Universitat Oberta de Catalunya

uoc.edu

Trabajo Final de Máster: Deckgeon

Albert Pagès Raventós
Máster en Diseño y Desarrollo de Videojuegos - B2.504
Área de Postgrado de Informática, Multimedia y Telecomunicaciones

Consultores: Jordi Duch Gavaldà, Heliodoro Tejedor Navarro
Responsable: Javier Luis Cánovas Izquierdo

06/2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Deckgeon*
Nombre del autor:	*Albert Pagès Raventós*
Nombre del consultor/a:	*Jordi Duch Gavaldà, Heliodoro Tejedor Navarro*
Nombre del PRA:	*Javier Luis Cánovas Izquierdo*
Fecha de entrega (mm/aaaa):	06/2019
Titulación:	*M Dis. y des. Videojuegos*
Área del Trabajo Final:	*B2.504*
Idioma del trabajo:	Castellano
Palabras clave	*Diseño de juego, interfaz móvil, juego de cartas.*

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

La finalidad de este trabajo es el desarrollo de un videojuego para móviles y la exposición de todos los pasos realizados hasta su publicación en una plataforma de distribución real.

El videojuego móvil carece de interfaz y jugabilidad atractiva para un perfil menos casual y más asidua al mundo de los videojuegos. Los juegos móviles tienden a ser simples y rápidos. Este proyecto trata de encontrar un punto equilibrado entre partidas complejas, pero rápidas, para jugadores asiduos y con un control orientado al dispositivo móvil.

Mediante el motor de desarrollo de videojuegos Unity3D se ha generado un núcleo principal con todas las mecánicas que se querían añadir para una muestra del producto final: pantallas y menús principales, exploración y bucle de juego.

El resultado es bastante alentador, y las pruebas con usuarios reales han ofrecido ideas y cambios interesantes, bastante comunes entre ellos, por lo que el producto final es claramente superior al entregado en las primeras etapas del proyecto.

Aún así, el juego final requerirá de un pulido y varias iteraciones de desarrollo más antes de ser un producto comercial, pero la calidad y jugabilidad se muestran bastante atractivas para los jugadores asiduos a los juegos de rol, puzles y videojuegos en general.

Abstract (in English, 250 words or less):

The purpose of this project is the development of a video game for smartphones and the explanation of all the steps made until its publication in a distribution platform.

Videogames designed for smartphones typically lack the attractive interfaces and gameplay depth that users beyond the most casual scope expect and demand, opting instead for short, fast-paced, simpler experiences. The aim of this project is to find equilibrium between the ease of use and pick up and play approach of mobile gaming and the complexity of traditional games.

Through the Unity 3D game engine, an application has been made containing all of the basic core mechanics to serve as a vertical slice of the final, complete product.

The conducted tests performed with actual users have been encouraging, and provided feedback used to improve and expand the scope of the project, resulting in a considerably superior product to the initial version.

The final product requires a few more iterations to be a great commercial application, but the quality and gameplay are quite attractive for demanding players in *RPG* games, puzzles and video games in general.

Índice

Lista de figuras

1. Introducción

1.1. Contexto y justificación del trabajo

En este trabajo de final de máster se realizarán las tareas relacionadas con el desarrollo de un proyecto real, aplicando todos los conocimientos adquiridos a lo largo del curso: diseño previo, bocetos, diseño de juego, interfaz, programación y gráficos. A pesar de que sea un trabajo más global, se va a hacer hincapié en las asignaturas que más interés han generado durante el curso: *game design*, interfaz y desarrollo en 2D.

El videojuego móvil carece de interfaz y jugabilidad atractiva para un perfil más acostumbrado a los videojuegos. En muchos aspectos, los juegos móviles tienden a ser simples, rápidos, para un público casual, o en el caso de intentar ser un poco más complejos, con muchos errores de *feedback* en el control del juego. En este proyecto se trata de encontrar un punto de equilibro entre ambos perfiles: partidas complejas y rápidas para jugadores asiduos en el mundo de los videojuegos y adaptadas al formato móvil.

1.2. Objetivos del trabajo

En este trabajo se quieren remarcar los principales puntos tenidos en cuenta para el desarrollo del videojuego y las decisiones tomadas durante el mismo, con los siguientes objetivos principales:

- Planificación realista.
- Diseño de juego: mecánicas, interfaz, etc.
- Desarrollo en general del trabajo.
- Producto final: APK para ***Android***.

1.3. Enfoque y método seguido

Se usará un motor de videojuegos de uso no-comercial ya existente para desarrollar el videojuego, usando *Assets* (material de terceros) para un nivel más audiovisual.

En esta versión del juego se ha partido y reutilizado código propio realizado en asignaturas anteriores para agilizar el proceso. Aun así, se han mejorado ciertas mecánicas que ya se tenían planeadas y revisadas que se van a exponer en este documento. Al ser un juego nuevo, es lógico que el resultado sea un producto totalmente nuevo creado para la ocasión.

1.4. Planificación del trabajo

Se ha estructurado el trabajo de manera que las entregas y tareas importantes se desarrollan cada semana durante el proyecto, salvo en ocasiones que existan otras PECs o tareas que puedan ocupar más tiempo por lo que se amplía en dos semanas

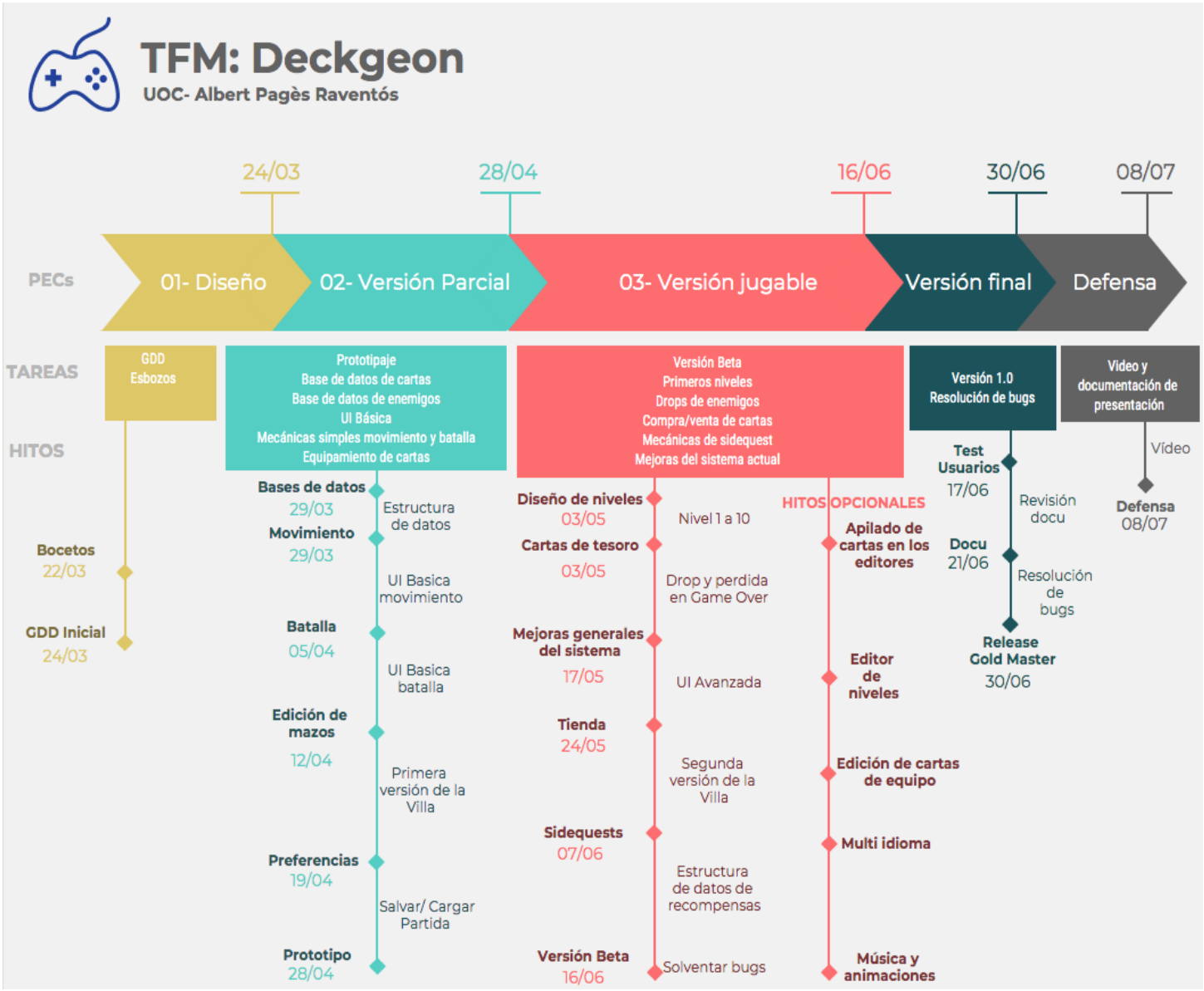


Fig. 1 Planificación del proyecto

La planificación ha variado ligeramente a lo largo del desarrollo, pero el uso de otras herramientas de gestión como ***Trello*** **[4]** han ayudado que las modificaciones no sean significativas.

Los hitos opcionales han quedado desplazados en segundo plano salvo en algunas ocasiones:

- La edición de equipo ha supuesto un gran esfuerzo, pero como se analizará en este documento, han supuesto una mejora al producto final.

- Las mejoras gráficas y sonido añadido han sido muy gratificantes de realizar e importar al producto final, y ha añadido un acabado más profesional.

- En el tramo final, y muy relacionado con las pruebas de usuario, se ha implementado un sistema de multi idioma.

1.5. Breve sumario de productos obtenidos

Se ha obtenido una versión del juego bastante avanzada pero no definitiva, una *beta* con todos los aspectos y herramientas importantes desarrollados. Dicho de otro modo, un *vertical slice* de lo que deberá ser el producto final y comercial.

Gracias a esta *beta* se obtendrá *feedback* de los usuarios de prueba y servirá también para publicitar el producto y darse a conocer en el mundo del desarrollo *indie*.

1.6. Breve descripción de los otros capítulos de la memoria

En este documento se presenta:

- El estado del arte del proyecto, una visión del género de los videojuegos en el mundo de la movilidad.
- La definición del juego, genero, diseño general del juego, historia y mecánicas.
- El diseño técnico, implementación y estructura.
- El diseño de niveles, mazmorra y villa.
- Pruebas de usuarios reales, y sus conclusiones tras jugar al juego.
- El manual de usuario y tutoriales incluidos en el juego.
- Herramientas externas utilizadas para la gestión y publicación del proyecto.
- Una breve justificación económica y planificación de futuro.
- Las conclusiones y siguientes pasos a realizar, futuro del proyecto.

2. Estado del arte

2.1. Revisión del género de juego

Los primeros teléfonos móviles *Nokia* incorporaban videojuegos simples en memoria, como la ***Serpiente*** o algún clon del ***Tetris***, realizados bajo la capa de ***Symbian OS*** y con ciertas limitaciones ya que disponían de poca capacidad de proceso, una minúscula pantalla y el propio set de botones físicos como interfaz de control entre el juego y el jugador. Debido a estas limitaciones, los videojuegos disponibles eran muy limitados y, salvo algunos intentos de acercar el mundo móvil a las videoconsolas portátiles (como es el caso de la ***N-Gage***), el videojuego en *smartphone* se estandarizó como un pasa ratos ideal.

Con la llegada y popularización de la telefonía móvil**[2]** a principios del 2000, y gracias a la salida de dispositivos como el iPhone en 2007 o los primeros dispositivos Android en 2008, el género del videojuego vio un interesante nicho de mercado a cubrir. A pesar de perder la entrada de *input* físico, disponía de una mejor pantalla y mayor potencia con la que mostrar mejores gráficos. Aun así, los videojuegos móviles seguían heredando ese aire tan casual, de pasatiempos y limitaciones en cuanto a su interfaz.

Con esta pérdida de *input* físico, se pueden encontrar juegos que requieren elementos de entrada de control más elaborados pero que incluyen un tedioso sistema de mando virtual en la propia pantalla.

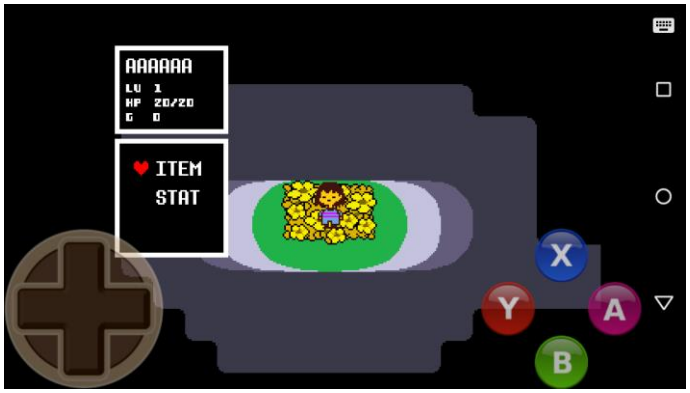


Fig. 2 Interfaz virtual de control

Las prestigiosas empresas ***King*** o ***Rovio*** vieron de manera inmediata que el juego debía adaptarse al control físico disponible y no al revés, y dieron paso a juegos con controles adaptados a las pantallas de los dispositivos móviles. Esto genera, como es obvio, no solo un control más sencillo sino una simplificación de las mecánicas de juego. Si bien ya existía el término casual (como videojuego que se utiliza de manera casual y usuario no asiduo a los videojuegos) la popularización de los móviles, de las pantallas táctiles y de estos videojuegos genera una atracción interesante para usuarios que nunca se habían atrevido a utilizar un videojuego.

Fig. 3 Interfaz táctil en Angry Birds

Sin embargo, el auge actual de los desarrolladores independientes (*indies*) aportan un golpe de aire fresco reinventando la entrada de controles independientemente de si sus mecánicas y su público objetivo son casuales o no. Una de las cosas que sí se debe que tener en cuenta es la barrera del tiempo y como esta se ha roto, pues un juego para dispositivos móviles puede ser utilizado en un corto periodo de tiempo o ser protagonista de un largo recorrido (como es el caso del fenómeno que resultó ser ***Pokemon Go!***). Aún así, las partidas o acciones que se requieren siguen siendo breves.

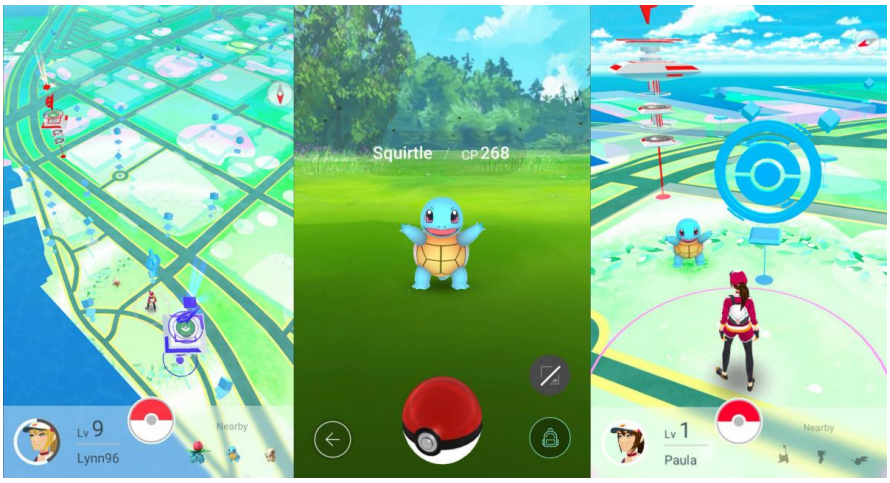


Fig. 4 Pokemon GO!

Los videojuegos de rol, o RPGs clásicos, orientados a jugadores más asiduos apenas tienen presencia en el mercado móvil, debido a sus complejas mecánicas que necesitan de un control físico en la pantalla. Famosas sagas como ***Final Fantasy*** o ***Dragon Quest*** se adaptan al control y tiempo para dar pie a versiones más simples que a sus versiones más clásicas de exploración, batalla y narrativa.

Fig. 5 Final Fantasy Brave Excius

Los videojuegos de puzles se han adaptado muy bien a los controles táctiles, sobretodo si se presenta una pantalla estática con un enigma a resolver. **The Room** presenta, en formato episódico, un objeto que debe ser abierto(una puerta, una caja, una mesa, …) para seguir avanzado en la historia. **Monument Valley** juega con la perspectiva para poder avanzar, con lo que resulta muy fácil de jugar en una pantalla táctil que permite mover elementos del escenario con el dedo. El caso de la empresa **Tiny Touch Tales,** realizan juegos sencillos con cartas que pueden suponer un reto para los jugadores más asiduos.

Fig. 6 The room y Monument Valley

Es con los juegos de estos y otros estudios independientes que se puede apreciar una adaptación del puzle más clásico y del uso de elementos diferentes para un control y narrativa totalmente orientadas al *feedback* de un móvil.

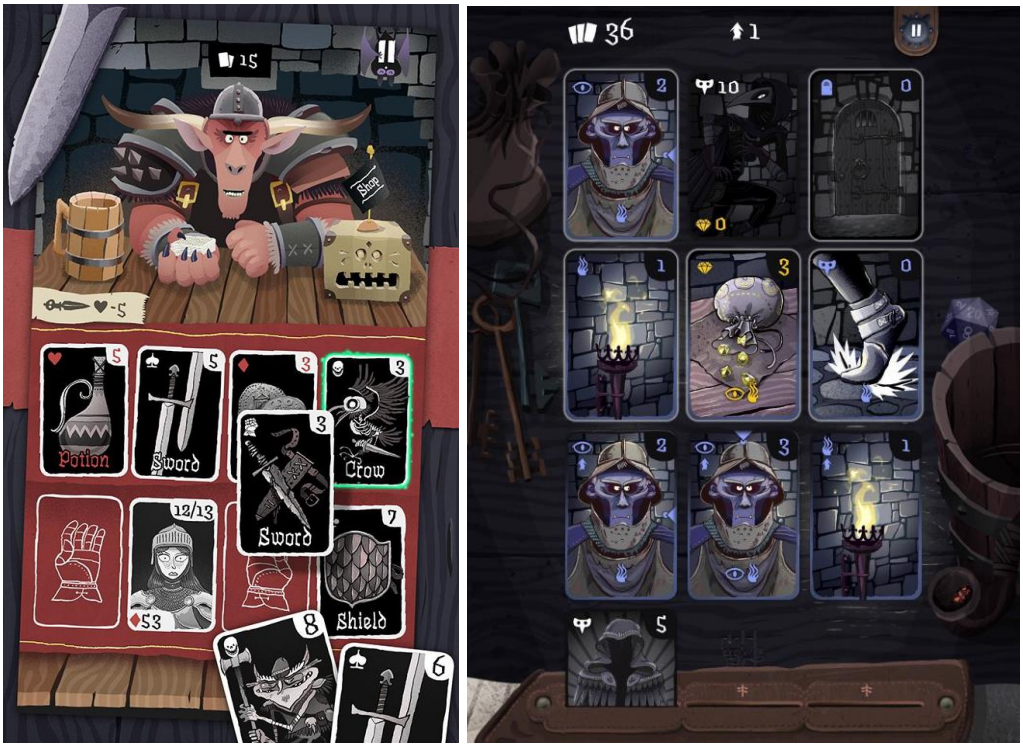


Fig. 7 Card Crawl y Card Thief, ambos de Tiny Touch Tales

2.2. Tecnologías más comunes

Para la realización de este proyecto y del desarrollo de videojuegos existen varias opciones, de entre las cuales se han tomado en consideración las 3 siguientes:

- ***Unity3D***: uno de los motores para desarrollar videojuegos más famosos del mercado, gratuito y con soporte para multitud de plataformas, entre ellas ***Android,*** que permite realizar ***debug*** si se utiliza algún dispositivo móvil físico. Desarrollo mediante C# o javascript y con muchas herramientas integradas. Además posee una de las comunidades más activas del mundo del desarrollo de videojuegos. Si el proyecto sale comercializado y genera más de 100.000$ de beneficios es necesario adquirir una licencia de *Unity*, de coste anual.

- ***Unreal Engine 4***: el motor de videojuegos multiplataforma de *Epic Games* destaca por su gran acabado visual con pocos pasos. Permite la generación de scripts en C++ o en su famoso sistema de *Blueprints*, esto es, programación visual, sin necesidad de ningún *plugins* externo. Si el proyecto sale comercializado, *Epic Games* obtiene un 5% de los beneficios de la obra cada trimestre.

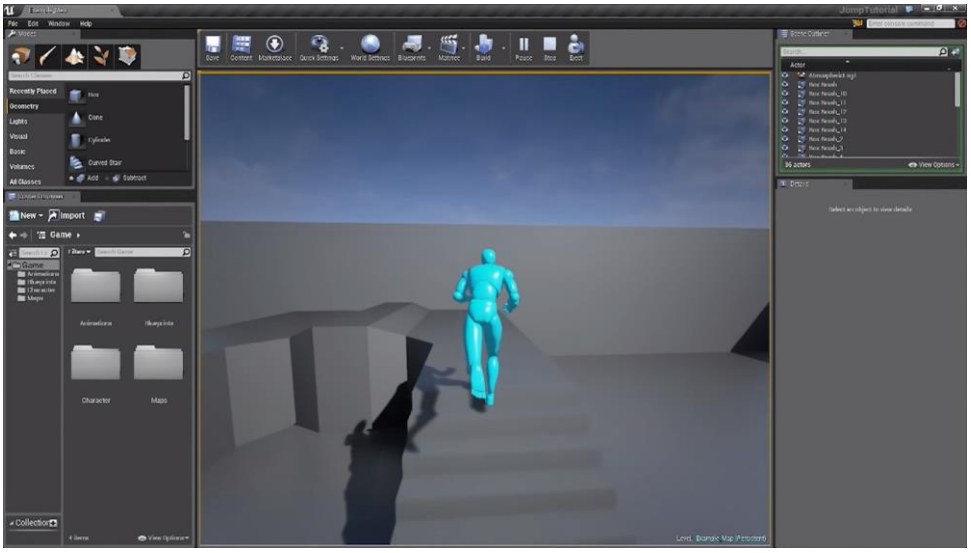


Fig. 8 Entorno de Unreal Engine

- ***GameMaker Studio***: motor de desarrollo de videojuegos orientado a juegos en 2D, con un sistema de scripts más visual, aunque permite el desarrollo de programación mediante GML (Game Maker Language). Por 99$ permite la publicación de videojuegos en las principales plataformas móviles.

Fig. 9 Entorno de Game Maker Studio

A parte del motor del videojuego se deben decidir otros elementos según si el juego será en 2D o 3D, *single player* o *multiplayer, offline* o *online,* o si el equipo de desarrollo es formado por una o varias personas.

3. Definición del juego

3.1. Descripción del juego

El jugador toma el rol de una aventurera para explorar los diferentes niveles de la mazmorra *Deckgeon* y completar todos sus niveles y secretos. El juego incluye exploración de mazmorras mediante una baraja de cartas de movimiento, y pelea de enemigos mediante una baraja de cartas de batalla y una vista en 2D con carácter colorido y simpático.

a. Género

Puzle y juego de cartas: selección de movimientos y acciones mediante una mano de 4 cartas por turno. Requiere recordar de qué cartas se disponen, que cartas se han descartado y que cartas pueden salir al robar una nueva mano.

Rol por turnos: exploración de mazmorra, turnos durante las batallas contra los enemigos y personalización del equipo. Requiere reconocer elementos básicos de los juegos de rol, como ataque, magia o defensa.

b. Historia

La mazmorra más famosa del lugar, *Deckgeon*, que había atraído a centenares de aventureros a lo largo y ancho de su historia, acumulando muchísimas leyendas a sus espaldas, ha acabado en el olvido en pos de otras zonas de aventuras más transitadas y famosas. El asentamiento alrededor de la mazmorra ha caído poco a poco en declive y la zona ha pasado a formar parte de la maleza, y ya pocos aventureros buscan tesoros en el lugar.

La protagonista de nuestra historia no se da por vencida y con la ayuda de los pocos habitantes que quedan alrededor, se aventura hasta lo más profundo de los niveles.

c. Referencias

Se han tomado algunas referencias tanto a nivel de *gameplay*, mecánicas, así como de estética.

- **The Legend of Zelda**, Nintendo (Nes). **[9]** Observaciones de interés: Estética, exploración de mazmorras y enemigos.

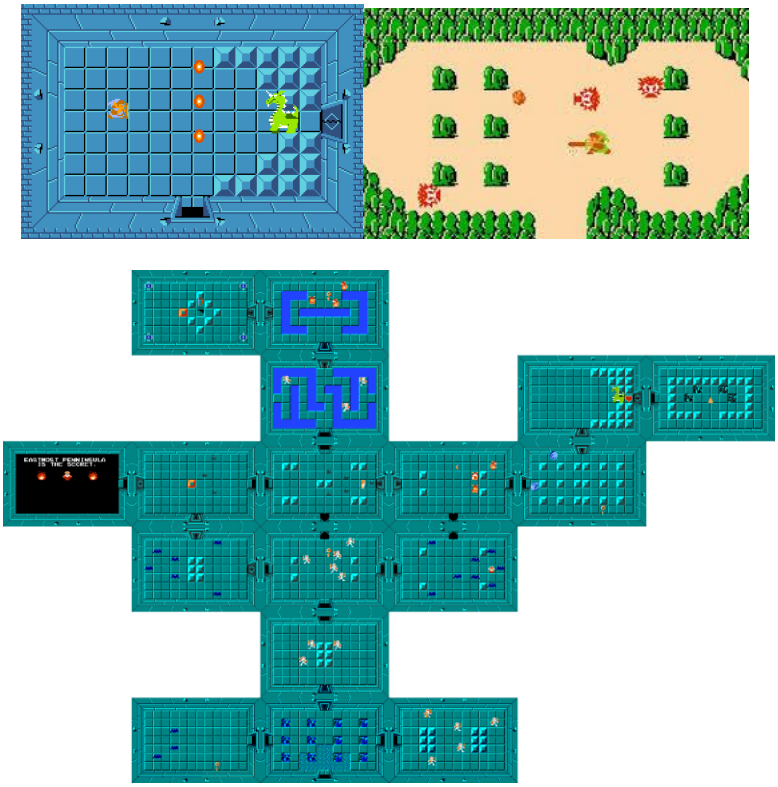


Fig. 10 Pantalla y mazmorra de Zelda

- **Magic the Gathering**, Wizards of the Coast (juego de mesa). **[12]** Observaciones de interés: Mazo y cementerio de cartas, combinaciones de cartas, estadísticas y distintos niveles en las cartas.

Fig. 11 Magic Arena PC

- **Card Crawl,** Tiny Touch Tales (Android). **[11]** Observaciones de interés: Juego con interfaz móvil, UI interesante, usable y accesible.

Fig. 12 Card Crawl en Android.

- **Dragon Quest Saga,** Square-Enix (Varias plataformas). **[10]** Observaciones de interés: Diseño de enemigos interesante y reconocibles, tono simpático.

Fig. 13 Slimes, enemigo clásico de Dragon Quest

d. *Mecánicas generales y su estructura*

i. Exploración y batalla

- Existen dos mazos distintos: mazo de movimiento y mazo de batalla.

- Es necesario utilizar **cartas de movimiento** para explorar un nivel de mazmorra moviéndose por las diferentes habitaciones:
    - *Game Over*, si se terminan las cartas del movimiento se reinicia el nivel.
    - Éxito si se llega a las escaleras de bajada.

- Si en una habitación hay un monstruo empezará la batalla, usando las cartas de batalla será necesario acabar con el monstruo para seguir avanzando.

- Siempre se reparten cuatro cartas, se usan o descartan tres a decisión del jugador y la cuarta se guarda para la siguiente mano.

- Se puede descartar toda la mano.

- En la última mano se deben usar o descartar todas las cartas.

ii. Combos

- En el juego cada carta pertenece a un tipo: Ataque (cortante, golpe, ….), Magia (fuego, hielo,...).

    - Si se acumulan 2 o 3 cartas del mismo tipo el jugador recibe una bonificación en el efecto deseado, pues el efecto se multiplica por el número de cartas de este tipo.

iii. Tesoro

- Durante la exploración el jugador puede encontrar **cartas de tesoro** eliminando a los diferentes enemigos. Estas cartas se descubren en la casa, donde:
    - Las cartas de moneda se convierten en oro automáticamente.
    - Las cartas de objetos especiales se guardan en el **Tesoro** para venderlas o entregarlas en una misión.

iv. Equipo

- Durante el juego (mazmorra, tienda,…) el jugador puede encontrar o comprar cartas de equipo. Estos objetos proporcionan mejoras en los atributos del personaje, suplantando el sistema de subida de niveles de los clásicos videojuegos de rol.

v. Villa

- La Villa hace de menú principal del juego. En la Villa el jugador podrá:
    - Modificar los mazos o equiparse nuevos objetos antes de bajar a explorar la mazmorra en la **Casa**. Además, también sirve para tomarse un descanso y revisar que cartas se han encontrado durante la exploración.
    - Comprar nuevas cartas y vender cartas que no se quiera la **Tienda**.
    - Acceder a consejos y *misiones* en la **Taberna**.
    - Acceder a la **Mazmorra**.
    - Acceder a las **Opciones del juego.**

e. Objetivos: recompensas y victoria.

El objetivo general del juego es llegar al último nivel de la mazmorra.

i. Condiciones de victoria

- Para cada nivel de exploración, llegar a las escaleras de bajada.
- Para cada batalla, reducir la vida del enemigo a 0.

ii. Condiciones de derrota

- Para cada nivel de exploración, no conseguir llegar a las escaleras de bajada.
- Para cada batalla, que la vida del personaje se reduzca a 0.
- Quedarse sin cartas en cualquiera de los 2 mazos.
- En estos casos de **Game Over**, se proporciona una pérdida de gran parte de las cartas encontradas (80% - 90%).

iii. Huida

- El juego permite huir de la mazmorra siempre, perdiendo la mitad (50%) del tesoro encontrado, a no ser que el jugador se encuentre en una sala con **escaleras**.

iii. Motivaciones

- Conseguir más cartas durante la exploración. Cuantas más cartas de tesoro se lleven encima, más probabilidades hay de conseguir más.
- **Para el movimiento:** conseguir más movimientos para seguir explorando.
- **Para las batallas**: conseguir mejores cartas para ganar en las peleas.
- Superar misiones de la población de la villa, que ofrecen mejores cartas y objetos

Con estas condiciones de victoria y derrota se consigue:

- Que el jugador deba decidir si lo más sensato es seguir explorando o huir.

- Que se premie al jugador valiente, pues el número de **Tesoro** actual otorga mayor probabilidad de encontrar más tesoros.

- Que se penalice al jugador demasiado ambicioso, ya que si explora demasiado sin salir de la mazmorra puede morir y perderlo todo.

f. Concept Art

- Villa

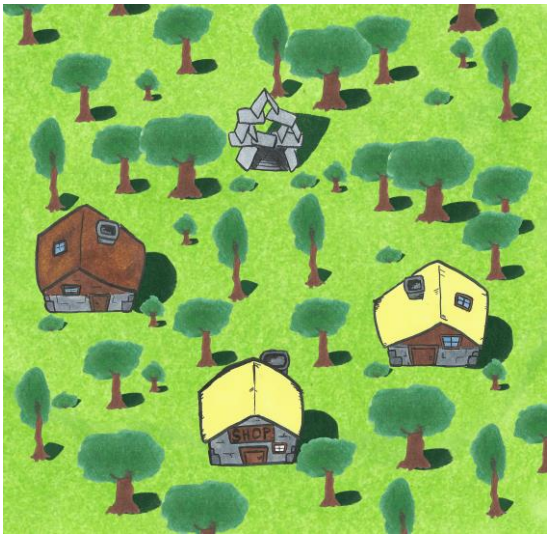


Fig. 14 Boceto del poblado inicial

- Personaje jugador

Fig. 15 Boceto del personaje del juego

- Diseño de la mazmorra: salas

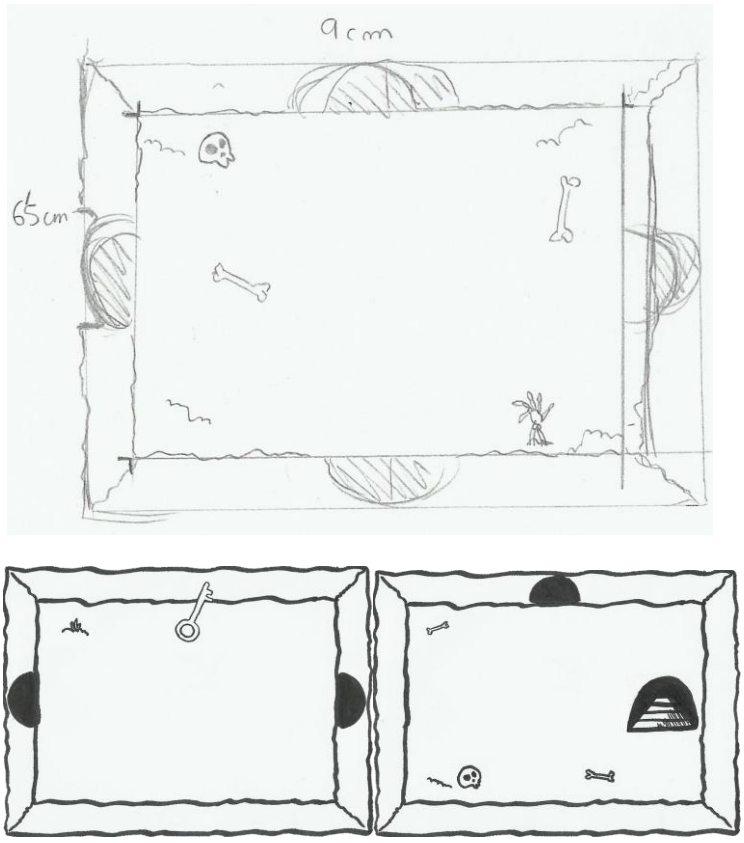


Fig. 16 Evolución de los bocetos de las salas con diferentes eventos

- Enemigos

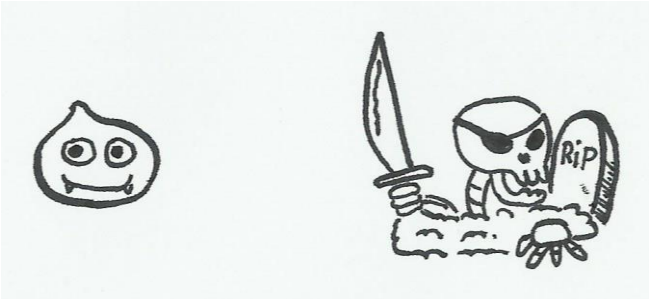


Fig. 17 Boceto de los enemigos

- Cartas

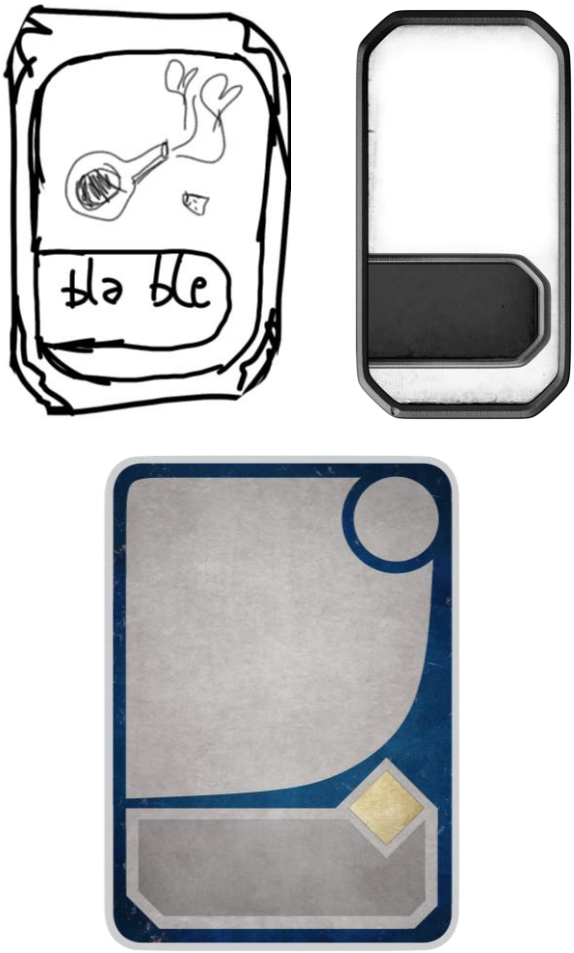
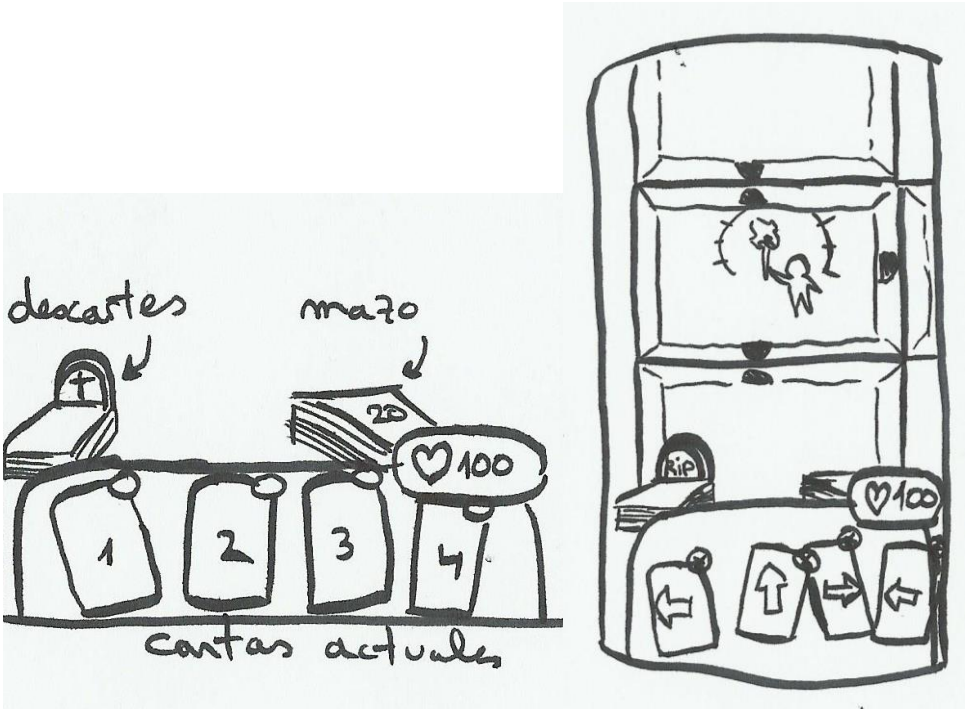


Fig. 18 Evolución del diseño de las cartas

- Bocetos del juego



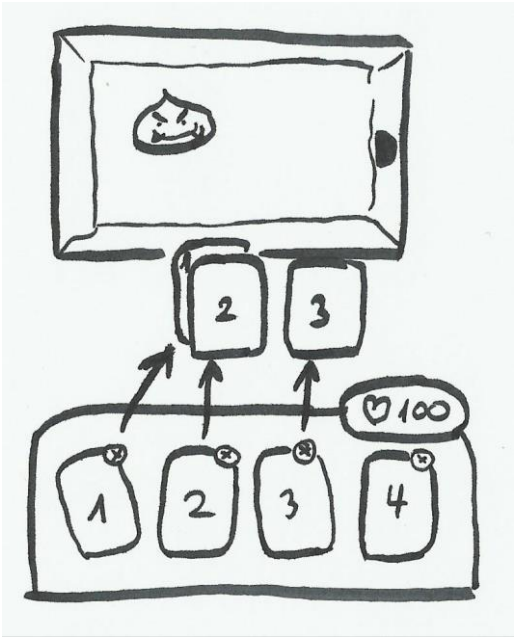


Fig. 19 Propuesta e ideas de interfaz y pantalla de juego

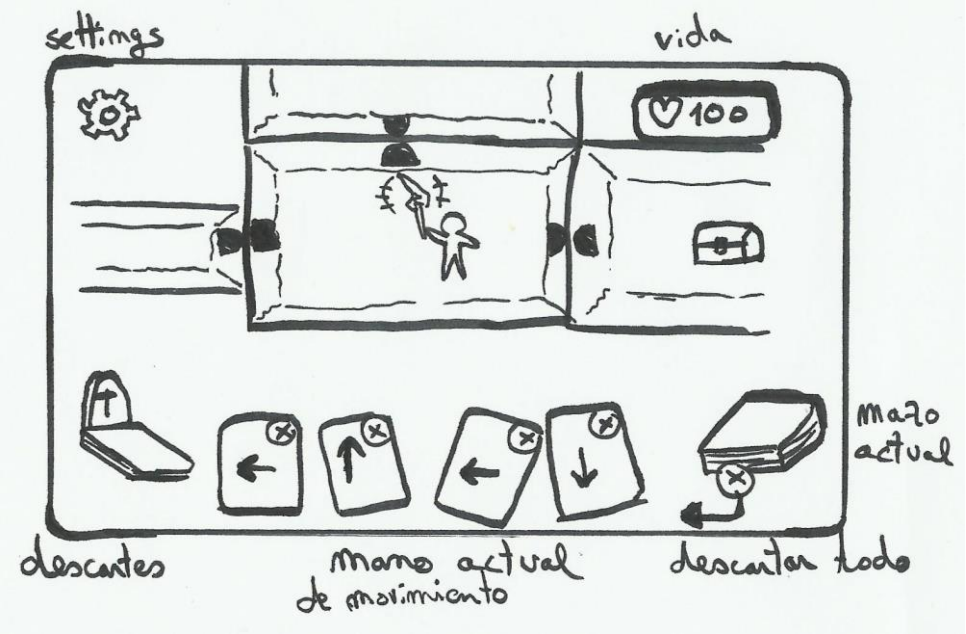


Fig. 20 Pantalla en horizontal, versión final

- Concepto de niveles

A continuación, se añade un ejemplo de un diseño de nivel, la entrada, los enemigos situados, los objetos, mini jefes y la salida del nivel.

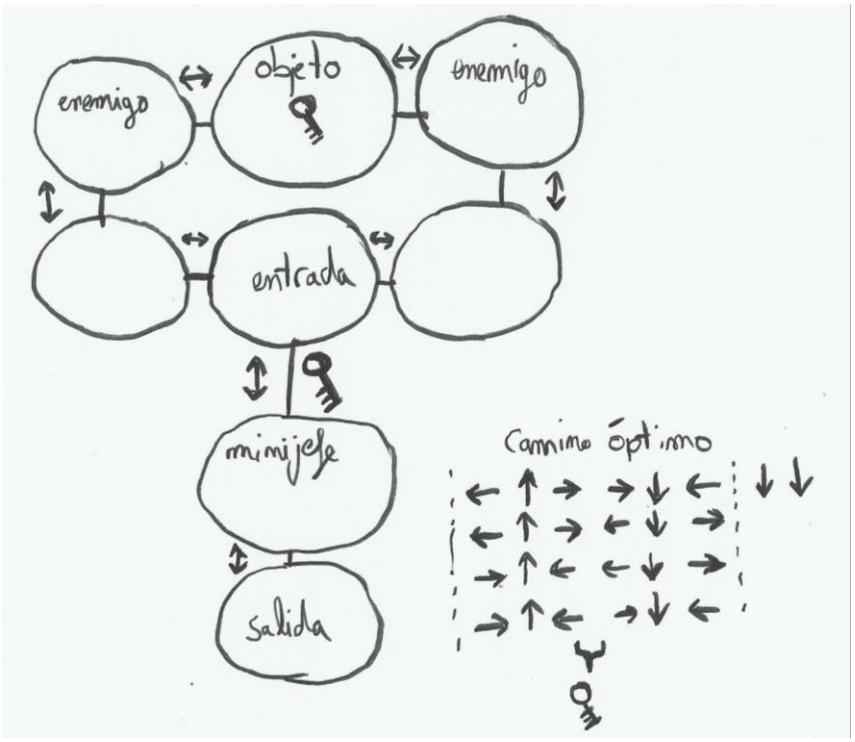


Fig. 21 Posible estructura de un nivel

4. Diseño técnico

4.1. Entorno y herramientas utilizadas

Para la realización del proyecto se ha optado por la versión de ***Unity3D v.*2018.3.9f1** ya que ha sido la opción usada a lo largo del aprendizaje del *Máster de Diseño y Desarrollo de Videojuegos*, y de la que se disponen más librerías, documentación y con la que se tiene más práctica. Además, se ha reutilizado código propio que ya se había desarrollado en C#, estructuras de datos y librerías propias. En vistas a una futura comercialización es una de las plataformas que ofrece un trato comercial justo con sus desarrolladores.

Durante el desarrollo de este trabajo, ***Unity Technologies*** actualizó su software a una versión estable **v.2019.1** con mejoras muy interesantes, pero se ha considerado un poco atrevido hacer la migración de versión en pleno desarrollo del trabajo. Aun así se prevé realizar esta actualización para seguir con el trabajo una vez realizada la evaluación del mismo.

Así mismo, se ha utilizado ***Visual Studio*** incorporado en el paquete de ***Unity*** para el desarrollo de los scripts, ya compatible con *Windows y Mac*. Este *IDE* substituye el conocido ***MonoDevelop,*** editor que se incluía en la instalación de ***Unity***.

También se han utilizado algunos *Packages* en *Preview* de ***Unity*** para la animación de personajes en 2D **[7]** :

- *2D PSDImporter:* Separa automáticamente las capas de los ficheros de imagen tipo *PSD* de *Adobe Photoshop* cuando son importados a *Unity3D,* para así trabajar la imagen por partes.

Fig. 22 Partes del fichero PSD de la protagonista

- *2D Animation:* Ventana y herramientas de animación para objetos 2D.

- *2D IK:* Herramientas útiles para que el movimiento de los huesos resulte más natural.

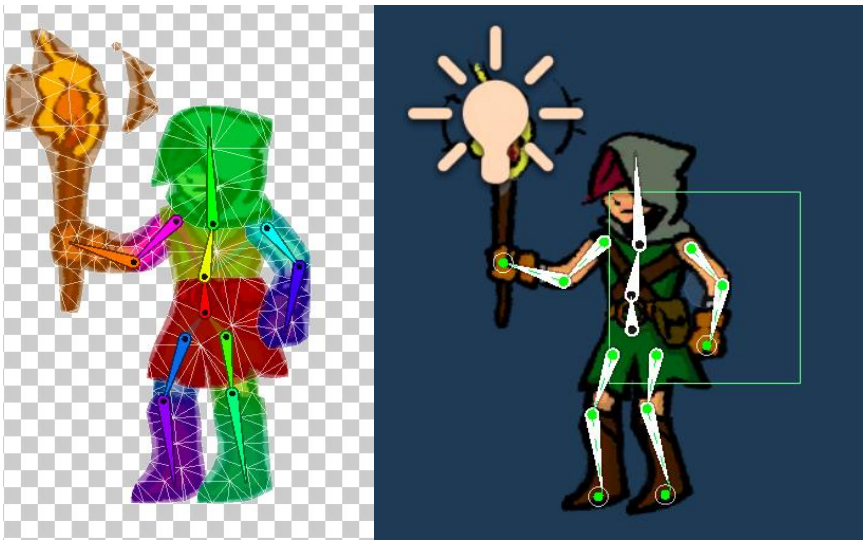


Fig. 23 Rigging y animaciones 2D

Se han usado gráficos propios realizados con el software de diseño y dibujo ***Adobe Photoshop CS6.***

Para el desarrollo de los modelos **FBX** se ha usado el software ***3D Studio Max 2018.***

4.2. Requerimientos técnicos

Se ha utilizado un *MacBook* Air 2012 para el desarrollo del motor del juego, con las siguientes especificaciones técnicas:

MacBook Air 13 pulgadas con MacOS 10.11.6 El Capitán

- *Procesador: Intel Core i7 2Ghz*
- *Discos duros: SSD 240GB SATA3*
- *Memoria: 8GB DDR3*
- *Gráfica: Intel HD graphics 4000 1536MB*

Se ha utilizado un PC para el desarrollo de gráficos y modelos, con las siguientes especificaciones técnicas:

PC con Windows 10 Pro

- *Procesador: Intel Core i7 2Ghz*
- *Placa Base: Asus*
- *Discos duros: SSD 240GB SATA3*

- *Memoria: 16GB 2x8GB DDR4*
- *Gráfica: GTX 1660*
- *Otros elementos (torre, fuente de alimentación)*
- *Pantalla, teclado, ratón, tableta Wacom.*

Para las pruebas reales del juego se han utilizado:

- ***Samsung Galaxy J3*** *con* ***Android 5.1.1***
- ***One Plus 3T*** *con* ***Android 8.0.0***

A parte, se prevé que las pruebas de usuario otorguen información interesante sobre el funcionamiento del juego en diferentes dispositivos móviles con varias versiones Android, que se analizan en el apartado correspondiente de esta memoria.

4.3. Assets utilizados

Para algunas partes gráficas y de interfaz se utilizarán kits de desarrollo adquiridos mediante la plataforma *Humble Bundle* que incluyen material que se puede encontrar en la *Asset Store de Unity* (por ejemplo: https://assetstore.unity.com/publishers/13229) y en *GameDevMarket* (por ejemplo https://www.gamedevmarket.net/asset/skill-icon-pack-8729/).

Estos elementos de terceros se incluyen en la carpeta ***Assets/Imágenes de Terceros*** del propio proyecto.

Para los otros elementos (fondos, dibujos de cartas, …), se ha dibujado directamente usando la herramienta ***Adobe Photoshop*** mediante una tableta de dibujo, con un boceto previo en papel de las ideas o elementos que se querían añadir.

Con estos bocetos se han diseñado unos enemigos que, al principio eran *Sprites* como la mayoría de imágenes del juego, pero que parecía gracioso que los monstruos tuvieran otra dimensión, por lo que para el desarrollo de enemigos se ha contactado con un modelador y animador que ha colaborado en el proyecto para realizar los ***FBX*** y animaciones de los dos enemigos.

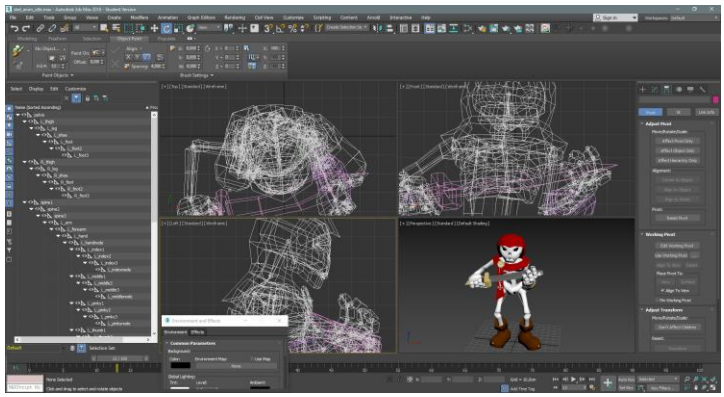


Fig. 24 Modelando al esqueleto

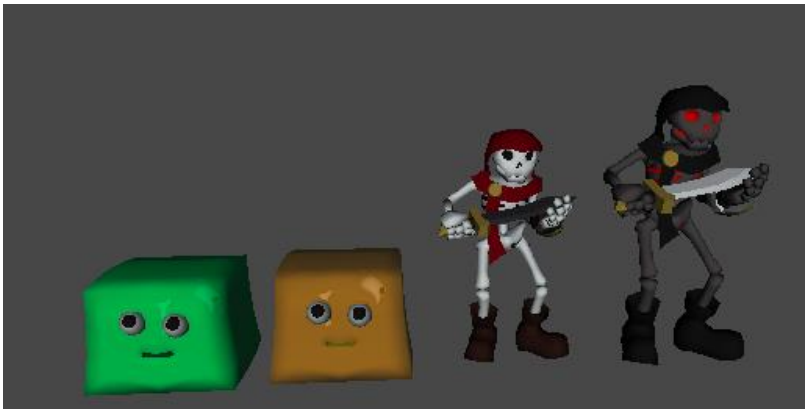


Fig. 25 Enemigos en Unity3D

Las animaciones en *Unity3D* incluyen eventos en puntos concretos de la animación para realizar acciones mediante llamadas a funciones secundarias asociadas al enemigo. Así, por ejemplo, un enemigo notifica al sistema que ha terminado la animación de morir para hacerlo desaparecer.

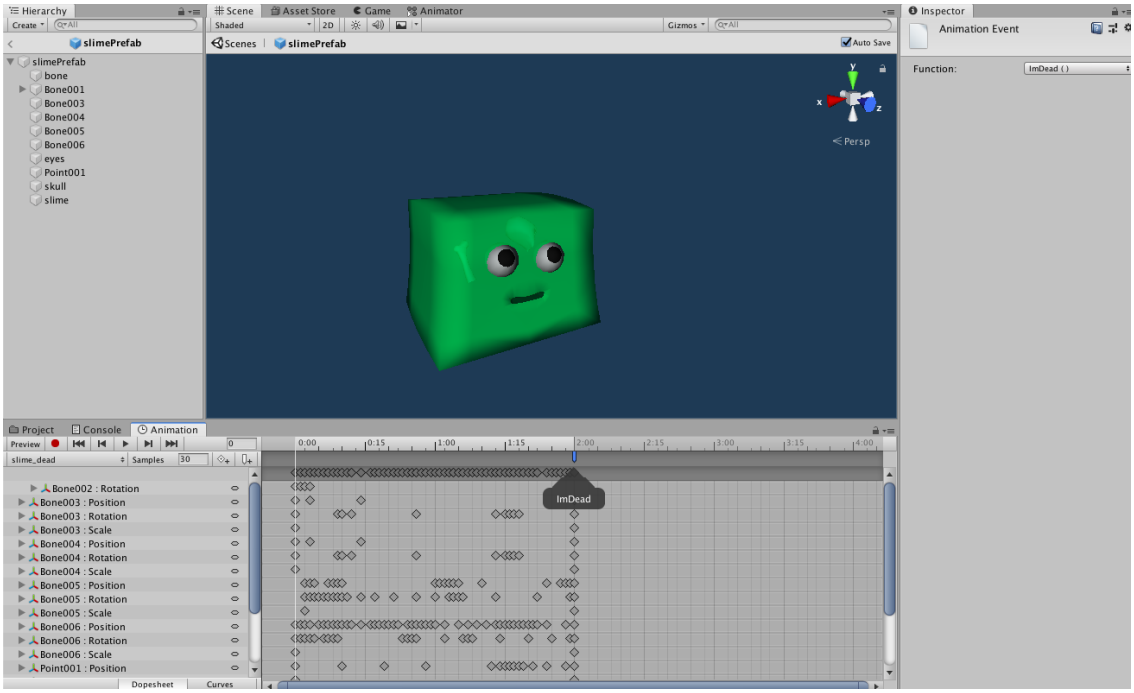


Fig. 26 Evento ImDead al segundo 2 de la animación de muerte

Me gustaría agradecer a **Héctor Ariza Sol** por la ayuda proporcionada para realizar estos modelos y sus animaciones, con su total autoría sobre éstos.

Para los Sonidos se han utilizado varios *sets* de efectos sonoros comprados, al igual que con los sets de gráficos, mediante la plataforma *Humble Bundle* **[6]** que ofrece ofertas de videojuegos y desarrollo de manera temporal, y que ofreció ofertas asociadas a *Game Dev Market* **[5]***:*

- https://www.gamedevmarket.net/asset/inventory-sfx-bundle-4864/
- https://www.gamedevmarket.net/asset/medieval-fantasy-sfx-pack-1061/
- https://www.gamedevmarket.net/asset/pro-sound-collection-6169/

Además dentro del control de volumen de ***Unity 3D*** se ha generado un *Audio Mixer* con dos canales: uno para la música y otro para los efectos de sonido.

De esta forma, en la configuración del juego será posible controlar estos dos volúmenes por separado y ofrecer al usuario la posibilidad de aumentar o disminuir los dos canales de manera independiente.

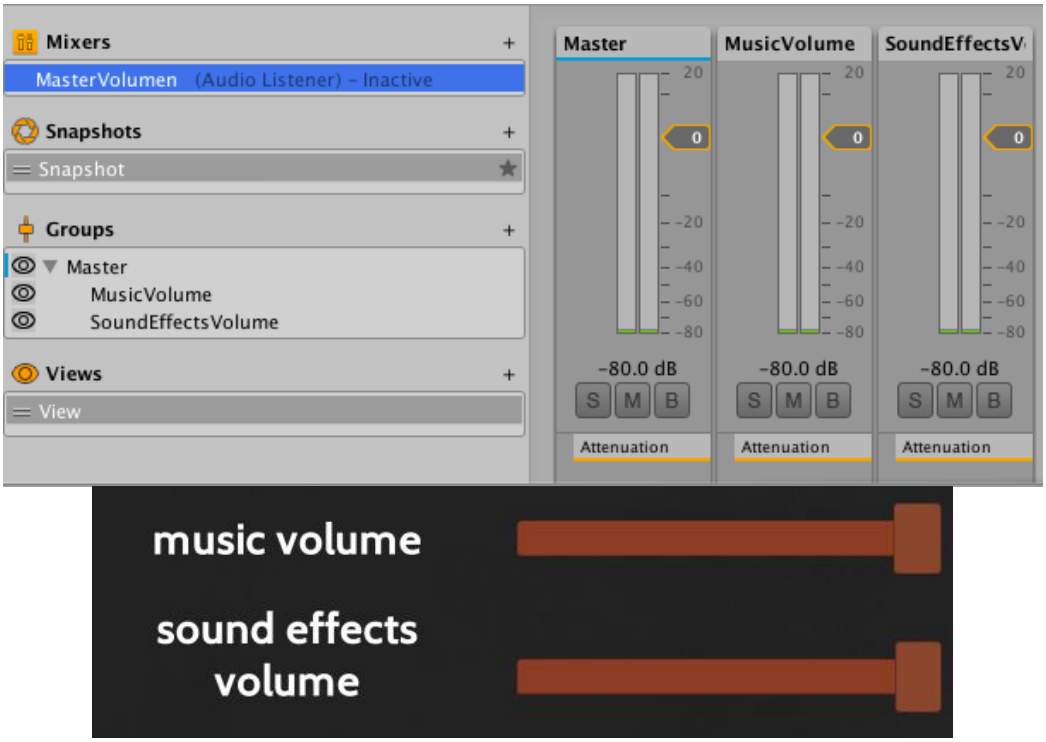


Fig. 27 Control de volumen de música y efectos de sonido en Unity y in-game

Respecto a las canciones de fondo, me gustaría agradecer a Kepa Izal por la ayuda proporcionada para realizar estas canciones, con su total autoría sobre éstas.

4.4. Arquitectura de juego

i. Estructura del proyecto

La estructura del proyecto en Unity3D es la siguiente:

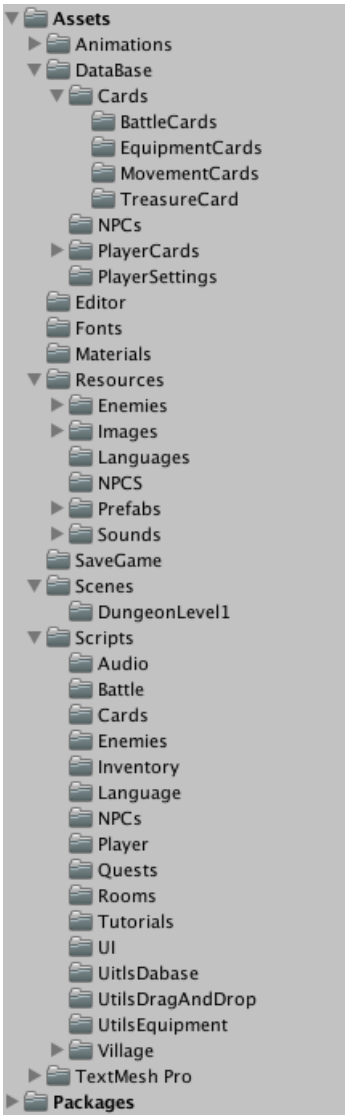


Fig. 28 Estructura del proyecto en Unity3D

a. **Animaciones**: Animaciones creadas para el proyecto, separando por elementos del juego que contienen clips de animación y controladores asociados a los *GameObject*. Estos son: elementos de interfaz, personajes o enemigos.
b. ***DataBase***: Archivos de tipo *Asset* de diferentes fuentes: Cartas, Personajes o Configuración del jugador. Este apartado se explica con más detalle en el sistema de bases de datos

c. **Editor**: Scripts que modifican la interfaz del propio Unity3D para generar Assets personalizados, necesario para la creación de la base de datos.
d. **Fuentes:** fuentes especiales usadas para este proyecto.
e. **Materiales:** materiales aplicados a distintos elementos del juego, en este caso sobretodo a elementos que se ven afectados por la iluminación.
f. **Resources:** Carpeta principal de objetos para instanciarlos en el juego y facilitar el uso y la ampliación del juego.
    i. Enemigos: Modelos propios y *prefabs* de enemigos.
    ii. *Images*: imágenes tanto propias como de terceros.
    iii. NPCS: Imágenes propias de personajes y prefabs de personajes
    iv. *Prefabs*: *prefabs* genéricos de interfaz y jugador.
    v. *Sounds*: Carpeta audio que contiene efectos de sonidos del jugador, ítems y enemigos. También contiene música de fondo.
    vi. *Languages*: Ficheros xml con la información de los distintos idiomas del juego (inglés y español).

g. **SaveGame**: Carpeta propia para guardar archivos de configuración y de la partida. Esta carpeta puede variar según el producto compilado, ya que por ejemplo ***Android*** almacena estos archivos en otra subcarpeta interna.
h. **Scenes**: Escenas utilizadas en el proyecto.
i. **Scripts**: Scripts utilizados en el proyecto, separados por su ámbito.
j. **TextMesh Pro y Packages**: Carpetas propias de *Unity3D* para algunos elementos utilizados durante el desarrollo, como por ejemplo texto enriquecido.

Se han implementado varias escenas en el juego:

a. **VillageScene**: menú principal.
b. **HomeScene**: escena de la casa, edición de mazos y objetos.
c. **WardrobeScene**: escena del equipamiento.
d. **TavernScene**: escena de la taberna, interactuar con personajes.
e. **StoreScene**: escena de la tienda, comprar y vender cartas.
f. ***DungeonLevel1***: carpeta que contiene los niveles actuales de la mazmorra.

A continuación se resumen los scripts más importantes:

- ***Battle***: Controladores de la batalla y comportamiento de enemigos, así como los ítems conseguidos.
- ***Cards***: Clases relacionadas con las cartas, su configuración, identificadores, control de mazos y mano de cartas.
- ***Enemies***: Configuración de enemigos, sus características y posibles ataques u objetos.
- ***Inventory***: Control de objetos durante la exploración de la mazmorra.
- ***NPCs***: Configuración de diálogos y personajes.
- ***Player***: Configuración jugador y sus características.
- ***Quests***: Configuración de las búsquedas secundarias asociadas a los NPCs.
- ***Quests***: Configuración de las búsquedas secundarias asociadas a los NPCs.
- ***Rooms***: Configuración de las salas de la mazmorra.
- ***Tutorials:*** Control del tutorial y su contenido.
- ***UtilsDatabase:*** Herramientas para el acceso a la base de datos.
- ***UtilsDragAndDrop:*** Herramientas para el movimiento de cartas por la pantalla y control de la zona donde son soltadas.
- ***UtilsEquipment:*** Herramientas auxiliares para las características de las cartas de equipo.
- ***Village:*** Control de Village, Home, Store, Tavern y de las configuración propia del juego (volumen, partida salvada,...).
- **GameDungeon:** Control del bucle de juego, la clase principal durante la exploración de la mazmorra.
- **OnBoard:** Control de las cartas que están en juego durante la exploración de la mazmorra.
- **Movement:** Control del movimiento del jugador.
- **UI:** Control de la interfaz gráfica e información de la exploración.
- **Language:** Controles para el diccionario de idiomas y lectura de ficheros XML.

ii. Sistema de bases de datos

Se ha utilizado una estructura interna de *Scriptable Objects* (en formato *Assets)* para almacenar datos persistentes, obviando el sistema de *Player Prefs* nativo de *Unity3D* que no es muy útil para almacenar muchos datos.

Para ello se ha añadido una carpeta */Editor* donde se añaden los *scripts* que generan un *Asset* propio: una lista de *CardClass, PlayerSettings o NPC*. Con ello, se podrán generar varias listas donde se ha separado:

- ***DataBase***: cartas creadas en la base de datos, todas las que existan en el juego deberían ir en estas bases de datos.

- ***Storage y Deck***: cartas que se encuentran en el almacenamiento y baraja del usuario.
- **PlayerSettings:** Características del jugador, su configuración (niveles de sonido) y otros elementos del juego, como misiones finalizadas o equipamiento.
- **NPCs:** Diálogos de los personajes.
- **Quests:** Diálogos de los personajes referente a alguna búsqueda, así como su recompensa.

Para gestionar esta base de datos se ha usado una única clase estática **DataBaseAccess**, que tiene referencias a todos estos *Assets* y es la encargada de leer y editar la información. Hay que tener mucho cuidado al editar los *Assets* de la base de datos, pero de esta manera se facilita mucho que el producto sea escalable haciendo muy sencillo generar nuevas cartas, *NPCs* o *Quests*.

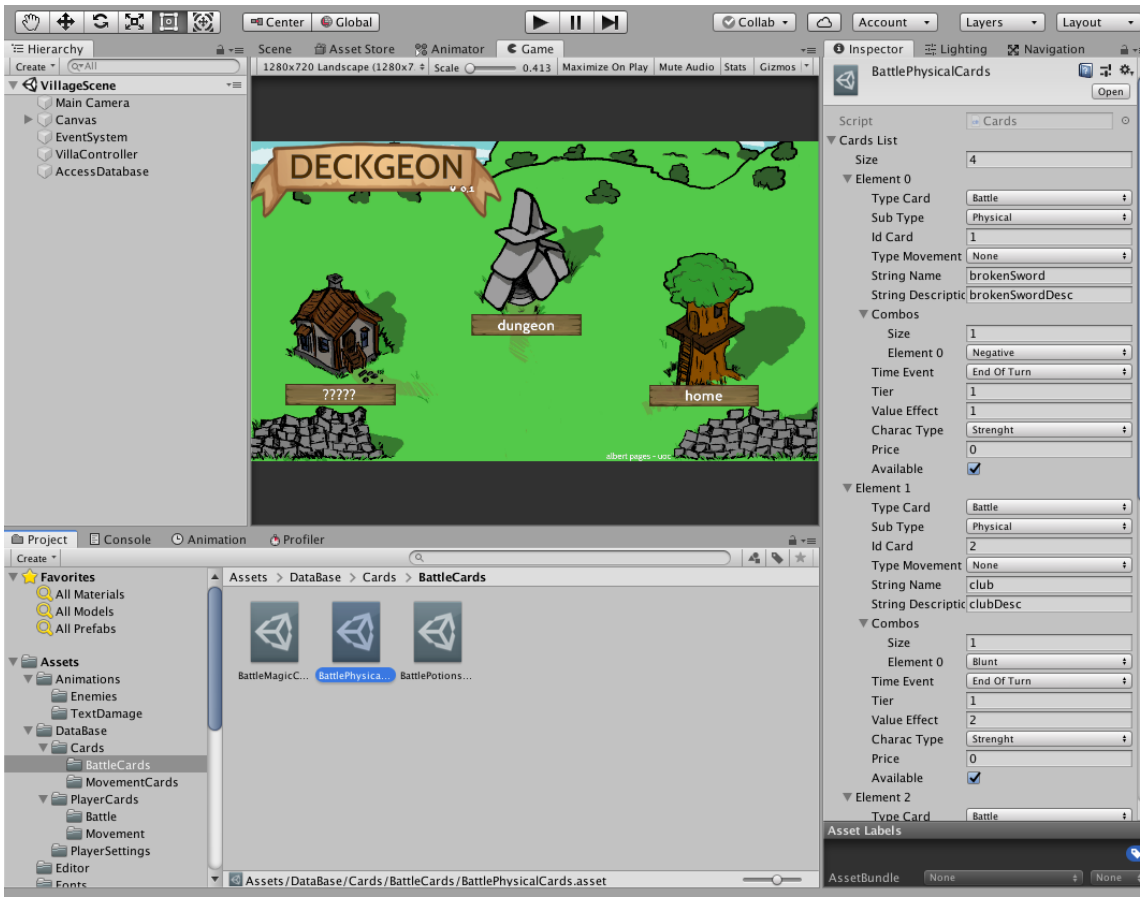


Fig. 29 Bases de datos

Aunque se quería evitar el uso de archivos externos, tras mucho estudio se ha llegado a la conclusión que los *Scriptable Objects* son muy útiles para ver los datos durante la instancia de juego, pero pierden consistencia entre ejecución y ejecución y vuelven a su estado original, razón por la cual las hace una herramienta perfecta para almacenar información solamente de lectura pero no de escritura ni avance de juego. Por ello se ha integrado un sistema de almacenamiento en archivos *JSON* para guardar

el avance el jugador y el uso de archivos XML para almacenar las diferentes traducciones.

Para el almacenamiento en estos archivos JSONs se utiliza un identificador de carta (llamado *CardIdHelper*) tal que:

tipo de carta _ subtipo de carta _ id de carta

De esta manera no se almacena toda la información de una carta sino su identificador. Esto es muy útil por dos razones:

- Si se modifica la carta en la base de datos, afecta a todo el juego, haciendo que el balanceo de cartas demasiado poderosas o demasiado débiles sea sencillo.

- No se almacena información extra y su lectura es más rápida.

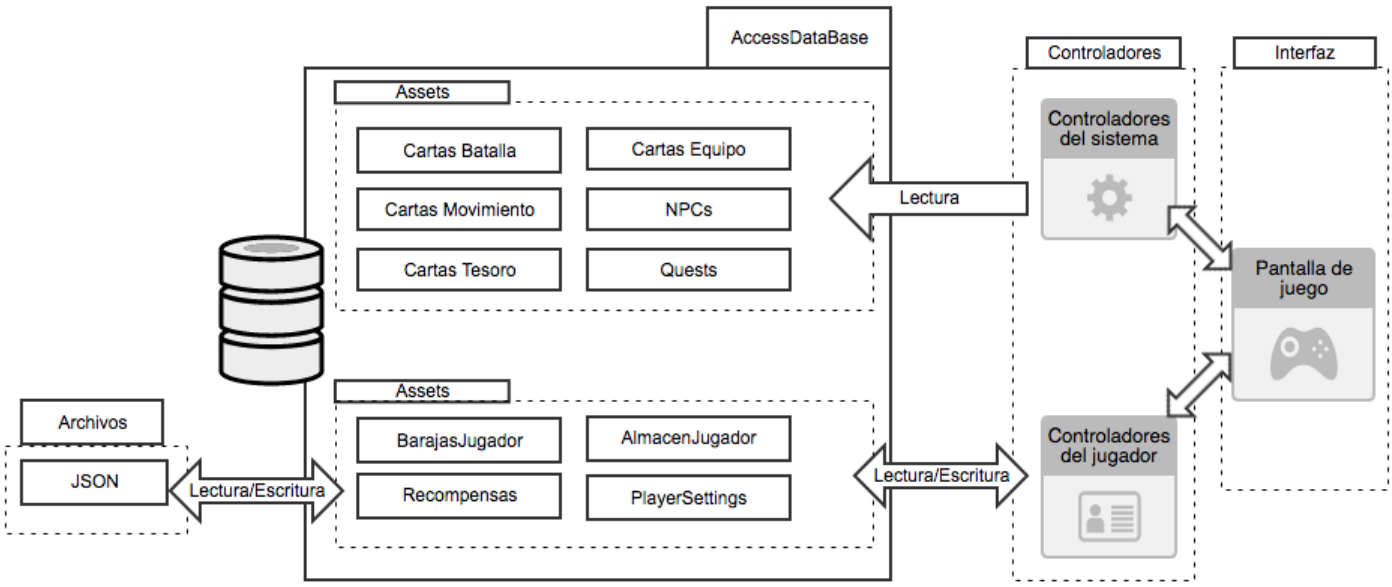


Fig. 30 Estructura de la base de datos

Todos los controladores del juego utilizan *AccessDataBase* para conseguir la información que quieran consultar o guardar en la base de datos, haciendo el diseño del juego lo más modular posible, y generando cohesión entre cada grupo de controladores sin introducir funciones propias de lectura o escritura.

Al ser una clase *singleton*, también es interesante que se encargue de pedir la traducción y devolverla al objeto que la pida. Por eso, *AccessDataBase* recibe estas peticiones, inicializa *LanguageCustomManager* con el idioma pertinente y pregunta por la palabra clave en cuestión.

iii. Controladores

Los controladores del juego tienen la función de gestionar cada uno de los distintos ámbitos (menús o mazmorra) e incluso generar el bucle principal del juego.

Respecto a la zona de menús se pueden encontrar los siguientes controladores:

- **VillageController**
  - Gestionar la opción seleccionada en la Villa.
  - Gestionar las opciones de configuración.
  - Mostrar, si es necesario, el menú de selección de nivel.
  - Inicializa las clases *Singleton* de acceso a la base de datos y de música, que usarán el resto de escenas.

- **HomeController**
  - Gestionar la edición de mazos y tesoro.
  - Gestionar las recompensas obtenidas.
  - Mostrar la escena *WardrobeController* para el equipamiento y estado del jugador.
  - Llamar al controlador de los tutoriales (*TutorialController*).

- **StoreController**
  - Gestionar la opción de compra y venta de cartas.

- **TavernController**
  - Gestionar los diálogos con personajes.
  - Gestionar las misiones y sus recompensas.

Respecto a la exploración de mazmorra y cartas se pueden encontrar los siguientes controladores:

- **GameDungeonController**
  - Gestionar el bucle principal del juego durante la exploración.
  - Gestionar movimiento y batalla (con ayuda de *MovementController* y *BattleController*)
  - Gestionar cartas usando los controladores de la mano de cartas, de los mazos y de las cartas que estén en juego.
  - Gestionar Game Over y sus razones: sin cartas o sin vida.
  - Añadir recompensas al terminar una batalla.
  - Gestionar la salida de la mazmorra.

- **Cards**
  En este punto agrupamos todas las funciones asignadas a las propias cartas, pilar principal del juego:

  - Recoger el *prefab* de la carta y le asigna todos sus valores, imágenes y descripción asociada.
  - Devolver información asociada a las dos barajas distintas del juego, inicializarlas y barajarlas cuando sea preciso (mediante *DecksController*).
  - Gestionar la mano actual de cartas, descartando y robando las cartas que sean necesarias, e incluso bloqueando el acceso a la mano de cartas si se está realizando alguna acción como moverse o atacar (mediante HandZoneController).
  - Permitir el *Drag and Drop* de cartas mediante la clase *CardDraggable*, y devolviendo la zona en la que ha sido soltada mediante otros controladores secundarios (*DropCardsController*) asignados a zonas específicas del juego, como por ejemplo en la tienda o en la edición de cartas.
  - *CardDraggable* también permite mantener pulsada en una carta para obtener información sobre ella, llamando a *HelpCardInfo.*

- **OnBoardController**
  - Controlador asociado al panel central del juego, se ocupa de ejecutar una única carta cuando es preciso(por ejemplo, movimiento), o todas las cartas cuando sea necesario (por ejemplo, en combate). Este controlador utiliza también *MovementController* y *BattleController* según el caso.

Existen también controladores principales comunes en todo el juego y usados con un patrón de diseño en específico para facilitar su acceso:

- **AccessDataBase**
  - *Singleton* que gestiona la base de datos y accede a las traducciones entre idiomas, explicado en el apartado anterior. Recibe peticiones para obtener información sobre listas de cartas, configuración del jugador u otros.

- **AudioCustomManager**
  - *Singleton* que tiene todas las referencias de los sonidos o temas musicales que debe sonar en cada ocasión. Recibe peticiones para reproducir un archivo de sonido.

Todos los controladores principales anteriores llaman a un conjunto de clases secundarias que gestionan otros elementos, como actualizar la UI o animaciones de los enemigos.

iv. Estructura y bucle de juego

Relacionado con los controladores descritos anteriormente, se ha orientado el bucle de juego de dos maneras, si se está en una pantalla de interfaz (menú principal, casa, tienda, taberna) o si se está en una exploración (mazmorra). También es muy importante entender la estructura interna de la carta, pues es un elemento vital del juego ya que contiene mucha información que se utiliza constantemente.

a. Estructura interna de la carta

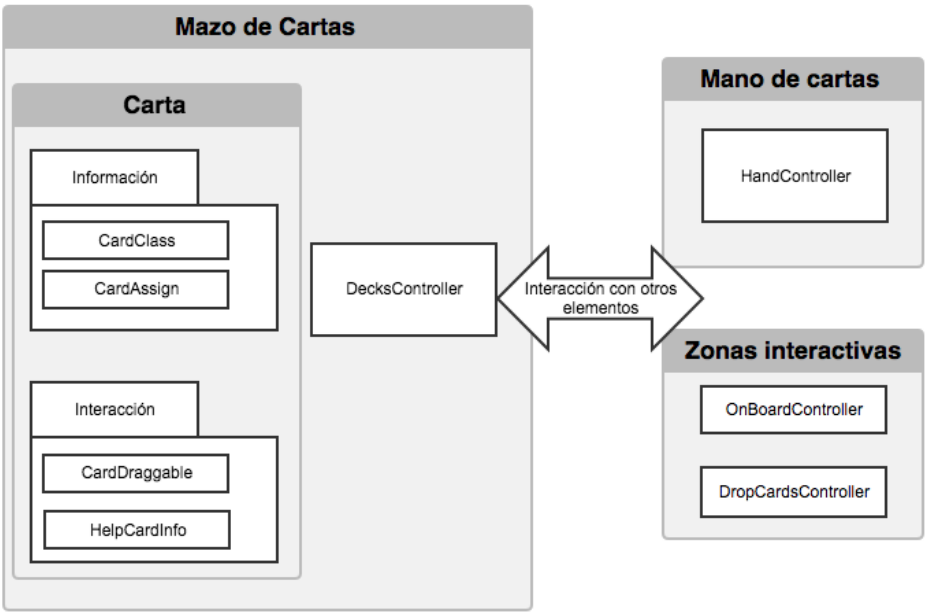


Fig. 31 Estructura de una carta e interacciones

Para la interacción y movimiento de las cartas se utilizan métodos e interfaces propias de *Unity 3D* para facilitar el trabajo: *IBeginDragHandler*, *IEndDragHandler.*

Las zonas interactivas y la mano contienen controladores que usan interfaces *IDropHandler* para detectar cuando un objeto es lanzado en ello y llevar a cabo distintas acciones. Un ejemplo de ello es la edición del mazo de cartas, donde se puede mover una carta del almacén al mazo y viceversa. Ambas zonas tienen asociadas un controlador con *IDropHandler* que gestionará donde se encuentra la carta.

Otro ejemplo interesante se puede encontrar en la edición de equipo, en *WardrobeScene*. La carta de equipamiento se puede soltar en una zona de equipo que representa el cuerpo (casco, guantes, cuerpo, cinturón, botas, accesorio), pero si no es del tipo asociado, la carta vuelve al almacén de equipo. Dicho de otra forma, el controlador detecta que la carta arrastrada no pertenece a esa ranura y la devuelve, pues por ejemplo no se puede equipar un casco a la ranura de botas.

b. Pantallas de menús: bucle de entrada de datos

En los menús generales del juego (ya puede ser en la Villa, la Casa, la Tienda, …) las escenas se limitan a esperar el *input* del jugador para realizar las acciones: pulsar un botón o arrastrar una carta a una zona determinada.

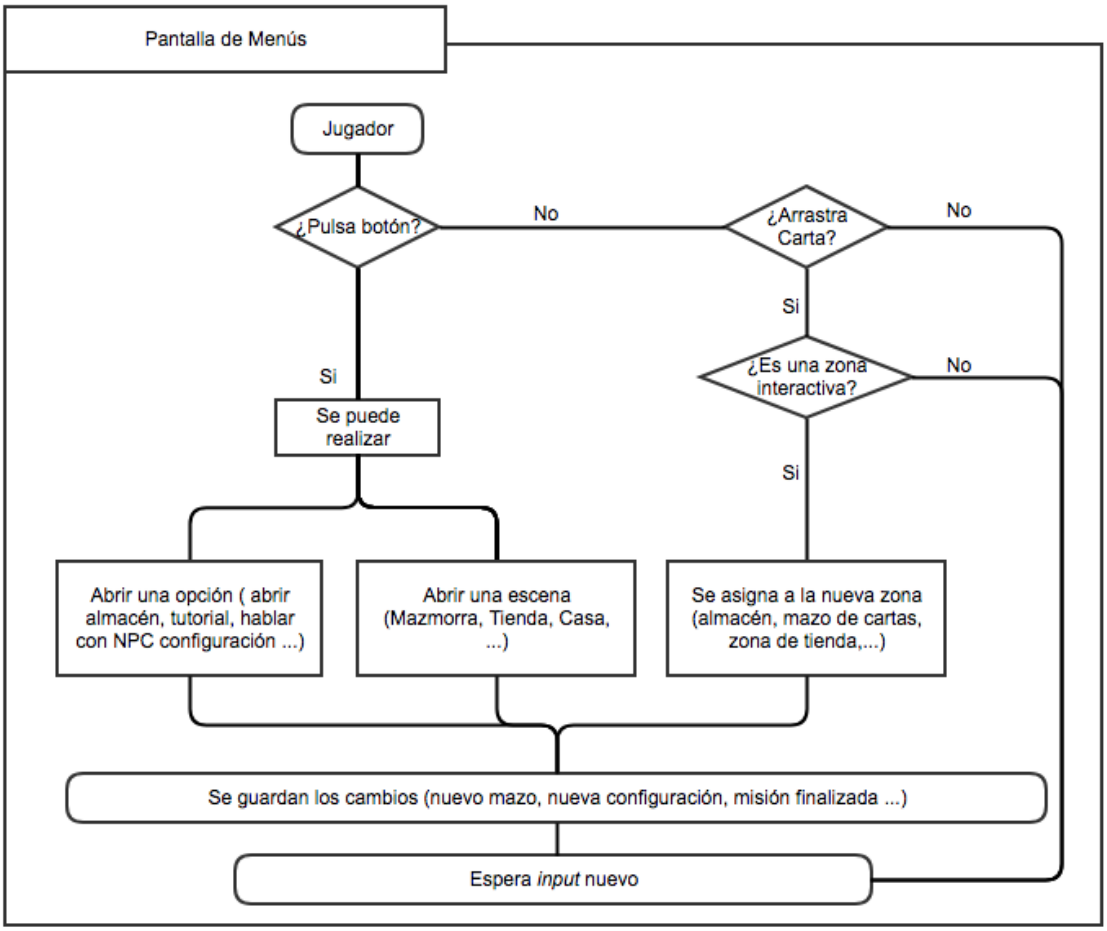


Fig. 32 Bucle de pantalla con menús

Estos cambios suelen almacenarse o de manera inmediata (por ejemplo, al terminar una misión) o a la espera que el jugador pulse el botón de *aceptar* o *cancelar.*

El bucle de juego en esta ocasión se repite siempre que se pulse un botón o se interactúe con algún otro elemento de la interfaz.

c. Exploración

Aunque en el caso de la exploración también se utilicen elementos de la interfaz cuyo bucle se ha explicado en el apartado anterior, como por ejemplo usar cartas o salir de la mazmorra, la exploración tiene un bucle propio dentro del juego. Este se puede encontrar en *GameDungeonController*, y se repite constantemente en cada *frame*.

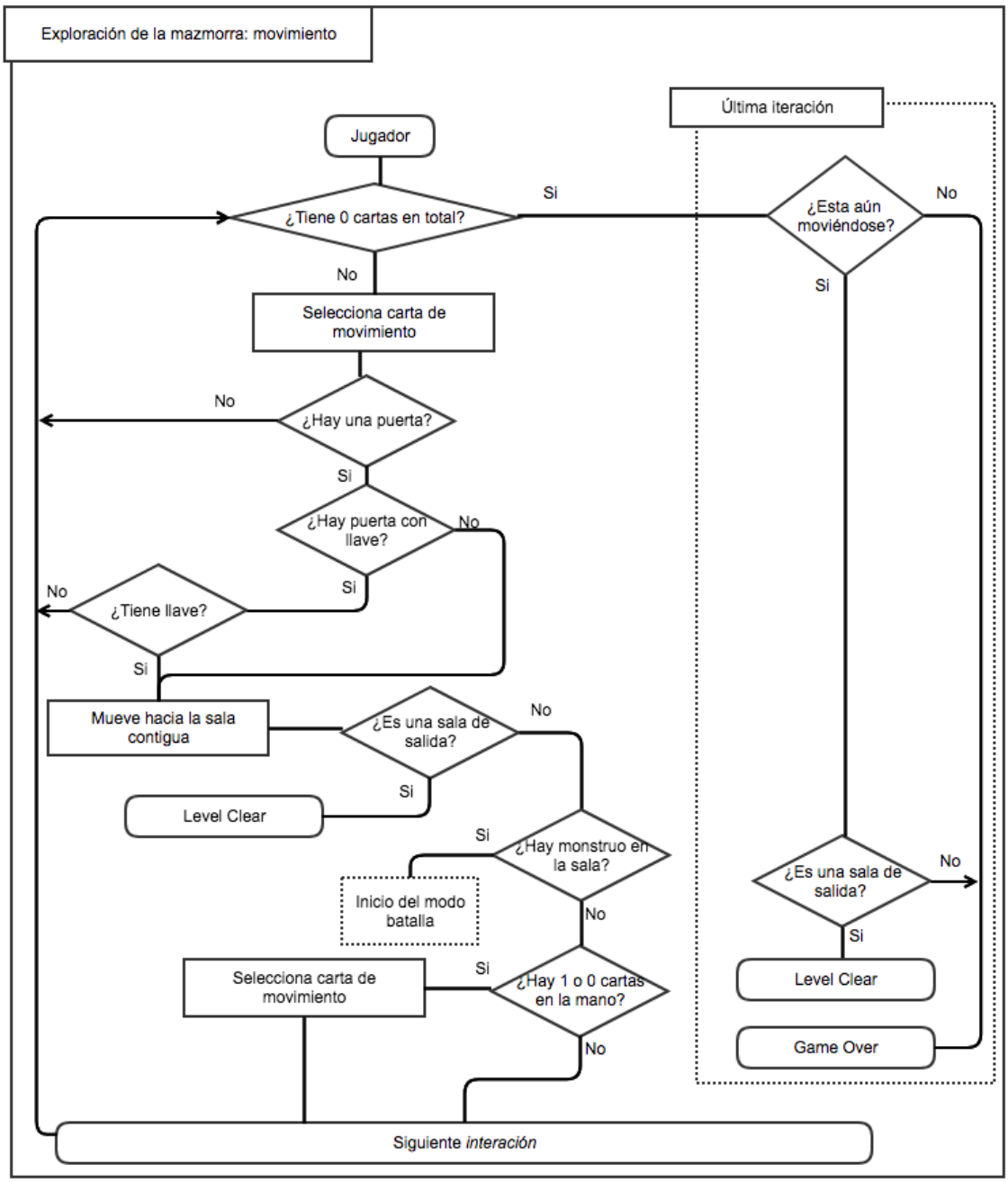


Fig. 33 Bucle de juego principal durante el movimiento

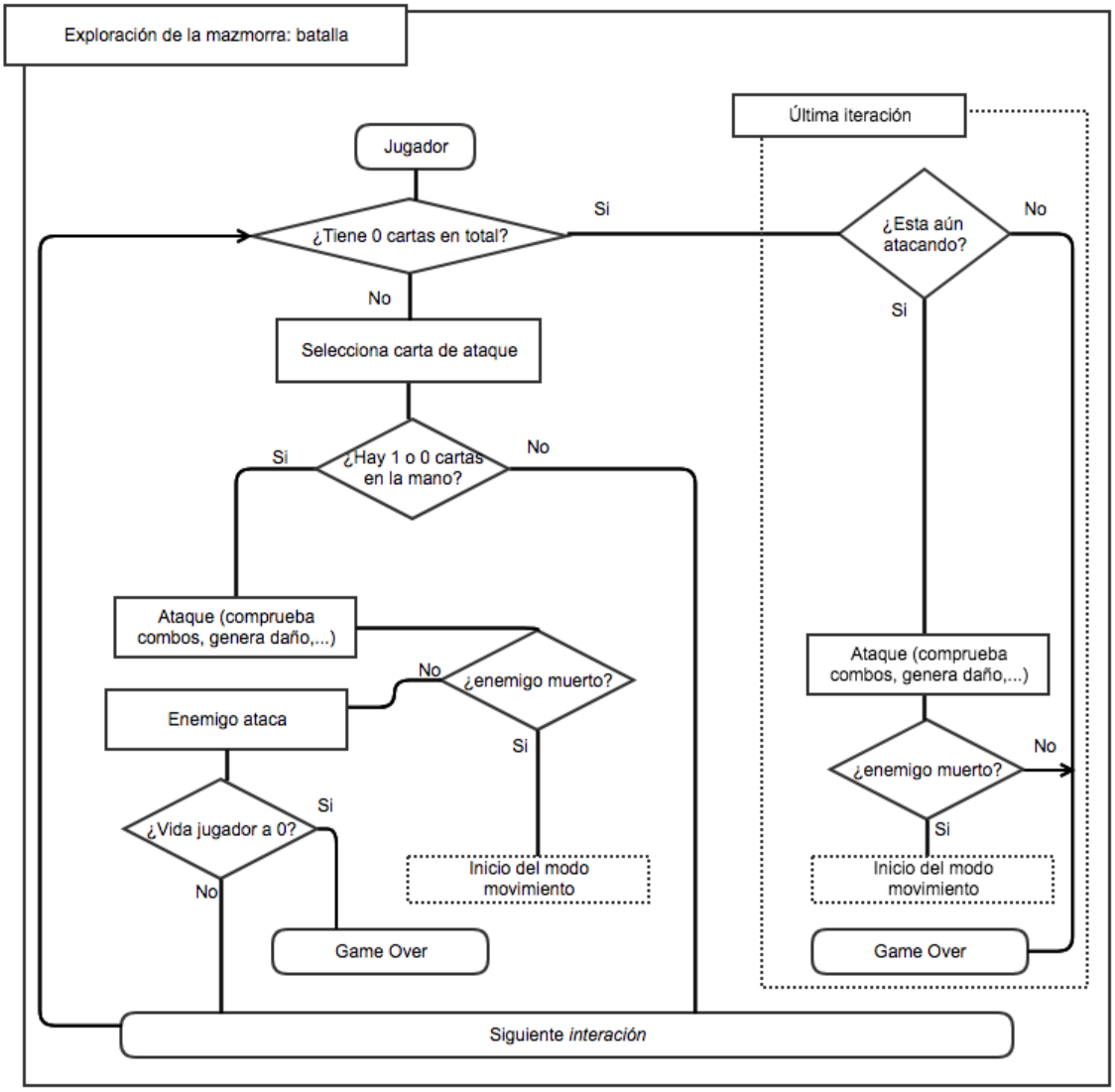


Fig. 34 Bucle de juego principal durante la batalla

La exploración siempre empieza en modo movimiento, pero durante el juego se cambia de modo de manera constante. Separar el bucle en dos clarifica y hace más comprensible las comprobaciones que el código escrito en *GameDungeonController* lleva a cabo. Además, en la **Selección de carta de movimiento/ataque** entra en juego los algoritmos definidos en la ***figura 32: bucle de pantalla con menús*****,** pues se arrastra cartas a la zona de juego, de descartes y se pueden pulsar varios botones.

Del bucle principal cabe destacar que tanto en movimiento como en batalla se debe tener en cuenta la última carta o mano jugada, pues se puede dar el caso que con la última carta el jugador llegue a la meta o acabe con el monstruo.

v. Interfaz

La interfaz ha sido diseñada para dispositivo móvil en formato horizontal (*landscape*) con una resolución fija de 1280x720 **[1]** (en formato horizontal), aunque también se ha comprobado que el escalado a 1920x1080, a 2160x1080 y a 2560x1440 funcionen correctamente. En la siguiente figura podemos ver que, en una pantalla de móvil, la zona coloreada de verde tiene fácil acceso, y la zona coloreada de amarillo cuesta más de llegar o ver. **[3]**

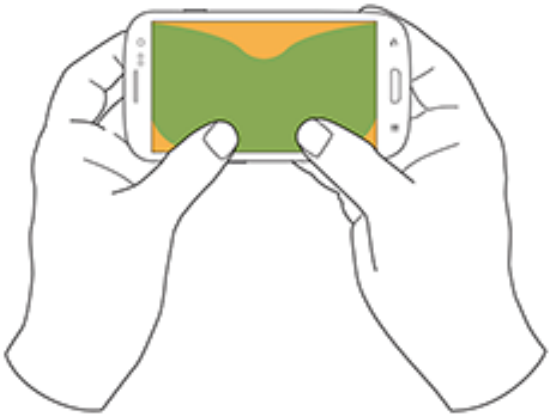


Fig. 35 Zonas de fácil acceso con los pulgares

Aun así, hay que tener en cuenta que la misma zona verde (y bordes) tienen mayor facilidad en ser tocados sin querer y activar alguna de las funciones que ahí se encuentren. Por eso la mayoría de las funciones no reversibles deben requerir una segunda confirmación por parte del usuario, ya que hay muchas pantallas con muchas opciones e información.

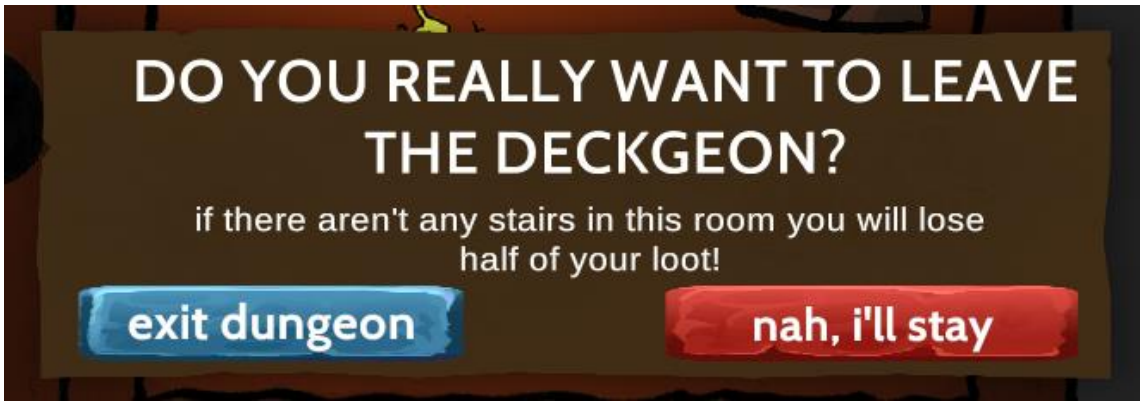


Fig. 36 Mensaje de confirmación

Por ejemplo, para salir de la mazmorra es requerido que se pulse Salir, y después que se indique que efectivamente se desea salir.

Para borrar la partida, en la configuración del juego, requiere de una segunda confirmación pues se va a reiniciar todo el juego y los datos salvados.

Para comprar y vender cartas, no basta con arrastrar la carta que se desee comprar o vender, también hay que pulsar el botón de comprar o de vender.

Estos elementos que añaden una doble confirmación a la acción que se quiere realizar evitan errores irremediables.

4.5. IA del juego

A pesar de que no existen algoritmos complejos en el juego, si se han añadido ciertos comportamientos que pueden simular "decisiones" sobre los personajes o mecánicas para compensar al jugador.

i. NPCs

Los personajes del juego disponen de un conjunto de frases de manera aleatoria que se gestionan mediante los controladores de diálogos**[8]** y almacenadas en su correspondiente base de datos. También pueden disponer de misiones (o *quests)* con una posible recompensa asociada.

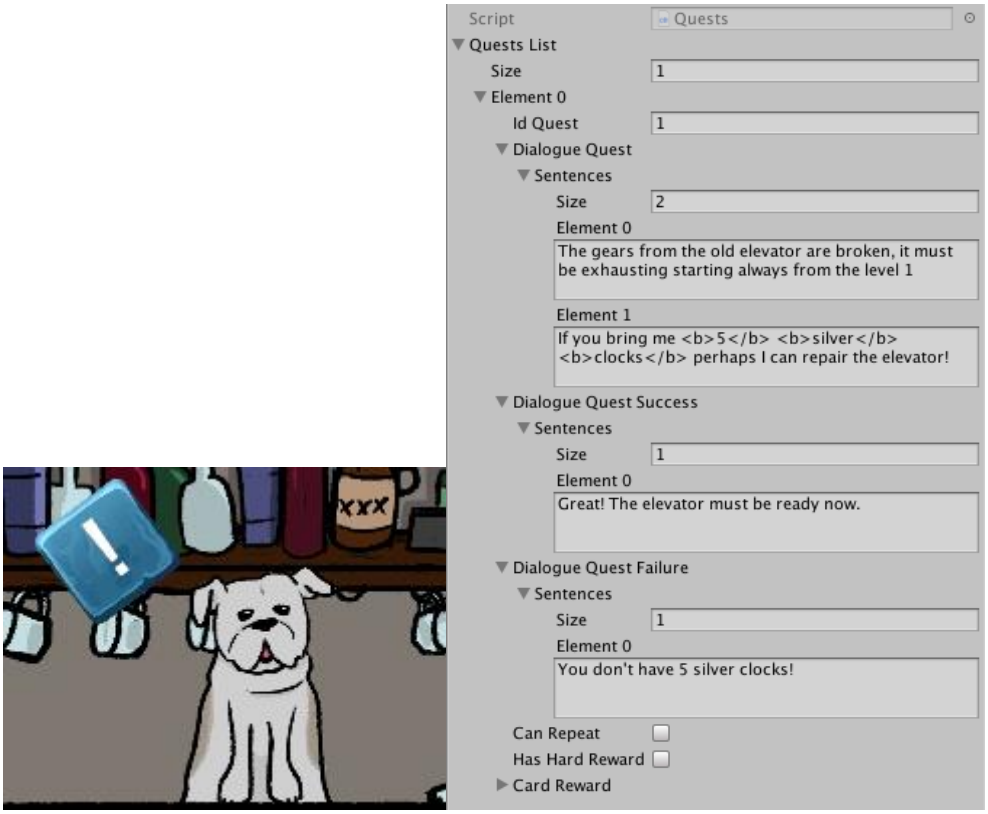


Fig. 37 Botón de inicio de misiones (izquierda) y su estructura interna (derecha)

El personaje dispone de un botón para terminar la *quest* en el propio diálogo. Si se cumplen los requisitos, el jugador recibe la recompensa. Si no se cumplen, el personaje ofrece una explicación de porqué no se cumplen.

Fig. 38 Finalizar la misión (quest)

Estas frases han sido suplantadas por códigos que hacen referencia a los textos en *XML* para su correspondiente traducción, pero que se leen mediante el mismo fichero de *Quests* como se ha mostrado anteriormente.

ii. Enemigos

Los enemigos disponen de una lista de ataques y un % de probabilidades de ser realizados, siendo una probabilidad alta los ataques normales y una probabilidad alta los ataques más fuertes.

Una buena idea para mejorar este sistema sería la implantación de decisiones con un conjunto de reglas simples, como por ejemplo:

- Si tiene menos de X vida, que realice una curación.
- Si tiene menos de Y vida, realiza ataques más poderosos a la desesperada.
- Si se acerca demasiado a 0 de vida, huye, o invoca a un nuevo enemigo.

Estas reglas se tendrán en consideración para una futura mejora.

iii. Sistema de *drop*: suerte

Los enemigos disponen de una lista objetos y un % de probabilidades de ser conseguidos. Aun así, este valor depende de la característica “suerte” del personaje y de los objetos que lleva encima el jugador (y que, por lo tanto, puede perder si muere o sale de la mazmorra).

Por ejemplo, en la figura siguiente, a las probabilidades de encontrar objetos se le añadiría un 10%.

Fig. 39 Mochila y suerte

Además, cuantos más niveles se hagan seguidos, la probabilidad de conseguir objetos aumenta. Este hecho no se le explica al jugador en ninguna pantalla o tutorial, pero éste tendrá mayor sensación de recompensa y de conseguir objetos cuanto más juegue.

Sin embargo, la suerte tiene un límite superior de 99%, de modo que siempre existe la posibilidad de no recibir nada por muchas mejoras que se hayan activado.

5. Diseño de niveles

5.1. Villa

La **Villa** hace de función de menú principal del juego. Permite navegar a todas las opciones del juego.

Fig. 40 Villa: menú principal

A la izquierda superior se puede acceder al **Menú de Configuración**, con todas sus opciones de volumen, borrar progreso (¿de verdad deseas eliminar tu partida salvada?), seleccionar idioma, ver créditos y salir de la aplicación.

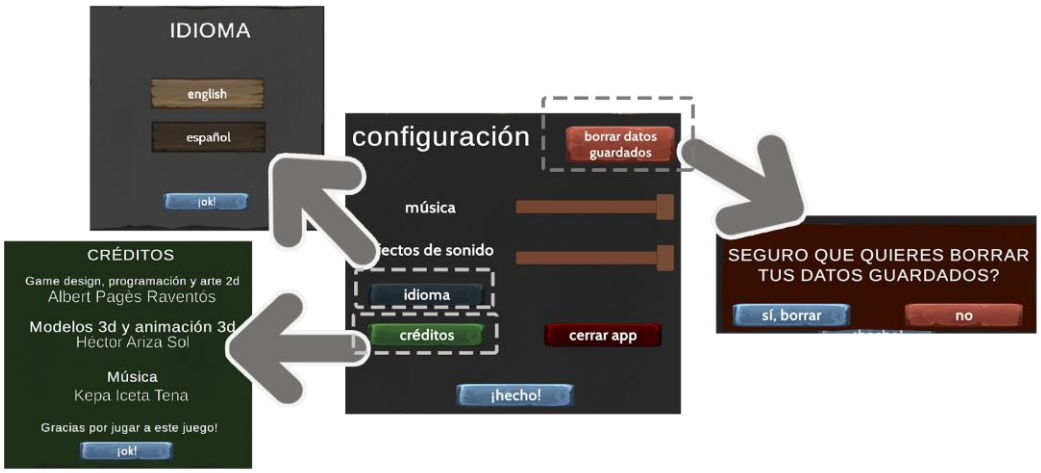


Fig. 41 Configuración del juego

La **Casa** hace de base para el jugador, permitiendo revisar el tutorial, personalizar los mazos que se usarán durante la aventura y ver el botín y tesoro conseguido.

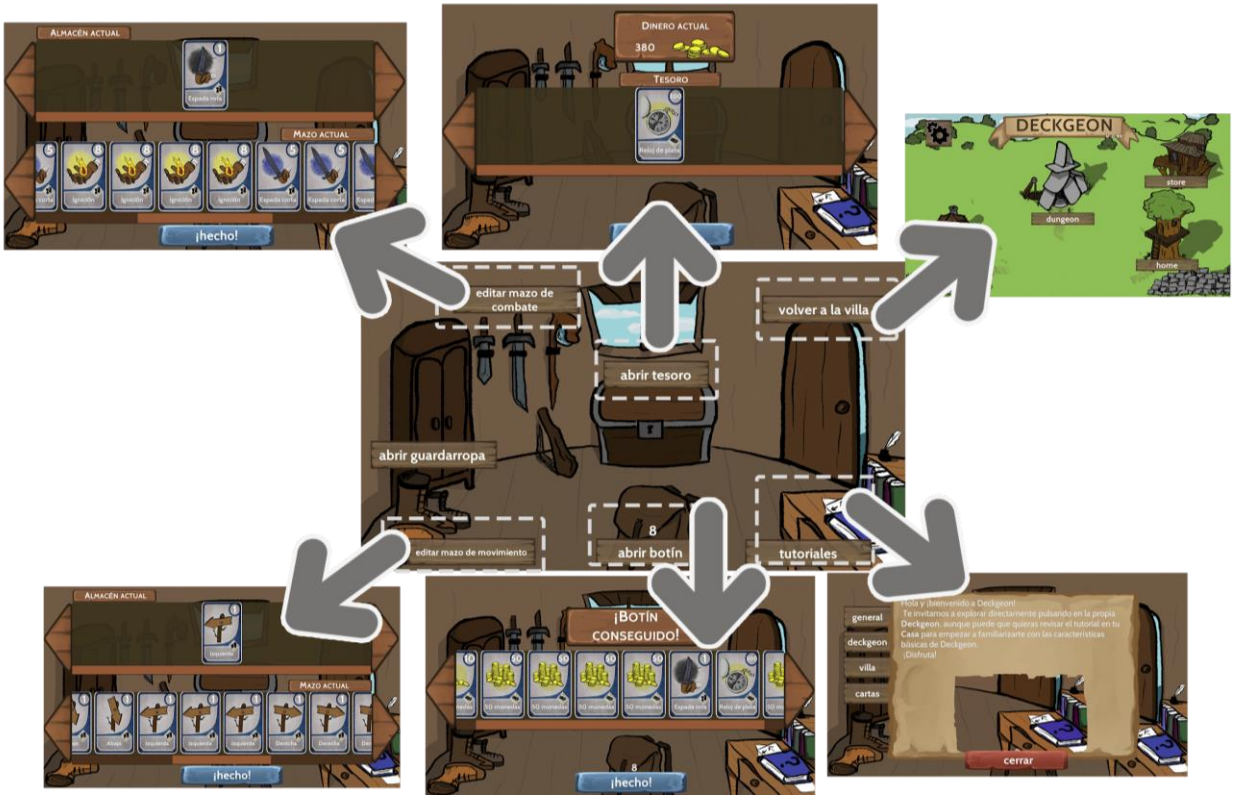


Fig. 42 Distribución de la casa

Además, permite acceder al **Guardarropa**, donde se pueden consultar las características del jugador y equipar cartas de equipo para mejorarlos.

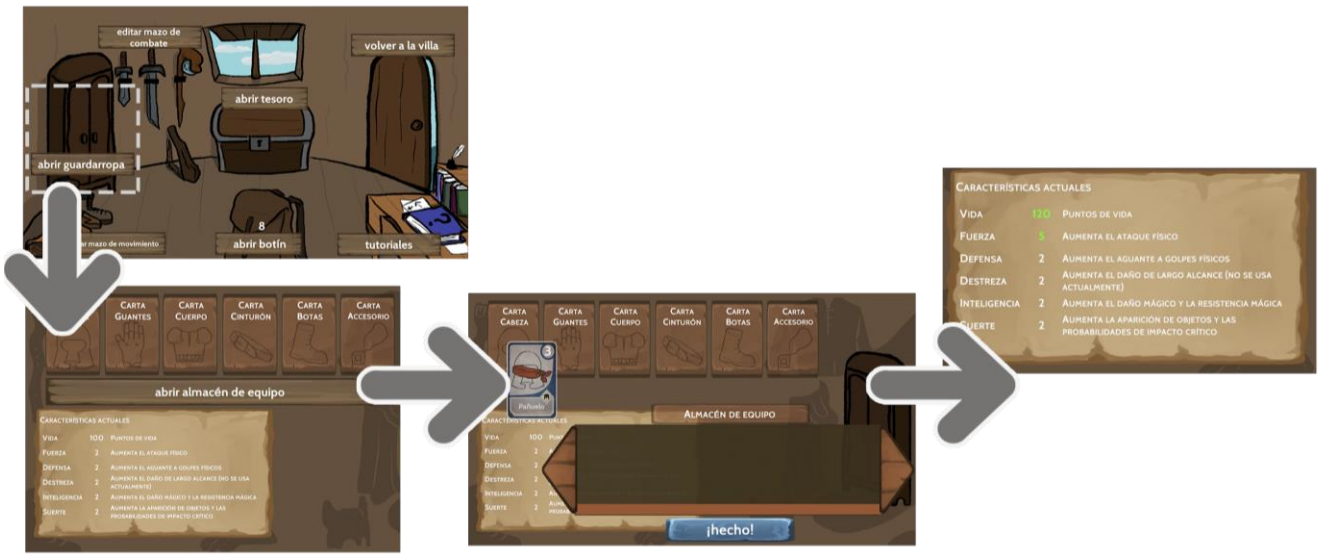


Fig. 43 Distribución del Guardarropa

La **Tienda** permite comprar y vender cartas, seleccionado la opción correspondiente. Las cartas compradas se almacenan en la Casa, por lo que se deben equipar. Las cartas vendibles son del almacén de cartas, pues representa que están en desuso.

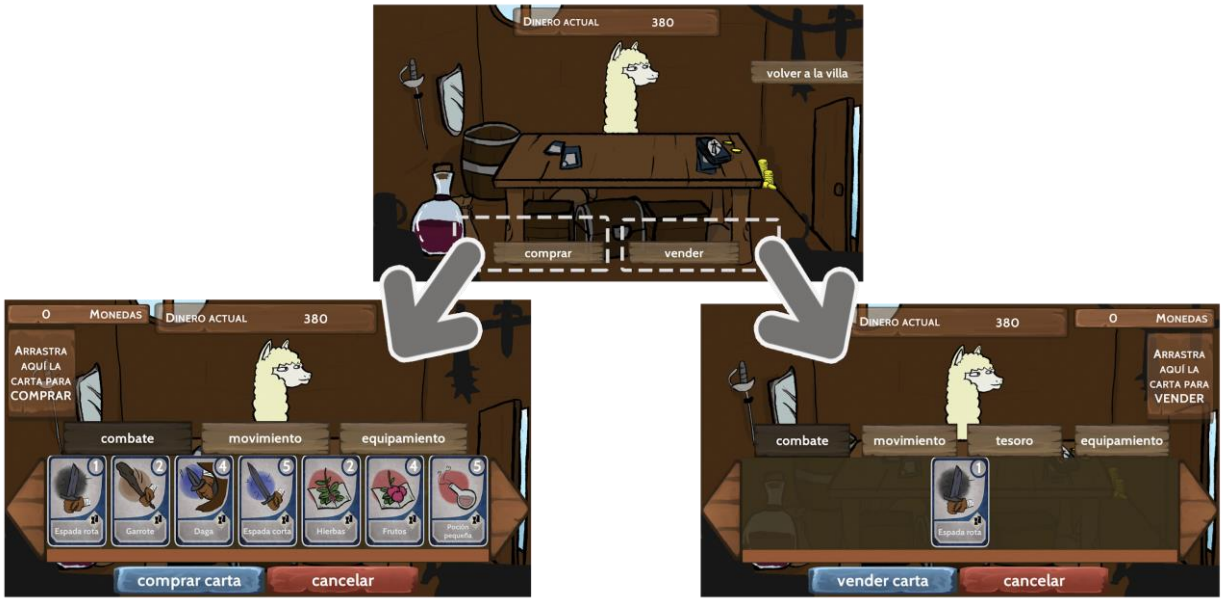


Fig. 44 Distribución de la Tienda

En la **Taberna** se puede hablar con los personajes, iniciar y finalizar misiones.

Fig. 45 Distribución de la Taberna

Para salir de todas las diferentes casas de la Villa, el botón siempre se encuentra en el lado derecho de la pantalla, al lado de la **puerta**.

Llegado a un punto del juego se tendrá acceso al ascensor, que permite seleccionar el nivel por el cual empezar

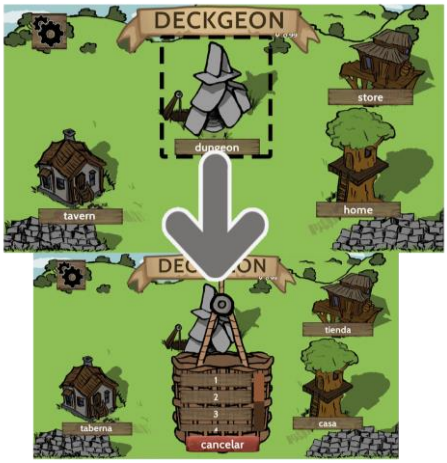


Fig. 46 Ascensor en la Villa

5.2. Mazmorra

La **Mazmorra** dispone de dos fases, el movimiento y el combate. Durante el movimiento se debe soltar una carta de movimiento en el centro para moverse hacia una puerta de la sala.

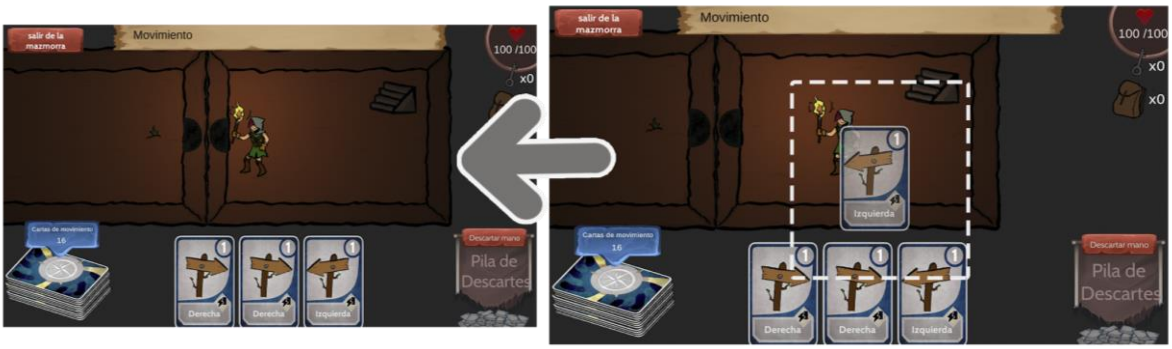


Fig. 47 Fase de movimiento

Las mazmorras suelen tener un diseño de 3x3 habitaciones de base, aproximadamente. Pero pueden crecer a medida que se vaya avanzando.

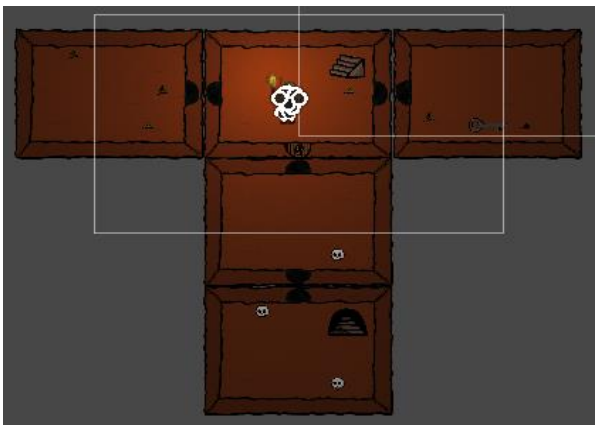


Fig. 48 Diseño de mazmorra inicial

Durante el combate se deben realizar ataques al enemigo. Para ello en muchas situaciones deben hacerse combinaciones de cartas del mismo tipo para hacer el suficiente daño superando así la defensa de los monstruos.

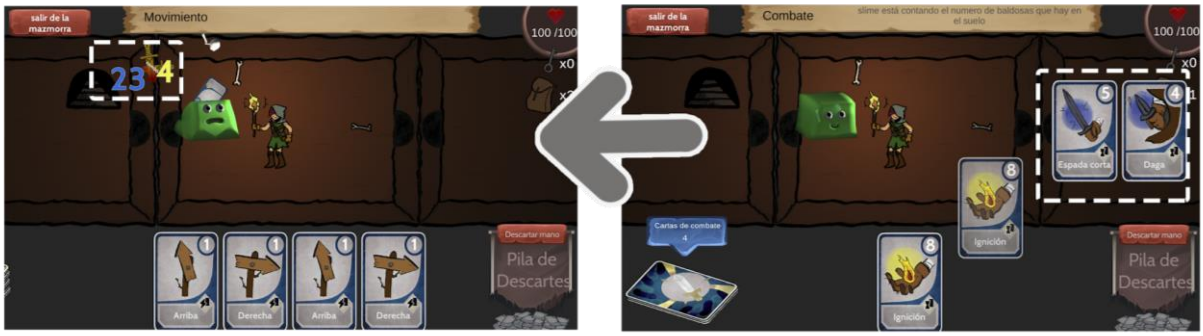


Fig. 49 Fase de Combate

Si se llega a una sala con escaleras de salida se supera el nivel, y el personaje seguirá bajando.

Fig. 50 Fase de Movimiento: escaleras y nuevo nivel

Se puede salir siempre que se desee, incluso en medio de un combate, sufriendo alguna penalización.

Fig. 51 Salir de la mazmorra

Para más información, consulten la sección de tutorial.

6. Pruebas con usuarios

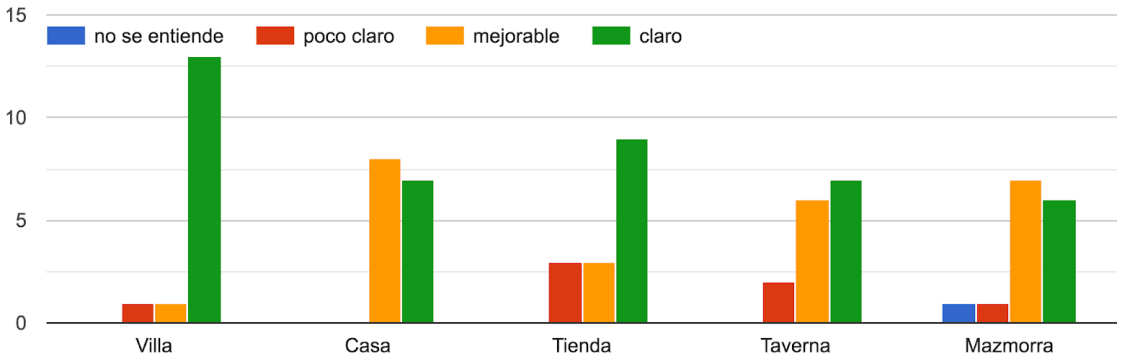
El juego es subido a ***Google Play*** en su versión v0.9 el día 15 de Junio de 2019, justo un día antes de la entrega de la *PEC03*. Se ha adjuntado un pequeño cuestionario *on-line* que se incluye en el ***Anexo A***.

https://forms.gle/zPwidvnLMygXuoRD9

Mediante el *feedback* de los usuarios se pueden extraer un conjunto de conclusiones interesantes, que se analizarán a continuación.

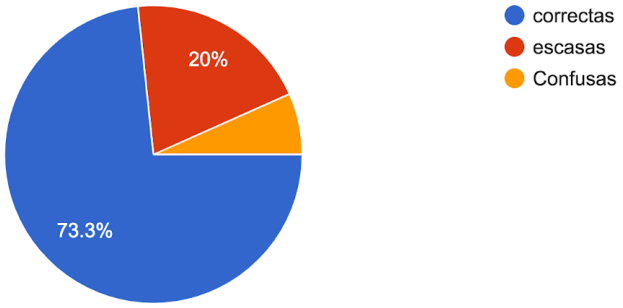
6.1. Villa

Por favor, puntua las pantallas de mayor a menor claridad:



Las opciones que me proponen me parecen...

15 responses



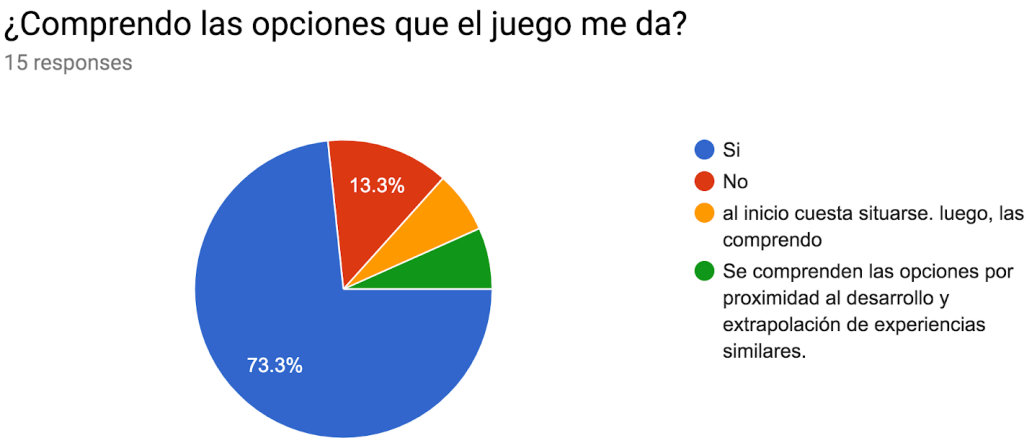
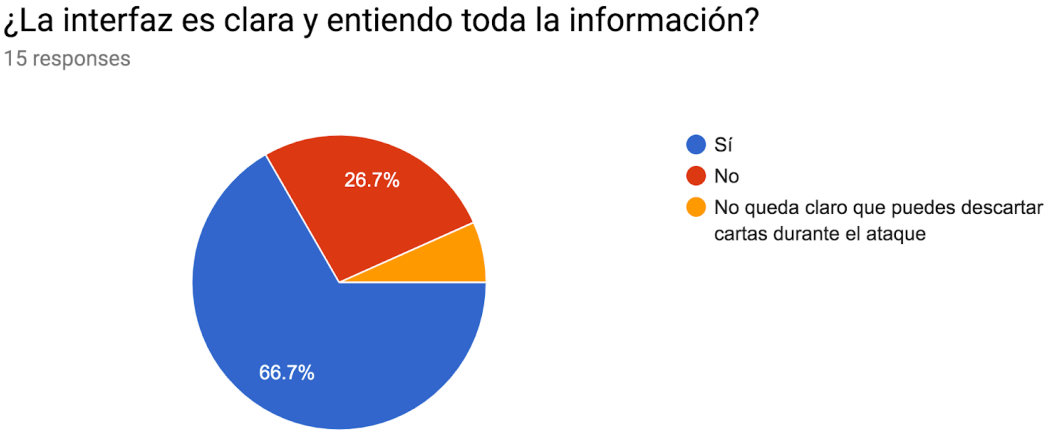


Fig. 52 Resultados de la encuesta de la Villa

Vistos los resultados, el diseño de la tienda y la taberna requieren una mejora tanto a nivel visual como de explicación de sus opciones. La mazmorra parece tener baja nota, pero es debido a la frustración de no entender como se juega y más adelante, en esta misma entrega, se intentará solventar

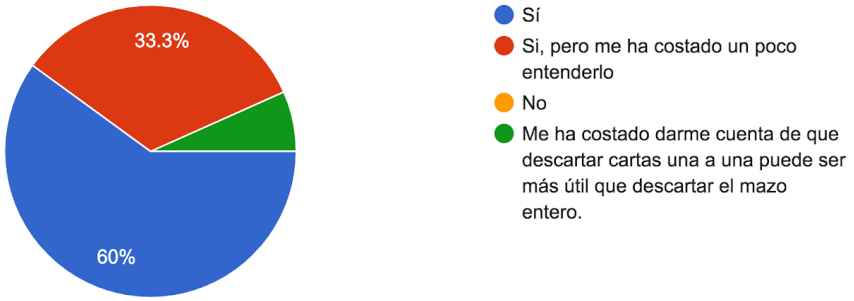
Aun así, la mayoría de los usuarios opinan que las opciones que da la Villa en general son comprensibles y adecuadas.

6.2. Mazmorra y mecánicas



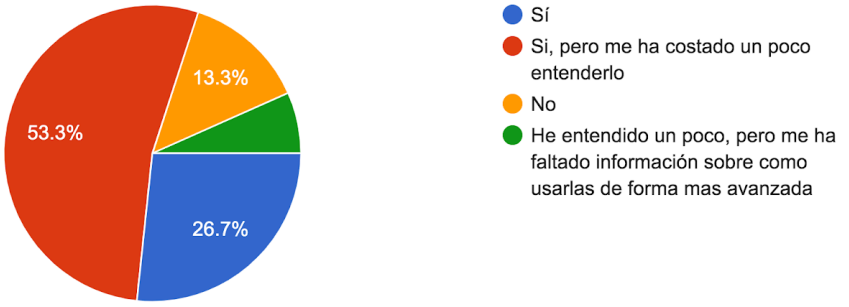
¿He entendido la mecánica de las cartas de movimiento?

15 responses



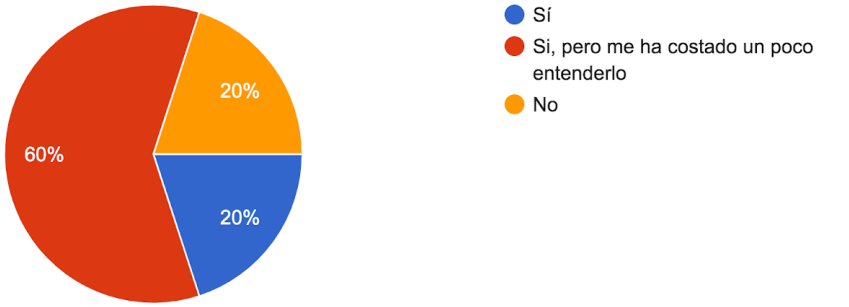
¿He entendido la mecánica de las cartas de batalla?

15 responses



¿He entendido la mecánica de los combos entre cartas?

15 responses



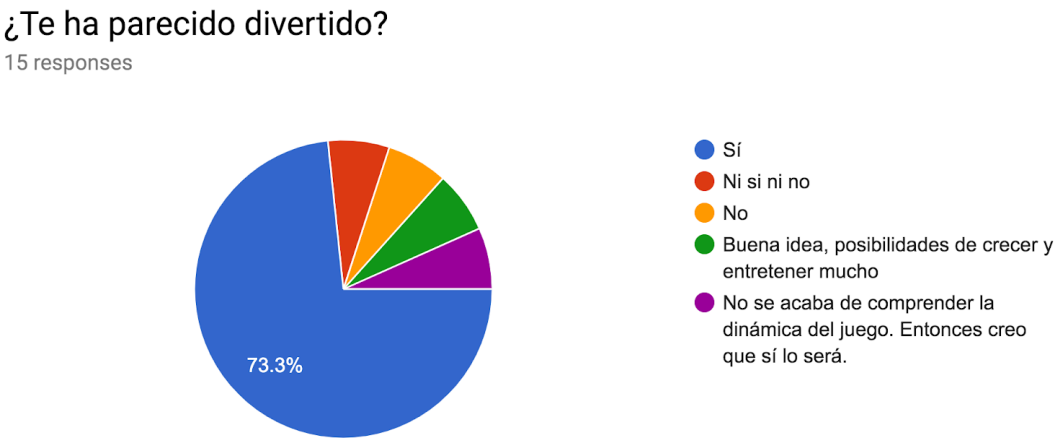
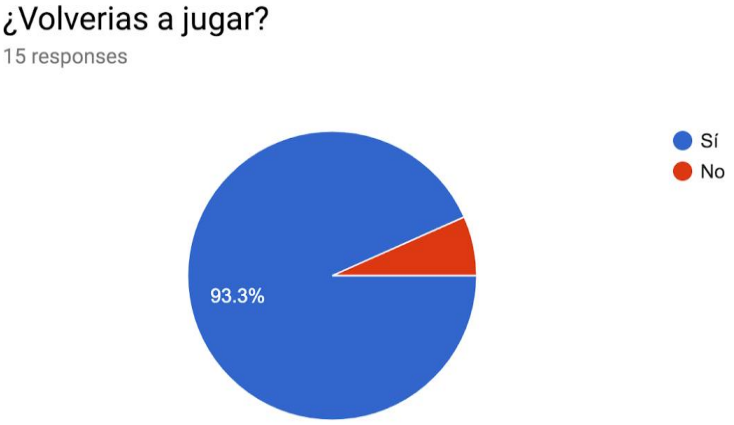


Fig. 53 Resultados de la encuesta de la Mazmorra

La mecánica de movimiento parece casi obvia, lo que es una buena noticia, pues la interfaz orientada a dispositivo móvil resulta entendible.

Sin embargo, la compresión de batalla y combos es crítica, siendo una de las características principales del juego no parece muy entendible. Aun así, la conclusión general es que parece divertido por lo que el proyecto resulta prometedor.

6.3. General



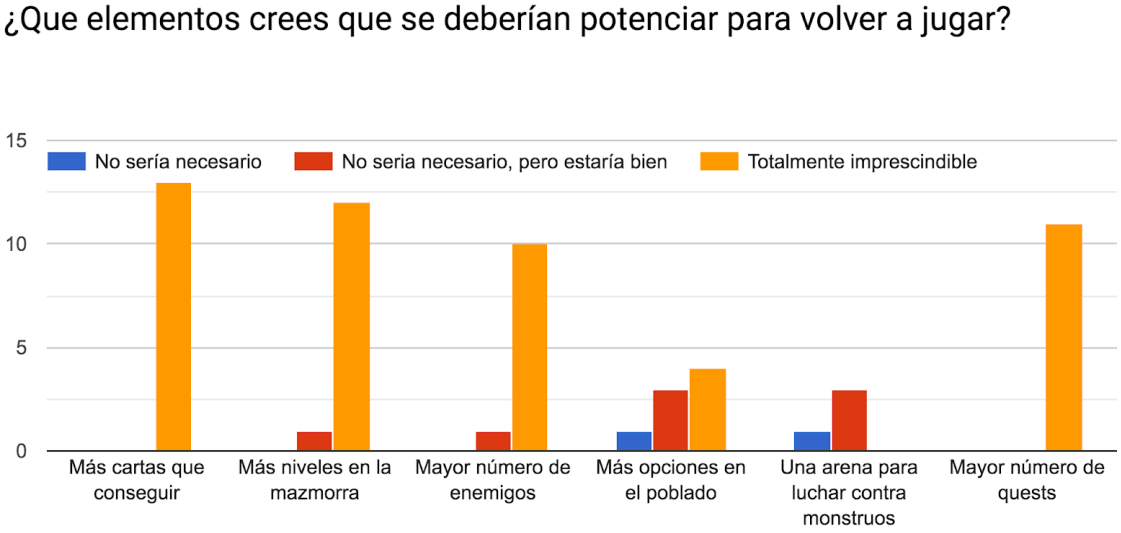


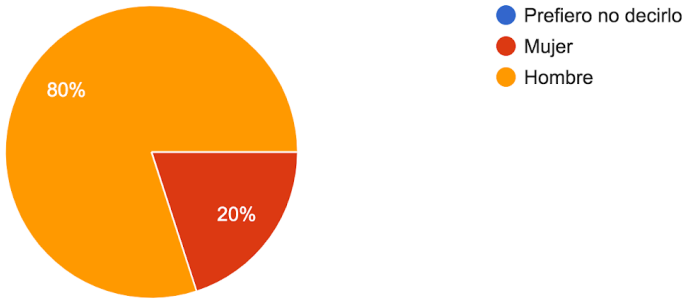
Fig. 54 Resultados de la encuesta del juego y precio

Los jugadores están dispuestos a volver a jugar, sin embargo el precio de pago único por la aplicación es bastante dispar, sobretodo por ser jugadores no habituados a jugar en dispositivos móviles. Lo que resulta un buen inicio, pues la idea de acercar el juego móvil a jugadores habituales y menos casuales parece dar sus frutos.

6.4. Sobre el jugador

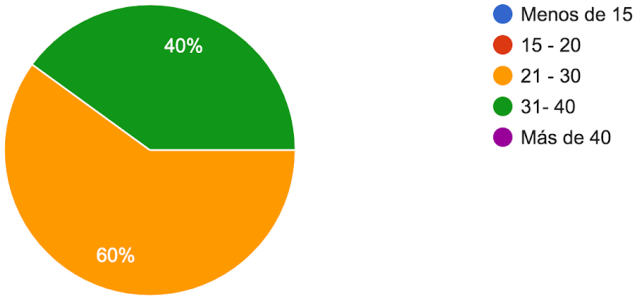
Sexo

15 responses



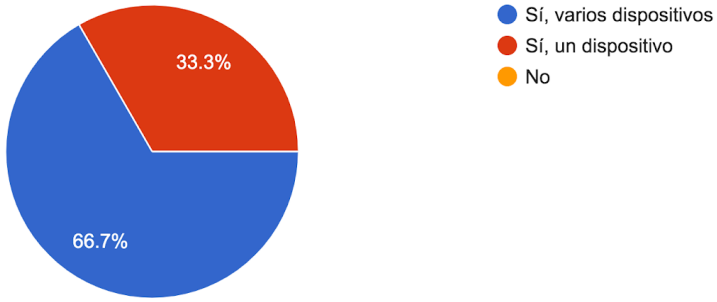
Edad

15 responses



¿Tienes consola o PC en casa?

15 responses



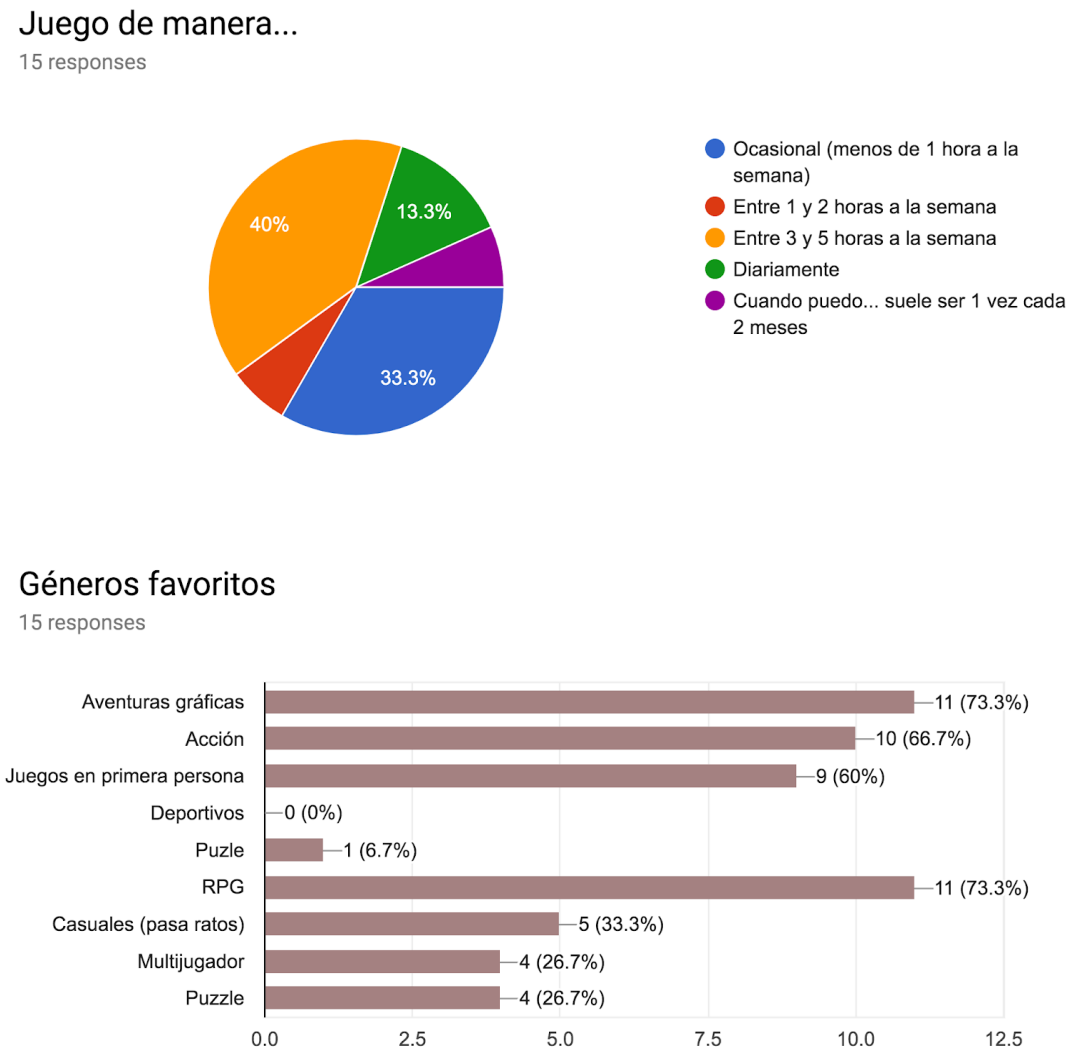


Fig. 55 Resultados de la encuesta sobre el jugador

Los jugadores tienen perfiles muy dispares, pero el grupo principal hombres de entre 21 y 40 años, con varios dispositivos en casa. También se pueden encontrar una gran variedad de géneros favoritos, donde destacan Aventuras, Acción y RPG.

6.5. Conclusiones y *feedback*

En general, los jugadores que sí han entendido (o acabado entendiendo) las mecánicas más propias del juego cumplen todos el mismo perfil:

- De entre 21 y 40 años.
- Jugadores de más de 1 hora semanal.
- Jugadores con más de 4 géneros de videojuegos favoritos.

- Jugadores con género de Aventuras, Rol y Puzle como mínimo.

De todas las conclusiones, quizá la más vital es que el juego no es intuitivo, pues muchos jugadores desconocen que se pueda descartar una o más cartas por separado (y no toda la mano). El sistema de combos y el *feedback* recibido por pantalla tampoco queda claro pues han habido preguntas y afirmaciones del tipo:

- *¿Porque sale 0 si en la carta ponía un valor distinto?*
- *No entiendo los combos o la batalla (aunque si el movimiento).*
- *En la última mano no me queda claro si debo o puedo usar todas las cartas, ¿es un error?*

Este hecho enseguida choca con la diversión del jugador, sintiéndose frustrado por no entender qué está pasando en la pantalla de juego.

Es por ello que se han tomado varias medidas para la versión final del juego:

- La primera vez que se visita la mazmorra se obliga el visionado del tutorial. No es óptimo pero es un parche aplicado para esta versión.

- El idioma de las explicaciones (y del juego) estaba en inglés por defecto, lo que puede echar para atrás a leerse el tutorial. Se ha llevado a cabo la traducción del juego entero.

- Si se mantiene pulsada una carta (sea donde sea del juego: mazmorra, tienda, casa, ….) se pueden ver las características de esta carta, lo que proporciona más información que quizá clarifique algunas dudas.

Fig. 56 Ayuda sobre las cartas

- Se ha introducido un video en lugar de una imagen estática en alguno de los apartados del tutorial, sobretodo en el caso de la batalla, para

remarcar las acciones se pueden hacer con las cartas.

- Cuando se consiguen objetos se ve una pequeña animación de una carta saliendo del enemigo. Esto se ha añadido en las nuevas versiones porque muchos jugadores no eran conscientes de haber recibido objetos.

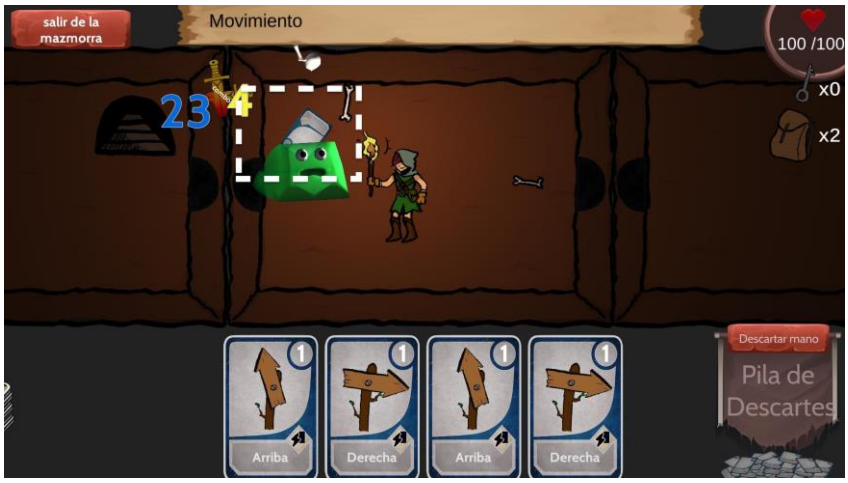


Fig. 57 El enemigo suelta una carta

Se valora la posibilidad futura de introducir un tutorial más guiado, con un mazo predefinido sabiendo que saldrá en cada momento y guiando al jugador de las acciones que puede realizar, sustituyendo el tutorial al inicio del juego.

7. Manual de usuario

Para jugar al juego es necesario disponer de un móvil ***Android*** con sistema operativo 4.1 o superior.

A continuación se muestran los tutoriales incluidos en el juego.

7.1. Tutorial de *Deckgeon*

- Tu principal meta es explorar la Deckgeon, hasta su último piso. Para hacerlo posible, usaras una baraja de cartas personalizable, usando una mano de 4 cartas cada turno.

Fig. 58 Tutorial Deckgeon_1

- Las cartas se pueden activar arrastrándolas y soltándolas en el centro de la pantalla Si tu mano actual no es de tu agrado, es posible descartarlas de manera individual arrastrándolas a la pila de descartes o deshacerte de todas ellas a la vez si pulsas el botón de descartar mano.

Fig. 59 Tutorial Deckgeon_2

- Si tu mano se reduce a 1 o menos cartas, automáticamente se reparten nuevas cartas desde tu mazo. Si no quedan cartas en tu mazo, lo que tienes en la mano es lo que queda, así que haz uso de todo lo que puedas!

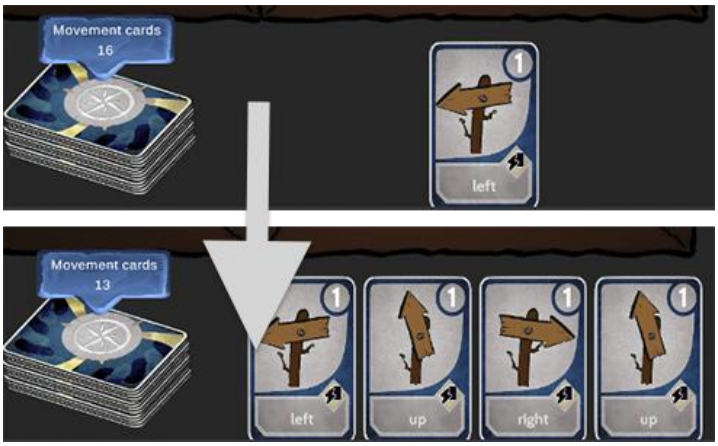


Fig. 60 Tutorial Deckgeon_3

- Durante la Fase de movimiento, usarás un único Mazo de movimiento por nivel, distinto al Mazo de combate. Las cartas de movimiento se usan de la misma forma, y se aplican instantáneamente cuando son soltadas en centro de la pantalla.

Fig. 61 Tutorial Deckgeon_4

- Durante el combate usarás tu Mazo de combate personalizado, que se reinicia después de cada pelea - ¡así que a por todas! Las cartas que se sueltan en el centro de la pantalla tendrán efecto al final de tu turno, después será el turno del enemigo y robarás nuevas cartas. Planifica bien la jugada descartando cartas no deseadas.

Fig. 62 Tutorial Deckgeon_5

- Durante el combate puedes realizar poderosos combinaciones de cartas con el mismo tipo y color de fondo. Sin embargo, si una carta con el fondo negro entra en juego, el combo no se realizará correctamente.

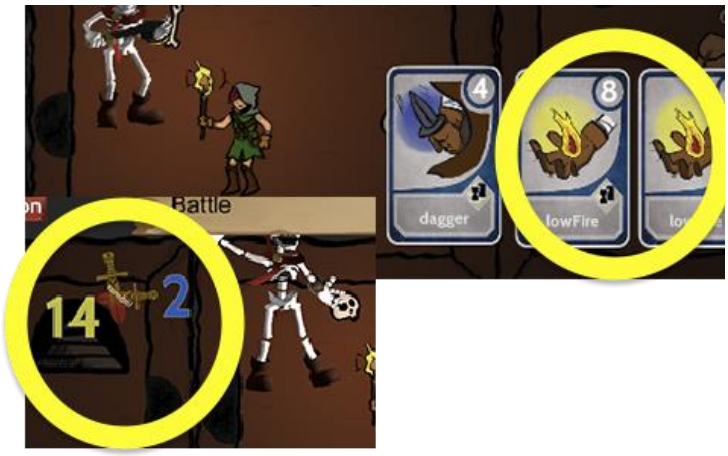


Fig. 63 Tutorial Deckgeon_6

- Si necesitas más información sobre cada carta, mantén pulsado encima de ella para ver los efectos y características que tendrá durante el juego.

Fig. 64 Tutorial Deckgeon_7

- En caso de derrota, ya sea por perder toda la vida o quedarte sin cartas en el mazo, la mayor parte del botín que lleves encima será arrebatado.

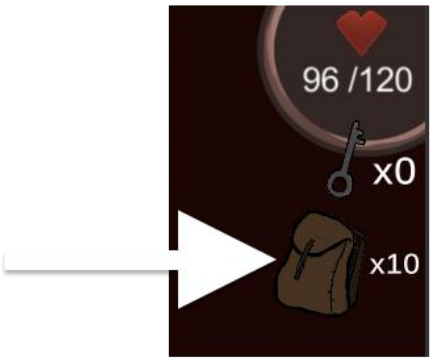


Fig. 65 Tutorial Deckgeon_8

- Puedes escapar de la Deckgeon siempre que quieras, pero si lo haces en cualquier habitación que no tenga escaleras te supondrá una pérdida de la mitad de tu botín. Juega sabiamente y escapa cuando sea adecuado.

Fig. 66 Tutorial Deckgeon_9

- El botín que ganes puedes ser identificado en la Casa, fuera de la Deckgeon.

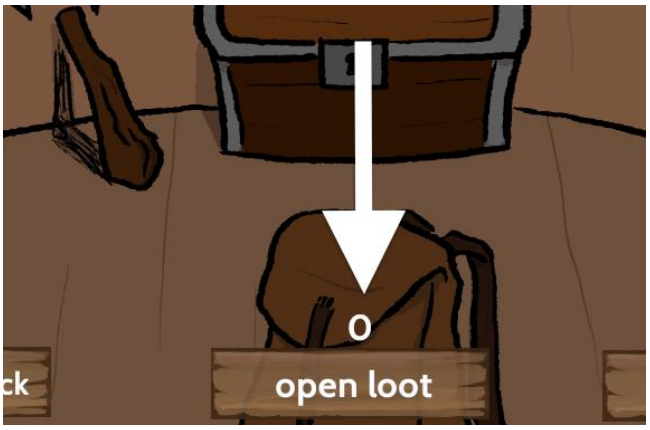


Fig. 67 Tutorial Deckgeon_10

7.2. Tutorial de la Villa

- La Villa ofrece muchas opciones útiles en tus aventuras.

Fig. 68 Tutorial Villa_1

- En la Casa, puedes personalizar el contenido de los mazos que utilizarás durante la exploración en Deckgeon. Mueve las cartas que quieras del Almacén al Mazo y viceversa, hasta que des con tu configuración óptima.

Fig. 69 Tutorial Villa_2

- Puedes comprobar el botín adquirido después de cada viaje a la Deckgeon en la Casa. Las Cartas de Moneda se convierten automáticamente en monedas después de ser identificadas, mientras que las Cartas de Tesoro se almacenarán en el Almacén de Tesoro.

Fig. 70 Tutorial Villa_3

- Tu primer impulso puede ser abrir la bolsa de botín cuanto antes mejor, antes de perderlo. ¡Quizá no hacerlo te proporcione mejores recompensas en un futuro!

Fig. 71 Tutorial Villa_4

- Puedes comprobar tus características en el Guardarropa, y pueden mejorarse mediante Cartas de Equipo. Arrastra y suelta cartas de equipo en su correspondiente ranura de equipo.

Fig. 72 Tutorial Villa_5

- La Tienda ofrece una variedad de cartas para añadir a tus mazos. Para comprar nuevas cartas es tan fácil como soltar la carta deseada en la ranura de compra y confirmar la transacción con el botón de Comprar carta.

Fig. 73 Tutorial Villa_6

- Además el comerciante de la misma Tienda está dispuesto a comprar las cartas que ya no necesites. Mediante el mismo proceso, arrastra las cartas que estén en tu Almacén hasta la ranura de venta y confirma con el botón de Vender carta. Ten en cuenta que la transacción es irreversible y no se puede vender cartas que estén en uso.

Fig. 74 Tutorial Villa_7

- Consulta la Taberna de tanto en tanto, pues interactuar con los habitantes de la Villa quizá sea beneficioso. ¡Algunos quizá tengan información importante y misiones que te ayuden en tu aventura!

Fig. 75 Tutorial Villa_8

7.3. Tutorial de las Cartas

- Existen diferentes tipos de cartas que puedes encontrar a lo largo de tu aventura por *Deckgeon*:

- Cartas de movimiento
- Cartas de combate
- Cartas de moneda y tesoro
- Cartas de equipo

Fig. 76 Tutorial Cartas_1

- Cada una de ellas tiene un valor, definido por el número superior derecho, que indica el efecto que realiza cada una de las cartas, como el movimiento, el daño causado, el precio o los beneficios que pueda proporcionar una carta.

Fig. 77 Tutorial Cartas_2

- Las Cartas de movimiento y combate pueden jugarse en la *Deckgeon*. El símbolo del centro indica si se aplica el efecto de la carta inmediatamente al ser jugada o al final de tu turno.

Fig. 78 Tutorial Cartas_3

- Las Cartas de tesoro y equipo solo pueden usarse en la Villa. En este caso, el símbolo central indica de qué tipo de carta se trata.

Fig. 79 Tutorial Cartas_4

8. Gestión del proyecto: herramientas externas

8.1. Trello

Se puede seguir la evolución de las tareas realizadas accediendo a la herramienta de gestión *Trello*:

https://trello.com/b/daWDj5gK/tfm-deckgeon

8.2. *BitBucket*

Se ha utilizado un repositorio en Bitbucket para almacenar y tener registrados todos los cambios del código, así como copia de seguridad del proyecto.

Recordar que para evitar errores la versión de ***Unity3D*** utilizada para el desarrollo del proyecto es la v. **2018.3.9f1**

https://bitbucket.org/albertpa/deckgeonrepo/

8.3. *Unity Connect*

Se ha añadido una página de Unity Connect que se irá editando según la evolución del juego, así como su *showcase* relacionado, con distintas pantallas del juego:

https://connect.unity.com/p/deckgeon

https://connect.unity.com/p/deckgeon-1

8.4. Ficha de *Google Play*

Se añade un link a la ficha en ***Google Play*** del proyecto publicado como *beta*, para que los jugadores puedan probarlo. Este paso es vital para la realización de las encuestas entre los usuarios.

https://play.google.com/store/apps/details?id=com.pages.Deckgeon

8.5. Otras *builds*

Se ha compilado una versión para Windows que se puede encontrar en el siguiente link:

https://drive.google.com/drive/folders/1eCYeX4WcWUIY6h5hhCLbwvBiyY-U8hVm?usp=sharing

8.6. Video de presentación

Se añade un link a la presentación del proyecto que se puede ver en *Youtube*:

Presentación del TFM: https://youtu.be/uJqV0LzjWNI

PEC02: https://youtu.be/metjBQas9rM

9. Gestión Económica

9.1. Coste del proyecto

Para el coste del proyecto se van a calcular las horas/programador y horas/diseñador de manera muy aproximada. No se va a tener en consideración costes de *hardware* ni *software* por lo que el cálculo puede ser irreal, por las siguientes razones:

- El uso de licencias de estudiante (***Adobe Photoshop y 3D Studio Max***) o directamente licencias gratuitas (***Unity3D***) para el desarrollo del juego.

- El uso de equipos de hardware antiguos (2012-2014) para el desarrollo del juego.
  En caso que se requiriese un nuevo equipo, se podría calcular en un equipo estándar valorado en **900€:**
  - *Procesador: AMD Ryzen 5 2600 3.9 Ghz*
  - *Placa Base: Asus B450M*
  - *Discos duros: SSD 240GB SATA3*
  - *Memoria: DDR4 3000 PC4-24000 16GB 2x8GB CL15*
  - *Gráfica: GTX 1650*
  - *Otros elementos (torre, fuente de alimentación)*
  - *Pantalla, teclado, ratón y tableta Wacom.*

Para subir la aplicación a *Google Play*, se debe contar con una cuenta de desarrollo que cuesta **25$ (23€ aprox)** en un pago único.

La compra de *Assets* mediante la plataforma *Humble Bundle* (varios conjuntos de *Assets* de gráficos y sonido) ha tenido un valor aproximado de **42€** en un pago único.

El tiempo de desarrollo del motor de juego y diseño ha sido 6 meses a media jornada cada día (excluyendo fines de semana) por lo que 123 días laborables a 4 horas cada día son un total de 492 horas. A 25 €/hora de programación y diseñador se calculan un total de **12300€** brutos (por lo que se debería descontar impuestos).

El tiempo de desarrollo de los modelos 3D y animaciones ha sido de 2 meses a media jornada cada día (excluyendo fines de semana) por lo que 42 días laborables a 4 horas cada día son un total de 168 horas. A 20 €/hora se calculan un total de **3360€** brutos (por lo que se debería descontar impuestos).

El coste del proyecto da un total de **16625€.** Cabe remarcar que es un coste aproximado ya que el precio/hora y el valor del equipo de *hardware* son orientativos.

9.2. Monetización

Para la monetización del producto se plantean varias posibilidades:

- **Un único pago**. Según los resultados de la encuesta, un 25% pagaría entre 3 y 5€ por una versión finalizada del mismo, un 19% entre 1 y 3€ y un 13% por menos de un 1€. El resto (43%) dependería del producto resultado o directamente no pagaría. Aunque visto así el pago único parece una buena opción, la muestra de la encuesta no es significativa (unas 15 personas) y, aunque haya interés en un único pago, no se puede asegurar que estos resultados sean del todo reales.

- **Publicidad *in-game***. Añadir anuncios usando las librerías de ***Google Ads*** en puntos clave del juego, como por ejemplo antes de bajar a la mazmorra. Esta es una de las opciones más plausibles para el diseño del juego actual. Aun así, existe el riesgo de cansar al usuario debido a la saturación de anuncios.

- **Free-to-play.** El jugador tiene acceso a todo el contenido, pero de manera limitada. En este caso, se podrían limitar las exploraciones a la mazmorra según un parámetro (energía) y que se pudiera comprar pociones de energía para poder seguir jugando, o esperar 8 horas a que el personaje descanse. Este modelo, parecido al **pay-for-win**, es muy criticado por los jugadores pues no permite jugar y corta muchísimo el flujo del juego.

Para realizar alguna de las 3 opciones propuestas, se podría añadir algún edificio nuevo en la Villa que resultara muy atractivo para las mecánicas del juego:

- Quiosco donde comprar sobres que contienen cartas al azar.
- Poder mezclar 2 cartas para que salga 1 nueva y más poderosa al azar.
- Disponer de potenciadores (power-ups) para la próxima aventura en la mazmorra: más vida, más ataque, ...

Estas acciones estarían limitadas 1 vez al día, o siempre que se quiera si:

- Se visualiza un anuncio.
- Se paga una única vez entre 3€ y 5€, pues es el valor más seleccionado entre los usuarios de prueba, desbloqueando el edificio sin limite de tiempo alguno.

Logrando así una mezcla entre **pago único** y **publicidad *in-game.***

Esta decisión ha de tomarse sin riesgo a que el juego quede desbalanceado entre la gente que haya pagado como la que juegue de manera gratuita, así que posiblemente la mejor opción sea la idea de mezclar 2, cartas ya que se pueden controlar las recetas de la mezcla mediante un algoritmo.

10. Conclusiones

Realizar este proyecto ha sido muy enriquecedor tanto a nivel personal como profesional. No solo se ha terminado el trabajo con una versión bastante definida publicada en ***Google Play*** sino que además se ha recibido *feedback* del juego por parte de usuarios reales. El juego es, como se ha propuesto desde buen principio, un producto para usuarios asiduos (y consolidados como jugadores) adaptado a la interfaz móvil y los límites que ésta ofrece.

Una de las primeras lecciones aprendidas es que la realización de un prototipo y extraer conclusiones de él resulta muy útil. Aunque pueda parecer una pérdida de tiempo inicial, ahorrará tiempo en el desarrollo de elementos que añaden poco o nada al proyecto o que deban modificarse desde buen inicio. Así pues, se ha seguido con la planificación al detalle, desviándose en contadas ocasiones y, es más, añadiendo puntos extra durante el desarrollo que han aportado calidad al producto.

De cara al final del proyecto, uno de los trabajos más costos ha sido la realización de un sistema multiidioma y adaptar todos los campos de interfaz a este sistema. Fueron días de trabajo que podrían haberse repartido durante el inicio del desarrollo, pero que no se tuvo en cuenta. En este sentido, debería haberse planificado durante el desarrollo del mismo y no como hito opcional.

Por parte de las pruebas de usuario, quizá ha sido una de las partes más enriquecedoras del proyecto, pues en algunos puntos es importante ver donde el juego falla y si se puede solucionar de una manera fácil, rápida y eficaz. En realidad, ha habido dos grandes grupos de *beta testers*: los 10 iniciales que jugaron a versiones entre v0.94 y v0.96 y los nuevos con la versión v0.96 y superiores, con cambios como el multiidioma o ayudas y videos en puntos donde los primeros se sentían más perdidos..

Lo que está claro es que como desarrollador, se acaba con unas malas prácticas y conocimientos que el nuevo usuario A ello se le suma que el sistema y algunas mecánicas son nuevas, no recuerdan a “nada” (no es un plataformas, no es un FPS, ni un juego de estrategia o construcción), por lo que jugar por asociación o saber que cartas aparecen o desaparecen es algo que el juego debe enseñar y el jugador debe aprender.

Pienso que el producto final es robusto, tiene posibilidades y, lo mejor de todo, es que algunos usuarios de test también lo piensan. En este punto se abren muchas posibilidades y, como muchos usuarios indican, llenar el juego de cartas, niveles, enemigos y, en general, contenido, hará que el juego sea una evolución de la *vertical slice* presentada. Al final, el objeto es generar un producto lo suficientemente atractivo como para que la gente lo adquiera.

11. Glosario

Asset: Elemento de *Unity3D* que tiene valor por sí mismo y es independiente del motor de juego actual, por lo que se puede usar en distintos proyectos: elemento audiovisual, base de datos, ...

Frame: Fotograma o imagen que se repite un número de veces por segundo en un videojuego, entre 30 y 60 *frames* por segundo. Implica la velocidad de refresco y sensación de fluidez en el juego.

GameObject: Elemento estructural de *Unity3D* que puede contener distintos subelementos o componentes y que describen su comportamiento: *scripts*, *sprites*, sonido, ...

IDE*:* Integrated *Development Environment,* entorno de desarrollo integrado.

NPC: *Non player Character,* personaje no jugador.

Prefab: Conjunto de *GameObjects* almacenado como un único objeto para poder ser utilizado o instanciado y que se usa como plantilla para generar otros objetos.

RPG: Role Playing Game, juego de rol.

Rigging: Añadir un sistema de control en un modelo 2D o 3D, normalmente conocido como huesos y su influencia en modelo, para su futura animación.

Singleton: patrón de diseño de programación, se genera una instancia única de un objeto y permite que todos los elementos tengan un acceso global a ella.

Script: fichero de código, en el caso de *Unity3D* en lenguaje C#.

Sprite: imagen en formato mapa de bits.

XML: Siglas de eXtensible Markup Language. Lenguaje con marcas que facilita la estructura y lectura del mismo fichero.

12. Agradecimientos

Primero de todo me gustaría agradecer a la gente que me ha ayudado a realizar este proyecto de una manera directa: A Héctor Ariza y Kepa Izeta por su trabajo, así como los usuarios de test, tanto los conocidos (Jesús Escribano, Alberto Hidalgo, Adriana Fernández, Andrea Sospedra y muchos otros) como los desconocidos.

A mis compañeros de trabajo de la UPC que están y han pasado por el lab, Marc Mateus, Fede Guede, Sergio Guerrero, Víctor Ferrer y Antonio López, que también han probado el juego y me han dado *feedback*, y que a veces han echo la vista gorda cuando me documentaba.

A mis compañeros de *La Universitat Invisible,* por aguantarme, muchos ya nombrados, pero que me dejo a Enric Garriga, Joan García y David Cruz.

A mis betatesters más roleros, Toni Garí y Joan Segovia.

A mis amigos de toda la vida que han hecho difusión del proyecto y me han regalado algunas ideas de manera consciente o inconsciente, Chema Peral, Jaime Agramunt y Adrià Castillón.

Finalmente, y mucho más importantes, agradecer a mi padre Ramón y mi hermana Elisabeth, también a mi pareja Gema Terol, por aguantarme y estar siempre ahí para mi cuando los necesito, y que a pesar de todo me siguen queriendo. Agradecer a los perros de la casa, que fueron como familia para mi aunque ellos no lo supieran, y que he intentado homenajearlos de la mejor manera posible. Agradecer a los familiares que no están ya conmigo, pero que me han ayudado a ser quien soy: os echo mucho de menos.

Y sobretodo a mi madre, que me ayuda aún sin estar.

I. Bibliografía

Literatura

[1]. Afilias Technologies Limited, (2019)[En línea] Most used smartphone screen resolutions in 2019 [Consultado en Marzo de 2019]
<https://deviceatlas.com/blog/most-used-smartphone-screen-resolutions >

[2]. Quora; Ryan Toh, (2017) [En línea]. When did smartphones become popular?. [Consultado el Abril de 2019]
<https://www.quora.com/When-did-smartphones-become-popular>

[3]. UX Matters; Steven Hobber Toh, (2013) [En línea]. How do users really hold mobile devices?. [Consultado el Marzo de 2019]
<https://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php>

Tutoriales y Software de interés

[4]. Atlassian, (2011-2019)[En línea] Trello [Consultado en Enero - Junio 2019]
<https://trello.com/b/daWDj5gK/tfm-deckgeon >

[5]. GameDev Network Limited, (2019)[En línea] GameDev Marketplace. [Consultado en Marzo de 2019]
<https://www.gamedevmarket.net/ >

[6]. Humble Bundle Inc., (2018)[En línea] Humble Bundle Store. [Consultado entre Enero y Abril de 2019]
<https://www.humblebundle.com/>

[7]. Youtube; Brackeys, (2018)[En línea] Unity tutorial:Animate 2D characters in Unity [Consultado en Junio de 2019]
<https://www.youtube.com/watch?v=eXIuizGzY2A>

[8]. Youtube; Brackeys, (2017)[En línea] Unity tutorial: How to make a dialogue system [Consultado en Mayo de 2019]
<https://www.youtube.com/watch?v=_nRzoTzeyxU>

Juegos

[9]. Nintendo EAD (1986). The Legend of Zelda.

[10]. Square Enix(1986). Dragon Quest.

[11]. TinyTouchTales(2018). Card Crawl.

[12]. Wizards of the Coast (1993). Magic The Gathering.

II. Anexos

A. Otro material consultado durante la realización del TFM

Arguello, Diego; Gamasutra, (2019)[En línea] Designing for deck-building in video games [Consultado en Marzo de 2019]
<https://www.gamasutra.com/view/news/339445/Designing_for_deckbuilding_in_video_games.php>

Gamasutra; Stonehouse, Anthony. (27 de Febrero de 2014) [En línea]. User interface design in video games. [Consultado en Febrero 2019]
<https://www.gamasutra.com/blogs/AnthonyStonehouse/20140227/211823/User_interface_design_in_video_games.php >

Gamasutra; Burke, Zach. (2015) [En línea]. The 5 golden rules of Input. [Consultado en Febrero 2019]
<https://www.gamasutra.com/blogs/ZachBurke/20151030/257920/The_5_Golden_Rules_of_Input.php >

Gamasutra, Leight Alexander. (2013)[En línea]. Designing better controls for the touchscreen experience for. [Consultado en Febrero 2019]
<https://www.gamasutra.com/view/news/205434/Designing_better_controls_for_the_touchscreen_experience.php >

Gliffy, Inc. (2019)[En línea] Gliffy [Consultado en Enero - Junio 2019]
<https://www.gliffy.com/>

Noun Project Inc.(2018)[En línea] The Noun Project. [Consultado el 2 de noviembre de 2018]
<*https://thenounproject.com/* >

Pennock, Jacob, (2012)[En línea] Use custom made assets as configuration files[Consultado en Abril de 2019]
<http://www.jacobpennock.com/Blog/unity-pro-tip-use-custom-made-assets-as-configuration-files/>

Pocket Tactics.(2017)[En línea] INTERVIEW: Card Thief Game Designer Arnold Rauers. [Consultado en Enero de 2019]
<*https://www.pockettactics.com/articles/interview-card-thief-game-designer-arnold-rauers/* >

StackOverflow; VV.AA., (2014)[En línea] Unity animation glitches in Build [Consultado en Abril de 2019]
<https://stackoverflow.com/questions/23106234/unity-animation-glitches-in-build>

Unity Technologies.(2019)[En línea] Unity Assets Store. [Consultado en Marzo de 2019]
<https://assetstore.unity.com/ >

Unity Technologies, (2016)[En línea] Object Pooling [Consultado en Abril de 2019]
<https://unity3d.com/es/learn/tutorials/topics/scripting/object-pooling/>

Unity Technologies, (2019)[En línea] Quality Settings [Consultado en Abril de 2019]
<https://docs.unity3d.com/Manual/class-QualitySettings.html>

Unity Technologies; VV.AA.,(2017)[En línea] Building script cannot find in menu item[Consultado en Abril de 2019]
<https://forum.unity.com/threads/build-script-cannot-find-menuitem.535889/>

Unity Technologies; VV.AA., (2016)[En línea] How do I access singleton pattern in Unity [Consultado en Abril de 2019]
<https://answers.unity.com/questions/1188170/how-do-i-access-singleton-pattern-script-in-unity3.html/>

Unity Technologies; VV.AA., (2018)[En línea] When adding prefab button to canvas with code, scale is wrong [Consultado en Abril de 2019]
<https://forum.unity.com/threads/when-adding-prefab-button-to-canvas-with-code-scale-is-wrong.514262/>

Youtube; *C00pala*, (2017)[En línea] Unity tutorial: float/ damage text [Consultado en Abril de 2019]
<https://www.youtube.com/watch?v=LkN4HvlPN84>

ZuluOneZero, (2019)[En línea] Violation of usage of Android advertising id policy[Consultado en Junio de 2019]
<http://www.zuluonezero.net/2019/02/11/violation-of-usage-of-android-advertising-id-policy-and-section-4-8-of-the-developer-distribution-agreement/>

B. Estructura de la encuesta realizada entre todos los usuarios

Preguntas	Respuestas
Sobre la Villa	
Por favor, puntúa las pantallas de mayor a menor claridad:	
[Villa]	● No se entiende ● Poco claro ● Mejorable ● Claro
[Casa]	
[Tienda]	
[Taberna]	
[Mazmorra]	
Las opciones que me proponen me parecen...	● Escasas ● Claras ● Otro
¿Comprendo las opciones que el juego me da?	● Sí ● No ● Otro
Sobre la Mazmorra	
¿La interfaz es clara y entiendo toda la información?	● Sí ● No ● Otro
¿He entendido la mecánica de las cartas de movimiento?	● Sí ● Sí, pero me ha costado un poco entenderlo ● No ● Otro
¿He entendido la mecánica de las cartas de batalla?	● Sí ● Sí, pero me ha costado un poco entenderlo ● No ● Otro
¿He entendido la mecánica de los combos entre cartas?	● Sí ● Sí, pero me ha costado un poco entenderlo ● No ● Otro

¿El tutorial es claro y correcto?	● Sí ● Sí, pero me ha costado un poco entenderlo ● No ● Otro
¿Te ha parecido divertido?	● Sí ● Ni sí ni no ● No ● Otro
Deckgeon: en general...	
¿Volverías a jugar?	● Sí ● No ● Otro
¿Pagarías por jugar a un juego con estas mecánicas?	● Sí, por más de 5€ ● Sí, entre 5€ y 3€ ● Sí, entre 3€ y 2€ ● Sí, entre 2€ y 1€ ● Sí, por menos de 1€ ● No
¿Que elementos crees que se deberían potenciar para volver a jugar?	
[Más cartas que conseguir]	● No sería necesario ● No sería necesario, pero estaría bien ● Totalmente imprescindible
[Más niveles en la mazmorra]	
[Mayor número de enemigos]	
[Más opciones en el poblado]	
[Una arena para luchar contra monstruos]	
[Mayor número de quests]	
¿Que otra mejora o mecánica añadirías?	*texto libre*
¿Te ha recordado a algún otro juego?	*texto libre*
¿Has encontrado algún bug o comportamiento extraño durante la partida? Si es así, por favor, descríbelo.	*texto libre*
¿En qué dispositivo y sistema operativo has probado el juego? (p.e.: Móvil Samsung J5, Android 8)	*texto libre*

Sobre ti	
Sexo	● Prefiero no decirlo ● Mujer ● Hombre
Edad	● Menos de 15 años ● 15-20 ● 21-30 ● 31-40 ● Más de 40
¿Tienes consola o PC en casa?	● Sí, varios dispositivos ● Sí, un dispositivo ● No
Juego de manera...	● Ocasional (menos de 1 hora a la semana) ● Entre 1 y 2 horas a la semana ● Entre 3 y 5 horas a la semana ● Diariamente
Géneros favoritos	❑ Aventuras gráficas ❑ Acción ❑ Juegos en primera persona ❑ Deportivos ❑ Puzle ❑ RPG ❑ Casuales (pasa ratos) ❑ Multijugador ❑ Otros