

DEMON'S STARWAY

SEGUNDO PROYECTO DE JUEGOS PARA WEB Y REDES SOCIALES

GRUPO 04

Contenido

DEMON'S STARWAY	2
DIAGRAMAS DE SECUENCIA.....	2
EQUIPO DE TRABAJO	3
GAME DESIGN.....	3
PRIMERAS IDEAS	3
DISEÑO DE NIVELES.....	4
MULTIJUGADOR	5
DISEÑO GRÁFICO	5
PLANTEAMIENTO DE LA ESTÉTICA DEL JUEGO	5
HERRAMIENTAS DE DIBUJO Y PROCESO DEL DESARROLLO DE CONCEPT ART	5
LOS PERSONAJES	5
EL ESCENARIO.....	8
LAS PANTALLAS	11
DISEÑO 3D.....	14
• REPRESENTANDO LA ESTÉTICA	14
• <i>SHADERS</i>	16
• CREANDO LAS TEXTURAS	16
• <i>ENVIRONMENT ASSETS</i> : CREANDO LA DECORACIÓN DE LOS ESCENARIOS.....	17
DISEÑO DE AUDIO	19
SISTEMA DE PARTICULAS	19
PROGRAMACIÓN	20
INTERFAZ	20
FUNCIONALIDAD MÁS IMPORTANTE DE LOS SCRIPTS MÁS DESTACADOS.....	22
CONTROLES.....	24
• PC:	24
• Dispositivo Portable:.....	25
OPCIONES.....	26
MEJORAS.....	26
CRÉDITOS	27
ENLACE A NUESTRO GITHUB	27
ENLACE A NUESTRO PORTFOLIO	27
ENLACE AL JUEGO WEB	27
ENLACE A VERSIONES DESCARGABLES	27

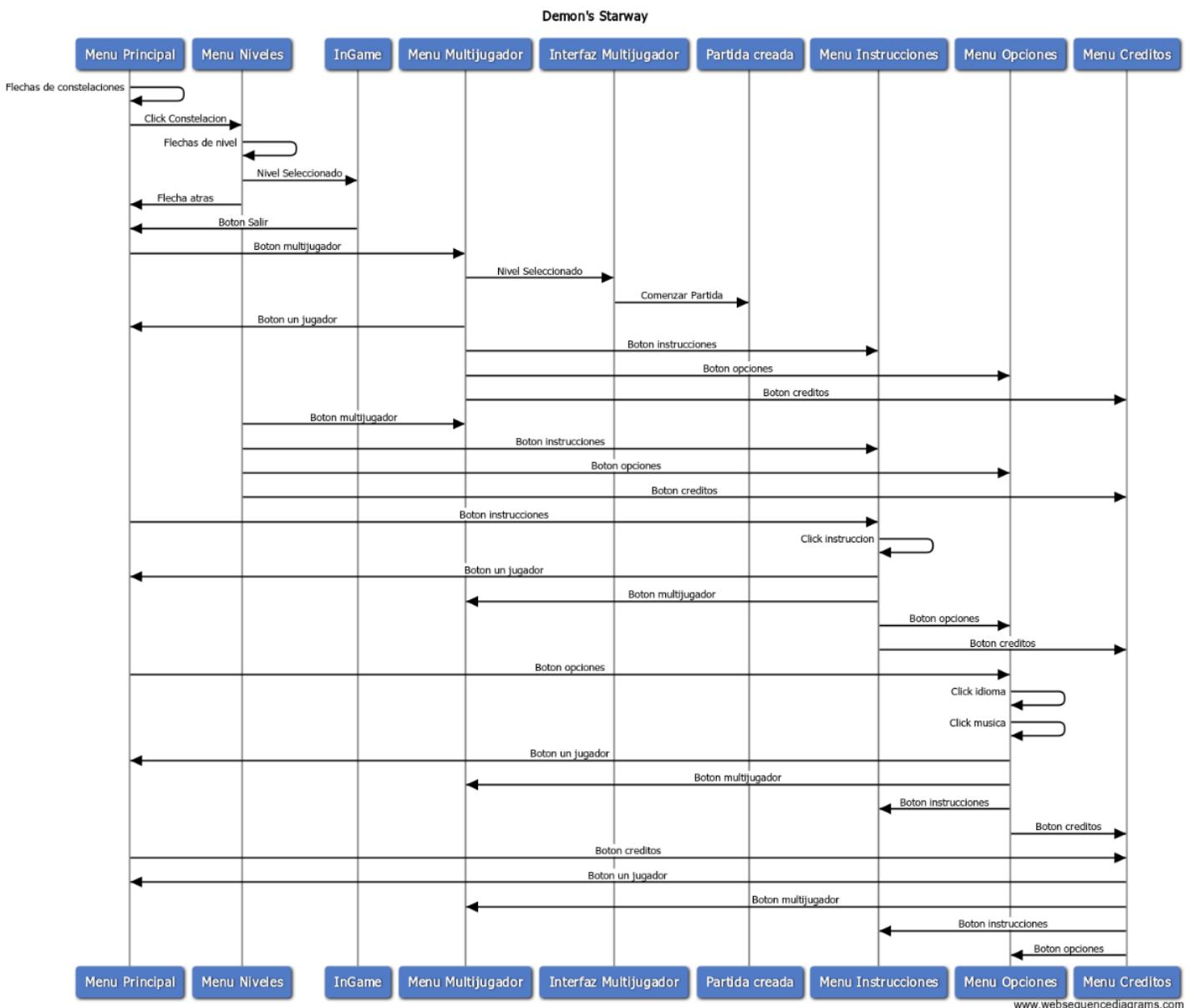
DEMON'S STARWAY

Demon's Starway es un videojuego de plataformas creado con Unity en el cual encarnaremos a Vicho, un pequeño demonio que deberá superar diferentes niveles en 3D sobre pequeños mundos cúbicos.

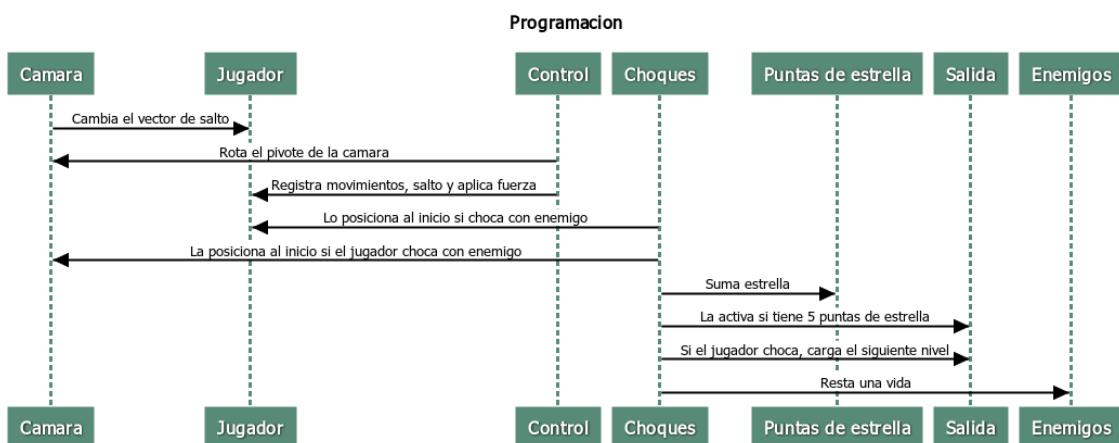
El Videojuego está disponible en versiones web que soporten *WebGL*, Android y Windows en dos idiomas, español e inglés. Además, podemos disfrutar de un modo multijugador en su versión Windows.

DIAGRAMAS DE SECUENCIA

Demon's Starway organiza sus pantallas según se muestra en el siguiente diagrama de secuencia:



En InGame, el juego sigue el siguiente diagrama de clases:



EQUIPO DE TRABAJO

Ismael Cabrera Casero: Game designer y programador

Sara Lozoya Sevilla: Artista conceptual y diseñadora 2D

Jesús Brey Fernández: Compositor y diseñador de audio

Jaime Calvo Amor: Programador

Alexander Matos Teodoro: Artista 3D

GAME DESIGN

PRIMERAS IDEAS

Demon's Starway desde el principio ha estado inspirado en mecánicas pertenecientes a títulos de la saga *Super Mario Galaxy* y *Captain Toad*, buscando jugar con la gravedad en saltos sobre planetas de tamaño reducido y la tranquilidad que te ofrecen los juegos sin combates directos y sin tiempos que atosiguen el cumplimiento de tu objetivo.

Además, Demon's Starway es un juego pensado para todos los públicos, algo que condicionará las decisiones tomadas a continuación.

De esta idea principal podemos ramificar el diseño del juego en tres partes:

-El universo de Demon's Starway y su historia

"En tiempos de la creación, Lucifer fue desterrado de los cielos por sus pecados. Descendió hasta el infierno por un camino de estrellas, el cual fue destruido por los ángeles a su paso para evitar su vuelta a toda costa."

Años más tarde, nuestro protagonista, por su condición de demonio tiene prohibida la entrada al paraíso, pero no está dispuesto a aceptar esa norma sobreimpuesta y se embarca en la aventura de reconstruir el camino a los cielos."

-Composición de niveles

Dicho camino mencionado en la historia no es otro que las 12 constelaciones. Cada constelación compondrá un estilo y temática de mundo.

Para completar las constelaciones y desbloquear las siguientes deberás completar todas las estrellas obligatorias que las conforman (considerándose cada estrella un nivel).

-Objetivo de los niveles

En cada estrella o nivel deberemos abrir el portal que fue destruido por los ángeles para avanzar a la siguiente estrella. Para ello deberemos recolectar 5 puntas de estrella que conforman el pentágono estrellado que abre dicho portal.

En los niveles encontraremos ángeles custodiando algunas de estas puntas para evitar que completemos nuestro objetivo los cuales nos quitarán una vida (De un máximo inicial de 3) y nos devolverán al punto de partida.

La partida acaba una vez perdamos 3 vidas (lo cual significa que tendremos que volver a intentarlo) o consigamos las 5 puntas de estrella y lleguemos al portal activado (y podremos jugar al siguiente nivel).

Explicados estos puntos, en la versión que se presenta encontraremos 3 niveles accesibles de la primera constelación (Aries) con diferentes assets y una curva de dificultad ascendente. Dicha curva no sería la original del juego ya que se ha querido mostrar niveles de dificultades diferentes.

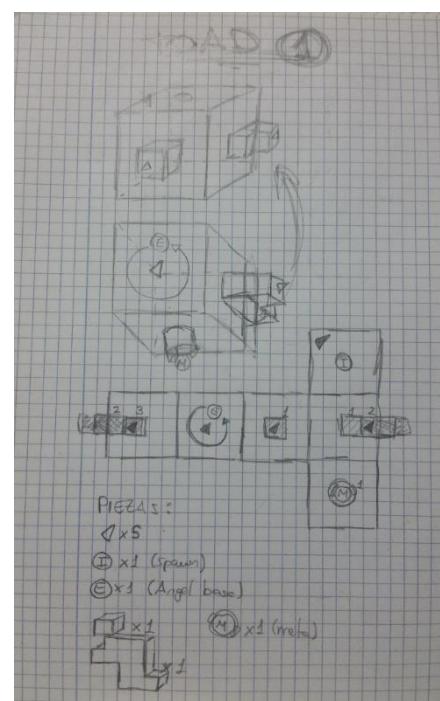
El principal objetivo del juego como se ha mencionado no es el estrés que te puede ofrecer un juego de acción, si no disfrutar de unos momentos intentando resolver los niveles, por lo que los ángeles no tienen tanto papel en la mecánica y son secundarios (en esta versión hay 1 tipo de ángel con 2 tipos de movimientos, pero en versiones futuras hay diseñados nuevos enemigos)

La historia mencionada forma parte de un trasfondo que no modifica la jugabilidad y sirve para ofrecer coherencia a la obra, por lo que en esta versión no se muestra ninguna cinematográfica o *cutscene* de la historia).

DISEÑO DE NIVELES

Los mundos se han diseñado para una fácil producción, aprovechando “planetas” o “mundos” cuadrados para cuadricular saltos y acciones y facilitar el diseño.

Como podemos observar en la imagen de la derecha, se diseñaba un boceto del nivel y se presentaban enemigos y una cantidad de objetos cúbicos para su posterior *concept* basado en su forma y finalmente su modelización.



MULTIJUGADOR

Siguiendo la temática del juego en el menú podemos encontrar el submenú “Ofiuco”, el cual simboliza la decimotercera constelación y nuestro modo multijugador.

Al igual que en el modo Individual, podremos encontrar diferentes niveles, con la diferencia que ahora competiremos contra un segundo jugador para alcanzar antes las puntas de estrella y llegar a nuestro portal.

DISEÑO GRÁFICO

PLANTEAMIENTO DE LA ESTÉTICA DEL JUEGO

Desde un principio quisimos que su apartado artístico y visual tuviese un diseño místico a la par que *cartoon* con referencias a juegos en 3D como “*The Legend of Zelda: Skyward Sword*” o “*Rime*”. Sin embargo, estos dos juegos que tuvimos de referencia eran de elevada complejidad a la hora de modelar y, como hemos querido ir un paso más allá en este proyecto y crear nuestros propios *assets*, se optó por un diseño más minimalista en cuanto a personajes y escenarios.

Por otro lado, dada la temática del juego, toda la acción se desarrolla en un espacio oscuro y estrellado. Esto hace que tengamos una paleta de colores reducida, puesto que en la noche lo que predominan son colores oscuros y apagados, y esto podría haber supuesto un problema a la hora de visualizar cualquier elemento en nuestro juego. Con esto en cuenta, se decidió utilizar una paleta de colores complementaria siendo los tonos azules para las pantallas y el personaje principal, y colores dorados y blancos para los enemigos y escenario. De esta forma, el contraste de ambos colores nos permite distinguir unos elementos de otros sin tener que forzar la vista.

HERRAMIENTAS DE DIBUJO Y PROCESO DEL DESARROLLO DE CONCEPT ART

Para este proyecto, el apartado artístico se ha separado el *concept art* y diseño de interfaz del modelado en 3D. Dicho esto, la herramienta principal del diseño artístico ha sido Photoshop CC debido a los elevados conocimientos del programa, así como por la licencia de estudiante que nos da la universidad para poder usarlo.

Con Photoshop, se han realizado todos los bocetos previos de personajes y escenarios, así como todas las vistas necesarias para modelar y las pantallas necesarias para los menús del juego. Cabe destacar que para crear ciertos efectos (por ejemplo: de estrellas y constelaciones), hemos optado por crear nuestros propios pinceles específicos.

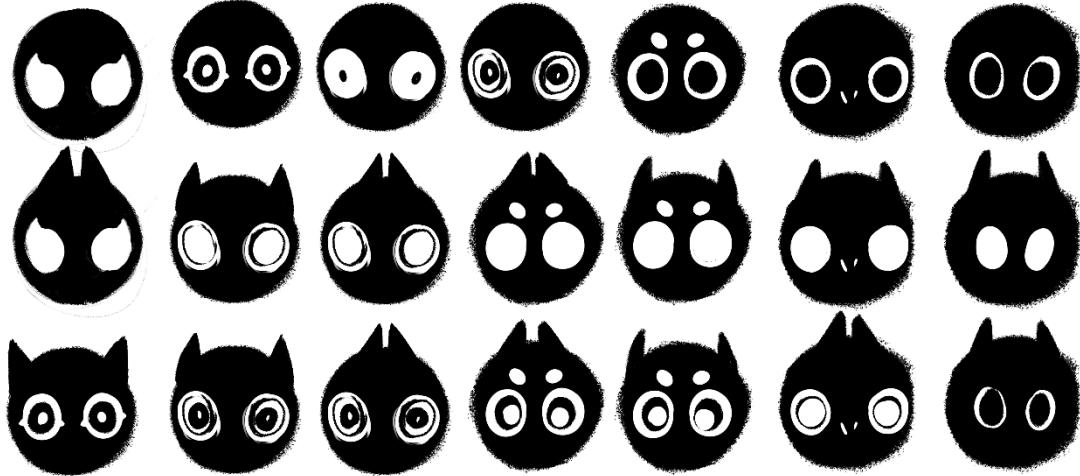
LOS PERSONAJES

- EL MINI DEMONIO

Para este proyecto se nos planteaba un reto en cuanto a su diseño: ¿cómo hacer ver al jugador que un demonio (un ser que por naturaleza es malo) es el bueno en esta historia?

Lo primero fue hacer su diseño como si se tratase de una bola. Esta decisión de hacer al personaje con formas circulares es debido a que éstas provocan una respuesta emocional en el receptor, de que nuestro personaje es amigable y afable.

A continuación, se muestran todos los bocetos del mini demonio que se sometieron a una votación entre los integrantes del grupo.



Tras dicha votación, se decidió que los diseños que más se acercaban al personaje que estamos buscando eran los de la línea de abajo, ya que éstos tienen unos mini cuernos que recuerdan a un demonio y además poseen ojos que hacen que el personaje sea más adorable para el jugador.

La imagen que se muestra a continuación, corresponde con el diseño final de nuestro personaje principal, con todas las vistas y anotaciones necesarias para su modelado en 3D.

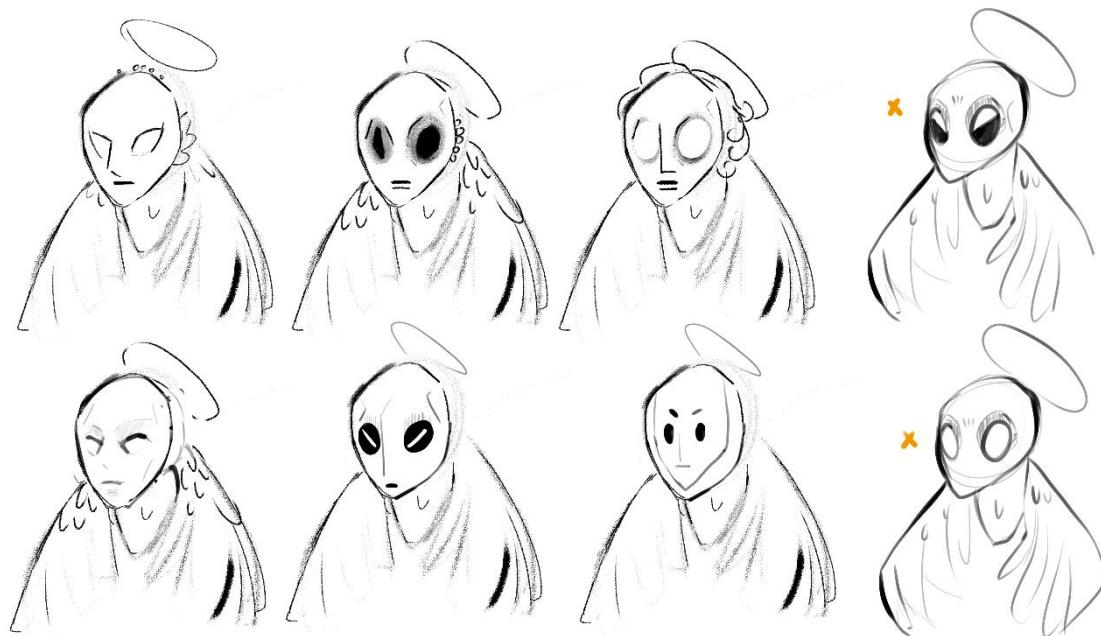


Cabe destacar que se desecharó la idea principal de que fuese de color negro puesto que el entorno en el que se iba a jugar es tan oscuro que no se distinguiría su silueta. Por este motivo, optamos por ponerle unas manchas en la parte frontal y trasera de su cuerpo para que el jugador pudiese verlo aun estando en la oscuridad.

- LOS ÁNGELES ENEMIGOS

Para el diseño de los enemigos se nos planteaba el mismo problema que con el personaje principal: ¿cómo hacemos que un ángel sea el malo de la historia?

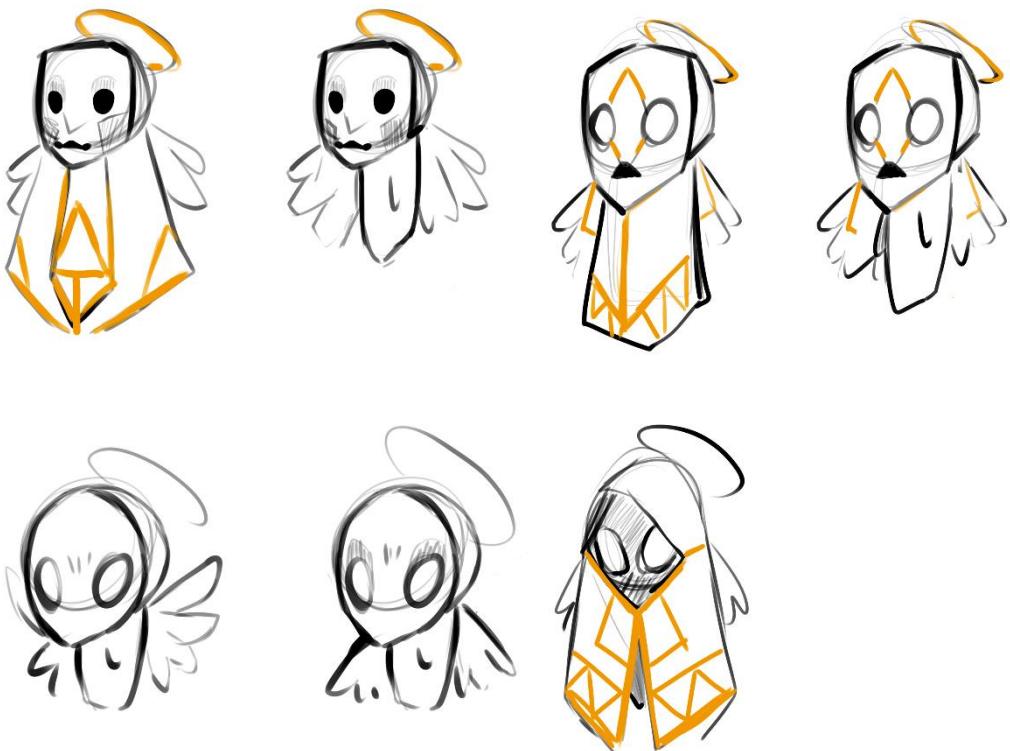
Con el ángel lo teníamos más complicado que con el mini demonio, puesto que queríamos hacer ver que era el enemigo, pero sin que perdiera su identidad de ángel. Desde un primer momento, experimentamos con su diseño, poniéndole una máscara blanca al estilo veneciano para que no tuviese ninguna expresión en su rostro y hacerle un personaje “distante”, evitando que el jugador simpatizase con él.



Sin embargo, los primeros bocetos, pese a que tenían ese aire “inquietante” que buscábamos, se alejaban mucho de poder ser identificados como ángeles y tenían un diseño tan complejo que no cuadraba con el del personaje principal. Además, las alas que se propusieron eran excesivamente grandes y podría resultar un problema a la hora de crear colisiones con ellas.

Por estos motivos, se decidió rehacer el diseño de los ángeles. Esta vez, se procuró que cada uno de los diseños cupiesen en un rectángulo vertical de manera que no sobresaliese ningún elemento de su diseño en grandes medidas. Además, el diseño de la máscara (así como el de su cuerpo y alas) cambió ligeramente, intentando que predominasen las formas puntiagudas o cuadradas para crear una respuesta emocional de tensión en el jugador.

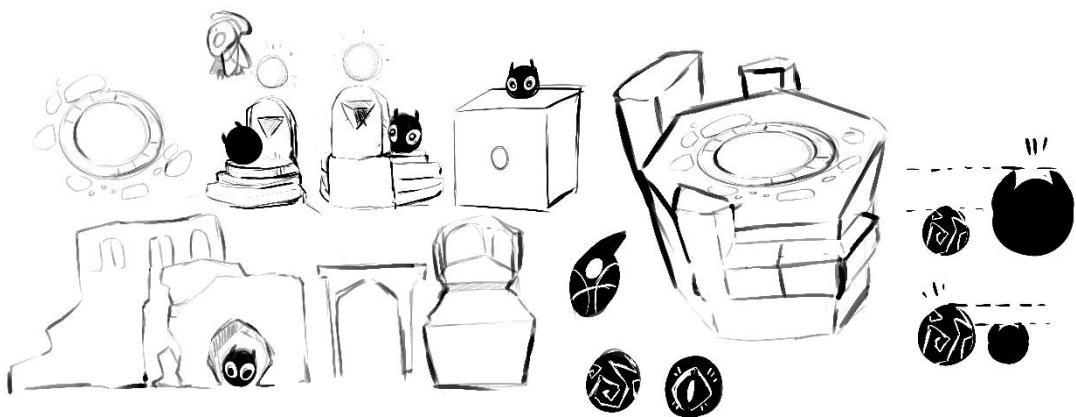
A continuación, se muestran los bocetos finales de los ángeles. Cabe mencionar que en un principio se diseñaron tres tipos de enemigos, cada uno más fuerte que el anterior y con mejoras añadidas (capas con detalles dorados).



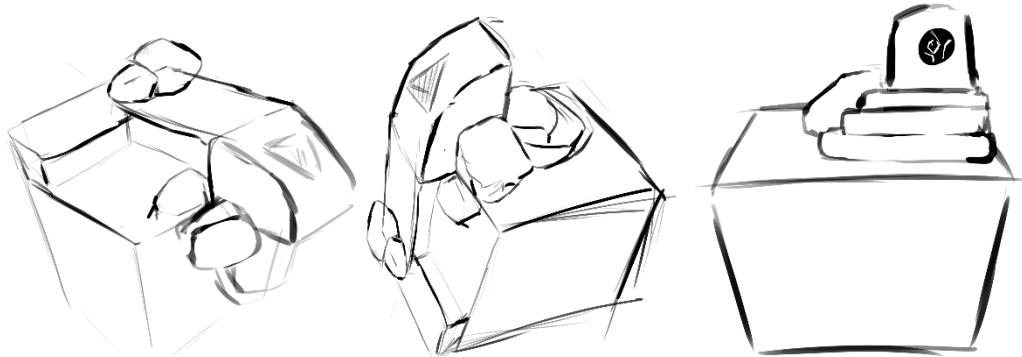
EL ESCENARIO

- LOS OBJETOS

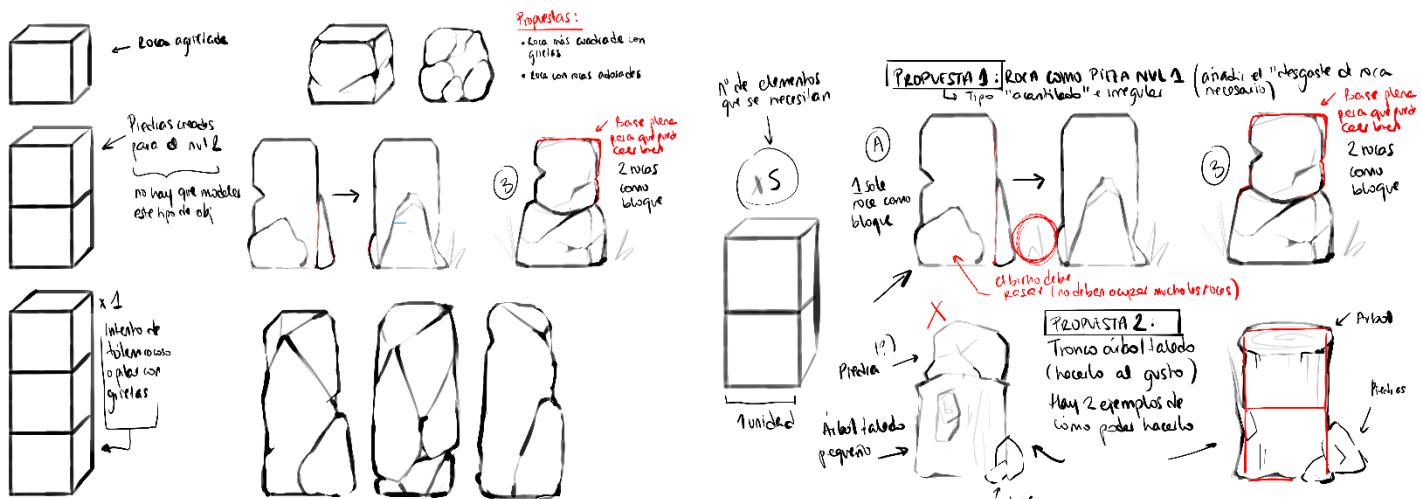
Para la creación de objetos, se comenzó realizando diversos bocetos de prueba que podrían aparecer en el escenario. Teniendo en cuenta que el mundo donde se desarrollaría la acción sería cúbico, los objetos no deberían ser excesivamente grandes y complejos puesto que al ser un mundo tan pequeño no se alcanzarían a distinguir los detalles.



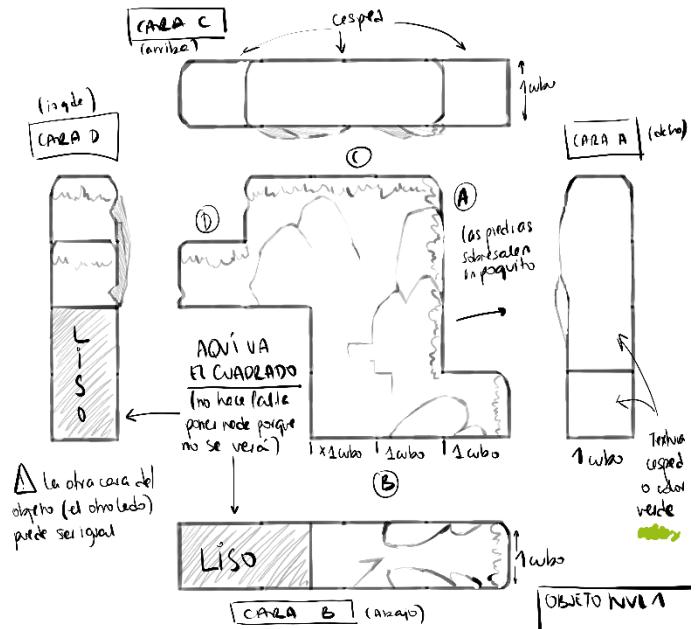
Teniendo en cuenta la forma del mundo, se consideró la idea de poder crear objetos que ocupasen más de una cara para poder dotar al juego de una mayor complejidad a la hora de resolver los puzzles planteados.



Cada nivel del juego posee sus propios objetos, los cuales dificultan el acceso de las puntas de estrella a recoger, éstos han sido creados en un principio con cubos, para comprobar en Unity la mecánica del propio nivel, y una vez que se decidían qué objetos eran necesarios, se pasó a diseñarlos de forma artística.

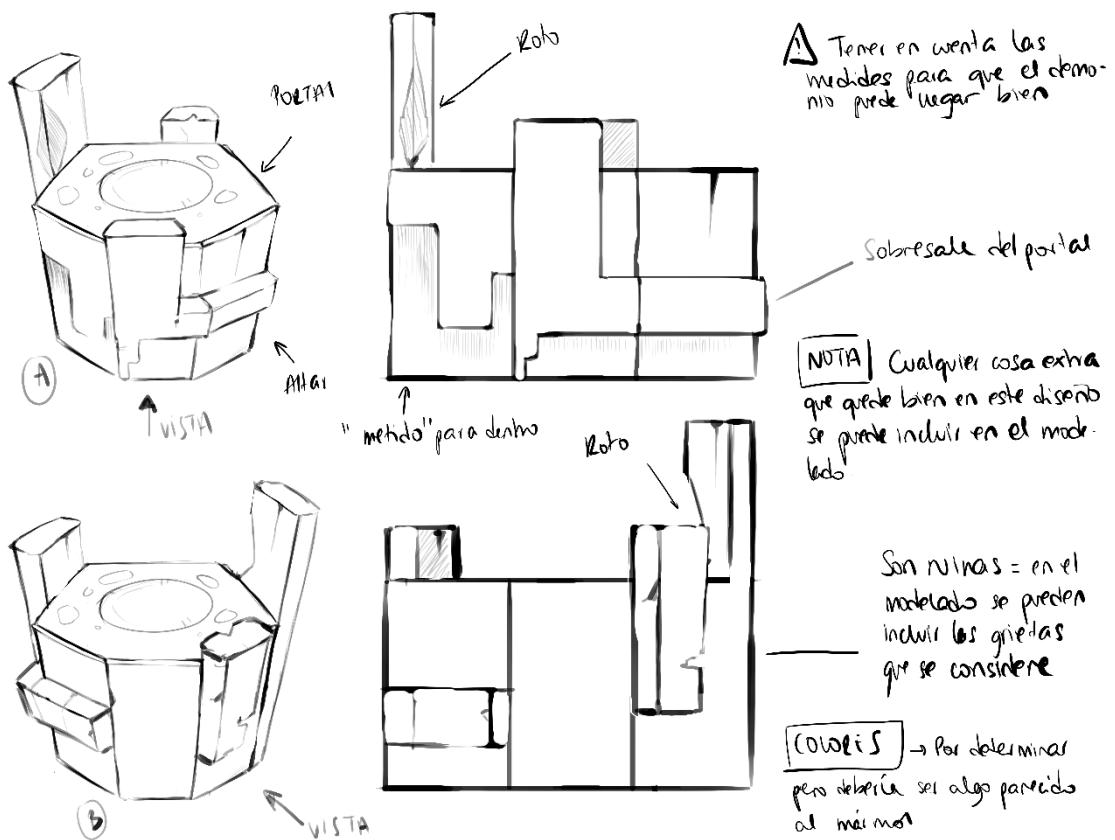


Cabe mencionar que, para cada uno de los diseños de los objetos, se han diseñado sus correspondientes vistas y anotaciones para facilitar el proceso del modelado tal y como se puede apreciar en el ejemplo que aparece.



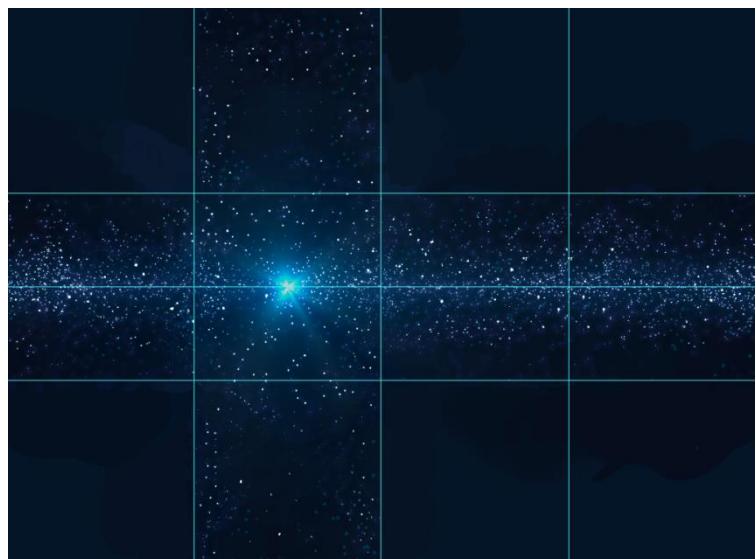
Como se ha mencionado, cada nivel posee sus propios objetos para completar el puzzle que se plantea. Sin embargo, todos los niveles tienen un objeto en común: el templo o altar. Es aquí donde el jugador deberá depositar las puntas de flecha que encuentre y, una vez reunidas todas, se abrirá un portal a través del cual podrá avanzar hacia el siguiente nivel.

Puesto que se trata de un objeto importante, se procuró que su diseño fuera lo más llamativo posible (sin salirnos de la estética planteada).



• EL SKYBOX

Para el escenario de cada mundo del juego, aparte de los objetos en 3D anteriormente descritos, se ha optado por crear un *skybox* desde cero. En nuestro *skybox* se muestra un espacio oscuro lleno de estrellas para que diera el efecto de más iluminación en la escena. Cada una de estas estrellas son fijas, puesto que se han pintado en seis caras que forman el *skybox* con la ayuda de Photoshop. A continuación, se muestra el *skybox* con las guías de Photoshop para distinguir cada una de sus caras.



LAS PANTALLAS

El juego consta de siete pantallas correspondientes con: el menú principal, el menú de elección de niveles, el de opciones, el de instrucciones, el de multijugador, el de créditos y el de *Game Over*. Cada una de estas pantallas ha seguido una misma estética basada en un mapa de constelaciones, es decir, para cada pantalla se ha diseñado una ruleta específica, o sea, un total de siete ruletas. A continuación, se pasa a explicar en detalle las más destacables.

- Menú principal (o menú de un jugador)

El menú principal consta de una ruleta con las doce constelaciones de los signos del zodiaco.



El motivo de este tipo de menú viene dado porque se nos ocurrió la idea de que el personaje principal pudiese viajar a cualquiera de las constelaciones y éstas representarían los distintos mundos que se pueden jugar. Sin embargo, por motivos de tiempo, para este proyecto se ha decidido que el jugador pueda acceder únicamente a la constelación de *Aries* y el resto permanecerán bloqueados con unos candados.



Cabe mencionar que, en este menú principal, se le dará la posibilidad al jugador de acceder a cualquier otra pantalla mediante los botones que aparecen en la imagen superior, justo debajo del logo del juego.

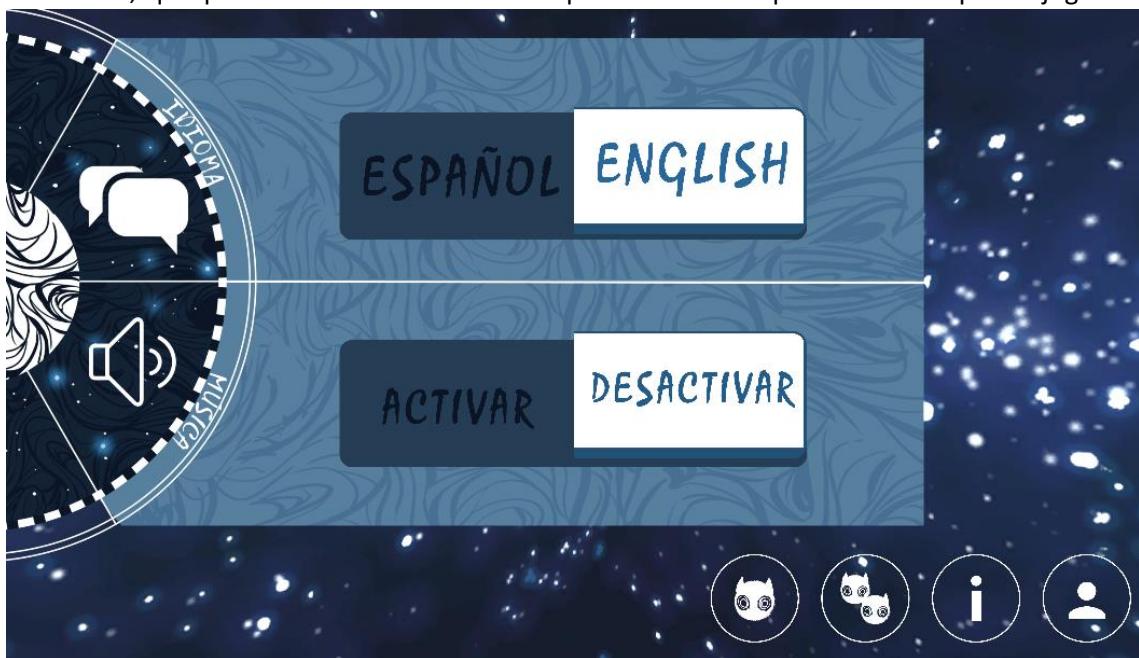
- Elección de niveles

De la misma forma que para el menú principal, se ha creado una ruleta para que el jugador elija un nivel, teniendo en cuenta que cada uno de estos niveles corresponde con las estrellas que posee cada constelación. Como para este proyecto solo se puede acceder a *Aries*, el jugador tiene la posibilidad de seleccionar tres de las cuatro estrellas que forman la constelación.



- Opciones

La pantalla de opciones cuenta con el cambio de idioma y la activación o desactivación de la música y sonidos. Ambas, se encuentran englobadas en una ruleta, de menor tamaño que las anteriores, que posee además unos botones para marcar la opción deseada por el jugador.



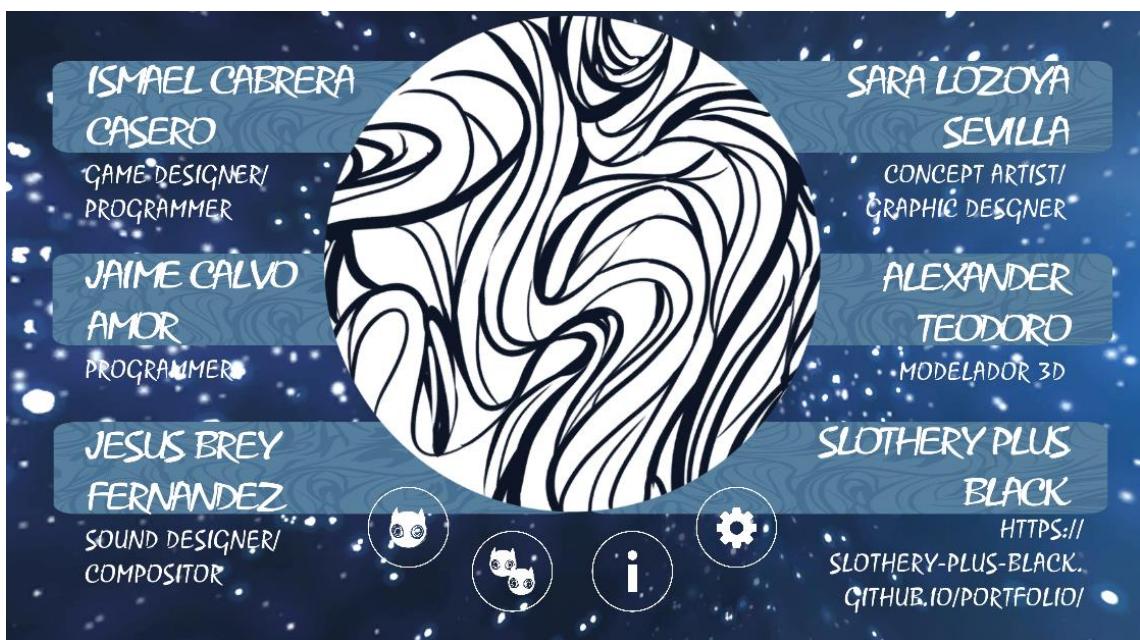
- Multijugador

Para la pantalla de multijugador, hemos querido hacer uso de la constelación *Ophiuco* puesto que se considera como el 13º signo zodiacal. De esta manera, el jugador cuenta con una nueva ruleta que le permitirá ir a un solo mundo correspondiente con la primera estrella de esta constelación. Cabe destacar que, en comparación con la ruleta del menú principal, ésta tiene los colores invertidos.



- Créditos

Por último, la pantalla de créditos muestra tanto los nombres de cada uno de los integrantes del grupo, así como sus roles. Para su diseño, se ha hecho uso de elementos que forman parte de las ruletas anteriormente mencionadas, pero sin llegar a sobrecargar en exceso la pantalla.



DISEÑO 3D

• REPRESENTANDO LA ESTÉTICA

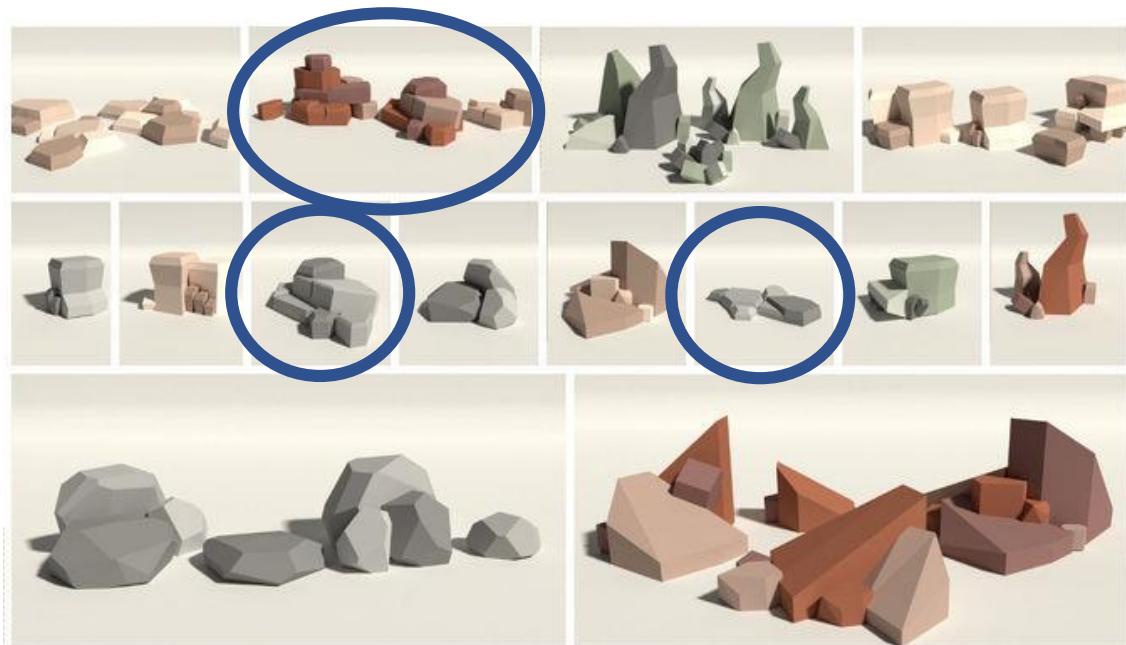
La primera pregunta que nos tenemos que hacer respecto al apartado 3D es: ¿Qué estética 3D utilizar y qué técnicas son las apropiadas? Nos encontramos ante un juego que va a requerir un alto número de assets para realizar todos los escenarios con las piezas que conforman el “puzzle” que es el mundo o nivel. Siendo la estética con rasgos *cartoon*, prima más representar la idea que ser fieles a la realidad, **caricaturizar**. Si se va a hacer un proyecto que potencialmente pueda ver la luz del mercado, la descarga de *assets* externos de modelos 3D no es una opción. Ya que todos están quemados y ya han sido usados en miles de juegos poco elaborados y llevan una mala fama con ellos. Por último, si el juego se va a exportar al mercado web y móvil, es necesario exigir el mínimo al dispositivo para poder acceder a un mayor número de clientes.

Teniendo todo esto sobre la mesa, la decisión que más se ajusta a las exigencias es sin duda la estética ***lowpoly***, ya popular entre estudios *indie* debido a que es **rápido, bonito y barato**.

Aunque en videojuegos todos los modelos son siempre *lowpoly*, en este contexto la estética *lowpoly* a llevar la reducción de polígonos al límite, con simplicidad de texturas y una iluminación acorde, esta representación es una caricatura y requiere decidir cómo se va a realizar, de esta manera podemos encontrar distintas estéticas:



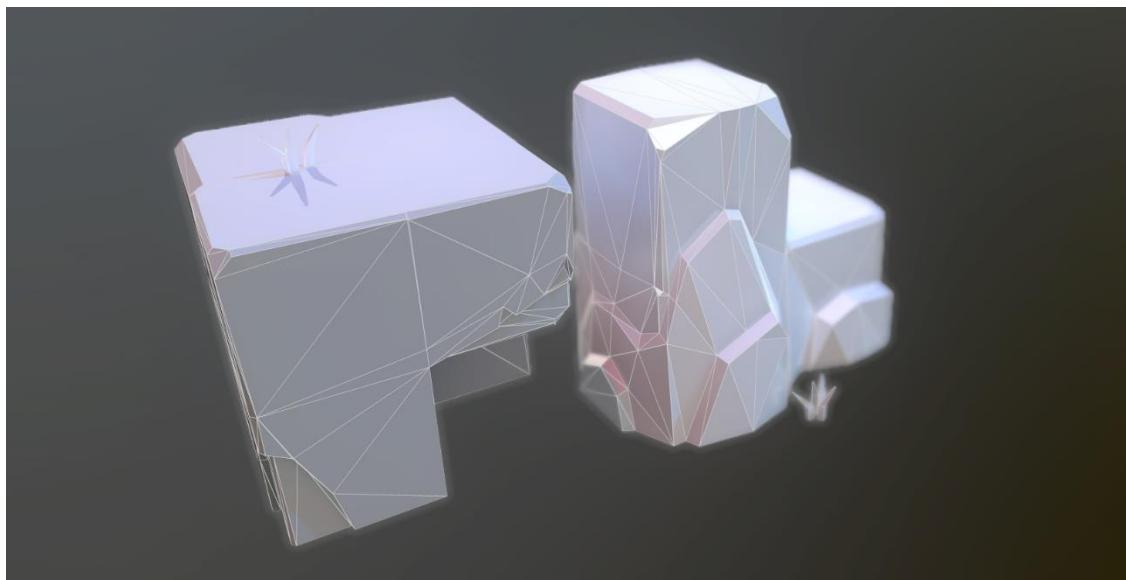
Src: <https://www.cgtrader.com/3d-models/plant/other/low-poly-trees--4>



En nuestro juego por la naturaleza de los puzzles, en cuanto a jugabilidad se refiere podríamos partir los mapas en cubos y prismas con las caras horizontales planas. De esta manera nuestra estética va a tener que respetar eso también. Evitando triangular los modelos y con las puntas poco pronunciadas (son formas más agresivas).

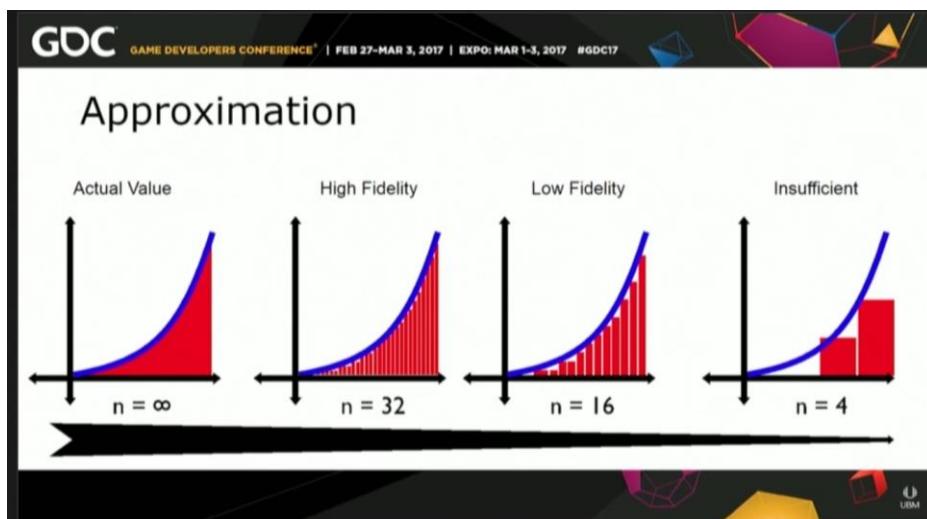
El software a utilizar será 3DS Max, debido a que ya se ha cursado anteriormente en otras asignaturas y además es el más adecuado para superficies no orgánicas y más técnicas.

De esta manera en los modelos que se han creado quedan de la siguiente manera:



Se puede observar que se respeta la horizontalidad y verticalidad, dando detalle y personalidad a los modelos en las zonas en las que nos saltamos esta regla (se pueden observar los detalles de roca y los rotos).

Adicionalmente, la última regla a seguir para mantener una consistencia en los modelos *lowpoly* es respetar la frecuencia de muestreo. Ethan Redd en la *Game Developers Conference* habla de que se pueden categorizar los gráficos 3D por su frecuencia de muestreo, de esta manera hay que mantener una relación en los detalles de los objetos en función del tamaño, un objeto debe tener X nivel de detalle y uno el triple de grande 3X.



Nuestros modelos respetan esta regla en la mayoría de los casos, saltándonos esta regla a la hora de querer centrar la atención en alguna zona. En las excepciones está la personalidad.

- **SHADERS**

Los *shaders* para la estética *lowpoly* son una ventaja computacional: No requerimos hacer cálculos de color/iluminación por cada fragmento (trozo de polígono asociado a un píxel en pantalla) la hora de mostrar cada objeto, si no por cada vértice es suficiente. Esto hace que el **coste computacional caiga de manera considerable**. Además, al usar colores planos no necesitamos texturas grandes 1024*1024px para los modelos, con una de 16*16px (tamaño mínimo soportado en *GPUs*) es suficiente. De esta manera también **se optimiza en la memoria**. La textura de un modelo pesa 4KB.

No nos ha sido necesario programar nuestros *shaders* ya que BrokenVector ofrece de manera gratuita los que necesitamos: Uno con cálculos de iluminación y otro sin iluminar nada (*unlit*).

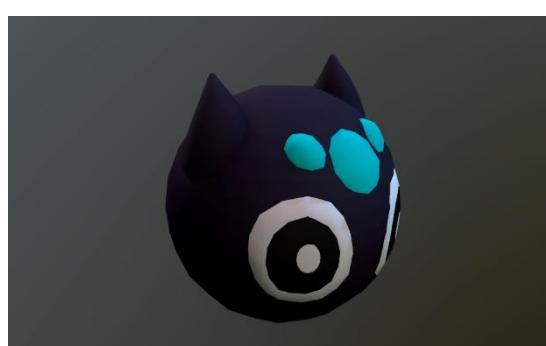
- **CREANDO LAS TEXTURAS**

Este apartado es en el que más se ha optimizado el trabajo. Las texturas son colores planos, con posibilidad de crear gradientes entre varios colores, esto tiene las ventajas que se van a poder observar a continuación.

Una vez se crea un modelo 3D, el modelador tiene que realizar el proceso de *unwrap* teniendo texturas de este tipo:

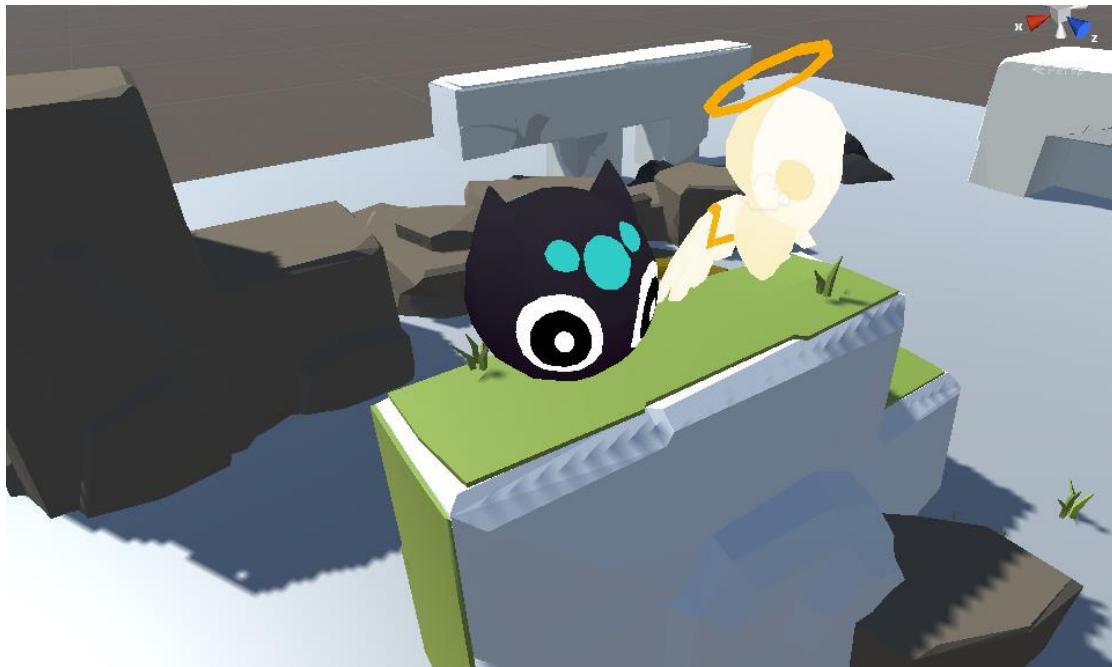


Una de las varias texturas que tiene un modelo convencional, se requiere estirar los polígonos del modelo para poder pintar todo sobre una imagen. Solamente el proceso de *unwrap* (ojo: no texturizar todavía) puede llevar horas. Nuestros modelos únicamente necesitan separar las zonas con distinto color en secciones, sin estirar, quedando como resultado una paleta de colores cuando se haga la textura:



El *unwrap* en cada modelo lleva entre 5-15 minutos dependiendo de la complejidad. La paleta resultante la puede proporcionar el/la Artista 2D, de esta manera la comunicación entre perfiles es muy sencilla.

Una vez hechas las texturas, se crean los materiales en Unity usando los *shaders*:

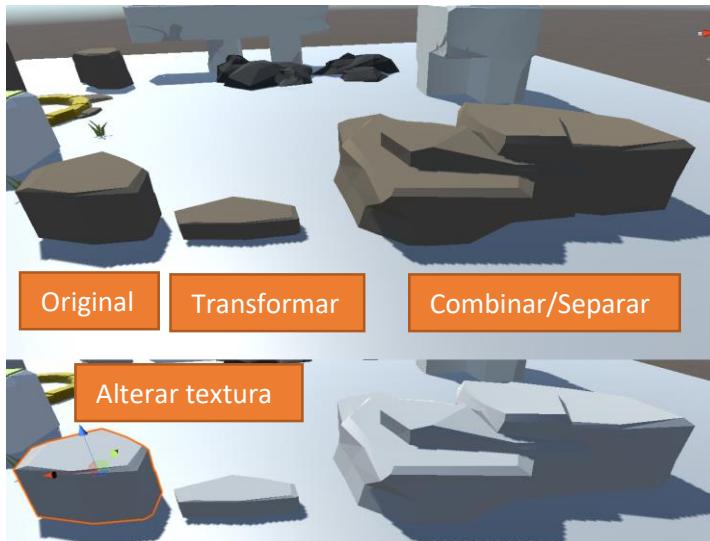


En los escenarios se utiliza el *shader* que aplica sombras a los modelos. A los personajes y enemigos en cambio se usa el que no proyecta sombras, es más óptimo, conserva el color en las zonas oscuras y le da la estética más *cartoon*, en los escenarios se buscan sombras ya que prima detectar el volumen.

- *ENVIRONMENT ASSETS: CREANDO LA DECORACIÓN DE LOS ESCENARIOS.*

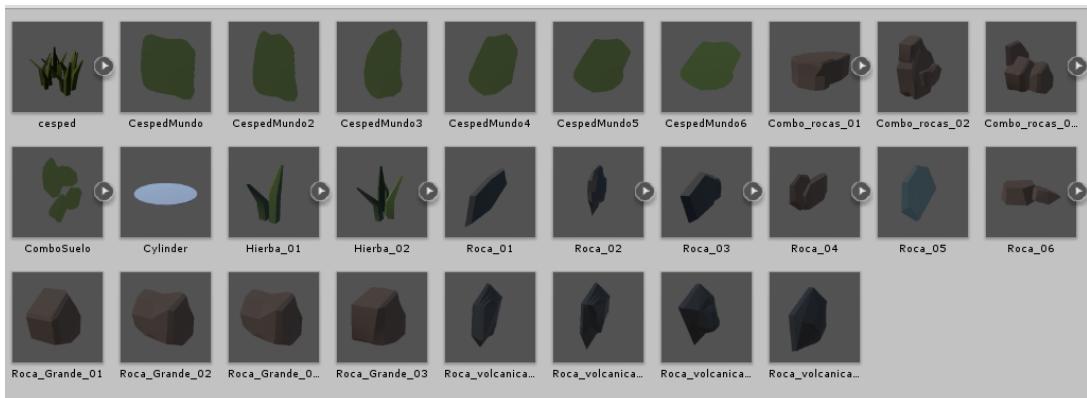
En el proceso de modelar, por un lado, hay que definir los modelos de todos los objetos necesarios para jugabilidad. Sin embargo, para dotar de vida al escenario es necesario crear una serie de *assets* que sirvan de decoración en los distintos niveles. Estos *assets* tienen las siguientes características:

- Modulares: Puedes juntar varios o separarlos si son un conjunto para formar un *asset* nuevo.
- Transformables: Puedes rotarlos, achatarlos, estirarlos, y formar otros distintos.
- Alterar la textura: Una vez creada la paleta, puedes cambiar el tinte para simular otros entornos o cambiar el material entero.

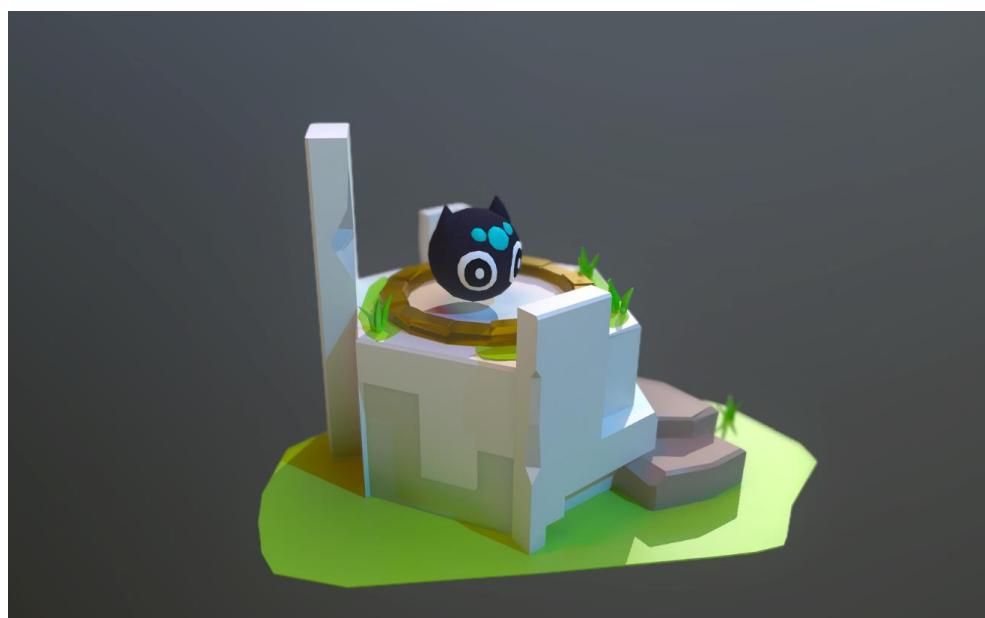


En resumen, crear un catálogo de modelos de manera que una vez importados, se puedan sacar muchas más combinaciones desde el editor del motor sin tener que volver modelar más. Este paso se realizó en las primeras fases de desarrollo, cuando se especificaron las temáticas de los niveles y se fue realizando de manera paralela a todo.

Este es el catálogo de *assets* base a partir de los cuales se han formado todas las combinaciones futuras a lo largo del juego. De suelo de césped eran necesarias muchas para poder dibujar todas las formas que se quieran.



En conclusión, se dispone de un apartado gráfico muy cuidado, que deja asentadas las bases a un futuro desarrollo si se quisiera continuar el juego, el conjunto de modelos y todo el arte creado puede crear una mejor inmersión y aportar una identidad a nuestro juego.



DISEÑO DE AUDIO

Para la música de este juego se buscaba algo que fuera acorde con la temática del astral y de lo onírico. Por lo tanto, en vez de componer temas muy complejos y con muchas notas, con melodías marcadas, se ha optado por unos sonidos y música más ambientales para crear una atmósfera de misterio. De este modo el jugador estará más inmerso en la partida y en el mundo astral del demonio.

Se ha compuesto un tema para el menú y unos sonidos que varían en tono cuando tocas los botones de éste, como evocando las diferentes estrellas que forman las constelaciones. Para los niveles de juego se ha compuesto un tema principal, el cual está acompañado por unos efectos de sonido como el del demonio cuando salta, o el sonido del ángel cuando tocas con él. Cada trozo de portal que recoges también reproduce un sonido que va aumentando de nota cada vez que se recoge un trozo nuevo, creando una escala musical. Al recogerlos todos, un sonido que evoca paz será la prueba de que el portal ha sido activado con éxito.

SISTEMA DE PARTICULAS

En el desarrollo del videojuego se ha implementado un sistema de partículas de Unity para el efecto del portal o meta cuando este está activado.

Dicho sistema se compone de una parte artística formada por un pentágono estrellado decorado y una circunferencia con las capas y filtros necesarios para el efecto deseado.

Estos componentes son añadidos como materiales para partículas aditivas en Unity.

El sistema completo se compone de 3 subsistemas de partículas.

Pentágono estrellado decorado:



Con una sola partícula por ciclo esta es modificada en el tiempo por sus valores de tamaño, giro y color.

Ondas en el suelo:

A cinco partículas por ciclo las circunferencias son modificadas en tamaño y color, haciéndolas desaparecer progresivamente al final del ciclo.



Ondas en el aire:

A cinco partículas por ciclo coordinadas con las ondas en el suelo son modificadas en posición (eje z) y color de la misma forma que las anteriores.



El conjunto de estos subsistemas forma el SP que podemos ver a continuación.



PROGRAMACIÓN

INTERFAZ

Para el diseño de la interfaz se buscaba una navegación más dinámica a la que estábamos acostumbrados, por lo que se hizo uso de un sistema de rotaciones para interactuar por las diferentes pantallas (mencionadas en el apartado artístico)

Encontramos 5 pantallas principales que, siguiendo el *lore* de Demon's Starway se han posicionado a 72 grados de diferencia respecto a los demás (la misma distancia que las puntas de un pentágono estrellado) y te puedes mover por ellos a través de botones.

Todos los componentes del menú exceptuando la cámara la cual es fija se encuentran dentro de un *GameObject* que hace de centro de masas, el cual gira en cada ciclo de la función de Unity *FixedUpdate()* hasta el ángulo deseado marcado por el botón pulsado. Además, la dirección de giro viene marcada por la posición final deseada buscando siempre el menor desplazamiento.

Como ayuda visual el *skybox* gira en sentido contrario a nuestro menú sobrescribiendo su posición con el comando `RenderSettings.skybox.SetFloat("_Rotation", newPos);`.



En el menú principal encontramos inicialmente el logotipo del juego, una rueda de constelaciones, 2 flechas y a nuestro personaje. La rueda gira respecto a su centro de la misma manera que el centro de masas anterior para acomodarse a la posterior o anterior constelación según la flecha pulsada.

A forma de ayuda visual nuestro personaje gira haciendo parecer que observa la ruleta girar en el mismo ciclo y volviendo a su estado inicial guardado en una variable *Quaternion*.

En la ruleta solo podremos pulsar sobre la constelación que esté mirando nuestro protagonista (si además está desbloqueada), esto hará que rotando sobre un *GameObject* posicionado con anterioridad entre en escena una nueva ruleta para elegir el nivel deseado.

Ahora los botones afectan a esta nueva ruleta y no a la anterior (la cual se encuentra debajo) por haber modificado el objeto al que tienen que mandar cambiar las variables que permiten girar la dirección de giro.

Además, este nuevo estado también afecta a nuestro protagonista que se adelanta como ayuda visual y deja paso en su posición a una nueva flecha que permite volver todo al estado anterior con su respectiva transición.

El resto de intención o *feedback* visual con el menú en cualquiera de los botones se produce modificando el contenido a mostrar por las funciones *OnMouseDown()*, *OnMouseUp()* y *OnMouseOver()* que nos proporciona Unity y las variables y funciones propias de los objetos como *SetActive(bool)*.

FUNCIONALIDAD MÁS IMPORTANTE DE LOS SCRIPTS MÁS DESTACADOS

Para controlar el juego usamos unos scripts básicos para controlar el jugador, la gravedad, la cámara, los choques del jugador. Siendo estos los más importantes del proyecto, explicaremos un poco su funcionamiento:

-Control del jugador:

Este script se encarga de aplicar fuerzas al avatar del jugador dependiendo de hacia donde este dirigiendo el jugador con los diferentes controles. Tiene variables públicas para configurarse, siendo las más importantes la fuerza de movimiento, fuerza de salto y fuerza en el aire.

La fuerza de movimiento es la fuerza que se le aplica para moverle por el suelo.

La fuerza de salto es una fuerza que se aplica como impulso hacia el eje de salto que tenga en ese momento.

Y por último la fuerza de movimiento en el aire, se usa para que el jugador no pueda controlar mucho al avatar en el aire para que así una vez haya saltado no pueda cambiar mucho el rumbo del avatar en el aire.

Además, este script reconoce si hay que usar un joystick para versión móvil, y lo habilita para mover el avatar.

-Control de la gravedad por partes:

Este script se ha diseñado para que cualquier objeto que entre en su zona de *collider*, este le aplique una fuerza de gravedad con un determinado vector. Es decir que este script se usa en las 4 diferentes caras del planeta, para que cuando el jugador salte de una cara a otra se le aplique la gravedad determinada de esa cara. Además, también le dice al jugador cuál es su vector de salto.

En esta imagen de la derecha se puede apreciar la zona en la que la cara superior del planeta aplicara gravedad hacia abajo, además las zonas no se solapan entre sí para que no apliquen gravedades múltiples.

Este script tiene como públicas dos variables, un vector3 (xyz), que indica hacia donde se aplica la gravedad, y la fuerza gravitatoria, que es la fuerza que se aplica en la dirección del vector. Para saber a qué objetos tiene que aplicar la gravedad se usan las funciones *OnTriggerEnter* y *OnTriggerExit*, que cuando entra un objeto en su zona le añadirá a una lista y si sale lo quitara. El script en cada *FixedUpdate* recorre la lista aplicando la fuerza a todos sus objetos.



-Control de la cámara:

Para controlar la cámara, se usa un *GameObject* que se usa como pivote para la cámara, siendo ésta su hijo. El pivote se posiciona en la posición del planeta, y éste es el que se rota para a su

vez rotar la cámara, y que así la cámara rote alrededor del planeta. Tiene dos variables públicas de velocidad, una para la velocidad en horizontal y otra para la vertical.

El script reconoce cuando hay que usar un joystick o no, haciendo así que la rotación del pivote sea por el *click* y arrastre del ratón o con el joystick definido para la cámara.

-Choques del jugador:

Este código se encarga de controlar los choques que el jugador hace con los enemigos, puntas de estrella y con la propia salida del nivel.

Lo que hace es usar el método *OnCollisionEnter* y *OnTriggerEnter*, y va comprobando el tag del objeto con el que choca reiniciar la posición del jugador y la cámara, además de restarle una vida si choca con un enemigo, sumar una punta y destruirla para que el jugador la “recoja” y activar el portal de salida con sus partículas cuando el jugador obtiene las 5 puntas de estrella.

Multijugador:

Para hacer el multijugador se ha desarrollado un lobbypersonalizado, que tiene cuatro componentes básicos, un botón para salir, un botón de buscar partida, una zona para poner el nombre de la partida y un *lobbymanager*, que se encarga de unirse o crear la partida y crear los avatares de los jugadores.

Entrando en detalle, el *lobbymanager* se encarga de manejar todo lo del multijugador. Es decir, crea la sala o te une a ella, inicia la partida si los jugadores necesarios están listos cargando la escena correspondiente. Se han reprogramado los métodos más importantes de la clase “*NetworkLobbyManager*” como *OnMatchList*, *OnMatchJoined* y *OnMatchCreate*, para que básicamente funcione como un *MatchMaking* automático, es

dicho, cuando pulsas buscar, el *lobbymanager* lo que hace es buscar si hay una partida ya creada con ese nombre, si es así te une, y si no la crea para que otros jugadores se unan automáticamente. Una vez en la sala con los jugadores necesarios para iniciar la partida, estos pueden pulsar “Ready” para decirle al *lobbymanager* que están listos para iniciar la partida. Y este la iniciara cuando todos estén listos.



Ilustración 1 - Imagen de lobby con partida creada

Una vez dentro, se usan unos scripts diferentes a los del juego normal, tales como ControlMulti y ChoquesJugadorMulti se han reprogramado para el multijugador, además se usa un *prefab* diferente que en el modo 1 Jugador.

Por ahora el multijugador está disponible en Windows y Android (sin controles aún). Siendo el objetivo coger las 5 puntas de estrella y salir el primero por el portal antes que tus compañeros, como una carrera, pudiendo entorpecer a tus enemigos.

El multijugador aún está en Beta porque no se controla por ejemplo las desconexiones, y que hay que hacer si estas pasan, los controles en Android aún no están...

El multijugador admite “*CrossPlay*”, es decir, que juegan usuarios de Windows y Android conjuntamente, pensado para ampliar las plataformas próximamente.

El multijugador no está disponible en WebGL porque es necesario que use websockets y el lobby no está pensado en un principio para usar websockets. Por lo tanto, el multijugador no está disponible en esta plataforma, solo pudiendo acceder a la versión de un jugador. En un futuro se mirará la posibilidad de usar websockets y así unir WebGL, Android y Windows en el multijugador “*CrossPlay*”.

CONTROLES

Demon's Starway dispone de dos formas de control dependiendo de la plataforma en uso.

- PC:
-Menú:

Ratón para desplazarse mediante los botones.

- InGame:

Ratón + Click Izquierdo para rotar la cámara.



Teclas A, S, D, W para mover a nuestro personaje y barra espaciadora para saltar.



- Dispositivo Portable:

-Menú:

Uso táctil para desplazarse mediante los botones.

-InGame:

Joystick virtual para rotar la cámara y mover nuestro personaje y botón virtual para saltar.



OPCIONES

El juego contempla las opciones de desactivar la música y sonidos, así como la posibilidad de cambiar el idioma entre español e inglés.



MEJORAS

Demon's Starway está pensado como un juego completo, aunque en esta versión no está completo por lo que cabe destacar ciertas mejoras en su diseño:

Submisiones: En cada nivel podríamos encontrar al acabarlo una sumisión oculta para dar el incentivo de volver a jugar y completar el juego al 100%. Dichas misiones podrían ser alguna como encontrar el objeto escondido, acabar en menos de X tiempo o llegar a la meta con 3 vidas.

Power-Ups: Aunque entretenida, a largo plazo la mecánica de solo saltar y moverse puede cansar, por lo que añadir cierto power-ups para la resolución de niveles es un punto muy considerado.

Bosses: Enemigos finales al acabar constelaciones que podrían estar implementado como en muchos juegos de Nintendo atacando indirecta o simplemente sobrevivir X tiempo para cansarlos.

Además, en un futuro juego completo se podrían añadir sistema de monetización como *skins* o expansiones.

CRÉDITOS

Todas las imágenes, modelados 3D, texturas, música y código han sido creados por Slothery Plus Black.

ENLACE A NUESTRO GITHUB

<https://github.com/slothery-plus-black/Demon-s-Starway>

ENLACE A NUESTRO PORTFOLIO

<https://slothery-plus-black.github.io/Portfolio/>

ENLACE AL JUEGO WEB

<https://slothery-plus-black.github.io/Demon-s-Starway/build/web/index.html>

ENLACE A VERSIONES DESCARGABLES

<https://github.com/slothery-plus-black/Demon-s-Starway/releases>