



# PROYECTO – JUEGO 3D

Ismael Aguilar Cabello Ismael Aguilar Cabello

# ÍNDICE

- [Introducción](#)
- [Justificación del Proyecto](#)
- [Objetivos del Proyecto](#)
- [Módulos, Herramientas y Lenguajes que serán usados](#)
- [Las fases del Proyecto](#)
  - [Inicio, instalación y preparación del entorno \(¿Qué cosas básicas necesitaremos?\).](#)
  - [Programación de las mecánicas del juego \(¿Cómo funcionan?\).](#)
  - [Diseño de los niveles y jugabilidad.](#)
- [Bibliografía](#)
- [Conclusiones, posibles mejoras y despedida.](#)

# INTRODUCCIÓN

En este proyecto se llevará a cabo la creación de un videojuego 3D en primera persona mediante el uso de Unity. La jugabilidad girará en torno a la idea del parkour, permitiéndole al usuario final disfrutar de una movilidad fluida a la par que satisfactoria.

El jugador deberá recorrer el entorno haciendo uso de las distintas mecánicas de las que dispondrá para alcanzar la meta. El juego acabará cuando el jugador llegue a la meta, pudiendo comprobar cuanto tiempo duró la partida, alimentando así las ganas de querer jugar de nuevo para batir su propio récord.

## JUSTIFICACIÓN

¿Por qué he decidido hacer este proyecto? En un principio este iba a ser mi plan original, pero acabé optando por otra alternativa que me llamaba más la atención. Dicha alternativa generó demasiados inconvenientes así que esta vez decidí volver a la idea original; Un Juego de Parkour.

Durante el curso confirmé que la herramienta que más me gustaba era Unity, el lenguaje (en C) no es excesivamente complicado, la interfaz es muy completa y ofrece muchas posibilidades, por no hablar del gran apoyo que esta plataforma recibe de la comunidad, pudiendo descargar todo tipo de Assets.

Además, me gustan mucho los juegos que mezclan la vista en primera persona con una amplia movilidad y fluidez. Este tipo de videojuegos tuvieron en su día un gran Debut, con excelentes críticas y grandes ganancias, Juegos como Titanfall lograron sobreponerse a otros competidores como la saga Call of Duty, que inmediatamente trató de imitar la verticalidad en sus mapas. De hecho, a día de hoy, muchos juegos se han reinventado o han implementado esta movilidad y verticalidad en sus FPS, como puede ser el caso de Doom, Overwatch o Dishonored.

Este juego se inspira sobre todo en **Mirror's Edge**, un apremiado videojuego de parkour en primera persona que crea a la perfección la sensación de libre movimiento y la idea de explorar rutas alternativas que sean más rápidas. Nuestro objetivo será obtener esa misma esencia. En el capítulo de objetivos puedes ver dos imágenes de Mirror's Edge para entender mejor el concepto.

# OBJETIVOS

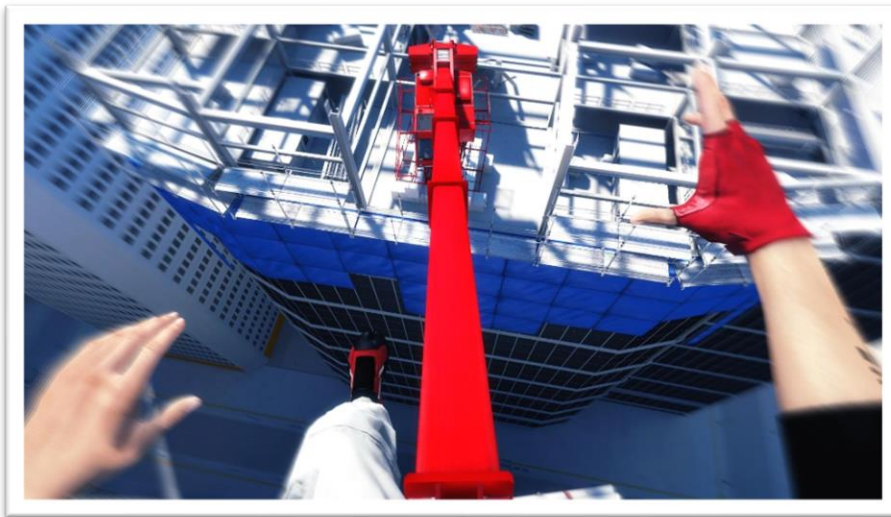
El objetivo es crear un juego que se sienta fluido, dinámico y desafiante, las mecánicas del mismo serán las grandes protagonistas, siendo el diseño artístico una meta secundaria. En orden de prioridad podríamos clasificar todos los objetivos así:

- 1- **Creación de las mecánicas de movilidad**, es vital lograr un movimiento en primera persona, que no genere conflictos, transmita fluidez y que permita al jugador completar el nivel. Las principales mecánicas serán:
  - **Correr**: aumentando la velocidad del jugador y dando pie a poder concatenar el resto de mecánicas.
  - **Salto**: parte fundamental de la jugabilidad, aplicando gravedad al jugador, permitiéndole caer y concatenar otras mecánicas como, por ejemplo, correr y luego saltar, alcanzando así una plataforma lejana, o saltar en dirección a una pared, pudiendo escalarla o correr por ella.
  - **Parkour**: Acciones que permitirán al jugador aprovechar su entorno para resolver el nivel. Las más relevantes son tres: correr por las paredes o por cualquier superficie vertical (pero de forma limitada, el jugador no podrá correr infinitamente de esta forma), por otro lado, el jugador podrá escalar aferrándose a salientes dentro de su alcance (habrá un límite de altura que no podrá escalar) y, por último, deslizarse y agacharse para pasar a través de aperturas más pequeñas o descender rápidamente por planos inclinados.
- 2- **Diseño de los niveles**, para poder hacer uso de las mecánicas descritas anteriormente, será necesario diseñar los distintos niveles del juego, los cuales contendrán en su escenario todo tipo de elementos para ser aprovechados por el jugador o para obstaculizarlo. Estos elementos pueden ser paredes lisas, plataformas, escaleras de mano, saltos de longitud, zonas de gran altura, caídas (tanto al vacío como hacia otras zonas), etc... Así mismo, una parte esencial del juego será la posibilidad de poder escoger distintas rutas para completar cada nivel, siendo algunas más fáciles, otras más difíciles, otras más rápidas y otras más lentas.
- 3- **Tiempo**: Se medirá el tiempo que el jugador tarda en completar el juego mediante un cronómetro situado en la parte superior del UI. El objetivo es alentar al jugador a querer

repetir el juego para batir sus anteriores marcas, fomentando la búsqueda de nuevas rutas más rápidas o directas e incrementado considerablemente la dificultad (ya que cada fallo supondrá una pérdida de tiempo).

- 4- **Diseño artístico:** Como último objetivo menos prioritario, se buscará recrear unos niveles que sean fácilmente reconocibles para el jugador, optando por un diseño más minimalista, compuesto por texturas planas de colores muy sólidos. Por otro lado, el decorado del mundo y el escenario “no jugable” si que tendrá más detalles y elementos inmersivos que darán profundidad al trasfondo del juego, sin llegar a interferir mucho con la jugabilidad y los niveles.

Algo similar a esto es a lo que se aspira:



# **HERRAMIENTAS, MÓDULOS Y LENGUAJES USADOS**

En este Proyecto intervendrán múltiples materias de las dadas a lo largo del grado, las cuales serían:

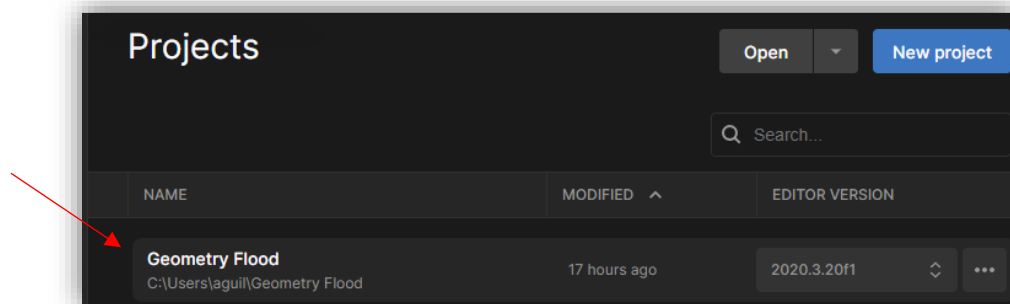
- Programación Multimedia y Dispositivos móviles: siendo esta la asignatura más importante ya que incluye:
  - El uso de Unity como principal entorno y motor para el videojuego.
  - El uso de C# como lenguaje de programación.
  - Nociones y conocimientos sobre el diseño de niveles y el entorno del juego.
- Programación: Como ya hemos dicho, en C#.
- Entornos de desarrollo: Unity.
- Inglés: Gran parte de la documentación usada para este proyecto proviene de fuentes de habla inglesa.
- Webs y Marketplaces para adquirir Assets creados por la comunidad de Unity, en la Bibliografía se incluye detalladamente los que se han usado.

Para este videojuego no hará falta ningún tipo de acceso a Datos, por ello no será necesario el uso de bases de datos.

# FASES DEL PROYECTO

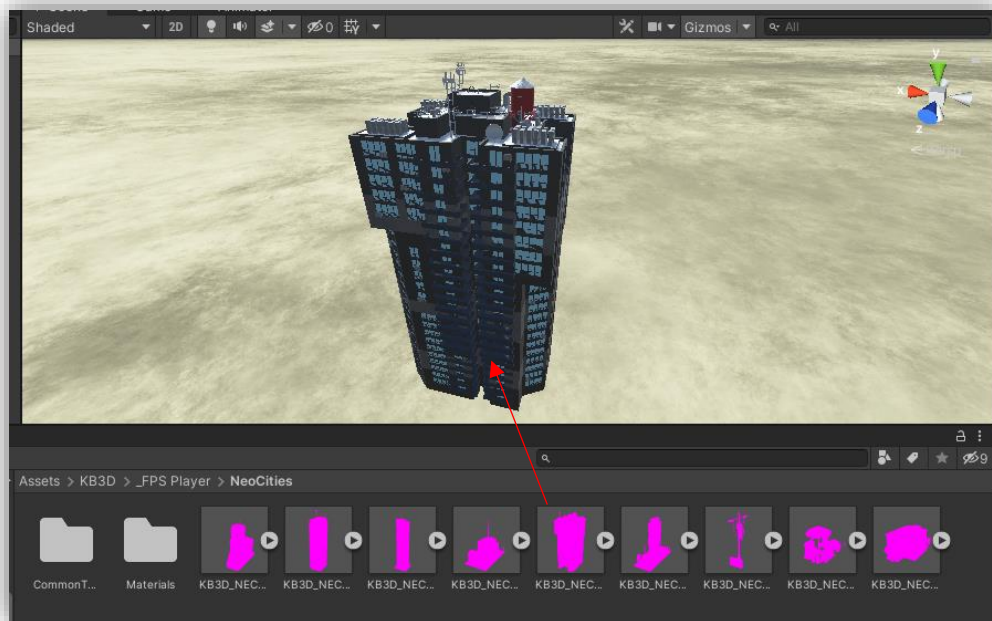
## INICIO Y PREPARACIÓN DEL ENTORNO

¿Qué recursos básicos se necesitarán para empezar con el proyecto? Necesitaremos esencialmente tres cosas, lo primero será configurar Unity y crear un nuevo proyecto con el nombre de nuestro juego. El juego se llamará Geometry Flood, un nombre que tendrá que ver con su trasfondo e historia (detalladas en el GDD).



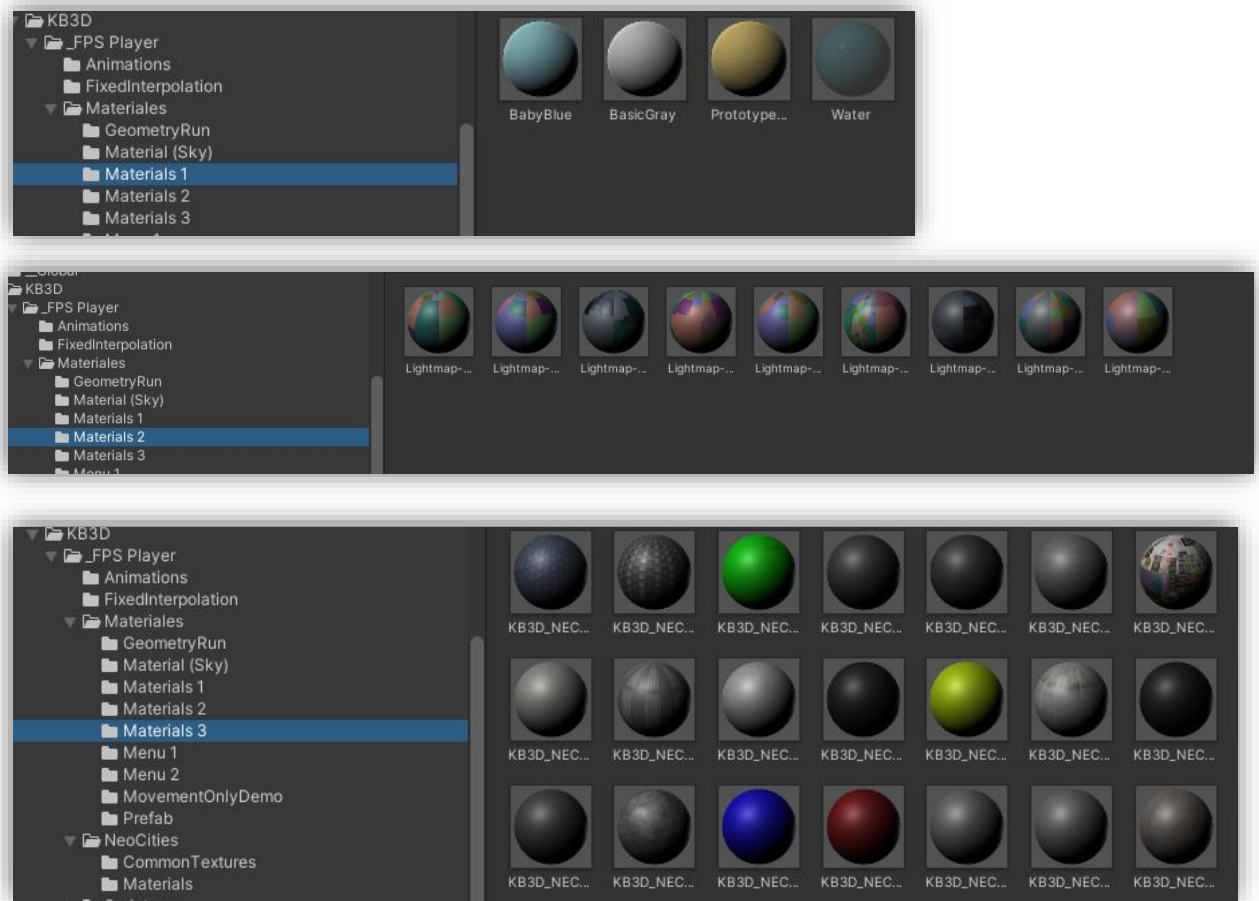
Lo segundo será importar una serie de assets que nos serán útiles a lo largo de la fase de diseño. Estos assets contienen texturas y materiales para las superficies, prefabs de edificios para el escenario e incluso texturas para cambiar el cielo (la Skybox).

### Prefabs de edificios:



Obtenidos de los assets de NeoCities, permiten ponerles las texturas y materiales que deseemos, dispondremos de más de nueve estructuras distintas para nuestro escenario.

**Materiales para superficies:** Concretamente usaremos 3 carpetas llenas de materiales, cada una contiene una serie de materiales determinados, siendo la primera para materiales muy simples, la segunda para materiales que mezclan muchos colores y la tercera para materiales de los edificios (de los Assets de NeoCities).



**Materiales para la Skybox:** Nos permiten cambiar la imagen del cielo:

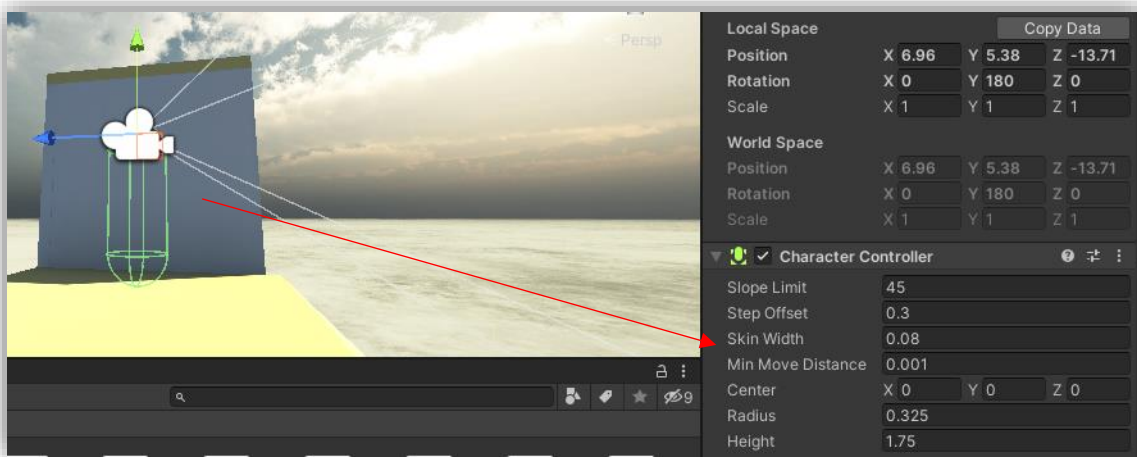




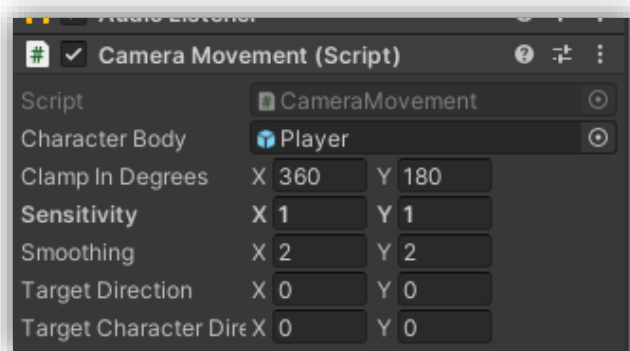
## PROGRAMACIÓN DE LAS MECÁNICAS DE JUEGO

¿Cómo funcionarán estas mecánicas? Lo primero es identificar todas las acciones que el jugador llevará a cabo, tal como ya hemos hablado en el capítulo de objetivos, queremos que el jugador pueda, caminar, correr, correr por las paredes, encaramarse a salientes, agacharse y deslizarse. Vamos a repasar cada uno de estas acciones y en general cómo funcionan.

**El jugador:** Este cuenta con un rigidBody y todo tipo de controladores (Character Controller) que le permitirán interactuar con el entorno. En la imagen de abajo podemos ver lo que sería el cuerpo del jugador y sus distintos parámetros básicos:



Su cuerpo es básicamente un cilindro cerrado con dos semiesferas tanto por arriba como por abajo. En su parte superior situamos la cámara, la cual cuenta con su propio Script para poder moverla a voluntad y poder cambiar parámetros como por ejemplo la sensibilidad:



**El Movimiento:** Para que el jugador pueda moverse y llevar a cabo todas las acciones de Parkour necesitará muchos scripts y parámetros. Detallar todo el código y funcionamiento de cada uno alargaría demasiado el capítulo, por lo que se explicará lo esencial y necesario para poder entenderlo.

Para empezar, el Script principal del movimiento será “**PlayerMovement (Script)**”, el cual almacenará las variables de velocidad principales, siendo estas **Walk Speed** (Velocidad al caminar), **Run speed** (velocidad al correr), **Crouch Speed** (Velocidad estando agachado) y **Jump Speed** (Fuerza del salto). Aparte de estos parámetros también controlaremos la gravedad del jugador, la condición de Grounded (es decir si está o no tocando el suelo) y el **Anti Bump Factor** el cual es una fuerza que interactúa con el jugador cuando este colisiona con un objeto, su objetivo es el de atenuar el impacto del jugador, dándole una mejor fluidez al chocar y eliminando en la medida de lo posible sensaciones ortopédicas.



El siguiente Script importante es el “**Player Input (Script)**” el cual básicamente recoge las distintas entradas de teclado y ratón, estas son usadas luego por el “**Player Controller (Script)**” que se encargará de distribuir las distintas acciones a sus respectivos Scripts especializados y de gestionar algunos parámetros. Los controles del jugador se resumirían así:

- Movimiento: teclas A S W D
- Saltos: Barra espaciadora.
- Correr: Click izquierdo del ratón.
- Agacharse: Tecla C.

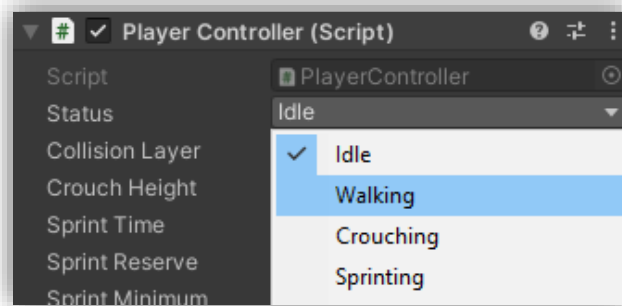
```
public bool run
{
    get { return Input.GetMouseButton(0); }
}

public bool crouch
{
    get { return Input.GetKeyDown(KeyCode.C); }
}

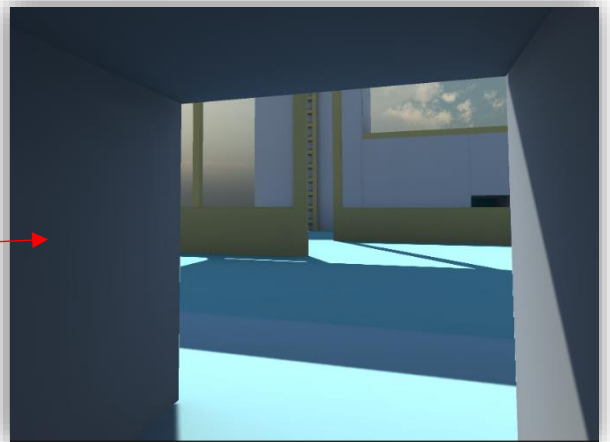
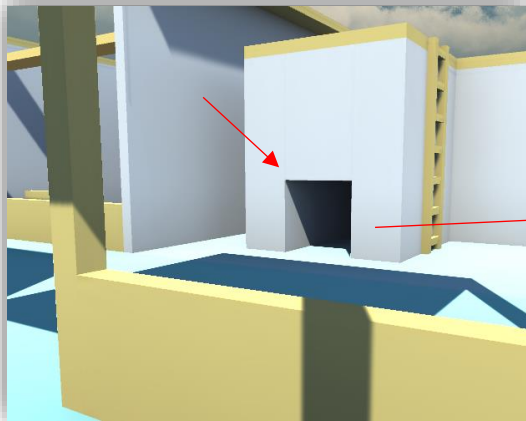
public bool crouching
{
    get { return Input.GetKey(KeyCode.C); }
}
```

Continuamos con el antes mencionado “**Player Controller Script**” el cual no solo dirige las funciones del jugador a otros scripts, si no que también controla los siguientes parámetros:

- **Status:** Parámetro que recoge las distintas animaciones del jugador, estas serían: caminando, quieto, corriendo, corriendo por la pared, agachado, deslizándose, escalando un saliente, quieto (aferrado) en un saliente y usando una escalera de mano.



- **Crouch Height:** Es la altura del jugador cuando este está agachado, la cual será obviamente menor, permitiendo así que pueda atravesar orificios de poca altura:



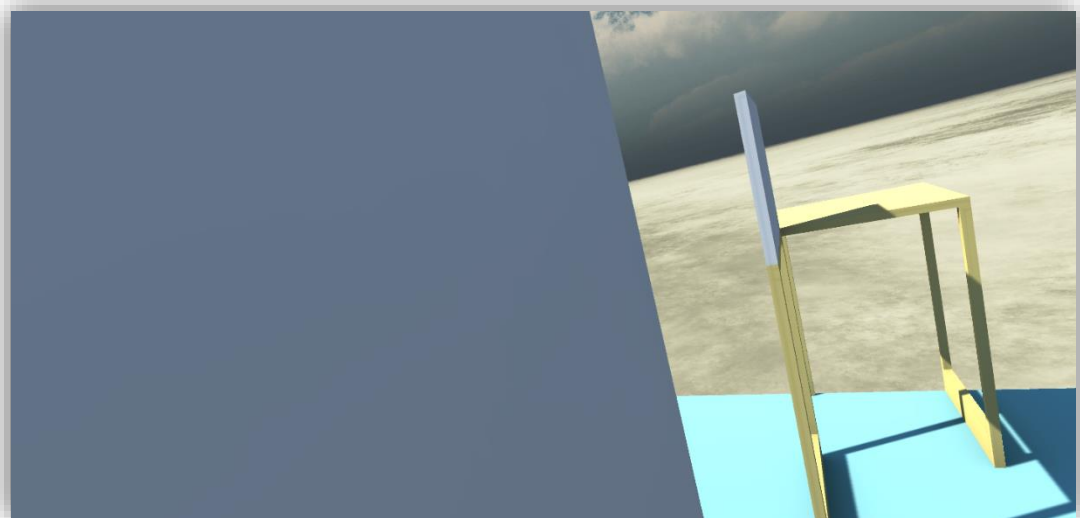
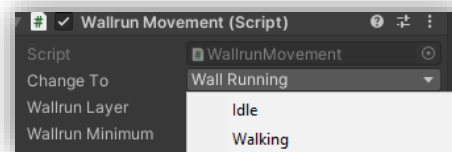
- **Sprint Time:** Sirve para imponer el tiempo que el jugador puede estar corriendo antes de “Cansarse” (es decir detener la animación de correr y pasar a la de caminar). En un principio daremos unos 60 segundos al jugador, esto le permitirá correr el suficiente tiempo como para no tener que descansar en mitad de cada recorrido entre distintas pruebas.
- **Sprint Reverse:** Simplemente es la velocidad a la que el jugador puede moverse en una dirección hacia la que no está mirando, algo así como correr de espaldas o hacia los lados sin mirar directamente hacia ellos. Esta velocidad será mayor que la de caminar, pero menor que la de correr (“hacia delante”).



Esos serían los Script principales, ahora toca ver los scripts específicos de cada acción de las mencionadas:

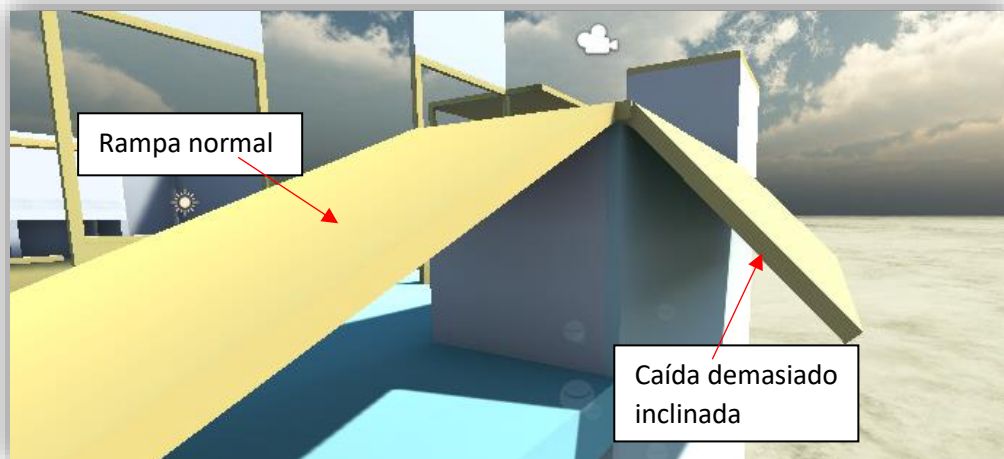
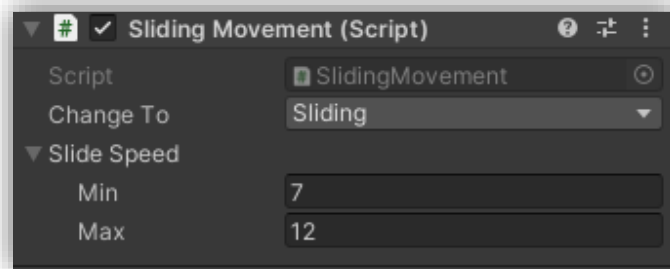
- **Wallrun Movement (Script):** El Script encargado de controlar el movimiento al correr por la pared, en esta parte considero importante detallar como se ha creado la sensación de este movimiento. El cuerpo del jugador tiene implementado tres figuras que no son visibles ni rígidas como tal, dos de ellas situadas a los costados y la otra saliendo desde el centro. Cuando las figuras de los costados entran en contacto con una superficie vertical y el jugador no está Grounded (tocando el suelo) se desencadena la acción de correr por la pared.

Dicha animación lo que hace es rotar la cámara para que se vea como si nos estuviéramos inclinando al apoyarnos en la pared. Aquí entra en juego la variable **Change To**, la cual funciona de manera muy similar a Status, lo que hace es cambiar de animación dependiendo de lo que suceda, pudiendo saltar desde la pared o desprendernos de ella (si dejáramos de movernos). Una cosa a tener en cuenta es que cuando corremos por una pared siempre reseteará el salto, esto permite al jugador saltar y ganar altura de manera relativamente ilimitada, siempre y cuando disponga de las suficientes paredes entre las que saltar. También existe una variable que mide la velocidad a la que nos estamos moviendo (Wallrun Minimum), si detecta que no nos estamos moviendo lo suficiente, el jugador caerá de la pared.



- **Sliding Movement (Script):** Este Script corresponde al movimiento mientras nos deslizamos. Cuando el jugador se encuentra en la animación de correr, puede pulsar la tecla C para agacharse y de esta forma realizar un deslizamiento. Este movimiento permite mantener la velocidad al mismo tiempo que traspasamos un orificio de baja altura. Las principales variables del Script serían (de nuevo) **Change To** y **Slide Speed**, esta última cuenta un **mínimo** y un **máximo** y es la responsable de mantener la velocidad de esta acción.

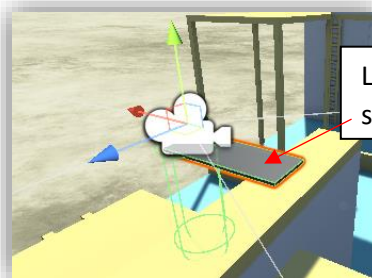
Cuando caigamos en una rampa (plano inclinado) demasiado inclinada, el jugador entrará automáticamente en modo deslizamiento durante todo el recorrido de la pendiente o hasta que este decida saltar. Esto también se puede hacer si nos deslizamos voluntariamente por cualquier rampa (basta con que este un poco inclinada). Cuanto más tiempo estemos deslizándonos por una rampa más irá aumentando la velocidad (hasta alcanzar el máximo)



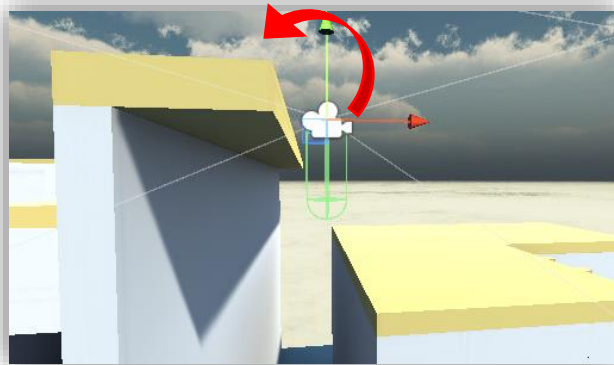
- **Ladder Movement (Script):** Este script controla el movimiento por las escaleras de mano, básicamente se puede subir o bajar siempre y cuando el jugador este en contacto con la escalera. Además, podremos pararnos sin problema en mitad del movimiento, pudiendo observar a nuestro alrededor o incluso saltar desde ella. En

este caso simplemente usamos los vectores verticales para subir y bajar, no hará falta ninguna velocidad, por ello este Script solo cuenta con la variable **Change To**.

- **Grab Ledge Movement (Script):** Este Script permite al jugador trepar por un saliente o una pared (siempre y cuando alcancemos el borde superior). Recordando explicaciones anteriores se menciona que el cuerpo del jugador tiene pegado 3 figuras, estando una de ellas en el centro. Esta figura sobresale hacia delante (observe la imagen), al estar en contacto con un borde o saliente y pulsar la barra espaciadora, se iniciará la animación de escalar, la cual hace que el cuerpo del jugador suba y después avance rápidamente para dar la sensación de haber trepado. En este Script tenemos de nuevo la variable **Change To**.



La figura central sería algo así

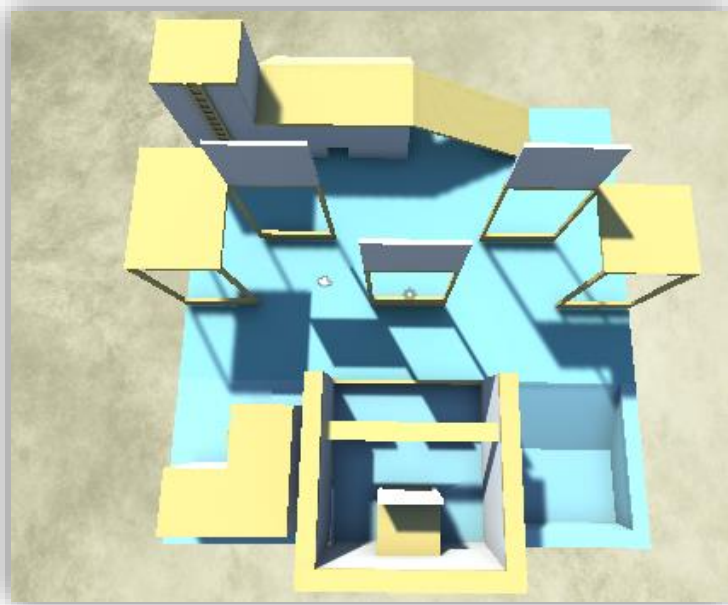


- **Climb Ledge Movement (Script):** Este Script está muy ligado al anterior, en este caso lo que hace es permitir al jugador quedarse agarrado a uno de los mencionados salientes. El jugador podrá observar tranquilamente desde esa posición, desde la cual podría desprenderse, saltar en una dirección o iniciar la animación de escalar. Al igual que antes, este Script cuenta con la variable **Change To**.

## DISEÑO DE LOS NIVELES Y JUGABILIDAD

El juego contará con dos principales opciones, la primera sería el modo tutorial, en ella se dispone un pequeño escenario que cuenta con muchos elementos que ayudaran al jugador a aprender las mecánicas de juego. A continuación, explicaremos el mapa:

Vista aérea:

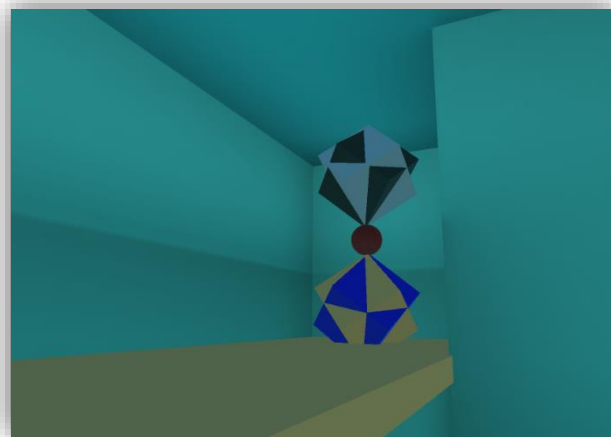


El jugador dispone de: **6 vayas para ser saltadas**, **3 rampas para deslizarse**, **2 escaleras de mano**, **2 huecos estrechos por los que pasar agachado** y **2 pruebas de saltar por las paredes**.





Aparte del circuito de práctica, hay una serie de túneles por debajo a los que se accede mediante uno de los agujeros estrechos. Al final del recorrido subterráneo, encontramos un “Easter Egg” o huevo de pascua con un curioso aspecto geométrico que cobra sentido dentro del trasfondo del juego (Consultar el GDD para más información).



Eso sería todo el tutorial, ahora toca analizar el juego de verdad, el cual tiene un considerable contraste estético con el tutorial. Esto se explica más en detalle en el GDD (Game Design Document)

El juego principal se compone de hasta 5 pruebas clasificadas en colores: **La prueba negra**, **la amarilla**, **la azul**, **la roja** y **la verde**. Cada una de estas pruebas tiene un propósito y diseño distinto al que el jugador tendrá que adaptarse.

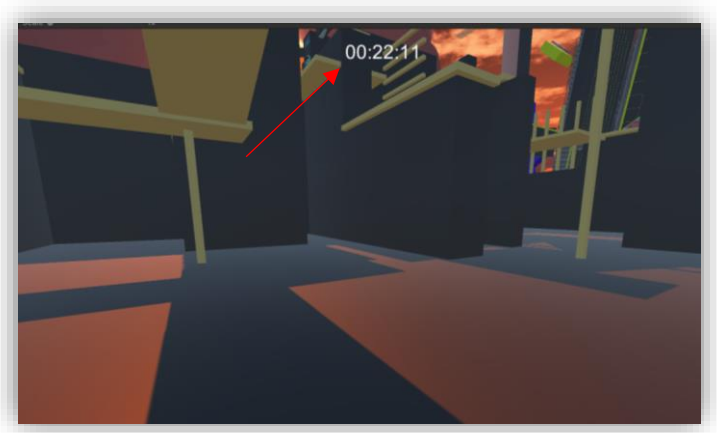
El objetivo de las pruebas es pasar a la siguiente mediante un portal que nos teleporta directamente. Para llegar a estos portales hará falta resolver cada circuito. En la UI del jugador podremos ver en la parte superior, un cronómetro que cuenta los milisegundos, segundos y minutos que tardamos en resolver el juego.



Este tiempo solo se detiene cuando llegamos al final, en un principio no supone ninguna condición para el jugador. Este puede tardar todo lo que quiera en completar el juego, puede tomárselo con calma y explorar todas las posibilidades, rutas y mecánicas sin preocuparse.

La idea es que una vez completado el juego, el jugador intente volver a completarlo batiendo su anterior récord de tiempo. Es aquí donde entra la verdadera dificultad del juego, pues si lo jugamos tranquilamente sin preocuparnos por el tiempo, notaremos que la dificultad no es muy alta, de hecho, se podría decir que es incluso fácil. Sin embargo, al intentar completarlo en menos tiempo, tenderemos a cometer más fallos, a tomar decisiones apresuradas, a escoger rutas más difíciles, pero más rápidas y en general será cada vez más complicado batir nuestras propias marcas.

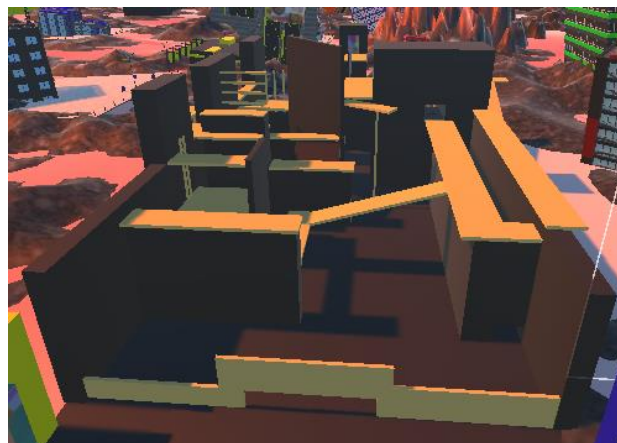
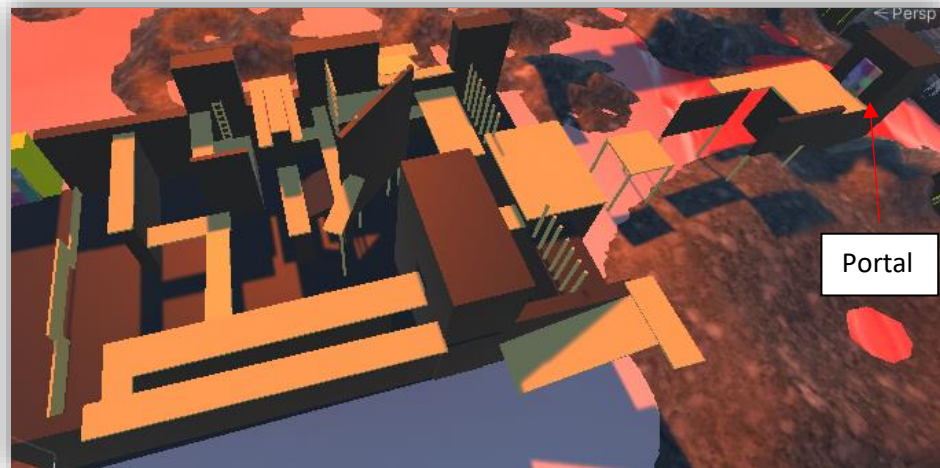
Aun así, el juego no está pensado para durar mucho, la idea es que sea rápido para poder repetirlo una y otra vez. Para que nos hagamos una idea, un jugador impaciente y con poca habilidad o que simplemente se tome su tiempo, podría tardar en su primer intento entre 10 y 20 minutos en resolverlo, mientras que jugadores más hábiles y acostumbrados a estos juegos podrían completarlo entre 5 y 10 minutos. En mi caso, mi **tiempo récord** es de **4 minutos y 7 segundos** teniendo en cuenta que cometí varios fallos durante el recorrido.



Otros amigos y conocidos que han probado el juego también suelen acercarse a los 5 minutos de tiempo récord. De las 5 personas más que participaron en el testeo del juego, solo 1 logró batir la barrera de los 4 minutos con un tiempo de **3 minutos y 43 segundos**. Siendo hasta la fecha el mayor record. Se puede observar en la siguiente tabla los tiempos de cada participante:

Nombre	Primer intento	Record
Fran	17 minutos 48 segundos	3 minutos 43 segundos
Samuel	34 minutos 15 segundos	4 minutos
César	14 minutos 39 segundos	6 minutos 17 segundos
Noa	22 minutos 24 segundos	7 minutos 48 segundos
Rafa	22 minutos 37 segundos	-----

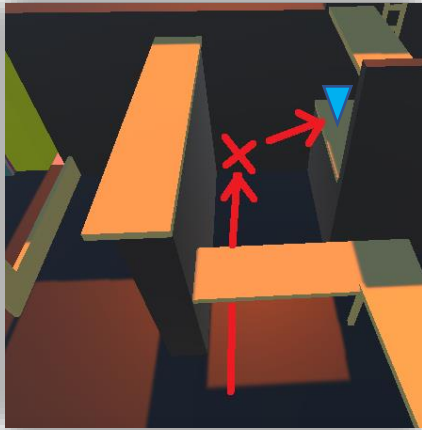
Empecemos analizando la primera prueba, la prueba negra. Tal como se puede observar en la imagen de abajo, predomina el color negro (para identificar la prueba). Esta consiste en un circuito algo similar al tutorial:



Esta primera prueba tiene el propósito de permitir al jugador resolverla con calma, explorando sus muchos recorridos para que este pueda trazar el que crea más eficiente. En cierto sentido, esta prueba es como un segundo tutorial, su dificultad no es muy alta, aunque no por ello es aburrida. La idea es transmitir tranquilidad y una primera toma de contacto asequible con el jugador.

La prueba contiene casi todos los elementos y usos de mecánicas posibles, saltos de longitud, saltos entre paredes, deslizamientos y zonas para agacharse, zonas donde hay que escalar, una escalera de mano, etc.

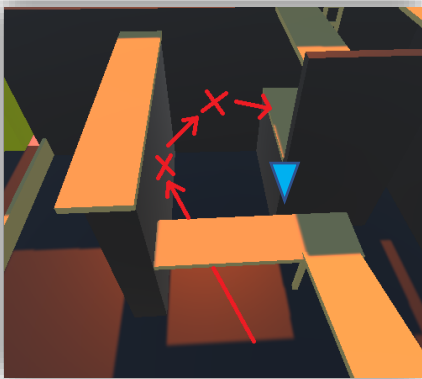
A continuación, explicaremos varias mecánicas o formas de completar este circuito. Una de las cosas más importantes en este juego es lo que yo llamo “Ganar altura”, una mecánica que consiste en aprovechar el reseteo de cada salto tras correr por una pared, acumulando más altura y pudiendo así alcanzar zonas altas. Veamos un ejemplo en este circuito:



Tal como vemos en la imagen de la izquierda, el jugador quiere llegar a la plataforma donde está situado el triángulo azul. Para ello intentará saltar en la pared de enfrente donde está situada la X roja y correrá por dicha pared para saltar y alcanzar la plataforma.

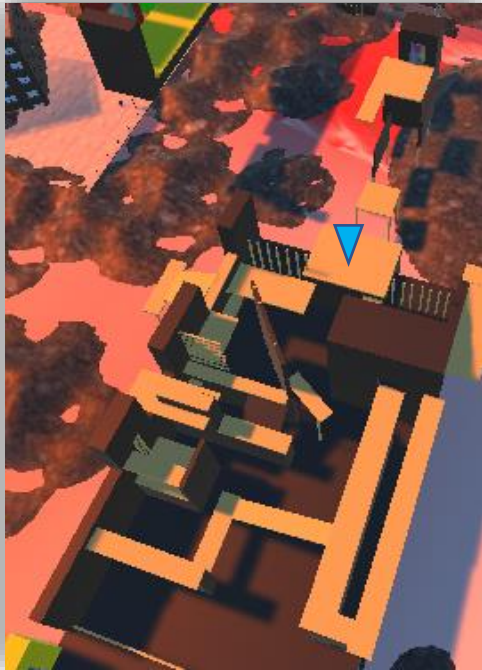
Por mucho que lo intente, el jugador no logrará alcanzar la plataforma de esta manera, ya que la altura que gana con un solo salto en la pared no es suficiente para alcanzar su objetivo.

Para poder llegar a la plataforma, el jugador tendrá que **acumular más altura**, para esto podría por ejemplo hacer algo como lo que se observa en la nueva imagen de la izquierda.



Lo que se hace, es aprovechar dos paredes en lugar de solo una, corriendo primero por la de la izquierda y saltando hacia la de enfrente. De esta forma ganamos la suficiente altura para llegar a la plataforma con el último salto.

Esta mecánica de “Ganar altura” estará, de una manera u otra, presente en todas las pruebas.



A la izquierda podemos ver una imagen en la que hemos señalado los dos principales objetivos. El triángulo azul indica la plataforma en la cual convergen todas las posibles rutas a seguir del circuito.

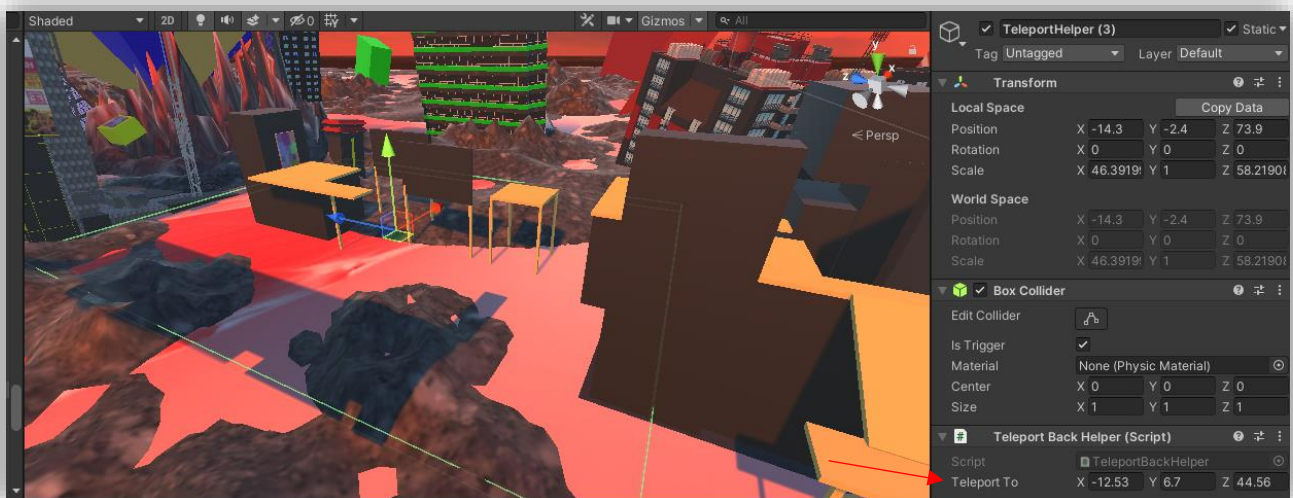
Una vez se llegue a dicha plataforma se deberá dar un salto de longitud, seguido de varios saltos por pared, tras lo cual podremos acceder a la segunda prueba mediante el portal.

Pero ¿Qué pasa si nos caemos? O ¿Qué pasa si el jugador intenta escapar fuera de los límites de la prueba?

En tal caso, será teletransportado al comienzo de la prueba o en su defecto a una plataforma que sirva como “punto de control”. En el caso de esta prueba el punto de control sería la plataforma del triángulo azul.

¿Cómo funcionan estas teleportaciones? Quizás debieran estar explicadas en el apartado anterior de programación, pero considero que es mejor explicarlas en este punto ya que forman parte de la jugabilidad y el diseño de niveles.

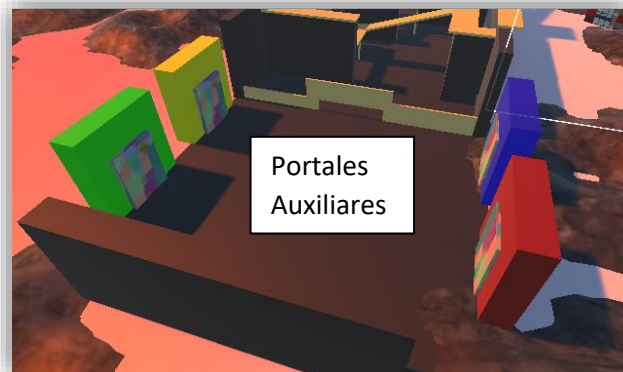
Estas funcionan mediante unas figuras invisibles llamadas **TeleporterHelpers** que reaccionan al entrar en contacto con el jugador, cambiando los parámetros de su posición en el mapa (las coordenadas X, Y, Z) por unas coordenadas establecidas en cada TeleporterHelper,



dichas coordenadas se establecen mediante la variable **Teleport To**. De esta forma funcionan tanto los portales como las caídas al vacío:

Además, todas las pruebas (excepto la prueba Roja), tienen en su zona inicial una serie de portales hacia el resto de pruebas. El propósito de estos portales es sobre todo el facilitar la defensa de este proyecto, pues esta solo dispone de 15 minutos para ser realizada, estos portales auxiliares simplemente son un seguro por si la falta de tiempo dificultara mostrar todas las pruebas o por si el tribunal quisiera volver rápidamente a alguna de las pruebas durante los 15 minutos posteriores a la defensa. Por otro lado, también serían útiles para los jugadores que simplemente quieran volver a una prueba concreta para analizarla.

Por supuesto, un jugador podría hacer trampas y usarlos para completar el juego desde la última prueba en cuestión de 1 minuto o menos. En tal caso no se le impedirá hacerlo, pues la filosofía de este juego es permitirle al jugador completarlo como el quiera, dotarle de libertad para usar el método que el desee, por muy “cuestionable” que este sea. Eso sí, este “exploit” (aprovechamiento de una función que no debería ser usada así) carecería de todo mérito y no ha sido admitido en los resultados de los records de tiempo durante los testeos.



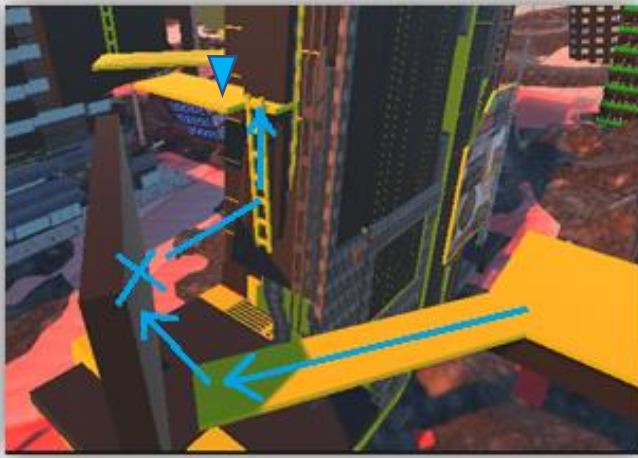
La segunda prueba sería **la prueba amarilla**. Justo tras haber sido teleportado desde la prueba anterior, apareceremos en una plataforma y tendremos que hacer un circuito que gira entorno a un edificio. La prueba va aumentando la altura conforme más nos acercamos a su final y tiene un recorrido muy “Lineal”.

En la anterior prueba el recorrido no era del todo lineal, ya que podíamos escoger multitud de rutas y formas para resolverlo. En este caso, la prueba amarilla si que plantea un recorrido claro y con pocas variaciones. El propósito de esta prueba es de ser: **“Breve pero Intensa”**, su duración es bastante baja si se realizan bien los movimientos, sin embargo, un fallo supondrá un gran riesgo, pues el jugador podría caer al vacío y ser teleportado al principio, teniendo que repetir la prueba una y otra vez hasta hacerla perfecta.



Por ello esta prueba podría considerarse la segunda más difícil, sobre todo para cuando deseemos hacer un **Speed Run** (completar un juego o nivel en el menor tiempo posible), pues la tensión y el miedo a fallar jugará en contra del jugador. Esto es algo que se pudo verificar completamente durante el testeo, todos los jugadores fallaban más en este tipo de prueba que en otras, pasando incluso más tiempo aquí que en otras pruebas con recorridos más largos, pero menos arriesgados.

Eso sí, una vez dominado el arte de la paciencia, la calma y la buena letra, se puede completar esta prueba en menos de 1 minuto y con una facilidad un tanto irónica.

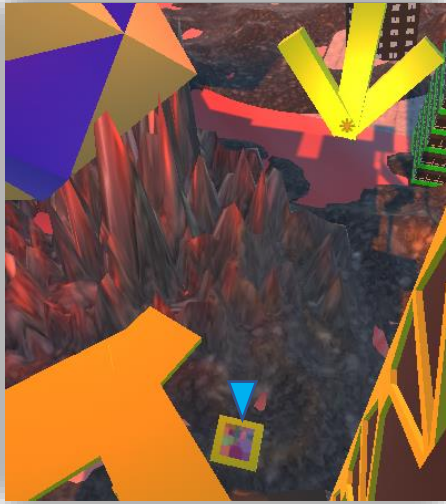


En esta prueba implementamos un nuevo tipo de desafío. Antes hemos visto la mecánica de “Ganar altura”, la cual también estará presente aquí, sin embargo, una nueva mecánica dificultará el paso: “El Salto de Escalera”.

Consiste en saltar hacia una escalera para encaramarse a ella, trepar, y continuar el recorrido tal y como vemos en la imagen de la izquierda.

Por supuesto este tipo de salto se complica mas cuando introducimos un salto previo desde la pared. Durante los testeos, estos saltos fueron sin duda los que más complicaciones supusieron para los jugadores. En la siguiente imagen se puede observar parte del recorrido:



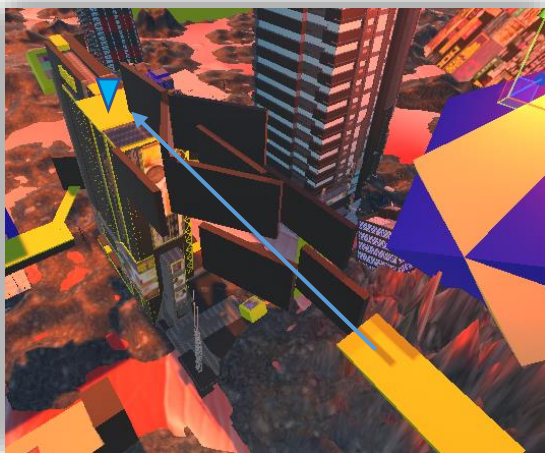


Tras pasar por varios saltos más, llegamos a esta zona que puedes observar en la imagen de la izquierda. Esta zona es llamada “El salto de Fe”, pues una enorme flecha nos indica que debemos saltar al vacío para caer en un portal que nos permitirá continuar con la prueba.

Esta zona es justo el último paso para llegar a un punto de control dentro de la prueba, fallar aquí supondría una gran pena. Considero que esta

zona representa mucho la filosofía de esta prueba, pues el salto es increíblemente fácil, lo único que hay que hacer es caminar hacia delante y ajustar un poco la dirección durante la caída.

Sin embargo, ponerse nervioso y llevar a cabo un salto o alterarse durante la caída y moverse demasiado provocará que no se atine en el portal y que tengamos que volver de nuevo al principio. Una vez hecho bien ese salto pasamos a la última parte de esta prueba.



En esta parte simplemente tendremos que ir saltando entre distintas paredes hasta llegar al triángulo azul, para ello habrá que ir acumulando altura y evitando caer al vacío.

En caso de caer apareceremos devuelta en la plataforma amarilla cercana (bastante tortura sería empezar desde el principio). La gracia de esta zona es que

también es muy fácil, pero el vértigo al estar a máxima altura y no tener plataformas en las que reposar crean de nuevo una inconveniente tensión para el jugador.

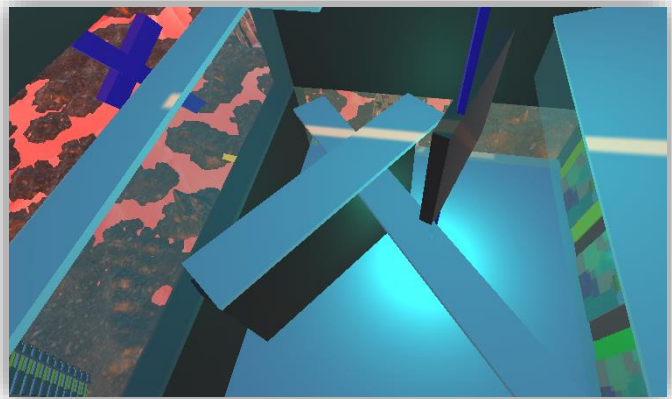
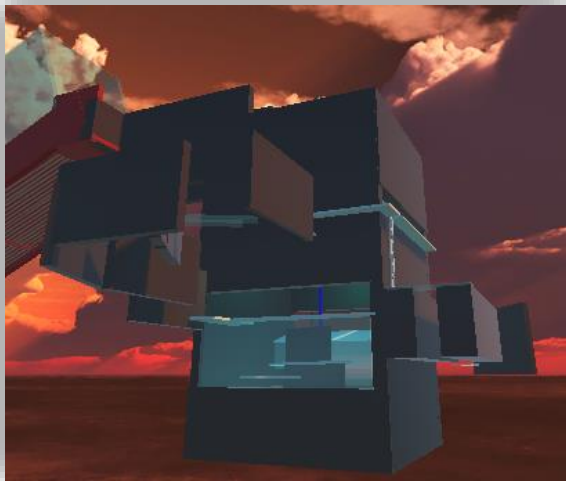
Al finalizar esta zona llegaremos al portal que nos transportará a la siguiente prueba, la **prueba Azul**. Esta prueba es sin lugar a dudas la de mayor dificultad de todo el juego, pues mezcla a la perfección los propósitos de las dos pruebas anteriores.

En la primera prueba teníamos un circuito con un recorrido “largo” pero calmado, con multitud de caminos que escoger, por otro lado, la segunda prueba era “Breve pero intensa”, con

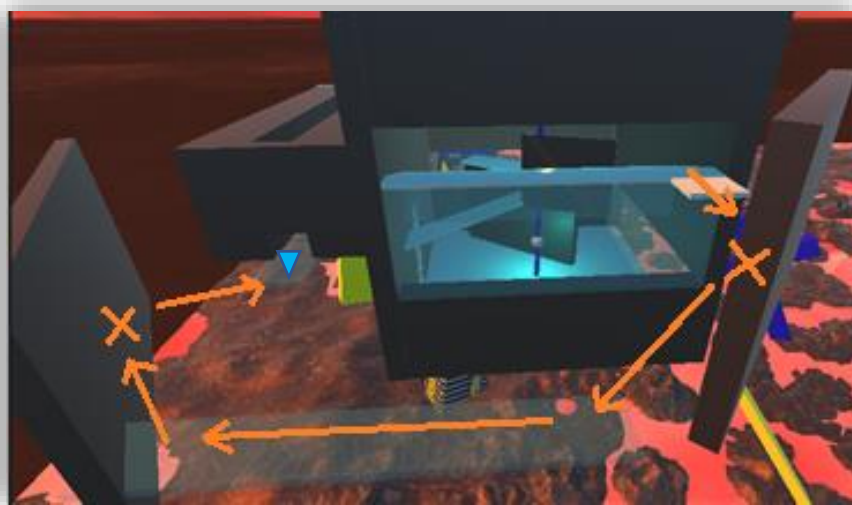
un recorrido mucho más corto pero lleno de elementos que hacían que el jugador dudase y cometiese errores simples.

Ahora, lo que se presenta en la tercera prueba es la mezcla de ambos conceptos, un recorrido “largo”, con distintos caminos a seguir y una dificultad incluso mayor que la de la segunda prueba. En palabras de uno de los participantes del testeo: “Ahora odio el color azul”.

Veamos en detalle esta prueba, se empieza dentro de un alejado edificio en el que habrá que resolver un circuito vertical en su interior:



En un momento dado habrá que escoger entre dos caminos, cada uno lleva a una cornisa distinta que da hacia el exterior del edificio, es aquí donde entra en juego la sensación de la prueba amarilla, pues tendremos que resolver recorridos con caídas al vacío y un nuevo tipo de desafío.

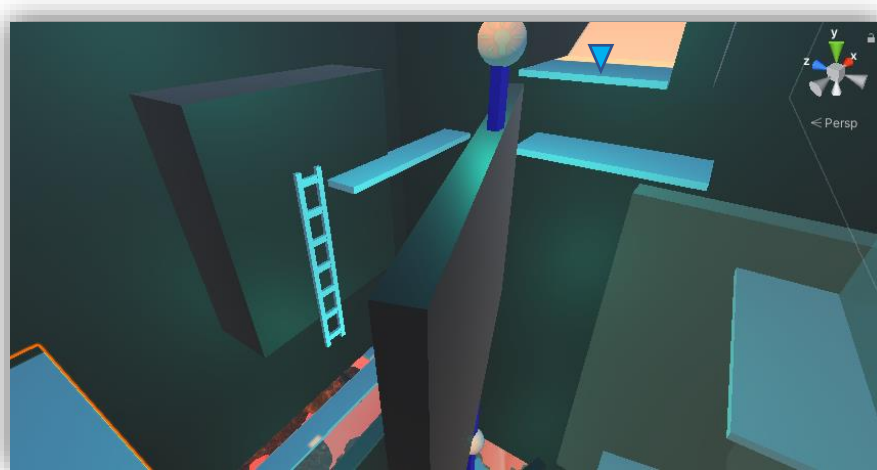


La imagen superior correspondería a la ruta derecha, en esta simplemente debemos hacer varios saltos de pared hacia plataformas que son un tanto transparentes.



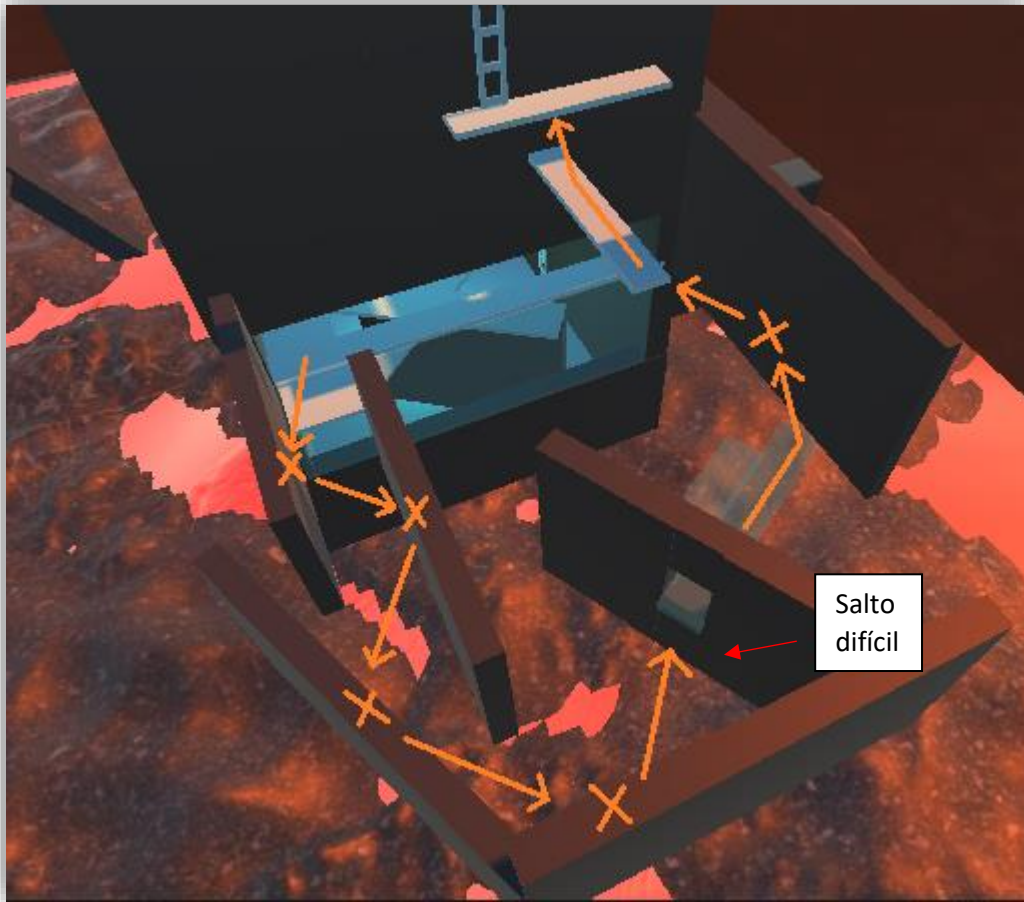
Estas plataformas provocan mayor inseguridad al mostrar la inmensa caída que hay por debajo. No solo eso, sino que ahora tendremos llegar a una pequeña apertura con suelo de cristal y con poca superficie de la que agarrarnos (véase en el triángulo azul). El salto no es difícil, pero al igual que ocurría con la prueba amarilla, los factores de riesgo jugaran malas pasadas al jugador.

Tras completarla y realizar alguna que otra prueba de acumular altura, se volvería de nuevo al interior del edificio mediante otra cornisa abierta. Una vez en el interior hay que resolver de nuevo un circuito vertical que llevará hacia la meta de la prueba azul. Lo interesante de esta última parte, es que está plagada de caídas hacia el interior del edificio, un fallo aquí podría suponer volver al inicio de la prueba entera. Esta parte se puede resolver de múltiples formas y ninguna es fácil (aunque tampoco difícil), la clave de nuevo es el miedo y el riesgo a caerse, pues o bien tendremos que acumular altura con múltiples saltos de pared poco intuitivos o bien tendremos que hacer uno de esos infames saltos de escalera:



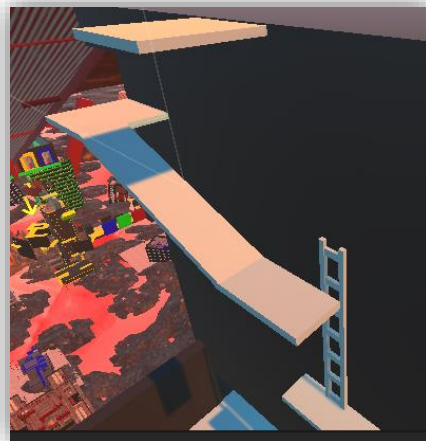
Otra forma de completar la prueba es mediante la segunda ruta (la de la izquierda), como ya se mencionó antes, esta daría hacia el exterior del edificio mediante una cornisa abierta. Esta ruta, es sin lugar a dudas la más difícil de todo el juego y a su vez, es más rápida que la anterior (la cual es relativamente más fácil).

La dificultad de esta ruta reside en que el jugador, no sabrá lo que se va a encontrar hasta que decida lanzarse a intentarla. En la imagen de la siguiente página podrás ver la parte importante de esta ruta. Tal como se observa en ella, el jugador va saltando de pared en pared hasta llegar a un punto en que debe saltar hacia un agujero estrecho. El no poder ver este agujero con suficiente antelación es lo que da a este salto el título del más difícil.



Una vez se llega a la escalera de mano deberemos realizar un pequeño salto desde ella y continuar el recorrido hasta el final de la prueba.

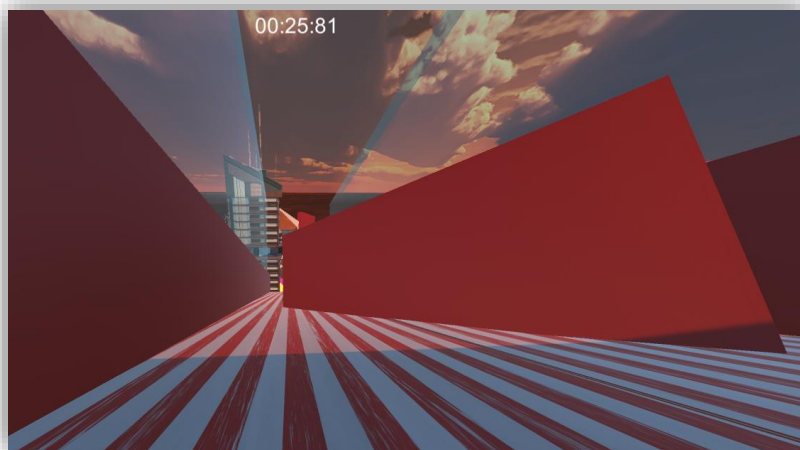
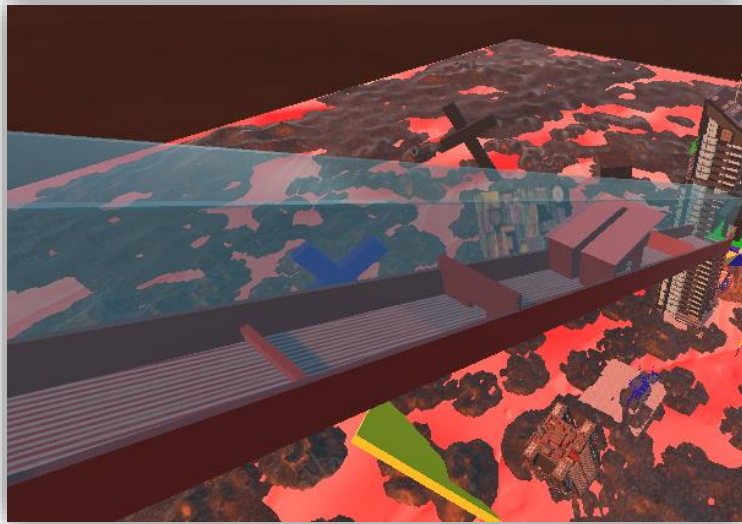
En resumen, esta prueba es la más larga y la que más castiga al jugador con cada fallo, teniendo además unos saltos y zonas que han desesperado a todos los participantes sin excepción. Eso sí, su satisfacción al superarla se hizo notar.



Acabada esta prueba, el jugador llegaría a la **prueba roja**. El juego ya ha expresado su máxima dificultad en la anterior prueba, donde se expresieron al máximo las posibilidades que las mecánicas y el diseño de niveles permiten. Por ello, el resto de pruebas subsiguientes no tienen como objetivo aumentar la dificultad, si no reducirla y poder experimentar otros conceptos muy distintos a las pruebas vistas anteriormente.

¿En qué consiste esta prueba? Dicho de forma resumida, se trata de un tobogán, la idea es que el jugador pueda deslizarse a gran velocidad por este, superando algunos obstáculos como, por ejemplo; saltar una valla, deslizarse por debajo de un orificio, una prueba de acumular altura y finalmente un nuevo tipo de desafío.

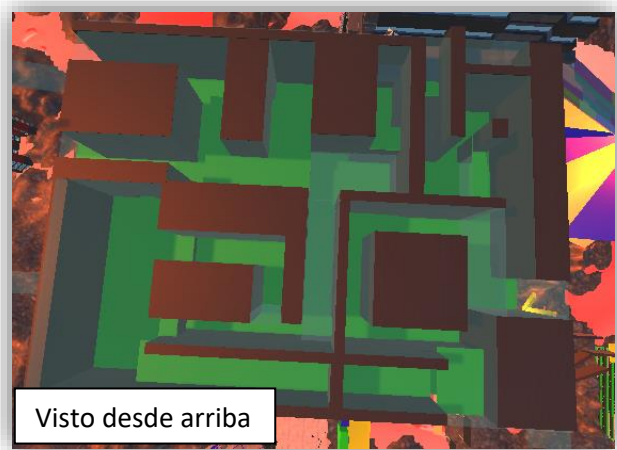
Este nuevo desafío no tiene mucha ciencia, simplemente se trata de un muro que gira continuamente mediante un Script que modifica su rotación. Si el jugador toca el muro será transportado de vuelta al inicio de esta prueba. El muro no gira muy rápido y además se puede pasar perfectamente caminando, la gracia es que la mayoría de jugadores que tratan de pasar a gran velocidad acaban por chocarse o teniendo que detenerse, obligándote a escoger entre riesgo y velocidad o seguridad, pero lentitud.



Al final del tobogán llegaríamos a la última prueba, la **prueba verde**, o como yo le llamo el Laberinto. Al igual que la prueba roja, su propósito no es el de ser más difícil que las anteriores, simplemente es una prueba que busca explorar un nuevo concepto que mezcle la resolución de un laberinto mediante el uso de las mecánicas de Parkour.

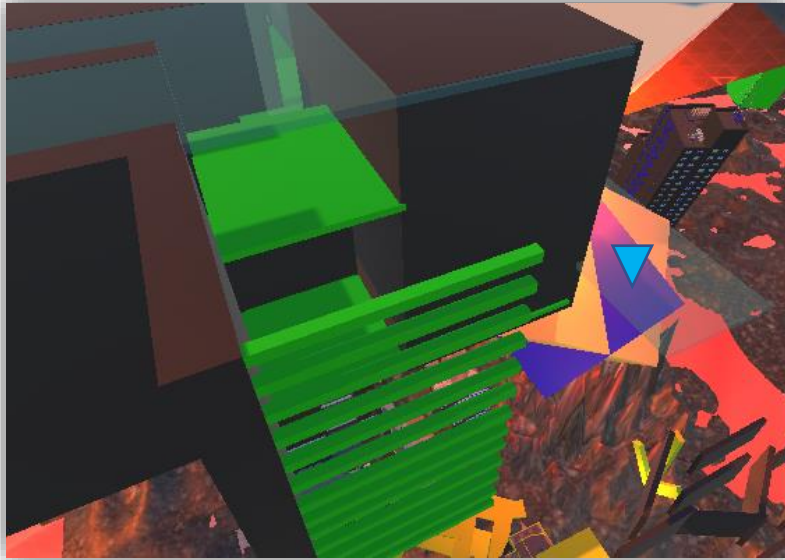
Al principio este laberinto le consumirá bastante tiempo al jugador, ya que deberá averiguar cuál es el camino dentro una zona cerrada en la que no solo tendrá que escoger entre caminos de izquierda o derecha, también deberá escoger entre caminos situados por arriba o por abajo, más todas las zonas donde poder ganar altura mediante el parkour.

A modo de curiosidad, este laberinto puede ser solucionado previamente por el jugador observándolo desde abajo (la prueba amarilla o desde un cristal oculto al que poder saltar). Esto permite observar cual va a ser la salida y por donde se debe ir para solucionarlo.

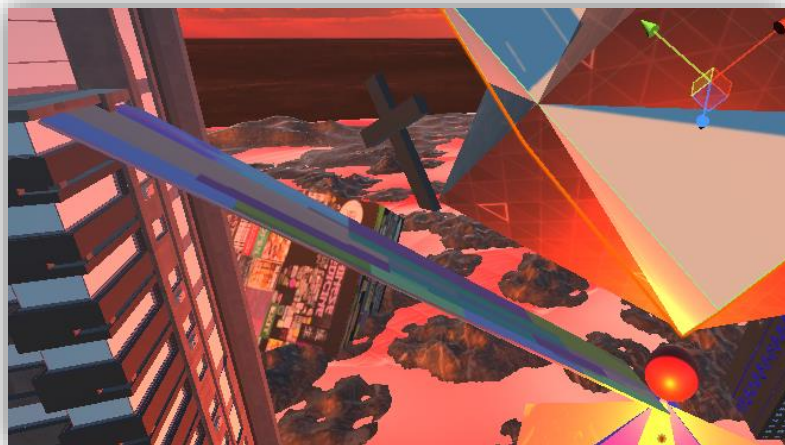




El objetivo es llegar a una zona situada en la esquina del laberinto la cual da al exterior de este, ahí, el jugador deberá observar detenidamente hasta encontrar un cristal a su izquierda, al cual podrá llegar corriendo por la pared de una forma no muy intuitiva.



Tras eso, el jugador continúa el recorrido del laberinto sin mucha pérdida hasta la salida, llegando a la zona final del juego donde simplemente tendremos que deslizarnos hacia la esfera roja brillante al final de la rampa. Al hacerlo será transportado al principio del juego y el tiempo se detendrá para que el jugador pueda comprobar cuanto tardo en completarlo. Se puede pulsar la tecla T para abrir de nuevo el menú (desde ahí podemos salir del juego, ir al tutorial o continuar con la partida).



Como dato extra, en cada prueba hay un Easter Egg igual que el del tutorial, os animamos a intentar encontrar los cinco.

# BIBLIOGRAFÍA

La realización de este proyecto a podido ser llevada a cabo gracias en parte a dos tutoriales de youtube que ayudaron en gran medida con la programación de las mecánicas de juego y los Scripts de Parkour:

- Por un lado, tenemos el video tutorial del canal inglés Colanderp:  
<https://youtu.be/2xjlfFOTtBc>
- Por otro lado, tenemos el video del canal inglés Will-Dev:  
<https://youtu.be/NuaKgcXI0J8>

En ambos videos se explican distintas formas de recrear el movimiento de parkour (correr por las paredes, sobre todo), siendo el primero más completo que el segundo. Lo que hice fue “Fusionar” la manera en la que se programa en ambos tutoriales ya que cosas como los deslizamientos, agacharse, o las escaleras de mano son explicadas en el primer tutorial, mientras que el segundo expone una forma de lograr correr por las paredes más dinámica y simple que el primer video. Mezclando lo aprendido en ambos tutoriales logré hacer buena parte de los Scripts de parkour.

Así mismo me apoye en otro video tutorial para programar el menú del juego y los cambios de escena (más que nada porque no recordaba bien cómo hacerlo):

- Canal en español de Don Pachi: <https://www.youtube.com/watch?v=M-7vzHsLCxU&t=181s>

Mencionar también la ayuda que encontré en un post de internet sobre la solución a un error de horneado de escenas en Unity. Este error hacía que al cargar una escena se oscurecieran mucho todos los elementos:

- Gracias al usuario “albertti”: [UNITY: Problema con Scene Manager Carga la escena oscura | Tips windows y web \(tipsww.blogspot.com\)](https://www.unity.com/unity-problem-scene-manager-load-scene-dark)

En cuanto a recursos de Unity usados:

- Kit de assets para los edificios: [Neo City Free Sample Kit \(Built-In\) | 3D Sci-Fi | Unity Asset Store](https://unity.com/unity-asset-store/asset/3d-sci-fi-city-kit)
- Kit de texturas para el cielo: [8K Skybox Pack Free | 2D Sky | Unity Asset Store](https://unity.com/unity-asset-store/asset/8k-skybox-pack-free)

Y por supuesto mencionar a Unity como la herramienta esencial para crear el videojuego y al aprendizaje que he podido adquirir en el Instituto Tecnológico Edix.

## CONCLUIONES, POSIBLES MEJORAS Y DESPEDIDA

Como conclusión final de todo el proyecto, tendríamos un juego en primera persona con mecánicas de Parkour fluidas que plantea todo tipo de desafíos de habilidad e ingenio al jugador.

Este juego nos enseña a ser pacientes, tomarnos nuestro tiempo, explorar distintas rutas, gestionar la frustración, mejorar nuestra habilidad mecánica (el famoso dicho gamer de “Git Gud”) y, sobre todo, a superarnos a nosotros mismos.

El tiempo que tarde en completarse es directamente proporcional a la habilidad del jugador, bien puede durar 4 minutos (o menos) o bien puede durar más de media hora. Esto lo convierte en un juego bastante ligero para entretenerse durante alguna tarde aburrida con la posibilidad de poder repetirlo para batir las anteriores marcas de tiempo.

En cuanto a posibles mejoras tenemos esta lista:

- **Implementación de música:** El juego no tiene música debido a que considero que supondría un deterioro para la jugabilidad. Normalmente se usa la música para crear una sensación de inmersión en el juego, sin embargo, este juego no necesita como tal o no tiene un estilo de música que lo acompañe favorablemente (probamos todo tipo de música con los participantes y estos concuerdan con que en general era mejor sin ella). Música calmada puede acabar sintiéndose repetitiva y fuera de lugar con la velocidad que podemos alcanzar y en cuanto a música más animada podría estar bien en algunas zonas, pero sería pesada y agobiante el resto del tiempo (y bastante te saca ya de quicio algunas zonas del juego como para empeorarlo). Quizás si modularizáramos cada prueba en una escena distinta si podríamos ponerle música a cada una, pero la idea de este juego es que todas las pruebas se encuentren en una misma escena.
- **Mejoras en el rendimiento:** Por desgracia, debido a la amplia carga de texturas del mapa, se pueden producir momentos de lag y caídas de frames, lo cual perjudicaría la fluidez del movimiento si disponemos de un equipo poco potente. Esto podría solucionarse de muchas formas, una de ellas y la más asequible sería reducir la cantidad de texturas y Assets del decorado, lo cual supondría sacrificar buena parte de la estética del juego.
- **Bugs:** El juego cuenta con algunos bugs o exploits que en un principio no perjudican la experiencia de juego. Estamos hablando de cosas como escalar infinitamente mediante el continuo salto en una esquina del escenario o escalar el edificio de la prueba amarilla

simplemente pulsando saltar. Estos bugs no han sido solucionados a propósito, pues un jugador que logre descubrirlos y aprovecharlos podría usar rutas muy alternativas para llegar antes. Los participantes que lo descubrieron se sintieron muy satisfechos y orgullosos de haberlos usado como beneficio para sus records. Aunque no por ello negaré que hacen que el juego parezca un tanto “cutre”, pero eso es un precio que estoy dispuesto a pagar si a cambio mejora la experiencia del jugador.

- **Futuras mecánicas y mejoras:** Me hubiera gustado implementar otras mecánicas que ofrecieran nuevas oportunidades, como por ejemplo balancearse por cuerdas, disparar ganchos, usar tirolinas, plataformas de cristal que se rompan o incluso un modo de juego de gran dificultad que cada vez que caigamos al vacío nos transporte al principio del juego. En cualquier caso, no ha dado tiempo a poder implementar todo esto y quizás en un futuro se publique una segunda entrega de este juego con más mecánicas, nuevos mapas, y modos de juego.

Por mi parte eso sería todo, especial agradecimiento a los participantes del testeo, los cuales ayudaron no solo dejando sus marcas, si no también descubriendo bugs, nuevos métodos y rutas para las pruebas y propuestas de mejora.

# FIN