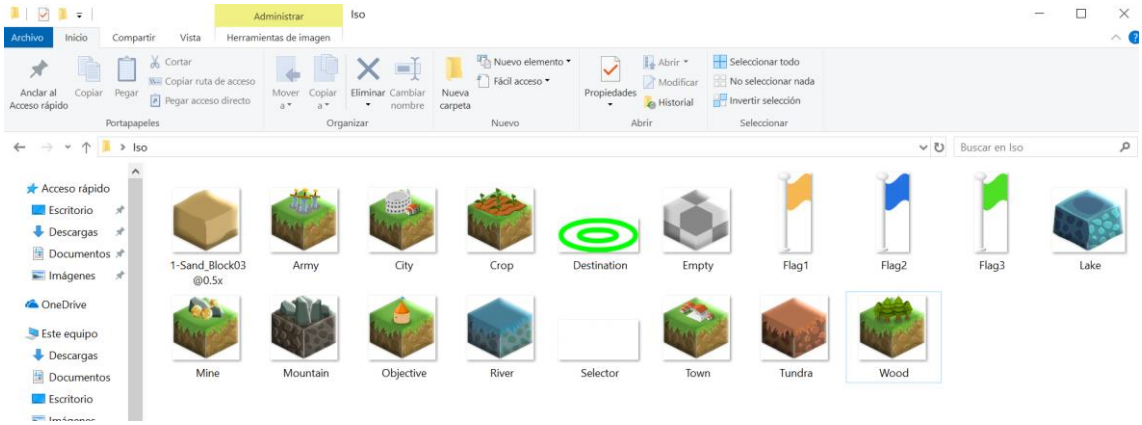
Cada etapa tendrá un punto de inicio y un punto final. El jugador debe recorrer eficientemente el tablero de juego para recoger recursos para mantener y mejorar su ejército. El caballo deberá pasar por puntos de control (objetivos) para superar la etapa que pueden provocar eventos con otros actores del tablero de juego. Si en algún momento el jugador no puede mantener a su ejército, no puede llegar a alguna casilla de control o es derrotado, el jugador pierde la etapa y debe empezarla de nuevo.

El jugador puede modificar las propiedades del ejército utilizando el oro disponible para reasignar recursos al ejército y seguir avanzando.

3.7 Concept art

El tablero estará compuesto por casillas isométricas i una ambientación de la época romana y la estepa rusa.

Figura 19. *Tilesets isométricos (propuesta 1)*



El modelo isométrico busca un atractivo visual en dispositivos móviles que otras aplicaciones del segmento no tienen además de conseguir una estética más actual y acorde a otros juegos estratégicos.

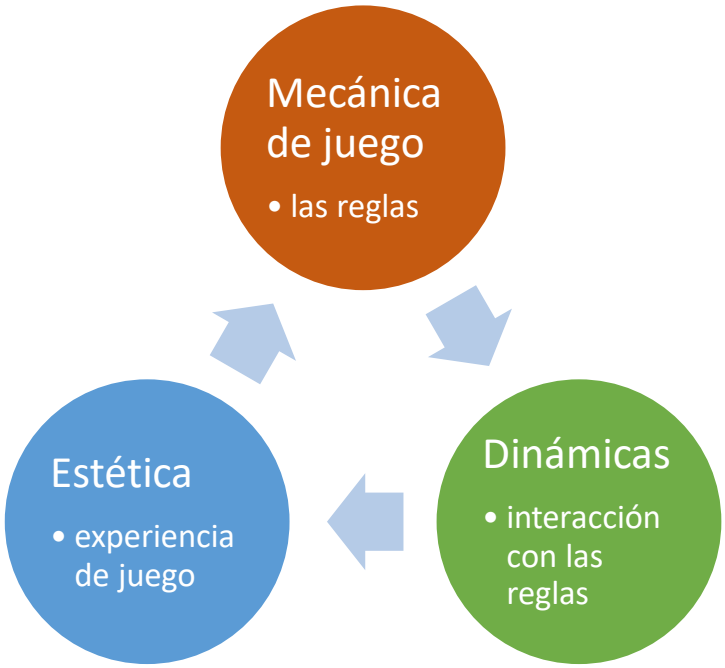
Así mismo se busca una imagen de acción que contrarreste los recelos de los jugadores más activos donde esperan encontrar unas mecánicas con más movimiento.

Figura 20. *Icono y logotipo del juego*

4. Diseño técnico

El diseño de un juego para dispositivos móviles se basa en tres premisas básicas que comparten con el diseño de cualquier juego.

Figura 21. *Flujo de juego*



Este *framework* condiciona todas las decisiones del diseño del juego. Además es necesario probar y evaluar cada decisión para comprobar si podemos mantener el *flow* en los jugadores o estos pierden interés.

Para ello ha sido necesario trabajar en el diseño de las mecánicas del juego en un entorno móvil, dinámico y muchas veces intermitente. A partir del trabajo de *Jesse Schell* (The Art of Game Design) se ha profundizado en el mundo de las interfaces para videojuegos y que tenerlo en cuenta al diseñar las pantallas del juego.

La interfaz de usuario es la herramienta que utiliza el jugador para comunicarse con el juego, es donde el jugador y el juego se unen. Cuando la interfaz falla, la experiencia que surge de la interacción jugador-juego no funciona. Es crucial entender cómo funciona nuestra interfaz de juego y hacerse tan robusta, y tan invisible como se pueda.

Lo que se busca en el juego es establecer esta conexión entre el juego y el jugador:

1. Extrema atención en una tarea.
2. Un sentido de control activo.
3. Fusión de la acción y la conciencia.
4. La pérdida de la conciencia de sí mismo.
5. La distorsión de la experiencia del tiempo.
6. La experiencia de la tarea de ser la única justificación necesaria para continuarlo.

4.1 Requerimientos técnicos

Para el desarrollo del juego se ha utilizado el **motor de Unity** aplicado al desarrollo de aplicaciones Android. Esto implica tener disponible varias herramientas de trabajo para poder crear los archivos de distribución y poder probar y distribuir un juego para la plataforma de Google. Referenciado por la propia herramienta tenemos:

Para desarrollo:

- OS: Windows 7 SP1+, 8, 10, 64-bit versions only; macOS 10.12+
- CPU: Soporte para el conjunto de instrucciones SSE2.
- GPU: Tarjeta de video con capacidad para DX10 (shader modelo 4.0).

El resto depende principalmente de la complejidad de sus proyectos.

Requisitos adicionales para el desarrollo de plataformas:

- **Android**: Kit de desarrollo Android SDK y Java (JDK); el IL2CPP scripting backend requiere Android NDK.
- **Plataforma Windows universal**: Windows 10 (64 bits), Visual Studio 2015 con componente C++ Tools o posterior y SDK para Windows 10

Para ejecutar juegos de Unity:

Por lo general, el contenido desarrollado con Unity puede ejecutarse bastante bien en todas partes. Qué tan bien se ejecuta depende de la complejidad de su proyecto.

Requisitos más detallados:

Escritorio:

- OS: Windows 7 SP1+, macOS 10.12+, Ubuntu 16.04+
- Tarjeta de video con capacidad para DX10 (shader modelo 4.0).
- CPU: compatible con el conjunto de instrucciones SSE2.
- El reproductor de iOS requiere iOS 9.0 o superior.
- Android: OS 4.1 o posterior; ARMv7 CPU con soporte NEON o CPU Atom; OpenGL ES 2.0 o posterior.
- WebGL: Cualquier versión de escritorio reciente de Firefox, Chrome, Edge o Safari
- Plataforma Windows universal: Windows 10 y una tarjeta de gráficos con capacidades DX10 (modelo de shader 4.0)

4.2 Herramientas de desarrollo

Además del motor Unity anteriormente justificado y definido, se requieren de otras herramientas para la gestión y el desarrollo del proyecto y cubrir todo tipo de necesidades.

Desarrollo:

- Unity 2018.3
- Visual Studio
- Notepad++

Gestón de versiones y flujo de trabajo:

- Git y SourceTree (gestor de versiones y respaldo en línea)
- GanttProject (creación y gestión de diagramas de Gantt)

Gráficos:

- Adobe Ilustrator (Herramienta de imágenes vectoriales)
- Adobe Photoshop (Herramienta de retoque fotográfico)
- Adobe Animate (Herramienta de animación)
- Paint.net (Editor simple de dibujo)
- Camtasia (Capturado y editor de vídeo)
- 4k video Downloader (Descarga de vídeos en línea)

En línea:

- Banco de imágenes Freepik
- Itch.io (graphics and music assets)
- Gamedevmarket (graphics assets)

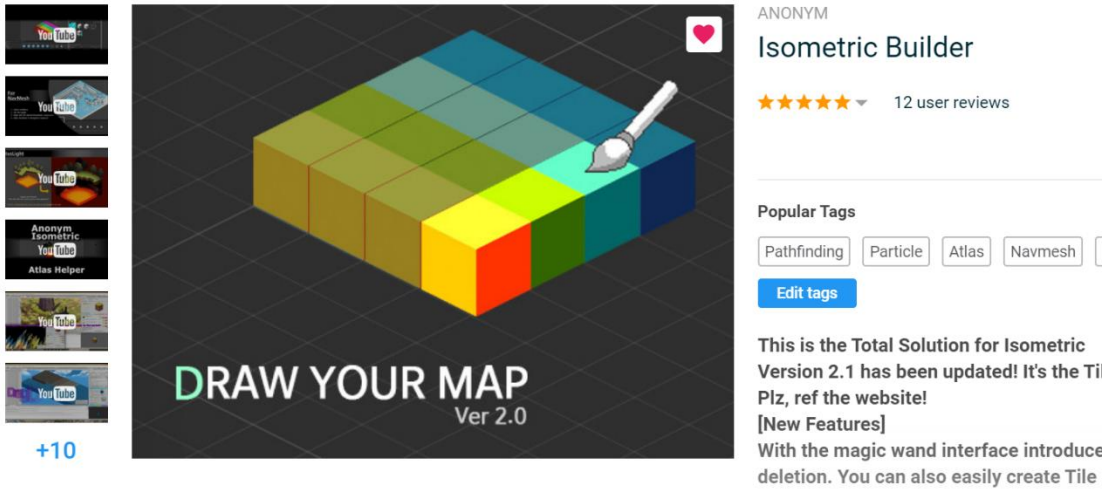
Estas son las herramientas utilizadas en los diferentes apartados del diseño y desarrollo del juego. Aunque algunas de ellas son herramientas de pago, disponen de versiones gratuitas limitadas en el tiempo, versiones para educación y otras de las cuales ya disponía por mi carrera profesional.

4.3 Assets y recursos del juego

Isometric builder:

La facilidad de uso y la separación de tilesets en el área de trabajo de unity para poder añadir clases asociadas a cada uno de las casillas del juego han decantado la balanza para utilizar Isometric Builder ya que trata las casillas como elementos independientes dentro del árbol de objetos de la escena y permite tener buen un control del escenario.

Figura 22. *Isometric builder*



I2 Localization:

En el juego que estamos desarrollando es primordial llegar al máximo público posible. Al ser un juego de estrategia es necesario explicar a través de un tutorial explicar las opciones que en cada momento tiene el jugador. Para ello es importante que le jugador pueda entender bien las instrucciones con lo cual llegar a través de un idioma conocido es importante.

Figura 23. *I2 Localization*

En el proyecto utilizo el asset de Unity creado por Inter Illusion: I2 Localization. Este asset nos permite cambiar, entre muchas otras cosas, dinámicamente el idioma del proyecto y localizar el juego para cada jugador. L2 puede utilizar el traductor de Google para traducir el juego a varios idiomas de forma dinámica y gestionar varias fuentes para lenguas no latinas. Todo ello a través de la interfaz integrada en el propio editor de Unity para máxima comodidad.

Easy Mobile:

Asset para la gestión publicitaria de la aplicación, interacción con Google Play Game, sistema de valoración de aplicaciones y notificaciones en el dispositivo móvil, recompensas, logros y tablas de puntuaciones.

Figura 24. *Easy Mobile*

2D Isometric Starter

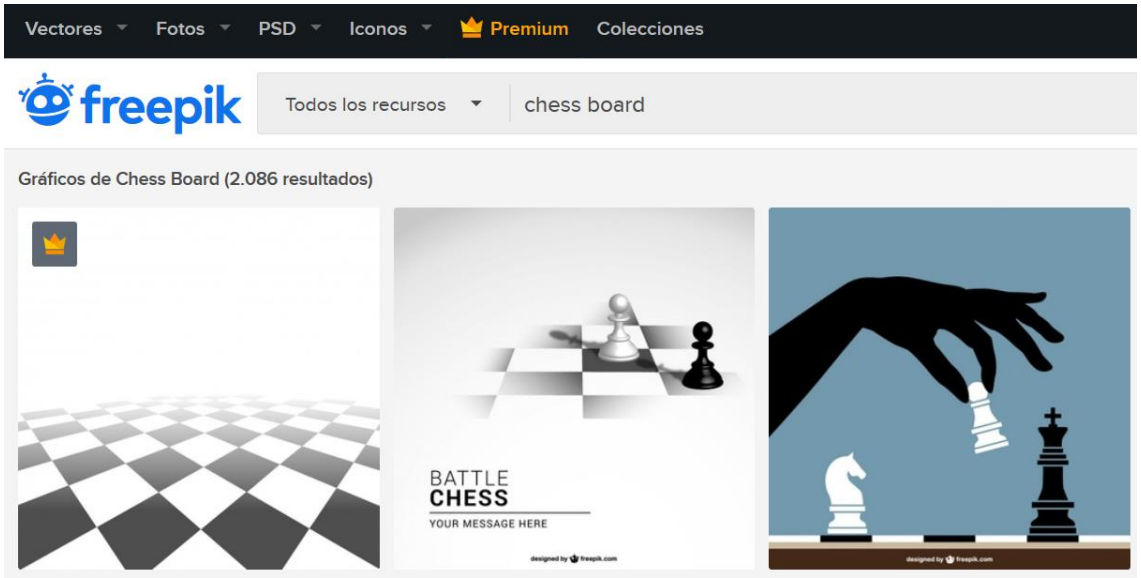
Recursos gráficos con un conjunto de tilesets isométricos. Este recurso ha sido muy útil para empezar el proyecto, aunque se ha tenido que reacondicionar para ajustarlo a la temática y necesidades específicas del juego.

Figura 25. *2D Isometric Starter*

Freepik (Banco de imágenes)

Todos los gráficos utilizados en el proyecto son elementos de los cuales se dispone de la licencia para uso y manipulación. Hace años que utilizo la página de **FreePik** (https://www.freepik.es) como repositorio de imágenes y recursos gráficos para los proyectos profesionales de la cual se paga una licencia anual para su uso.

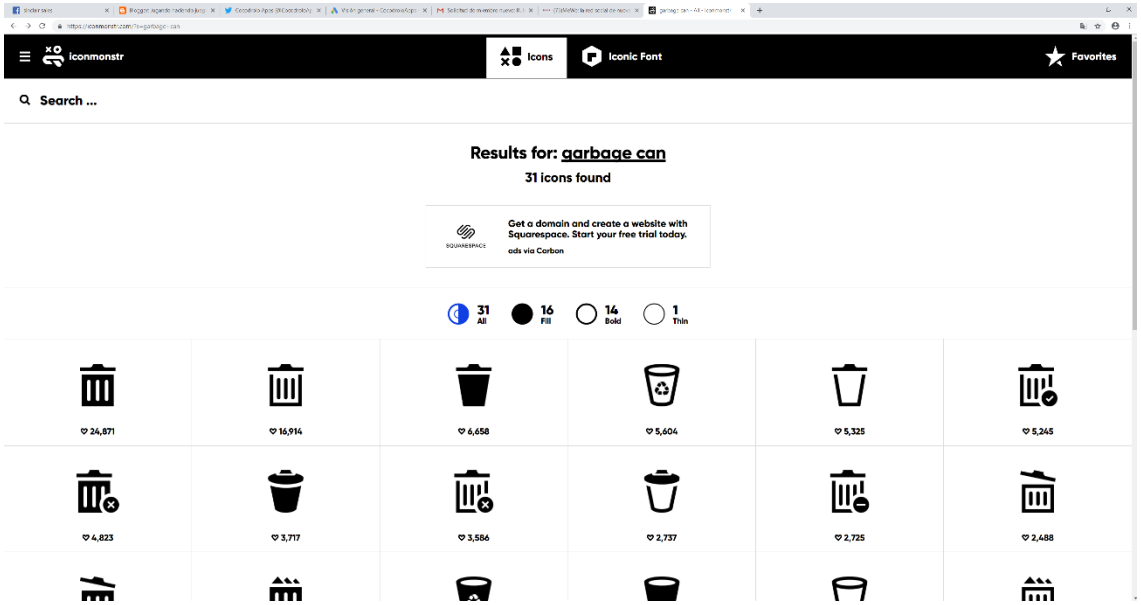
Figura 26. *freepik*



Iconmstr

Página web de iconos gratuitos para uso personal y comercial. Página especializada en iconos monocromáticos de buena calidad, uniformes y de diseño compacto.

Figura 27. *Iconmstr*



4.4 Los gráficos y elementos visuales

Combinando estos tres últimos recursos se ha construido todos los elementos gráficos del juego. Utilizando herramientas de dibujo vectorial y fotográfico y animaciones se han construido los elementos gráficos del juego (logo, iconos, animaciones, tiles, etc.) La mayor parte de los gráficos que se pueden encontrar son composiciones, distorsiones o elementos parciales de otros elementos gráficos creando los gráficos actuales con una continuidad artística que de homogeneidad a todo el apartado gráficos del juego.

Figura 28. *Ejemplo de composición gráfica*

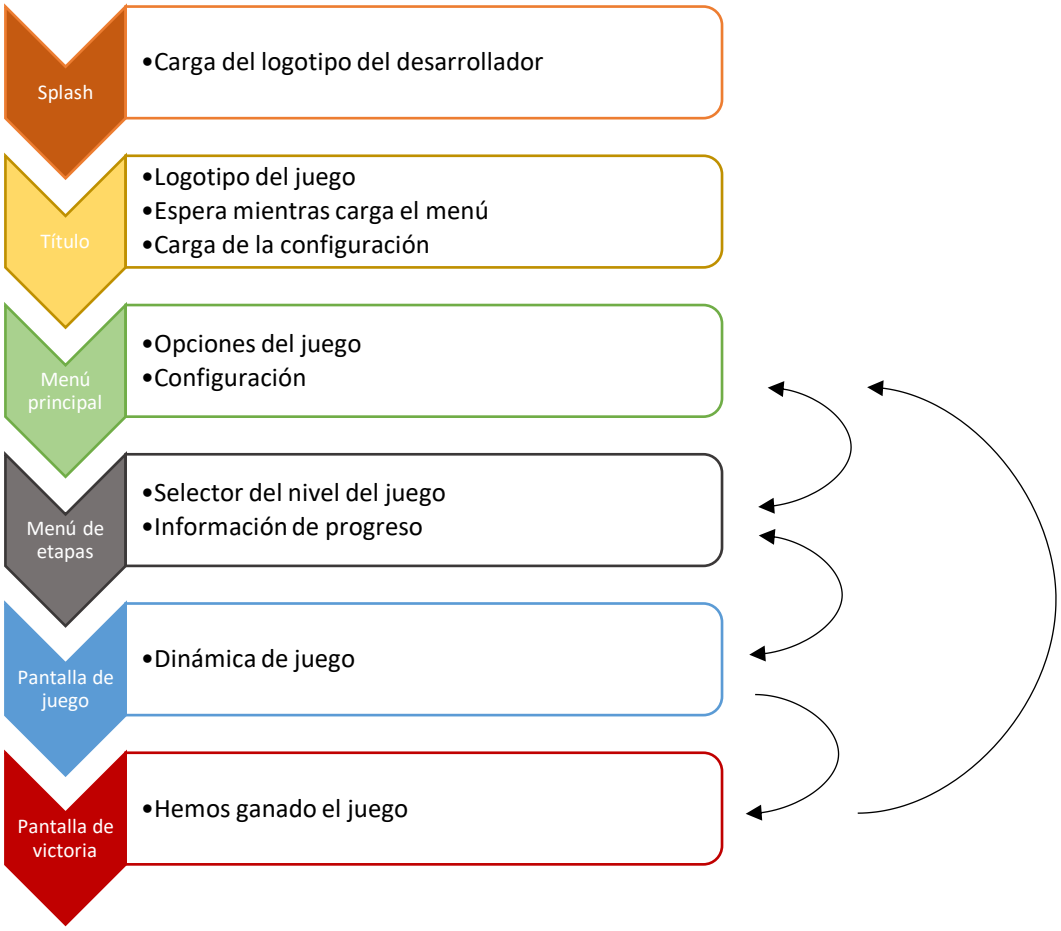
Elementos base

4.5 Arquitectura del juego

Flujo de juego

El juego consiste en un conjunto de escenas que configuran el flujo por donde el jugador interaccionará con el mundo virtual diseñado. Algunas de estas etapas pertenecen a la arquitectura básica de una aplicación (juego) para dispositivos móviles, ya sea por convenio, tiempo de carga o necesidad de procesar datos que en otros entornos serian innecesarios o virtualmente instantáneos.

Figura 29. *Flujo de juego*



En este caso se utiliza la escena de título para cargar la configuración del jugador mientras cargamos la escena del menú del juego. En esta segunda escena el jugador ya percibe el estado de sus tropas y se le muestran las opciones comunes que puede hacer en un juego de estas características

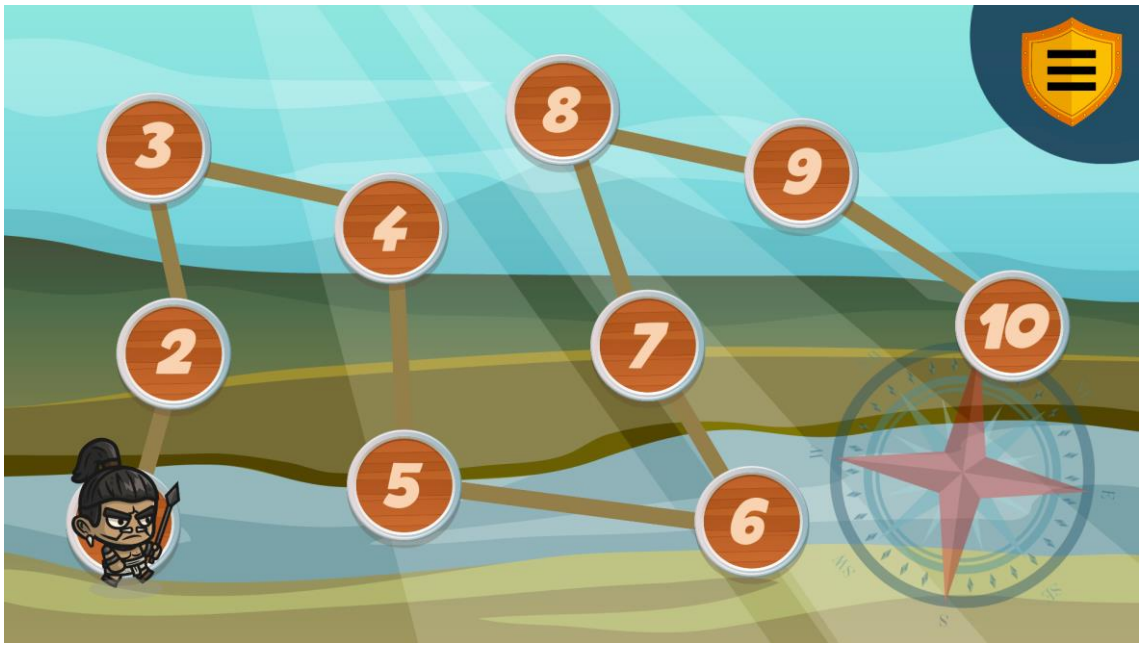
- Jugar
- Cambiar el idioma (6 opciones disponibles).
- Ver las estadísticas de juego
- Mercadear gemas (puntos) por oro
- Acceder a las opciones
    - Sonido ON/OFF
    - Reiniciar juego
    - Compartir aplicación
    - Valorar el juego
    - Ver las condiciones legales

Figura 30. *Menú principal y opciones*

Una vez accedemos a jugar pasamos a una escena de selección de la etapa actual del juego representada por una figura de un guerrero huno sobre el número de nivel actual.

Este recurso de usabilidad es muy utilizado en todo tipo de juegos que desean mostrar la progresión del jugador a través de múltiples niveles. Este recurso es fácil de entender para cualquier jugador que juegue a juegos para dispositivos móviles segmentado en niveles. Una vez pulsamos sobre el guerrero huno, accedemos a la escena de juego.

Figura 31. *Selector de etapas*



En la escena de juego debemos progresar por el tablero a través de las mecánicas descritas anteriormente.

Mecánicas de etapa:

1. El jugador empieza en una posición del tablero
2. Hay casillas objetivos que deben ser eliminadas
3. Ataca la casilla final
4. Avanza hasta la siguiente etapa

Figura 32. *Movimiento de sobre el tablero*

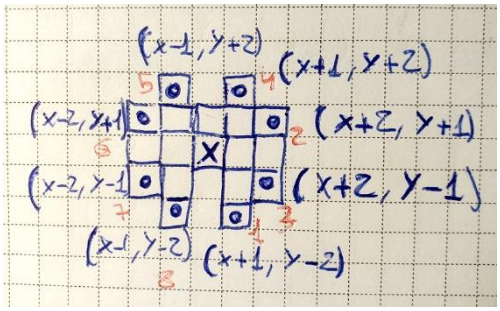
Mecánicas de movimiento:

6. El jugador mueve el caballo
7. Recoge los recursos de la casilla destino
8. Interactúa con la casilla si es necesario y puede modelar su ejercito
    a. Recoge los recursos de dicha casilla
    b. Provoca una batalla entre las fuerzas hunas y la resistencia romana.
9. Modifica las propiedades del ejército en función de la interacción
10. La casilla de origen se destruye.

Consideraciones:

1. Si la casilla destino es un objetivo la descontamos.
2. Si la casilla es final pero aún quedan casillas objetivo por destruir no podemos avanzar.
3. Si no quedan casillas objetivo y estamos en una casilla final… ejecutamos el evento y si es superado terminados el nivel.

Figura 33. *Posibles movimientos del jugador*



Estas dinámicas principales en el núcleo del juego intentan potenciar aquellos aspectos que aumentan la jugabilidad a través de dinámicas de núcleo como y mantienen al jugador conectado con el juego.

- Adquisición territorial (la sensación subjetiva de sentirse poderoso)
- Predicción (intentar predecir el resultado de las acciones)
- Razonamiento espacial (interpretar los espacios para ubicar los elementos)
- Supervivencia (el jugador juega parar seguir vivo)
- Destrucción (el jugador se reafirma al destruir cosas)
- Comercio e intercambio (potenciamos la estrategia y la negociación)

Estas condiciones de movimiento **son el principal motor motivacional** del juego, donde debemos combinar los movimientos estratégicos para la consecución de los objetivos y las casillas finales.

Cuando el caballo accede a una casilla sin eventos asociados podemos, explícitamente, ver los recursos obtenidos por el movimiento a dicho lugar. Pero si el movimiento va destinado a una casilla que genera un evento asociado puede desencadenarse una batalla. **A los jugadores les gustan las batallas**. En primer lugar deberemos ajustar los parámetros de la batalla (fuerzas, formación, recursos asignados) manual o automáticamente y después añadiendo un factor de azar veremos el resultado de la misma.

Figura 34. *Resultado de una batalla*

Simultáneamente podemos ver bajo los recursos los movimientos restantes que nos quedan antes de que los recursos actuales se agoten y no podamos continuar. Esta relación de equilibro entre ganancias (movernos por diferentes casillas) y costes (mantener al ejército) es el **tercer motor** del juego. Comerciar, intercambiar y reasignar recursos puede ser un valor estratégico importante para determinar si conseguimos pasar el nivel y seguir adelante.

Figura 35. *Obtención de recursos*

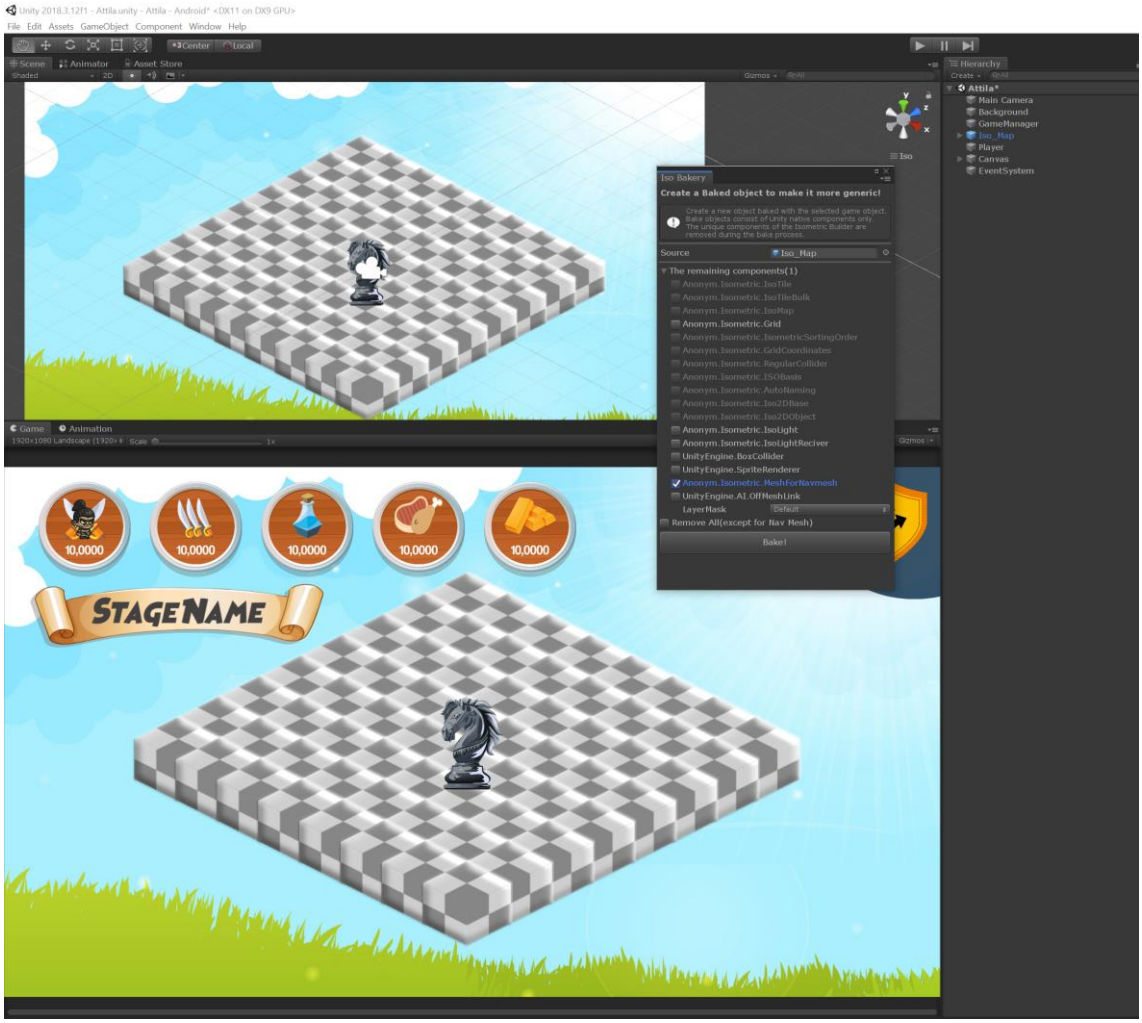
Si finalmente podemos superar todas las etapas del juego habremos conquistado Roma, cosa que Atila, rey de los hunos nunca pudo hacer.

Figura 36. *Atila conquista Roma*

Tablero de juego en entornos isométricos

El primer paso fue diseñar el tablero máximo para distribuir los elementos de juego (tablero HUD, controles) en el espacio. En este caso se ha utilizado un gráfico isométrico genérico para implementar las casillas vacías.

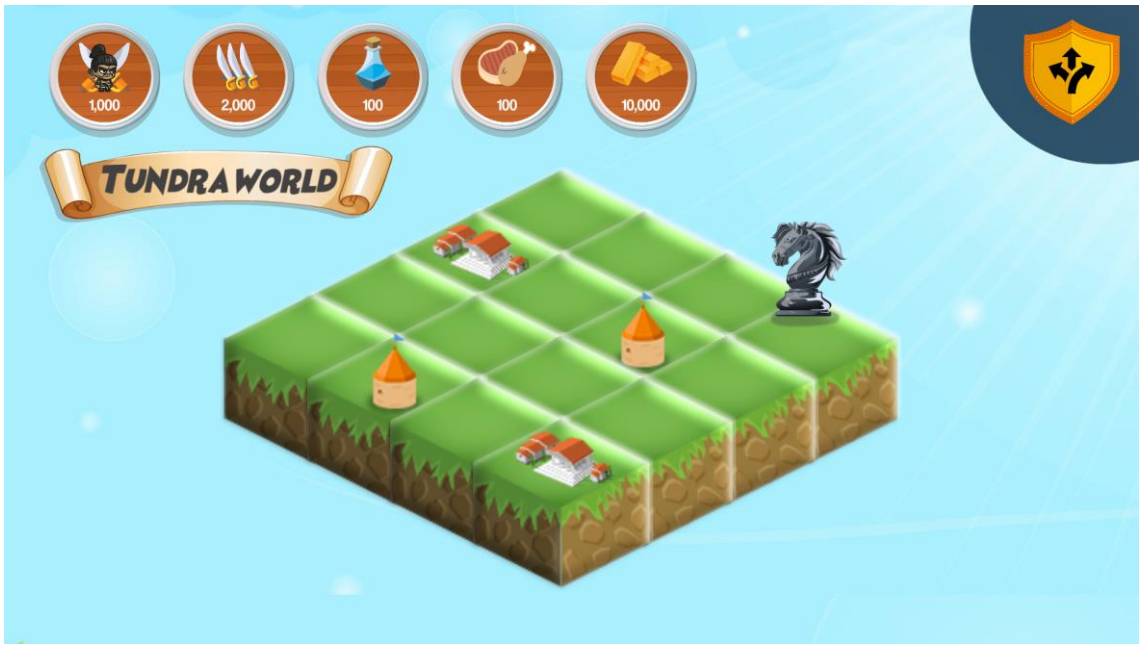
Figura 37. *Diseño del tablero isométrico*



En concreto se ha utilizado **Isometric Builder** para para implementar las escenas isométricas ya que permite, más adelante, tratar cada casilla como un objetos independiente y en este caso añadir clases para cada casilla independiente.

Una vez creado el tablero máximo, dinámicamente el juego substituye las casillas vacías por las casillas correspondientes generando la escena correspondiente a la etapa actual del juego.

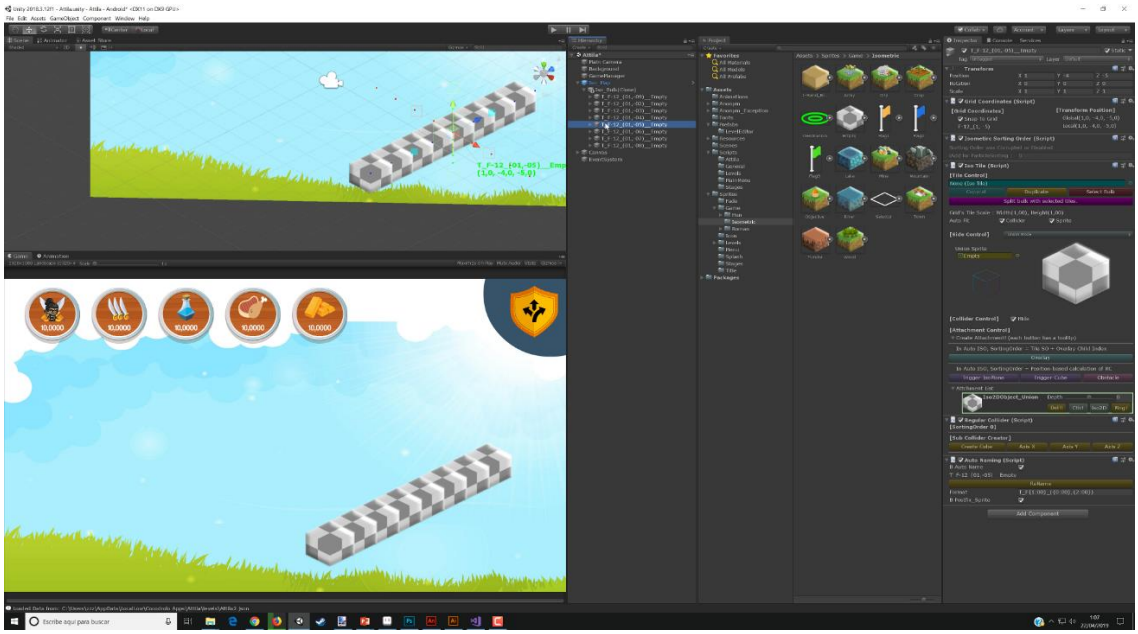
Figura 38. *Creación dinámica del tablero de juego*



Además se ha podido implementar un *Navmesh* para que el movimiento pueda realizarse automáticamente utilizando las funciones internas de Unity. Esta implementación justifica la implementación casilla a casilla de clases asociadas para cada elemento y así controlar los posibles movimientos y eventos asociados.

Este sistema, además permite asociar a cada casilla los elementos sobreexpuestos para cada casilla como marcadores de destino, bandera de objetivo, casilla final, efectos, etc.

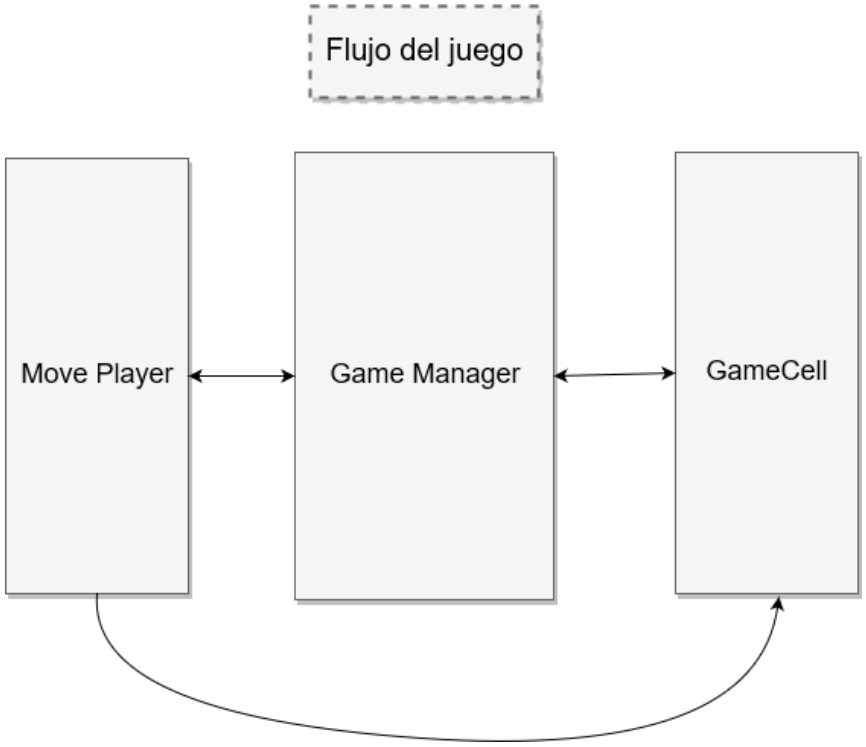
Figura 39. *Asociación de clases, sprites, selector y banderas.*



Una vez creadas las estructuras necesarias el flujo del juego pivota entre 3 clases del proyecto de Unity. Por un lado, el ***GameManager*** centraliza la secuencia del juego, el control del jugador sobre la pantalla al hacer *click* en una casilla disponible. La clase es la encargada de llamar a funciones de la clase ***MovePlayer*** para gestionar el movimiento del caballo, calcular las nuevas casillas disponibles, que así mismo llamará a las funciones de la clase ***GameCell*** para destruir la casilla de origen una vez llagado al destino, etc. Una vez procesado el movimiento es necesario procesar los eventos que se derivan de la casilla destino interaccionando con el caballo.

Básicamente se pueden producir 3 tipos de eventos en función de la casilla de destino. ***GameManager*** puede ejecutar la secuencia de batalla, deducir que el jugador se ha quedado bloqueado sin más casillas disponibles o que el ejército no tiene suficientes recursos para seguir adelante.

Figura 40. *Flujo del juego a través de las clases*




```
void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            Ray screenRay = Camera.main.ScreenPointToRay(Input.mousePosition);
            RaycastHit hit;
            if (Physics.Raycast(screenRay, out hit))
            {
                if (hit.collider != null)
                {
                    if (hit.collider.gameObject.name == "Regualr_Collider_Union"
                        && GlobalInfo.isPlaying == true
                        && GlobalInfo.isPlayerMoving == false
                        && GlobalInfo.isShowingInfo == false)
                    {
```

```
                        if
(DestionationAvaliable(GameObject.Find(hit.collider.gameObject.transform.parent.n
ame)))
                        {
                            MoveHorse("Cell" + GlobalInfo.playerPos.ToString(),
hit.collider.gameObject.transform.parent.name);
                        }
                    }
                }
            }
        }
    }
```

El objetivo de la función es detectar la pulsación sobre una casilla del tablero y a través de la función *MoveHorse()* actuar en consecuencia.

Es evidente que la asincronía de la pulsación sobre el tablero obliga a controlar los diferentes estados del juego para que el movimiento sólo sea posible cuando tengamos casillas disponibles, los eventos anteriores estén terminados y no se esté en medio de un movimiento.

```
public void Move(string mCell)
    {
        GameObject dest = GameObject.Find(mCell);
        if (dest != null)
        {
            //Don't move at the same position
            if ("Cell" + GlobalInfo.playerPos.ToString() != mCell)
            {
                NavMeshAgent agent = GetComponent<NavMeshAgent>();
                agent.destination = dest.transform.position;
                GlobalInfo.isPlayerMoving = true;
                GlobalInfo.isOldCellDestroyed = false;
                destination = mCell;
                dest.GetComponent<GameCell>().SetMoveables();
                GameObject origin = GameObject.Find("Cell" +
GlobalInfo.playerPos.ToString());
                origin.GetComponent<GameCell>().HideMoveables();
                StartCoroutine(DestroyCell(origin));
            }
        }
    }
```

Condiciones de victoria y derrota

Cualquiera de las **3 dinámicas motivacionales** puede llevar a la derrota al ejército de Atila, y sólo jugando bien en las tres obtendremos la victoria para seguir adelante.

Sólo ganamos una etapa si:

1. No hay casillas objetivos
2. Atacar la casilla final y ganar la batalla

Derrota por movimiento: Si nos quedamos sin movimientos disponibles y no podemos seguir hacia ningún lugar y habremos perdido la partida y se nos indicará que debemos empezar de nuevo el nivel.

Figura 41. *Derrota por movimiento*

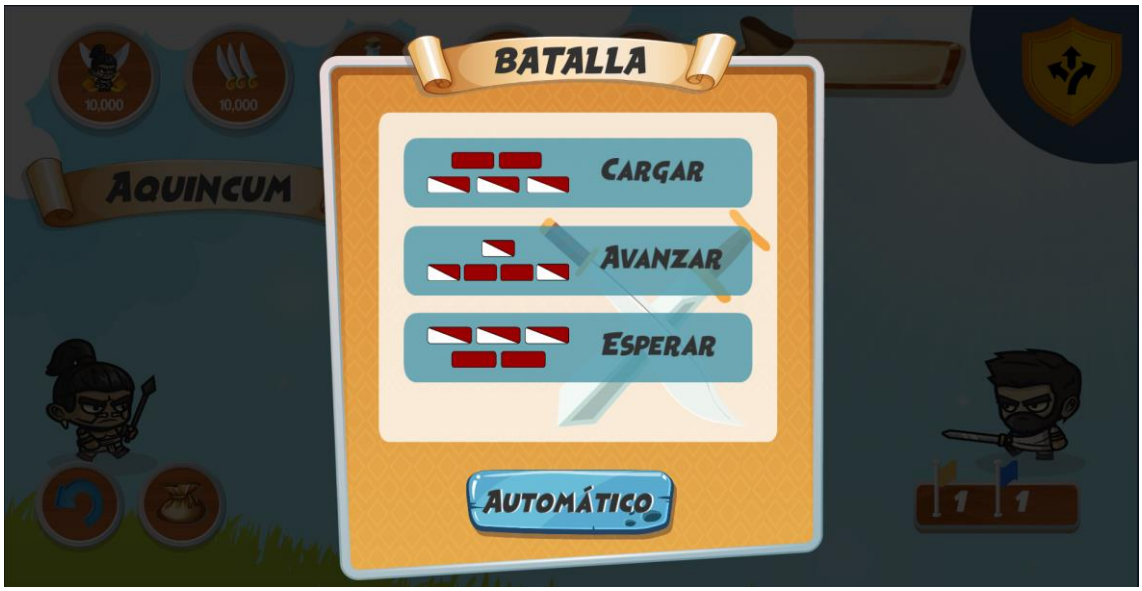
Derrota en el campo de batalla: Si tenemos menos tropas que los romanos, nuestra disposición de combate es superada o el azar nos presenta unas malas cartas perdernos la batalla y deberemos volver a empezar el nivel.

El sistema compensatorio para minimizar el factor de azar consiste en la capacidad de decidir nuestra disipación de batalla o antes de la batalla invertir nuestro preciado oro en potenciar nuestras fuerzas de ataque.

El resultado de la batalla dependerá de factores matemáticos, disposiciones del jugador y el azar, con lo que nunca habrá dos partidas iguales ya que el resultado de una batalla puede afectar a otros aspectos del juego.

Figura 42. *Derrota en el campo de batalla*

Figura 43. *Disposición de las tropas antes del ataque*

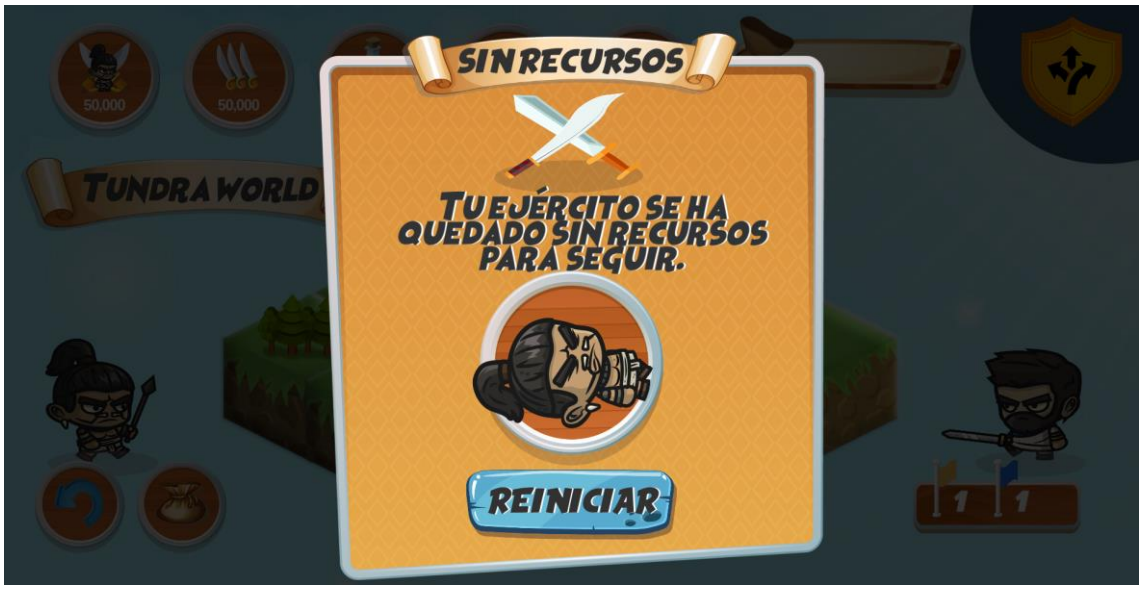


En resumen, una batalla depende de:

1. Disposición de las tropas
2. Número de tropas de los dos bandos
3. El azar

Derrota por falta de recursos: Si nos quedamos sin recursos de algún tipo y no podemos dar de beber o comer a nuestros soldados nos abandonaran masivamente y deberemos empezar otra vez el nivel. Pasará lo mismo si no podemos pagar a nuestras tropas y mercenarios salario en oro.

Figura 44. *Derrota por falta de recursos*



Esto significa que se debe estar pendiente de los recursos del ejército. Para simplificar los cálculos y aumentar la usabilidad se ha dispuesto un sistema de caculo para que el jugador sepa el número de movimientos que puede realizar en función del número de tropas actuales (circulo azul debajo del recurso). Si el número de tropas aumenta o disminuye, los recursos necesarios para mantener el ejército también varían, y el número de saltos cambiará dinámicamente.

Para minimizar este aspecto en el juego y no penalizar en exceso un error estratégico que puede afectar a etapas ya realizadas, al finalizar una etapa el jugador puede acceder al mercado de mercenarios y mercaderes locales y utilizar parte de su oro comprando recursos para seguir avanzando. Así mismo, si el jugador gasta demasiado no podrá pagar a sus tropas y se perjudicado en el futuro.

Figura 45. *Cálculo de saltos disponibles*

Usabilidad

La usabilidad es el primer paso para ofrecer una excelente experiencia de usuario porque intenta comprender cuando realizar tareas simples en un juego puede crear importantes obstáculos que pueden, en casos extremos, llevar a los jugadores a dejar de jugar el juego. En el mejor de los casos, un juego con mala usabilidad se experimentará como molestias o puede impedir que los jugadores descubran o disfruten de algunas características y en el peor de los casos, un excelente juego puede volverse totalmente imposible de jugar. Para asegurar que un juego tenga una buena usabilidad se trata de eliminar las frustraciones innecesarias e indeseadas en aquello que el jugador percibe.

En este sentido se ha trabajado la consistencia, la proximidad entre el sistema y el mundo real, la prevención de errores y el reconocimiento en lugar de la memorización.

Figura 46. *Tutorial (1)*

En el caso de juegos gratuitos, como es el caso, la facilidad de uso es aún más crítica porque los problemas podrían afectar el compromiso inicial o retención porque el compromiso de los jugadores para jugar es bajo cuando no tienen que aportar ningún dinero por adelantado. Pequeñas frustraciones podrían impedirles ir más allá de los primeros minutos de tu experiencia de juego. Esto también podría impactar dramáticamente sus ingresos si los jugadores, por ejemplo, al no entender por qué o cómo comprar un artículo en la tienda virtual del juego.

Figura 47. *Tutorial (2)*

La parte más crítica del proceso es el diseño del tutorial de juego. Sin duda que el juego esté traducido a 6 idiomas permite que las instrucciones de juego puedan ser comprendidas en un lenguaje cercano al jugador, mejorando la usabilidad del proceso.

Además el tutorial consiste en un conjunto de *triggers* que se activa en el momento que por primera vez debemos conocer una información relevante y necesaria, minimizando el proceso de memorización y mejorando el aprendizaje automatismos.

Tanto como ha sido posible se ha seguido el **principio de percepción de Gestalt**. Este principio se ha utilizado para explicar cómo la mente humana percibe y organiza el medio ambiente. Por ejemplo, de acuerdo con la ley de proximidad de la Gestalt, los elementos que están cerca entre sí se interpretarán como pertenecientes al mismo grupo. La aplicación de los principios Gestalt puede ayudar a los diseñadores a organizar la interfaz del juego de manera que los jugadores entiendan correctamente. Hay que asegurarse de que toda la iconografía que utiliza en el juego no sea confusa.

Figura 48. *Usabilidad sobre la pantalla de juego*

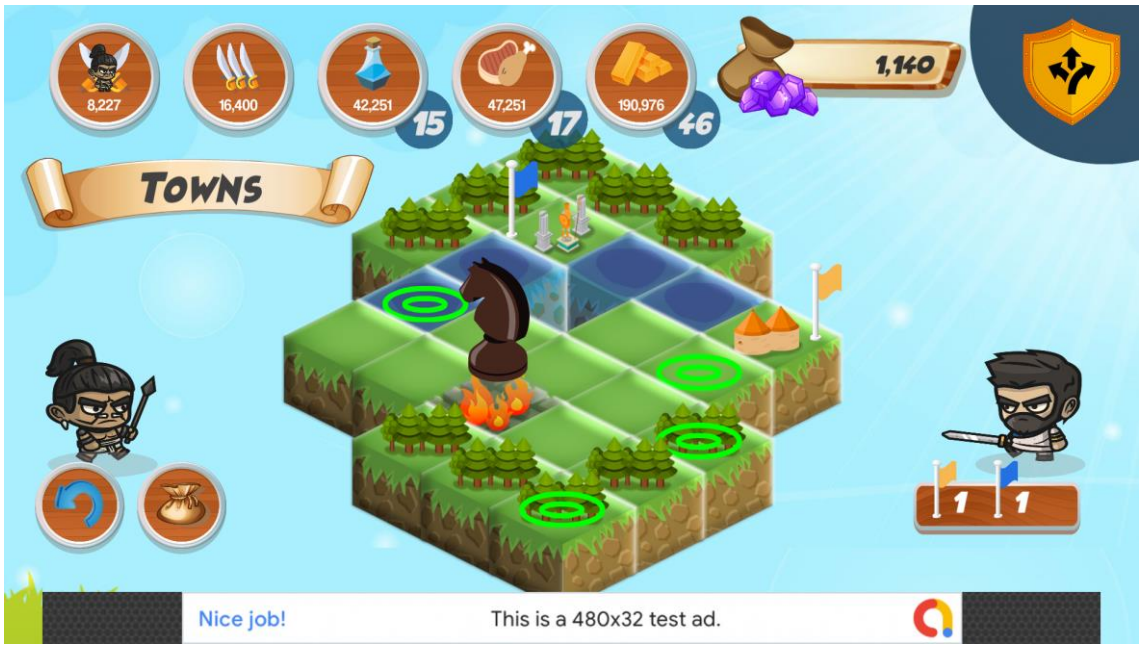
Interfaz de usuario

El objetivo de una interfaz es hacer que los jugadores sientan el control de la experiencia de juego, además puede ser bonita y fluida, pero la sensación de usuario es lo primordial. En el diseño de juegos, la interfaz no es simple ni fácil de describir. Interfaz puede significar muchas cosas: un controlador de juego, un dispositivo de visualización, un sistema de manipulación de un personaje virtual, la forma en que el juego comunica información al jugador y muchas otras cosas.

En este contexto la interfaz debe posibilitar que el jugador se comunique con el juego para conseguir sus objetivos. Cuando la habilidad para jugar es demasiado baja y el objetivo demasiado difícil, el jugador se vuelve ansioso. Alternativamente, si el objetivo es demasiado fácil y habilidad demasiada alta, el jugador se aburre. Sin embargo, cuando la habilidad y la dificultad son aproximadamente proporcionales, las personas entran en estados de flujo.

En este caso la experiencia de usuario está estrechamente relacionada con el diseño de los niveles. La Interfaz está posicionada sobre una pantalla horizontal (teléfono inteligente o tableta) con los controles distribuidos a derecha e izquierda para facilitar el control de la partida. El control de movimiento se sitúa en el centro de la pantalla simbolizada por los círculos concéntricos que indican los saltos disponibles en cada momento.

Figura 49. *La interfaz de usuario*



Todo ello ha estado reforzado con elementos multimedia para ambientar, informar y subrayar aspectos determinados del juego para centrar en cada momento al jugador en las cosas que son realmente importantes en cada momento.

Además es de obligada atención remarcar que todo ello (gráficos, interfaz, multimedia) tienen en cuenta que es un juego para dispositivos móviles con lo que es imperativo simplificar algunos procesos y tareas para centrar al jugador en lo que es realmente importante.

Al fin y al cabo el objetivo de estas decisiones es conseguir que el jugador pueda:

- Extremar la atención en una tarea.
- Un sentido de control activo.
- Fusión de la acción y la conciencia.
- La pérdida de la conciencia de sí mismo.
- La distorsión de la experiencia del tiempo.
- La experiencia de la tarea de ser la única justificación necesaria para continuarlo.

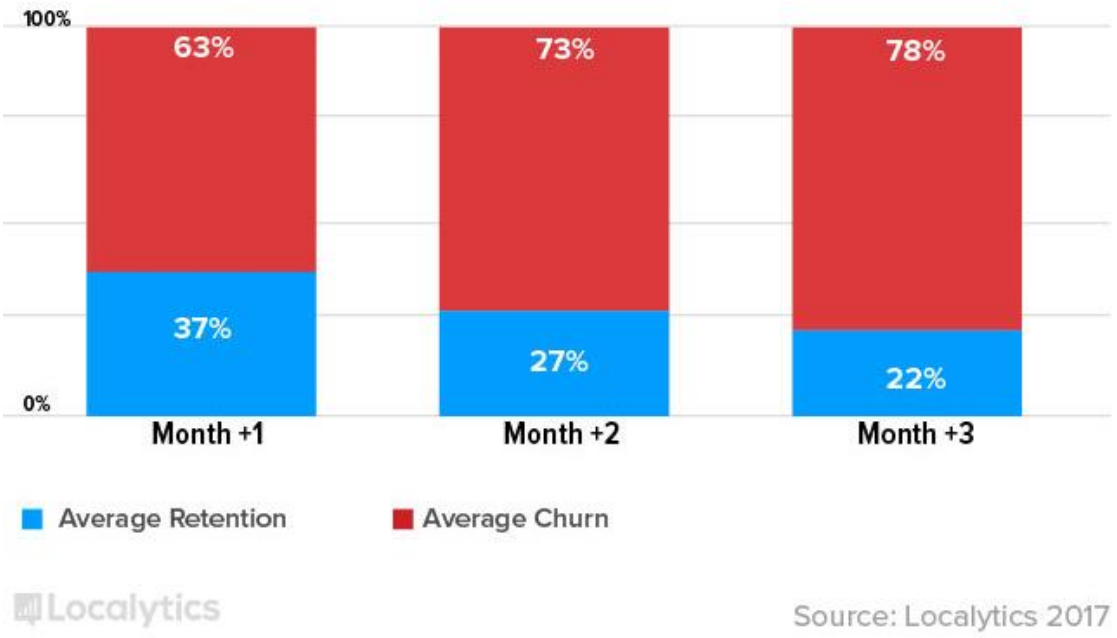
4.6 El juego como elemento empresarial

El juego diseñado, y estrechamente ligado a los aspectos de la usabilidad y la interfaz, está pensado como un producto de consumo. Este hecho influye en aspectos del diseño nucleares de la estructura lúdica de la aplicación.

Según la **empresa de marketing online Localytics** la tasa de abandono de una aplicación móvil cae un 63% durante el primer mes. Localytics ha comprobado que la tasa de retención promedio es de un 37% al mes después de la instalación. Ésta caería al 27% al segundo mes y se precipitaría al 22% al tercero. Asimismo, se puede ver que la tasa de cancelación o abandono es de un 78% a los noventa días de la descarga.

Ya en 2013, un informe de la empresa **TechCrunch** demostraba que el 16% de los usuarios descargaban la aplicación y nunca más la utilizaban por segunda vez. Y de allí hasta este entonces, un patrón de conducta se mantiene inerte: en esencia, a la gente le gusta ver lo que la marca tiene para ofrecer y saber si le interesa.

Figura 50. *Retención del usuario de aplicaciones móviles*



Localytics se ha centrado también en lo que ocurre en las dos primeras semanas tras bajar una aplicación. El primer día un 30,7% vuelve a usar la app, pero este porcentaje se reduce drásticamente a medida que transcurren las jornadas. El segundo día hay un bajón importante: solo un 22,8% abre la aplicación. La trayectoria descendente continúa hasta el día 14, cuando solo un 11,7% de los usuarios utiliza la herramienta.

La consultora calcula que el promedio de veces que una app se abre por semana son 6,93, con los usuarios dedicando 4,58 minutos en cada sesión y 31,7 minutos semanalmente. Al mes cada app se usaría de media 15,89 ocasiones, con 4,63 minutos por sesión para un total mensual de 73,57 minutos.

Estos datos ya nos hacen prever que una campaña post instalación es imprescindible para rentabilizar el esfuerzo empleado para el desarrollo del juego. Pero lo que muchos diseñadores no tienen en cuenta es que este esfuerzo post instalación empieza en el

propio diseño de la juego para móviles e incluso puede ser crucial para la rentabilización del coste de producción.

Las opciones más comunes que se tienen al alcance de la mano serian publicar la aplicación en alguna de las tiendas virtuales más importantes que hay en el mercado. Hay que tener en cuenta que la licencia para publicar en Google Play es de 25$ USD (una sola vez) y la licencia para Apple Store es de 100$ USD anuales. Publicar el juego no es garantía de éxito ni de recuperación de la inversión ni de obtener cuantiosos beneficios.

Teniendo en cuenta que los usuarios descubren 7 de cada 10 apps en los propios buscadores de las tiendas de aplicaciones hay que tener clara que estrategia se quiere seguir parar posicionar bien la aplicación. De esto se encarga ASO (*App Store Optimization*) y SEO (*Search Engine Optimization*) para apps.

Esta realidad, de un modo u otro, afectan al diseño del juego y alteran la estructura lúdica en juegos para móviles. Especialmente se ha diseñado un conjunto de acciones y recompensas que además de potenciar aspectos psicológicos en el jugador, también permiten monetizar la aplicación durante el tiempo que el jugador la tendrá instalada en su dispositivo.

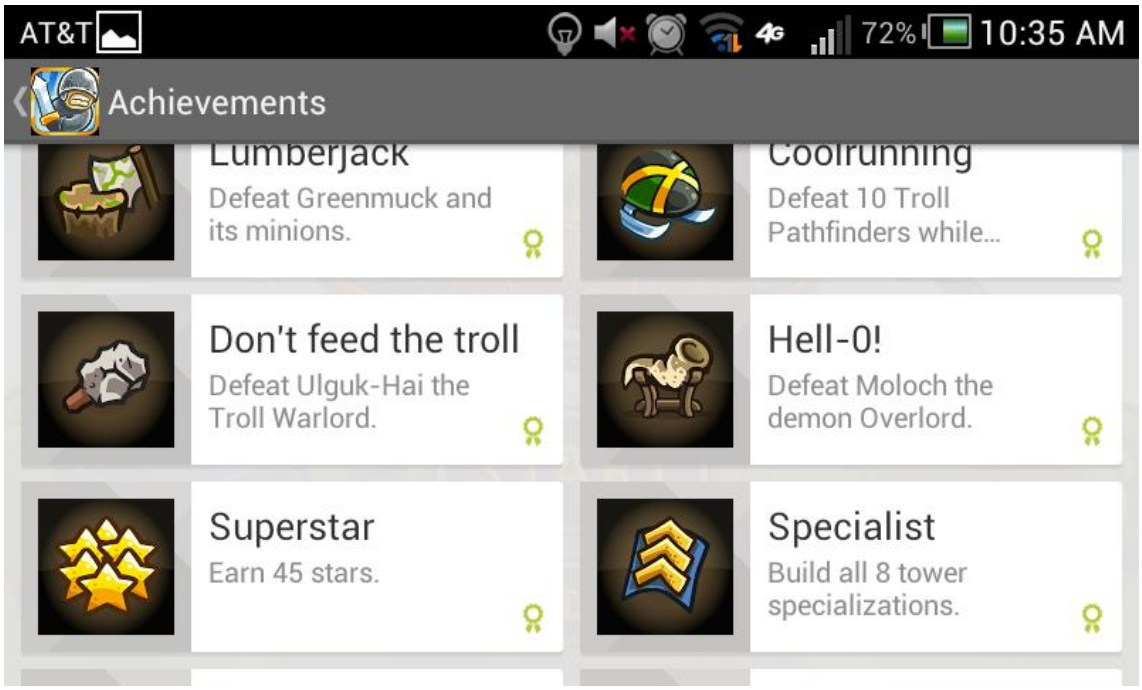
Figura 51. *Recompensas virtuales a cambio de monetizar el juego*

En la primera opción de la pantalla de comercio, podemos obtener 50.000 monedas de oro a cambio de ver un anuncio publicitario.

Por un lado el juego tiene reservados espacios de la interfaz de juego para la inserción publicitaria. Anuncios Intersticial entre algunas pantallas cada *n* veces que ejecutamos el juego. Además podemos aumentar las ganancias poniendo recompensas por la visión de anuncios publicitarios que afectan a la dinámica del juego, pero incentiva al jugador a mantener sus recursos virtuales y a la vez monetiza la aplicación.

Además como sistema de recompensas se puede utilizar el propio sistema de Google Play para gestionar las puntuaciones globales (potenciando la competición entre jugadores) y los logros (reforzando la búsqueda de la superación personal).

Figura 52. *Sistema de logros de Google Play Games*



5. Diseño de niveles

Al ser un juego de estrategia con múltiples niveles que necesitan ser ajustados minuciosamente se requería el diseño y programación de un editor de niveles. El problema principal es que el sistema de archivos de en Android es diferente al sistema de archivos de un ordenador de sobremesa. Para la inclusión de los niveles del juego se ha utilizado una carpeta en *Resouces* para garantizar que al ser distribuida la aplicación la estructura de ficheros se mantiene.

Por otro lado, la principal ventaja de construir una escena independiente para ordenador de sobremesa como editor de niveles es que permite diferenciar entre los niveles en fase de diseño y los que tenemos en producción.

5.1 Editor de niveles

El editor debía ser práctico y funcional a la vez que debía trasladar a una visión editable lo que posteriormente seria el tablero de juego isométrico. En este caso se han implementado las funciones básicas de edición (creación, edición y guardado) para a través de una cuadrícula de 64 casillas representar el tablero.

Figura 53. *Editor de niveles*

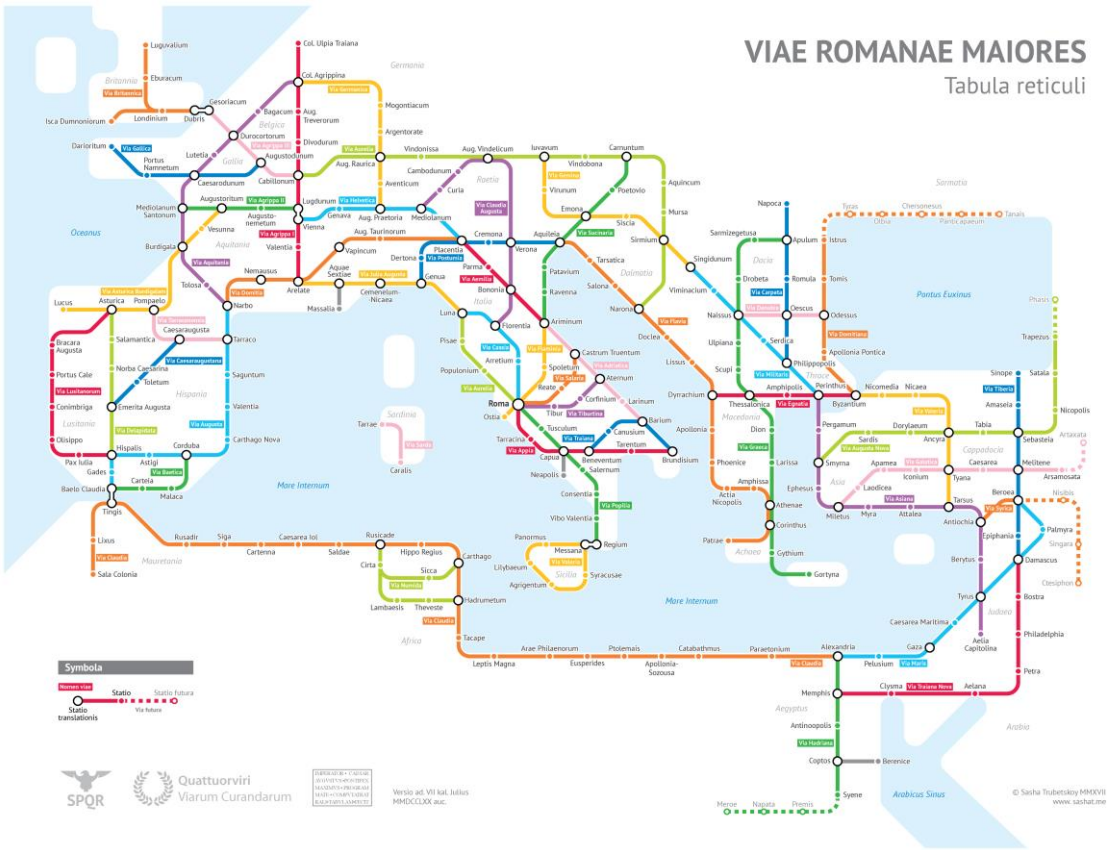
El editor, en sí mismo, funciona como una aplicación dentro de la aplicación. No tiene ninguna dependencia con ninguna otra escena del juego ni existe vinculación con el proyecto parar Android aunque la escena pertenezca al mismo entorno de trabajo de la aplicación. Además está pensada para ser ejecutada en un equipo de sobremesa ya que utiliza funciones que en sistemas que el acceso a archivos no son compatibles en el entorno del juego. Así mismo este sistema nos da la ventaja de trabajar siempre con una copia de los archivos de niveles, para después de su aprobación puedan ser copiados a la carpeta de recursos y formar parte de la versión de producción del juego.

Con el editor se puede externalizar el trabajo de crear y editar los niveles del juego e ir calibrando la dificultad para que el flujo de juego sea correcto.

5.2 Contexto histórico

Otro aspecto importante para crear los niveles es el contexto histórico del juego. En cierto modo el juego es una recreación de la las conquistas de Atila, rey de los hunos.

Figura 54. *Principales vías romanas*



En este aspecto el juego ha intentado ser lo más fiel a los hechos históricos. Atila realizó 3 campañas sobre el imperio romano y en este caso el juego también se compone de 3 campañas de 10 niveles cada una. La primera campaña es la que corresponde a la incursión sobre el imperio romano de oriente, terminando en Constantinopla. La segunda campaña se centra en la conquista de la Galia hasta la batalla de los campos Cataláunicos. Y en tercer lugar tenemos la campaña de la invasión de la península itálica donde fue detenido sobre el rio Po por el papa León I, pero en nuestro caso, podemos llegar a la misma Roma, donde Atila tenia fijado su objetivo.

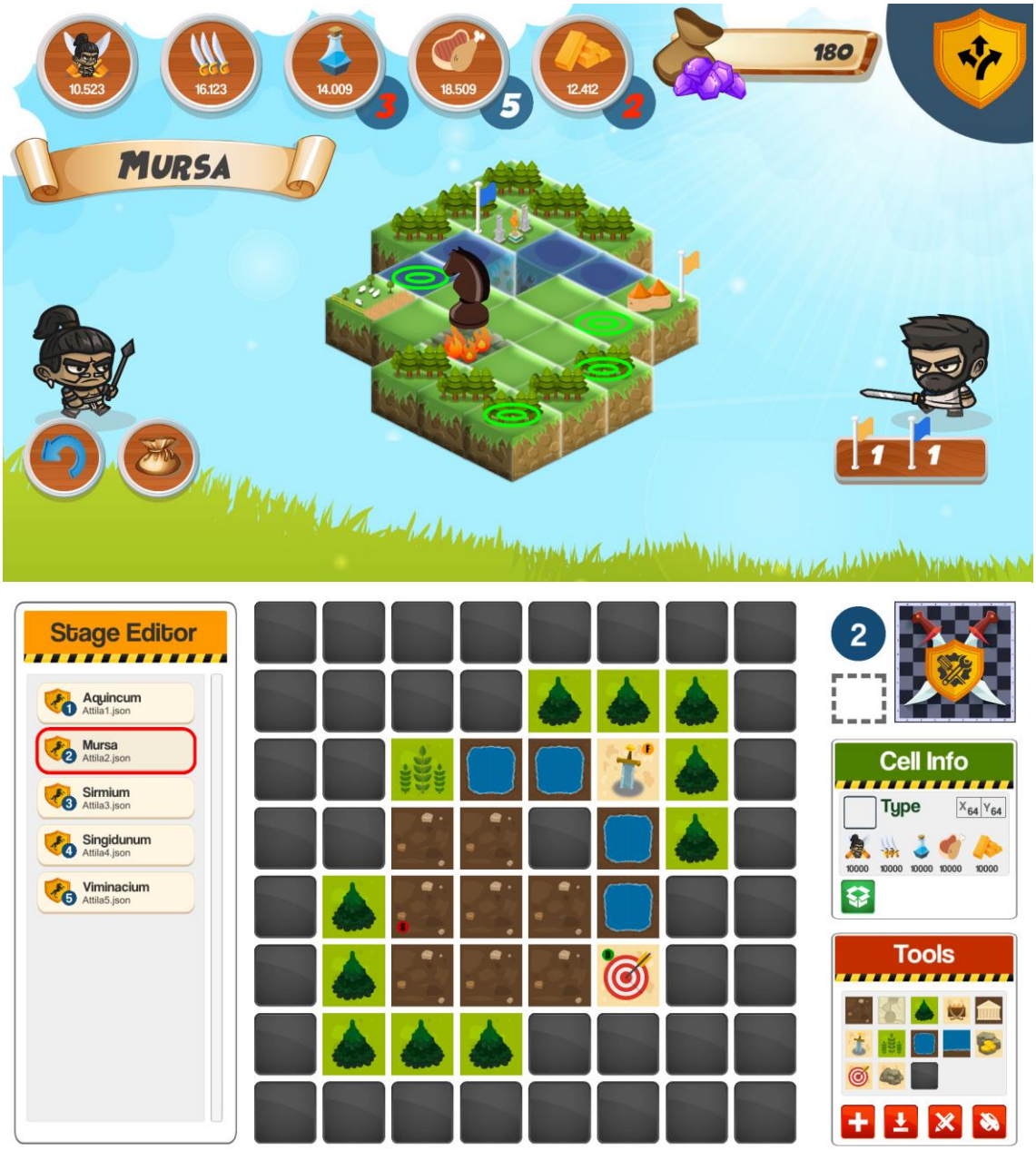
5.3 Análisis de un nivel

Cada nivel tiene el nombre de una ciudad romana que sigue el camino trazado por Atila en sus conquistas. El aspecto de cada nivel intenta recrear las condiciones orográficas de la zona, con evidentes licencias creativas. Así miso cada nivel representa un reto pensado para establecer una relación entre los recursos estimados que puede tener el ejercito de Atila, el recorrido mínimo para llegar a los objetivos y a la casilla final y el desgaste (consumo) de recursos que conllevara este recorrido y las fuerzas opositoras que desembocaran en una batalla entre el ejército del jugador y las defensas romanas.

1. Recursos disponibles
2. Recorrido sobre el tablero
3. Fuerzas de resistencia

Véase un ejemplo completo. En la siguiente figura pueden ver el nivel 2 del juego, que representa la ciudad romana de Mursa.

Figura 55. *Nivel 2 del juego: Mursa*



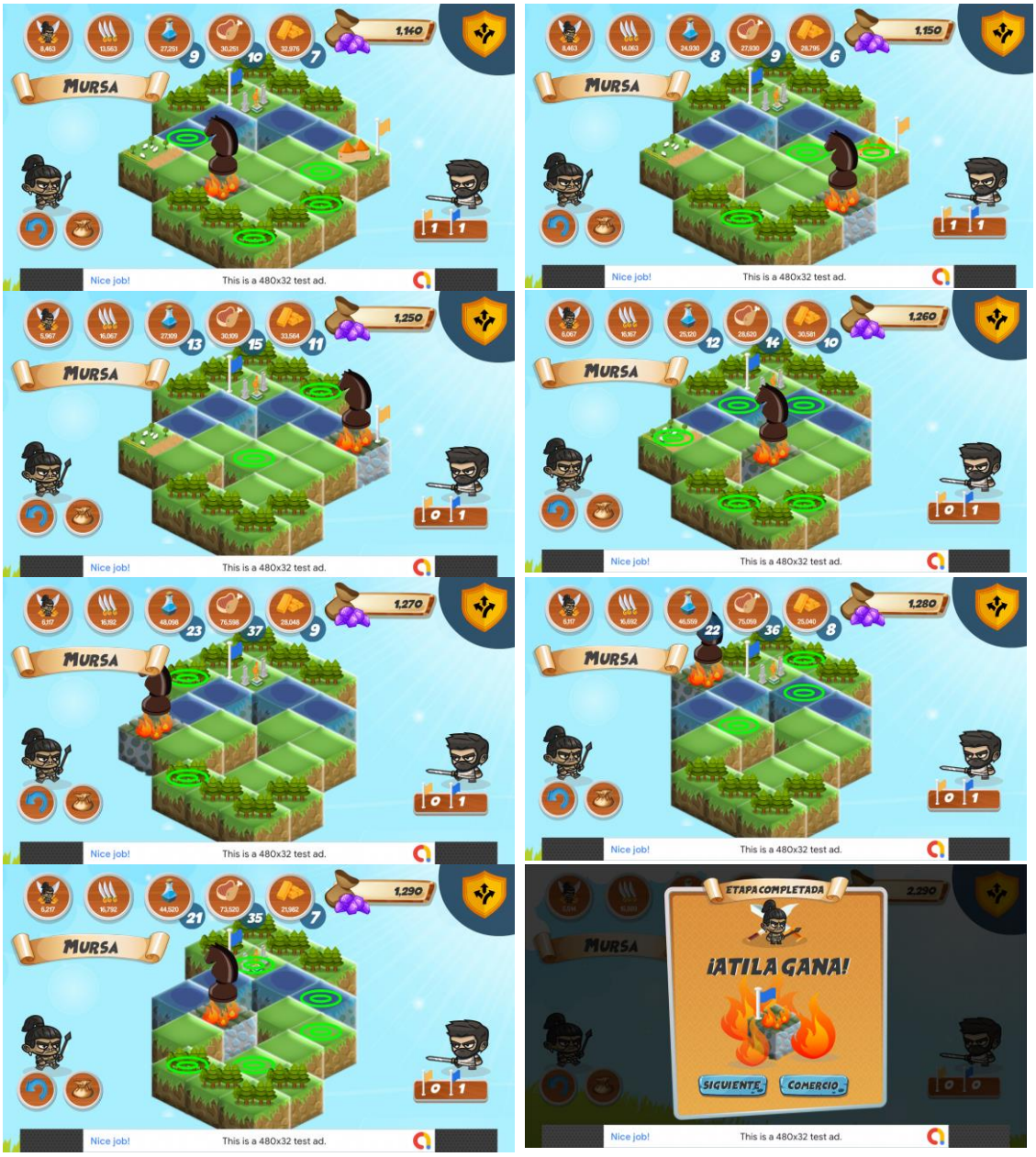
Podemos ver la representación del mapa en la versión isométrica del juego y la cenital del editor de niveles.

Precondiciones:
Este es el segundo nivel del juego. Como puede verse en la figura 54. Tenemos problemas de recursos (agua y oro) que nos limitan la capacidad de movimiento. Este valor es aproximado ya que depende de los sucesos del nivel 1, así mismo, está claro

que como parte de la estrategia común en este nivel es necesario reforzar aspectos de las propiedades del ejército para no quedarse sin recursos.

Recorrido:
Para llegar desde la posición inicial, hasta el primer objetivo y posteriormente hasta la casilla final se puede realizar en 8 pasos. Evidentemente podemos realizar el recorrido en más pasos, con el riesgo de quedarnos atrapados, pero con la confianza de recoger recursos necesarios para seguir avanzando.

Figura 56. *Resolución del nivel 2*



Así mismo es muy recomendable que al terminar el nivel invirtamos una parte de nuestro oro en reforzar los suministros de las tropas ya que en futuras campañas nos podemos ver obligados a terminar el juego por falta de recursos.

Figura 57. *Compra de suministros al final del nivel*

6. Manual de usuario

El juego ha sido en varios dispositivos móviles sin problemas. Con el nuevo sistema de empaquetado de Google Play el archivo del juego ocupa entre 25 y 27 Mb en función de la arquitectura del dispositivo. La interfaz ha sido ajustada para teléfonos y tabletas sin variaciones importantes en función de la resolución o aspecto de la pantalla.

Por el propio sistema de compilación de Unity con mis NDS y SDK instalados la versión mínima es el SO Android 4.1.

Figura 58. *Requisitos en Google Play*

Tipo	Código de versión	Tamaño de descarga de la aplicación	
Android App Bundle	2	Entre 27,4 y 28,9 MB	ELIMINAR

EXPLORAR APP BUNDLE

Archivos APK de configuración	Generados en: 3 plataformas nativas
Dispositivos Android compatibles	11723 dispositivos
Niveles de API	16+
SDK objetivo	28
Diseños de pantalla	4 diseños de pantalla: small, normal, large, xlarge
Funciones	1 función: android.hardware.screen.landscape
Permisos necesarios	3 permisos: android.permission.ACCESS_NETWORK_STATE, android.permission.INTERNET, android.permission.WAKE_LOCK
Versiones de OpenGL ES	2.0+
texturas OpenGL	todas las texturas
Plataformas nativas	todas las plataformas nativas

6.1 Instrucciones del juego

El juego consiste en una recreación estratégica de las campañas de Atila, rey de los hunos. Debemos superar 30 niveles hasta llegar a la ciudad de Roma y conquistarla. Para mover nuestro ejército usaremos un caballo de ajedrez.

Figura 59. *Como jugar a Atila, Tierra quemada*

En estas imágenes podemos ver las reglas auto explicadas del juego para entender las dinámicas básicas.

6.2 Enlaces

GitHub: https://github.com/CocodroloApps/attila/tree/UOC/Attila
Ejecutable: https://github.com/CocodroloApps/attila/tree/UOC/Attila/Android
Formato: Android AAB
Unity Connect: https://connect.unity.com/p/attila-tierra-quemada

Video general del proyecto (13 min): https://youtu.be/hU1w_fGyYjI
Mecánica del juego (20 sec): https://youtu.be/G2eiq231ge8
Video explicativo Editor de niveles (6 min): https://youtu.be/1bmo_osoraU
Trailer (41 sec): https://youtu.be/4rfkGO6v22w

7. Conclusiones

Durante estos meses de intenso trabajo es importante hacer notar lo importante que es la planificación y lo costoso que es hacer una planificación profesional formando equipo consigo mismo. Posiblemente este ha sido el mayor reto de todo el trabajo.

El trabajo de estos meses ha sido apasionante y nada mejor que la experiencia real para acelerar el proceso de aprendizaje. Estoy satisfecho con el trabajo realizado y la calidad del producto obtenido.

Así mismo durante la fase de pruebas he podido experimentar lo complejo que es rectificar o modificar decisiones de diseño que en la fase de testeo has recibido como respuesta sugerencias que implican hacer cambios en alguna mecánica del juego.

Muchas veces la carga de trabajo va en detrimento de la calidad, y justamente la calidad era uno de los objetivos principales del TFM. También ha sido interesante descubrir los costos reales de hacer algunos detalles y de cómo una formación interdisciplinar es básica para crear un producto de calidad y darse cuenta, a veces, a veces el coste visto a través de los ojos del usuario son insignificantes, pero importantes a la vez para la inmersión en la experiencia de juego.

Otro aspecto importante es el reto en probar cosas nuevas. Podría haber diseñado un juego dentro del espacio trabajado durante estos dos años, pero para mí era un reto personal salir de mi zona de confort y hacer, de principio a fin, un producto cerrado, completo y publicable en tecnologías que hasta hace poco eran desconocidas o poco exploradas para mí.

Personalmente me ha apasionado el potencial que tienen los entornos isométricos en juegos para plataformas móviles, ya que estos dispositivos al no disponer de una gran potencia gráfica podemos recrear juegos 3D en entornos 2D. Ha sido un reto interesante.

Después de este Master me gustaría empezar una nueva vida profesional en este sector y más allá de lo aprendido creo que es importante poner en valor el aprendizaje más íntimo, el aprender a aprender y ser capaz de explorar espacios nuevos que por la complejidad y amplitud del mundo de los videojuegos no siempre se pueden trabajar al 100% en un entorno académico como este.

Como en todo proceso importante me han surgido dudas y preguntas al avanzar en el TFM que me generan nuevas inquietudes para nuevos y futuros proyectos con todo lo aprendido y las nuevas puertas que han aparecido en el viaje mientras desarrollaba este Atila, Tierra quemada. Espero que igual que yo lo saboreen y lleguen a la única conclusión valida... Que ha sido divertido.

8. Glosario

ASO (App Store Optimization) – Conjunto de estrategias para posicionar una aplicación en una tienda de apps móviles.
Asset – Conjunto de herramientas, recursos o funciones que se pueden añadir o usar en el entorno de trabajo para complementar una función o aspecto del juego.
Click and go – Mecánica que consiste en pulsar un punto de la pantalla para direccionar un elemento del juego.
Gráficos isométricos - Una proyección isométrica es un método gráfico de representación, más específicamente una axonometrica cilíndrica ortogonal Constituye una representación visual de un objeto tridimensional en dos dimensiones, en la que los tres ejes ortogonales principales, al proyectarse, forman ángulos de 120, y las dimensiones paralelas a dichos ejes se miden en una misma escala.
Google Play – Tienda de aplicaciones móviles para sistemas Android.
Unity – Entorno de trabajo para el desarrollo de juegos en varias plataformas.
SEO (Search Engine Optimization) – Conjunto de estrategias para posicionar una un elemento en un lugar preferente de un buscador virtual.
Tileset – Conjunto de gráficos que podemos utilizar para componer una escena de un juego.
TFM – Trabajo final de Master
Trigger – Reacción que tiene el juego cuando un suceso ocurre en la escena.

9. Bibliografía

Durgan A. Nallar. **Diseño de videojuegos en América Latina**. Ed. CreateSpace y Amazon. 2015

Schell, Jesse. **The Art of Game Design: A Book of Lenses, Second Edition**. CRC Press. 2014

ASOdesk Search Index Games Germany: Brand and Non-Brand Searches
[https://medium.com/blog-asodesk-com/asodesk-search-index-games-germany-brand-and-non-brand-searches-a2adde79686c] (2 de junio de 2019)

Atila, el rey de los hunos, la pesadilla de Roma
[https://www.nationalgeographic.com.es/historia/grandes-reportajes/atila-la-pesadilla-de-roma_6288/7] (2 de mayo de 2019)

El imperio huno de Atila en Europa
[http://www.historiadelascivilizaciones.com/2014/06/el-imperio-huno-de-atila-en-europa.html] (2 de mayo de 2019)

Game assets for indie game developers
[https://www.gamedevmarket.net/]] (18 de abril de 2019)

Get your apps ready for the 64-bit requirement
[https://android-developers.googleblog.com/2019/01/get-your-apps-ready-for-64-bit.html] (22 de mayo de 2019)

Iconmonstr – Free simple icons for your next project
[https://iconmonstr.com/] (18 de abril de 2019)

Indie games assets and tools
[https://itch.io/game-assets] (18 de abril de 2019)

Localytics
[https://www.localytics.com/] (9 de junio de 2019)

Requisitos del sistema para Unity
[https://unity3d.com/es/unity/system-requirements] (6 de junio de 2019)

TechCrunch – Tecnology news
[https://techcrunch.com] (9 de junio de 2019)