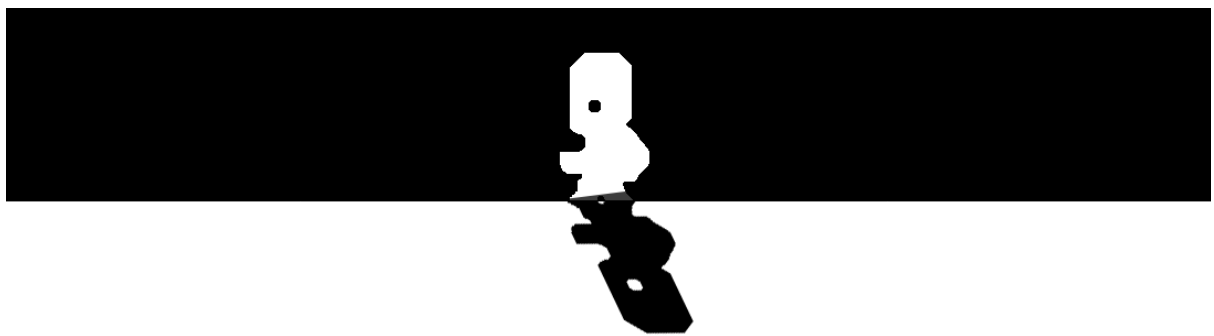


INFORME Y ANÁLISIS DE LAS PRUEBAS DE USUARIO - MYRROR

<https://github.com/evalucas/UAJ/tree/main/PoyectoFinal>



Usabilidad y Análisis de Juegos

Curso 2021-2022.

Eva Lucas, David Carmona, David Godoy, Jesús Ramos, Jorge Bello, Tomás López

ÍNDICE

| | |
|---|-----------|
| ÍNDICE | 2 |
| INFORMACIÓN Y OBJETIVOS DE ESTUDIO | 3 |
| METODOLOGÍA DE LA PRUEBA DE USUARIO | 4 |
| Sistema de telemetría | 5 |
| Formulario | 6 |
| Distribución de las pruebas y recogida de datos | 7 |
| Interpretación de los datos | 8 |
| RESULTADOS DEL ESTUDIO | 8 |
| ANÁLISIS DE LOS RESULTADOS | 16 |
| CONCLUSIONES FINALES | 18 |
| ADENDA | 19 |

INFORMACIÓN Y OBJETIVOS DE ESTUDIO

Myrror es un videojuego de plataformas de avance lateral realizado en una *game-jam* en el año 2019 por una parte del equipo de realización de este estudio. A día de hoy, hemos decidido rescatarlo para hacer un estudio de la dificultad del mismo, tanto de las mecánicas más importantes como de la dificultad general de los niveles disponibles en el videojuego. Esto nos puede ayudar para realizar retoques y, en general, rediseñar de manera más elaborada una idea que por las condiciones en las que se desarrolló no gozó de pruebas de usuario, y que se puede beneficiar de más iteraciones para ajustar la dificultad.

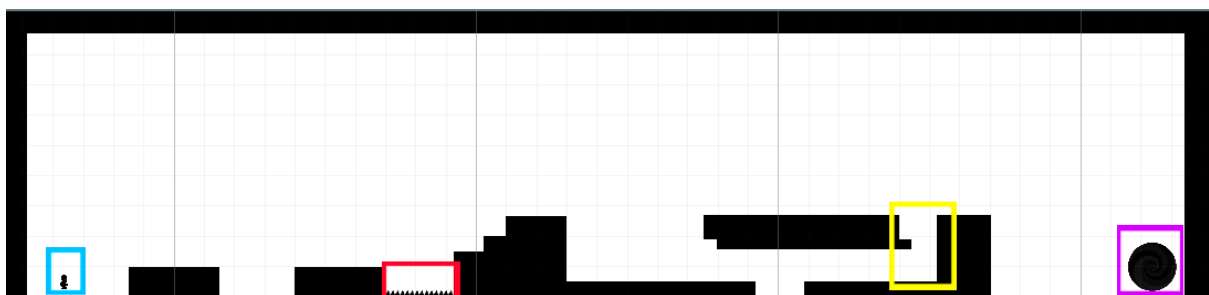
Myrror tiene cuatro mecánicas principales: el movimiento, el salto, el salto en paredes (de aquí en adelante *wall jump*) y los cambios de gravedad.

insertar gifs de las tres cosas (alguien sabe si un pdf soporta gifs? ajsdjaja)

Hemos definido de manera previa cuáles son nuestros objetivos y una serie de preguntas de investigación que nos ayudarán a hacer un análisis de la dificultad del videojuego, preguntas que deberemos responder en el análisis final de los datos:

1. ¿Es capaz el jugador de hacer los *wall jumps* sin problemas?
2. ¿Es capaz el jugador de hacer los cambios de gravedad sin problemas?
3. ¿Cómo de difícil es cada nivel?
4. ¿Qué caminos decide tomar el jugador a lo largo del nivel?

Hemos preparado una *build* que contiene cuatro niveles. El código de colores de los diagramas es el siguiente: **azul** para puntos de aparición, **rojo** para puntos donde el jugador puede morir (volver al punto de aparición), **amarillo** para zonas en las que es necesario utilizar el *walljump*, **verde** para zonas en las que se usan los cambios de gravedad, **morado** para el final de un nivel.



Nivel 1



Nivel 2

1. Sistema de telemetría

El sistema de telemetría trata de recoger las siguientes métricas que hemos planteado pensando en qué datos de una partida necesitamos para sacar la información necesaria para la respuesta de cada una de las preguntas de investigación:

1. ¿Es capaz el jugador de hacer los *wall jumps* sin problemas?

Tiempo que tarda el jugador en superar un *wall jump*, veces que salta para superar un *wall jump*

2. ¿Es capaz el jugador de hacer los cambios de gravedad sin problemas?

Veces que pasa por cada cambio de gravedad

3. ¿Cómo de difícil es cada nivel?

Posición de las zonas en las que muere, número de muertes por nivel, tiempo que tarda en finalizar cada nivel

4. ¿Qué caminos decide tomar el jugador a lo largo del nivel?

Posición del jugador a lo largo del tiempo

El sistema se trata de un package para Unity que hemos integrado en el proyecto del videojuego original. Está estructurado en cuatro scripts diferentes: *Eventos.cs*, *Serializacion.cs*, *Persistencia.cs* y *Telemetria.cs*.

1. *Eventos.cs*

Recoge la definición de la clase base para un evento, y los eventos que hemos creado a partir de ella para la recogida de los datos. Los eventos enumerados en este archivo son los siguientes:

- **Gestión de sesión:** Inicio de una sesión (*SessionStartEvent*), final de una sesión (*SessionEndEvent*)
- **Gestión de nivel:** Inicio de un nivel (*LevelStartEvent*), fin de un nivel (*LevelEndEvent*), pausar el juego (*PauseEvent*), reanudar el juego tras una pausa (*UnpauseEvent*)
- **Gestión del jugador:** Muerte (*DeathEvent*), saltos (*JumpEvent*), posición del jugador (*PlayerPosEvent*), colisión con un tag (*CollisionEvent*)

2. *Serializacion.cs*

Este script escribe todos los parámetros públicos que tengan los eventos recogidos en un archivo con un formato específico. Para ello, creamos una interfaz *Serializer*, la cual implementarán los diferentes tipos de serializadores. Hemos implementado serializadores para archivos *.json* y *.bin*, pero por comodidad, para la prueba solo utilizaremos el serializador de archivos *.json*.

3. *Persistencia.cs*

Decide dónde se van a guardar los archivos escritos en la serialización. Hemos decidido implementar un sistema de persistencia síncrono, que irá añadiendo líneas al archivo según se vayan escribiendo eventos. Además, guardará los archivos en disco, no en un servidor, ya que como explicaremos más adelante, nos ayudaremos de los formularios de *Google* para recibir el archivo con las trazas recogidas.

4. *Telemetría.cs*

Se trata de la clase principal, la cual será llamada desde los diferentes puntos del videojuego donde se deban enviar los mensajes. Utilizamos el patrón de diseño *fluent interface* para ayudar a la legibilidad del sistema a la hora de utilizarlo. El funcionamiento es el siguiente:

1. Desde el videojuego, en los scripts necesarios, añadimos la referencia al *Unity Package* para llamar al sistema de telemetría y enviar un mensaje. A la hora de crear la instancia que guarde el sistema de telemetría, se llama al constructor del mismo, que creará una referencia al sistema de persistencia y serialización que deseemos usar, los cuales en este caso son un sistema de serialización de *.json* y un sistema de persistencia en disco.
2. Se envía un mensaje. Para ello, *Telemetria.cs* tiene implementado métodos (siguiendo el patrón *fluent interface* mencionado antes) que devuelven el tipo de mensaje requerido. Ejemplo de un envío de mensaje:

```
Telemetria.Instance.TrackEvent(Telemetria.Instance.SessionStart())
```

3. El sistema de telemetría envía el evento al sistema de persistencia, y este se lo envía al sistema de serialización. El sistema de serialización devolverá al sistema de persistencia el mensaje serializado para que éste lo escriba donde toque (en el disco).

En resumen, este sistema de telemetría escribirá en un archivo *.json* todos los eventos que ocurren en la partida de un jugador, y será este quien nos envíe el archivo con todos los datos mediante *Google Forms*.

2. Formulario

El formulario ha sido creado con la herramienta *Google Forms*, con el detalle de que nos sirve como herramienta para alojar los datos recogidos por el sistema de telemetría, gracias a la posibilidad de crear una pregunta en el formulario en la cual el usuario pueda subir un archivo, que se aloja en una carpeta de *Google Drive* que nosotros hemos creado. Así, podemos utilizar una misma herramienta para recoger todos los datos necesarios (los de la telemetría y las respuestas del formulario), con la ventaja de no tener que crear un servidor propio para alojar estos archivos.

Hemos decidido que el formulario sea corto y conciso, con preguntas que nos ayuden directamente a responder las cuestiones de investigación previamente planteadas. A la hora de usar escalas de valoración, hemos decidido usar un número par (4 valores) para evitar que los usuarios respondan con un valor intermedio, que no nos daría ningún matiz ni positivo ni negativo que nos ayude a responder a las preguntas de investigación; nos darían unas respuestas tibias. Las preguntas del formulario son las siguientes:

1. **Resultado del test:** Pregunta que consta únicamente de un botón para subir el archivo generado por el sistema de telemetría.
2. **Nombre del archivo subido. Ejemplo (ID_61205...):** Pregunta en la cual el usuario debe escribir el nombre del archivo subido, necesario para poder relacionar los resultados de la sesión de juego con las respuestas del formulario.


3. **¿Cuál es tu edad?:** Posibles respuestas:
 - a. 0-19
 - b. 10-19
 - c. 20-29
 - d. 30-39
 - e. 40-59
 - f. 60+
4. **Define tu nivel de experiencia jugando a videojuegos (EN GENERAL):** Respuestas en forma de escala lineal de 1 a 4 (Nada experimentado - Muy experimentado)
5. **Define tu nivel de experiencia jugando a videojuegos de PLATAFORMAS:** Respuestas en forma de escala lineal de 1 a 4 (Nada experimentado - Muy experimentado)
6. **Resulta fácil realizar un wall-jump (saltos de una pared a otra):** Respuestas en forma de escala lineal de 1 a 4 (Muy en desacuerdo - Muy de acuerdo)
7. **Resulta fácil realizar un cambio de gravedad:** Respuestas en forma de escala lineal de 1 a 4 (Muy en desacuerdo - Muy de acuerdo)
8. **¿Cuál ha sido tu percepción de la dificultad del videojuego en cada nivel?:** Respuesta en forma de cuadrícula de varias opciones, en la cual tenemos cuatro filas de respuestas (Nivel 1, Nivel 2, Nivel 3 y Nivel 4) y la escala de valoración en las columnas (Muy fácil, Fácil, Difícil, Muy difícil)

3. Distribución de las pruebas y recogida de datos

Para la distribución de las pruebas de usuario, hemos redactado un archivo .pdf (porque es un estándar que ayuda a que la gente a la que le llegue la prueba lo lea sin problemas) en el que detallamos en qué consiste la prueba, los requisitos para poder realizarla, los pasos a seguir y por último, cómo pueden mandarnos los resultados.

La prueba va dirigida a todo tipo de público, ya sean jugadores habituales como no habituales, independientemente del rango de edad. Hemos distribuido el archivo por redes sociales y a nuestros círculos cercanos para poder conseguir la mayor cantidad de pruebas posibles. Gracias a *Google Forms*, podremos recoger los resultados fácilmente, ya que tendremos alojados en una carpeta de *Google Drive* todos los archivos .json generados por el sistema de telemetría, y en un archivo .csv todas las respuestas enviadas por los usuarios.

Cabe destacar que al utilizar este tipo de alojamiento de los datos, pese a que ganamos en tiempo y usabilidad (ya sea para nosotros como para



GUÍA PARA LA REALIZACIÓN DE LA PRUEBA DE USUARIO
 Descarga el juego [aquí](#)

El objetivo de esta prueba es que cada usuario juegue al videojuego Myrror y avance en él todo lo que pueda hasta el final de la prueba. Los puntos que se van a evaluar en esta prueba no dependen de la habilidad de cada usuario, sino que vamos a evaluar el videojuego en sí.

La prueba tiene una duración de 15 minutos desde que se empieza a jugar. Para la realización de la prueba, es necesario lo siguiente:

1. Un ordenador con Windows
2. Un teclado y un ratón

Pasos para la realización de la prueba de usuario:

1. **Abrir [este enlace](#)** (o el superior) para acceder a la descarga del juego mediante Google Drive. Una vez ahí, descargar el archivo y situarlo en la carpeta que el usuario prefiera.
2. **Descomprimir el archivo descargado** con cualquier programa que lo permita (Winrar, 7-zip, etc) o abrirlo con el explorador de Windows (doble click → Extraer todo → Extraer).
3. **Abrir la carpeta descomprimida "Myrror"** y hacer doble click en el archivo "Myrror.exe". Esto ejecutará el videojuego.
4. Una vez abierto el videojuego, **hacer click en "Start" para comenzar la prueba**. A partir de aquí comenzará la sesión de prueba, que durará 15 minutos.
5. Al acabar el tiempo o completar todos los niveles, aparecerá la pantalla final en la que se indica el fin de la prueba, y se cerrará el juego automáticamente.
6. A continuación, **abrir [este enlace](#)** para acceder al cuestionario. En la primera pregunta, el usuario deberá subir un archivo con los resultados de la prueba generados automáticamente al terminar de jugar. Estos resultados se encuentran en la carpeta "Myrror/Resultados de la prueba".
7. Contestar al resto de preguntas y una vez respondidas, pulsar en "enviar cuestionario".

*Nota: Si por cualquier razón, el usuario desea terminar la prueba antes de los 15 minutos, puede pulsar el botón "ESC" para salir del juego. En dicho caso, rogamos no seguir con el cuestionario, sino empezar otra prueba desde el paso 1 en otro momento.

*Nota 2: Para el correcto funcionamiento de la prueba, rogamos no crear ni eliminar ningún archivo o directorio dentro de la carpeta descomprimida.

los usuarios, que en general estarán más habituados a los entornos de trabajo de *Google*), por la naturaleza de *Google Drive* y *Forms* hemos tenido que optar por la opción de que sea el propio usuario quien suba tanto el archivo como el nombre del mismo para poder identificarlo, pudiendo caer en equívocos a la hora de realizarlo. Por ello, prevemos un pequeño margen de error en la recogida de datos, pero que por la extensa descripción de los pasos de realización de la prueba se reducirá a un margen muy pequeño de los datos finales. En el apartado de análisis del estudio detallaremos cuántas de estas respuestas acabarán desechadas y cuántas no.

4. Interpretación de los datos

Para interpretar los datos hemos utilizado un jupyter notebook que lee los datos tanto de los formularios como del sistema de telemetría y va procesando los datos de ambas fuentes para acabar la ejecución del proceso con un array de *players* que contiene toda la información necesaria para sacar las gráficas, como el número de muertes y tiempo de completado de los niveles; o el nivel de experiencia del jugador o su edad.

RESULTADOS DEL ESTUDIO

A continuación, vamos a exponer los resultados del estudio como base del análisis de los mismos que haremos en el siguiente apartado.

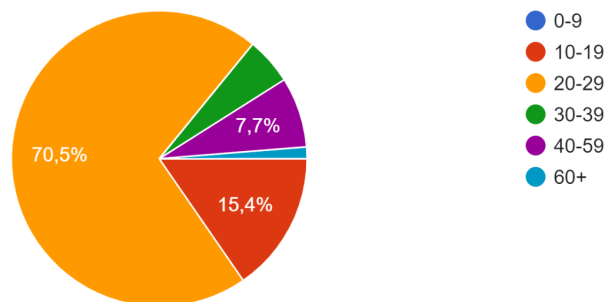
Actualmente, el formulario consta de 78 respuestas proporcionadas. No obstante, debemos hacer una criba de los informes erróneos. Pueden ser erróneos por varios factores: que el archivo subido sea incorrecto, que el nombre indicado del archivo no se corresponda con el mismo, que haya habido algún error en el formulario o cualquier modificación en los .json que cause un error en los cálculos necesarios.

Teniendo en cuenta el número de pruebas recibidas (78), vemos que el porcentaje de acierto es del 94.7%, siendo por lo tanto un error bastante pequeño, como esperábamos.

A continuación, mostraremos primeramente los gráficos con los datos que nos ofrecen los formularios de *Google*, y después utilizaremos estos datos y los de las pruebas recogidas por el sistema de telemetría para hacer otros gráficos diferentes.

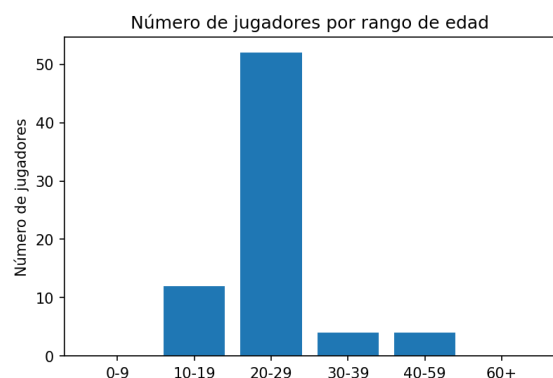
¿Cuál es tu edad?

78 respuestas



Como podemos observar, hemos tenido un amplio volumen de pruebas en un rango de edad de entre 20 y 29 años, dando menos oportunidad para probar con usuarios de diferentes generaciones, pero al mismo tiempo, acercándose más al público objetivo de un videojuego como el nuestro, por lo que resulta bastante útil y representativo. La diferencia de cantidad de usuario por grupo no nos causará más molestia que el hecho de tener menos precisión en los diferentes cálculos.

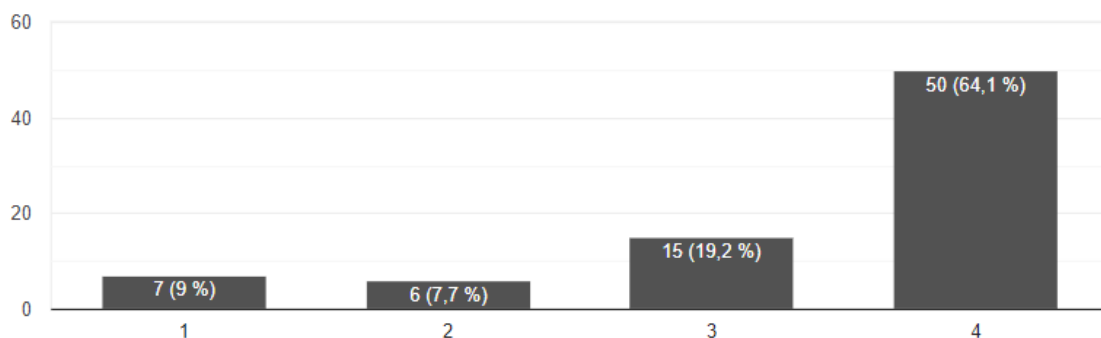
Estos resultados no tienen en cuenta los datos no válidos, sino la participación en general. Teniendo en cuenta solamente los datos válidos, obtenemos que no tenemos ninguna prueba no sólo de usuarios entre 0-9 años, sino que tampoco tenemos de 60+. Es lógico que no haber conseguido usuarios del primer o último grupo nos da menos información final de la que nos habría gustado, pero viendo que aún así hubo bastante volumen de pruebas, nos es suficiente para valorar las preguntas de investigación.



Define tu nivel de experiencia jugando a videojuegos (EN GENERAL)

 Copiar

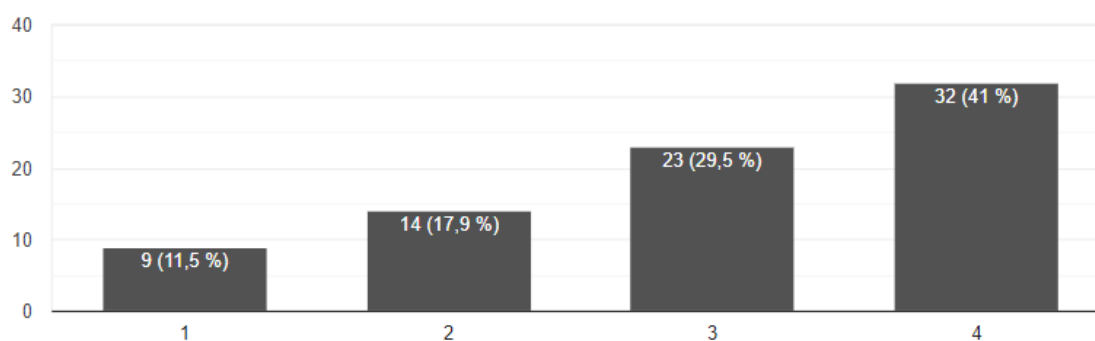
78 respuestas



Define tu nivel de experiencia jugando a videojuegos de PLATAFORMAS

 Copiar

78 respuestas

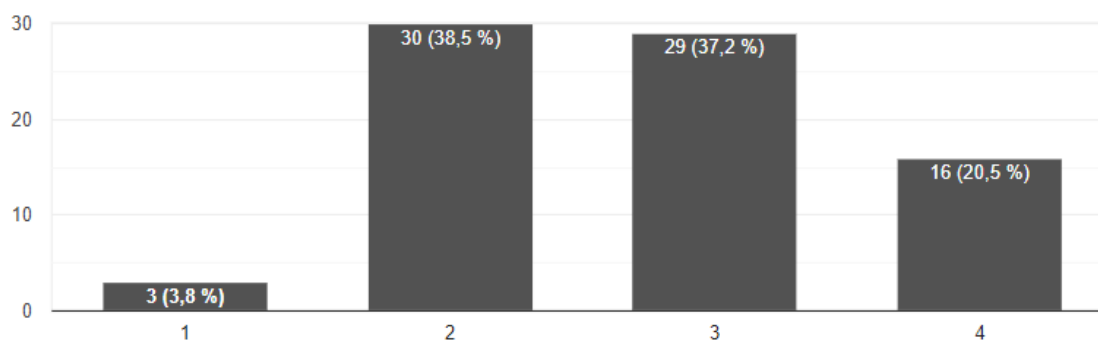


En cuanto a los niveles de experiencia, podemos ver que una mayoría de nuestros probadores han definido que son muy experimentados, mientras que varía más cuando se trata de la experiencia concretamente en los videojuegos de plataformas.

Resulta fácil realizar un wall-jump (saltos de una pared a otra)

 Copiar

78 respuestas

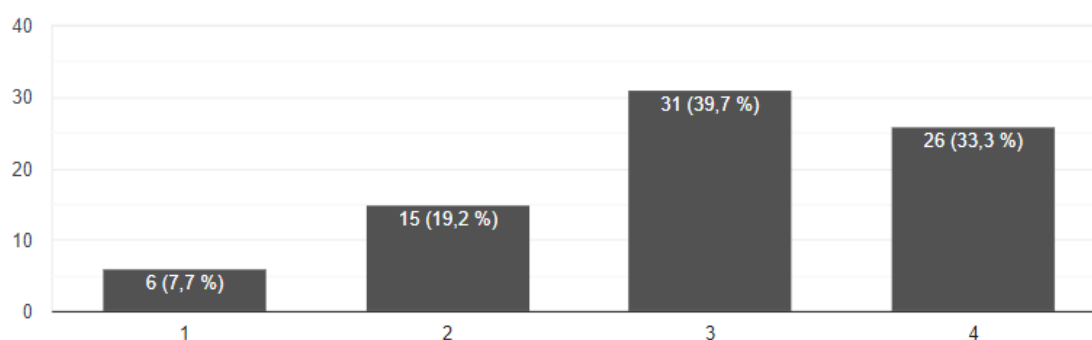


En este gráfico podemos observar que el gran volumen de respuestas encuentran que realizar un *wall jump* tiene una dificultad media, entre poco fácil (38.5%) y poco difícil (37.2%). Pocos usuarios han pensado que sea muy difícil (3.8%) realizar esta mecánica, y aunque menor, también es significativo los que piensan que es muy fácil (20.5%).

Resulta fácil realizar un cambio de gravedad

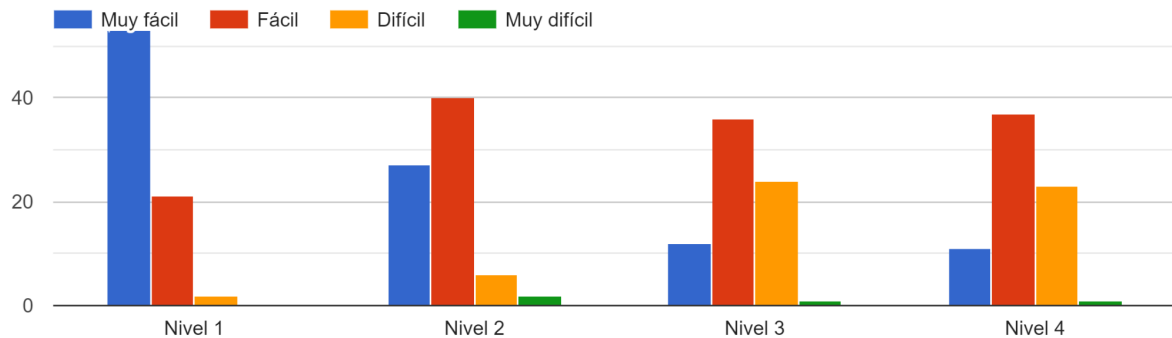
 Copiar

78 respuestas



En cuanto a los cambios de gravedad, los resultados quedan más dirigidos hacia el extremo derecho, el que indica que es fácil o muy fácil realizar un cambio de gravedad (39.7% y 33.3% respectivamente)

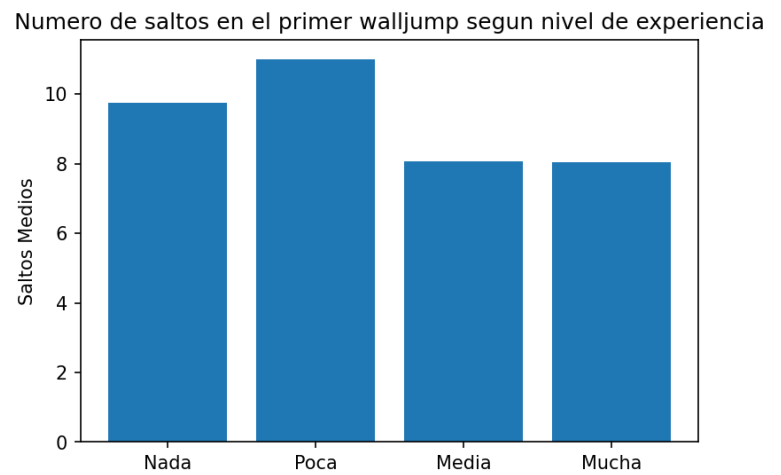
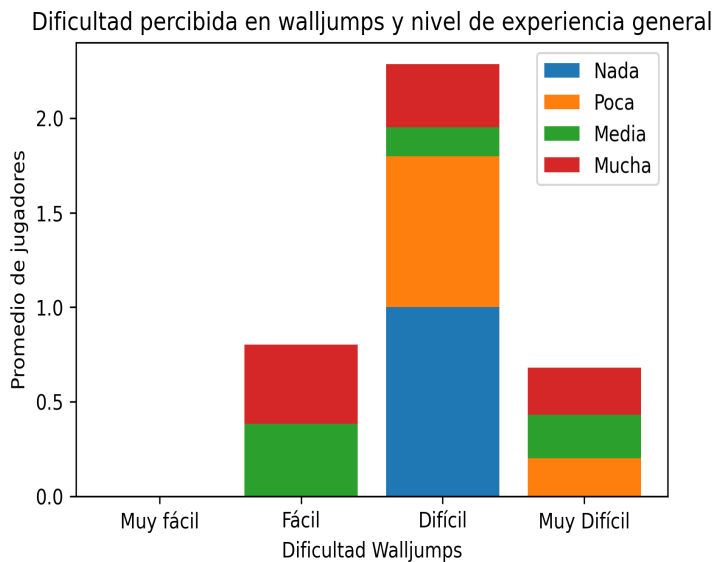
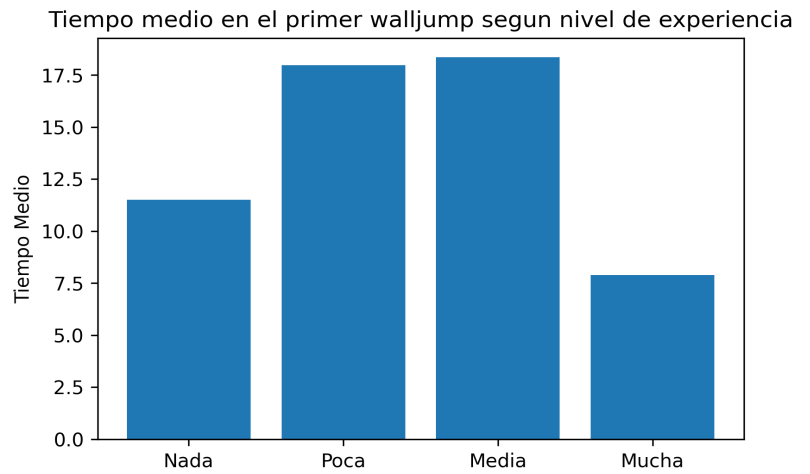
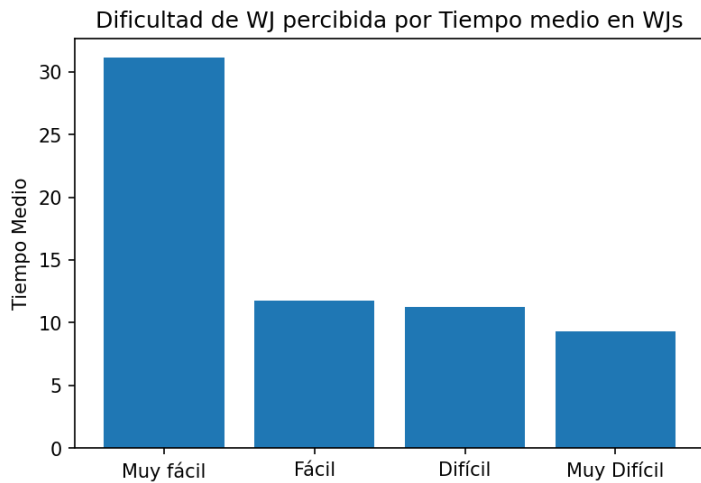
¿Cuál ha sido tu percepción de la dificultad del videojuego en cada nivel?



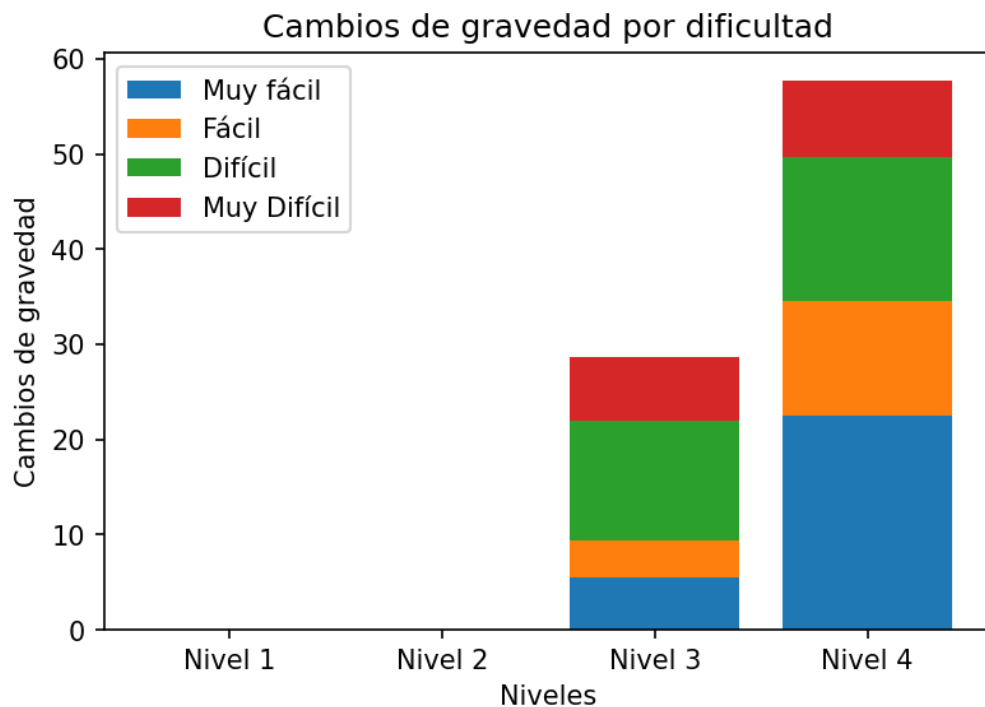
Por último, en cuanto a la dificultad percibida por cada uno de los niveles, vemos que apenas ha habido la apreciación de que fuese muy difícil, pero sí de que son fáciles o muy fáciles, con énfasis en el primer y segundo nivel.

A continuación, vamos a agrupar las diferentes gráficas recogidas utilizando principalmente los datos de las pruebas de telemetría, comparándolos según convenga con los datos de los formularios. Se agruparán según cada una de las preguntas de investigación, y en el siguiente apartado se analizarán con más detalle junto con los gráficos mostrados anteriormente.

1. ¿Es capaz el jugador de hacer los wall jumps sin problemas?

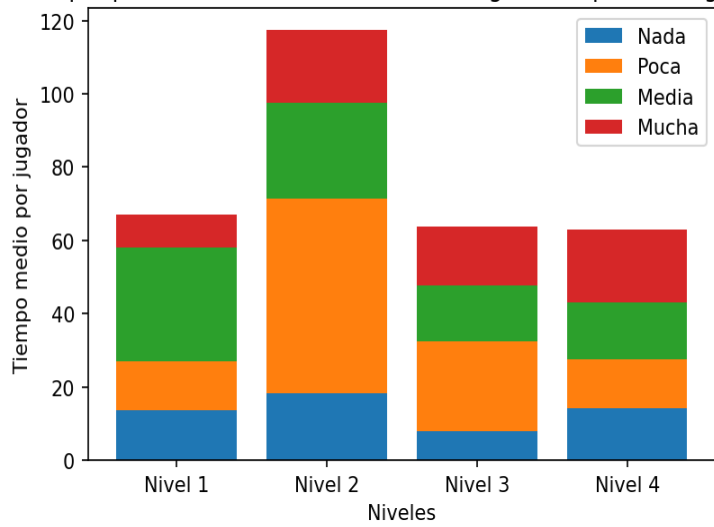


2. ¿Es capaz el jugador de hacer los cambios de gravedad sin problemas?

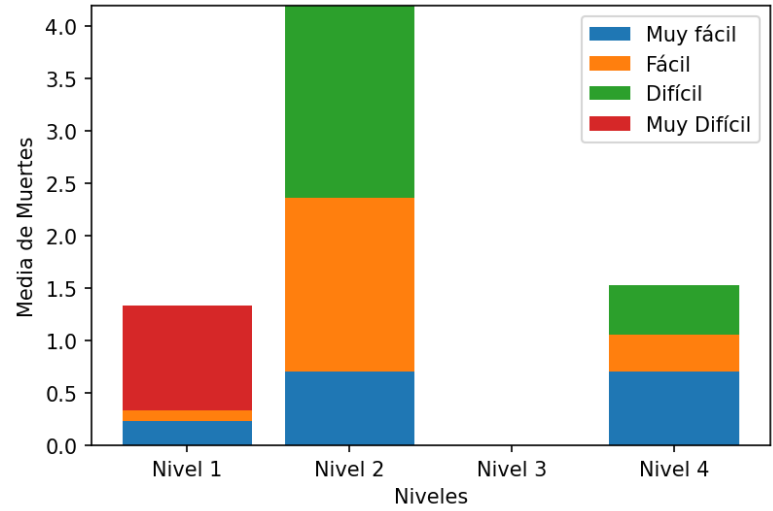


3. ¿Cómo de difícil es cada nivel?

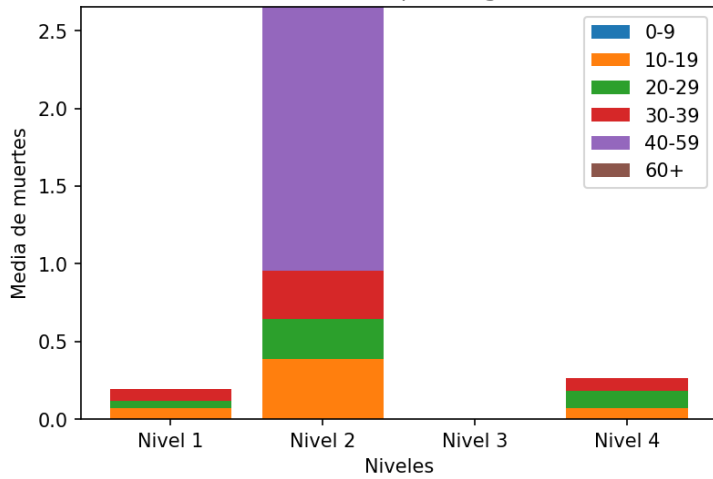
Tiempo que tardan en terminar el nivel según la experiencia general



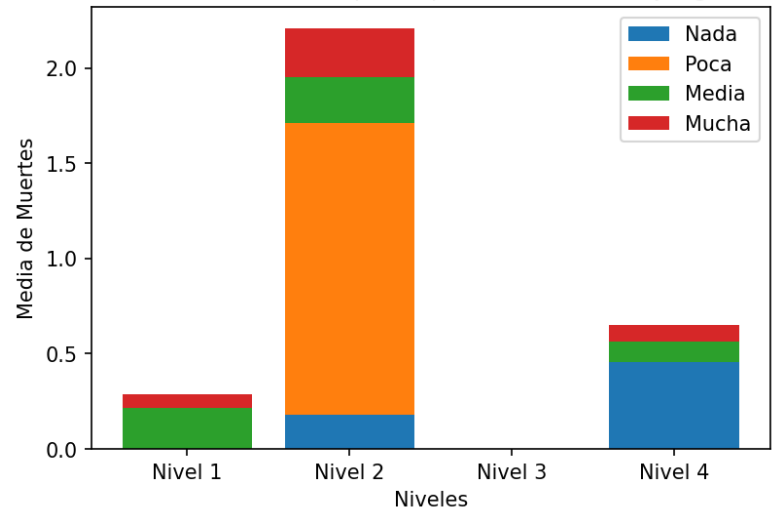
Media de muertes según dificultad percibida



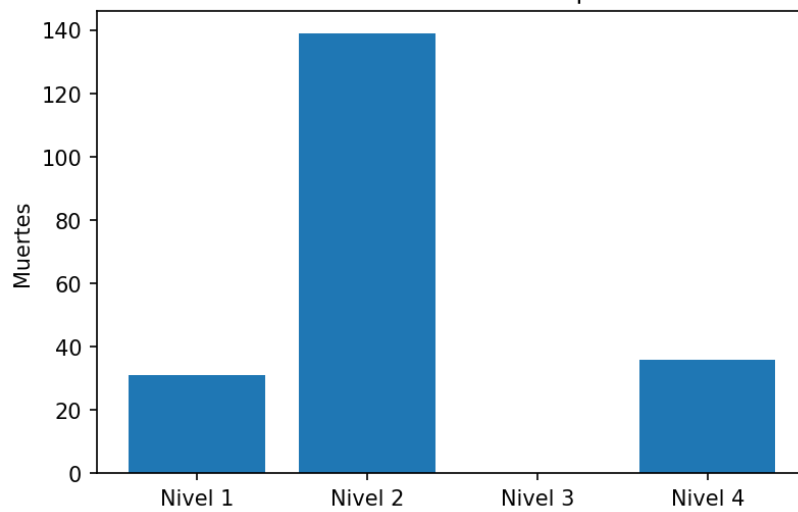
Media de muertes por rango de edad



Media de muertes por experiencia en videojuegos

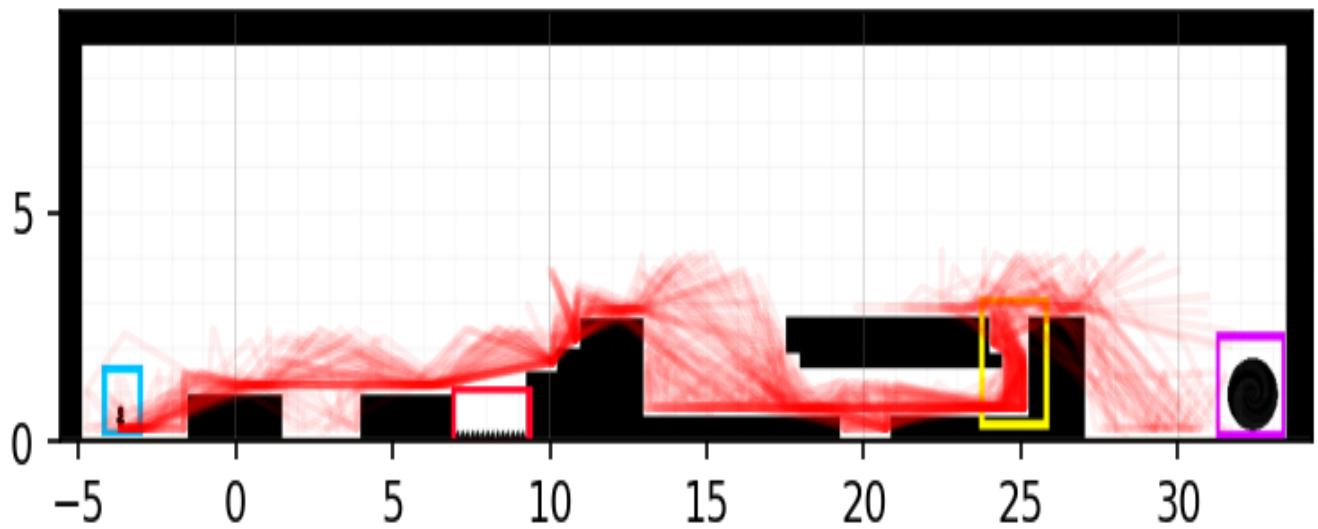


Número de muertes totales por nivel

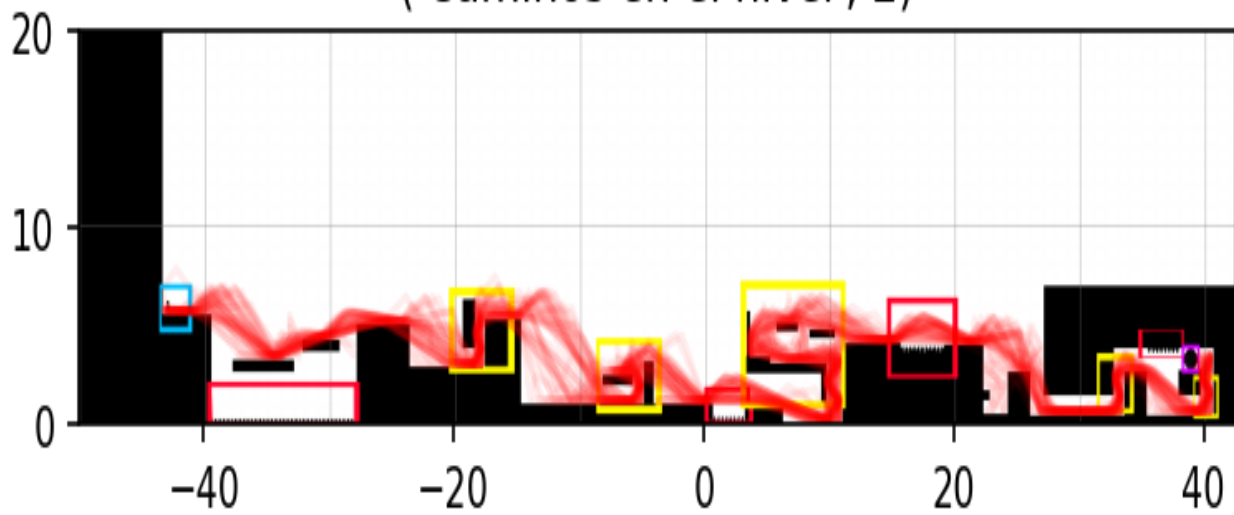


4. ¿Qué caminos decide tomar el jugador a lo largo del nivel?

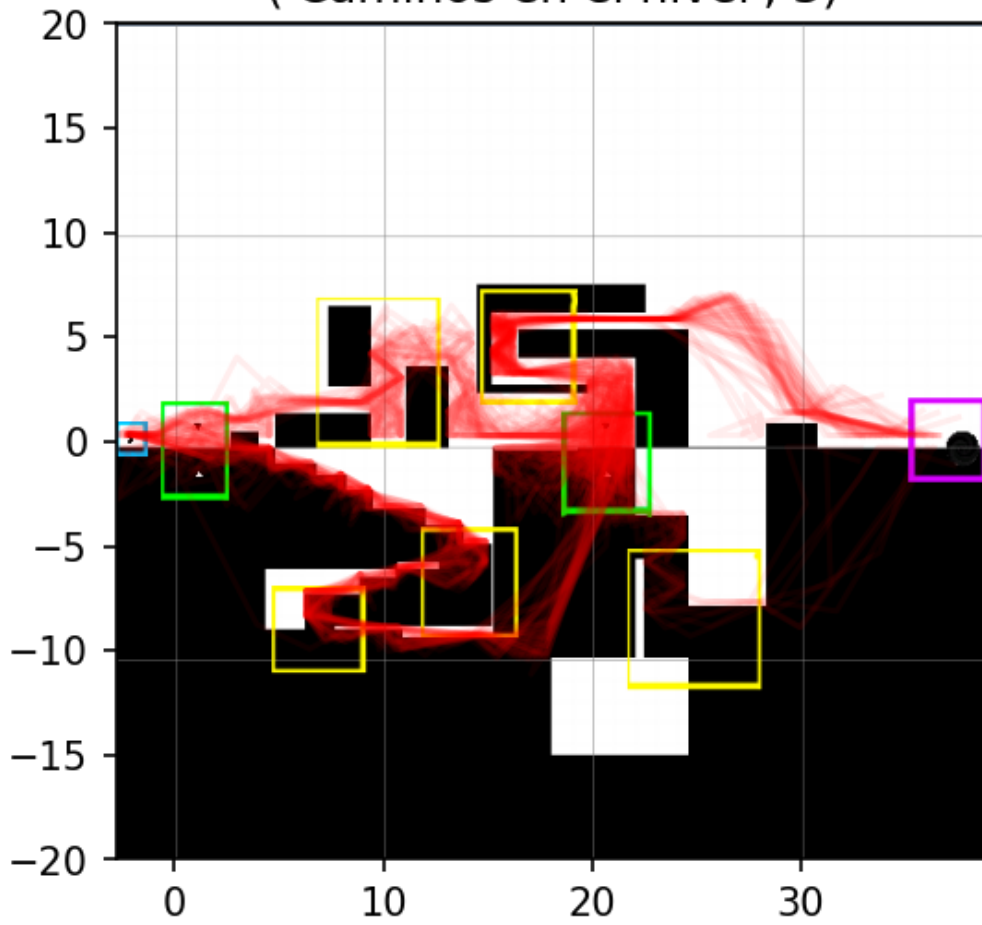
('Caminos en el nivel', 1)



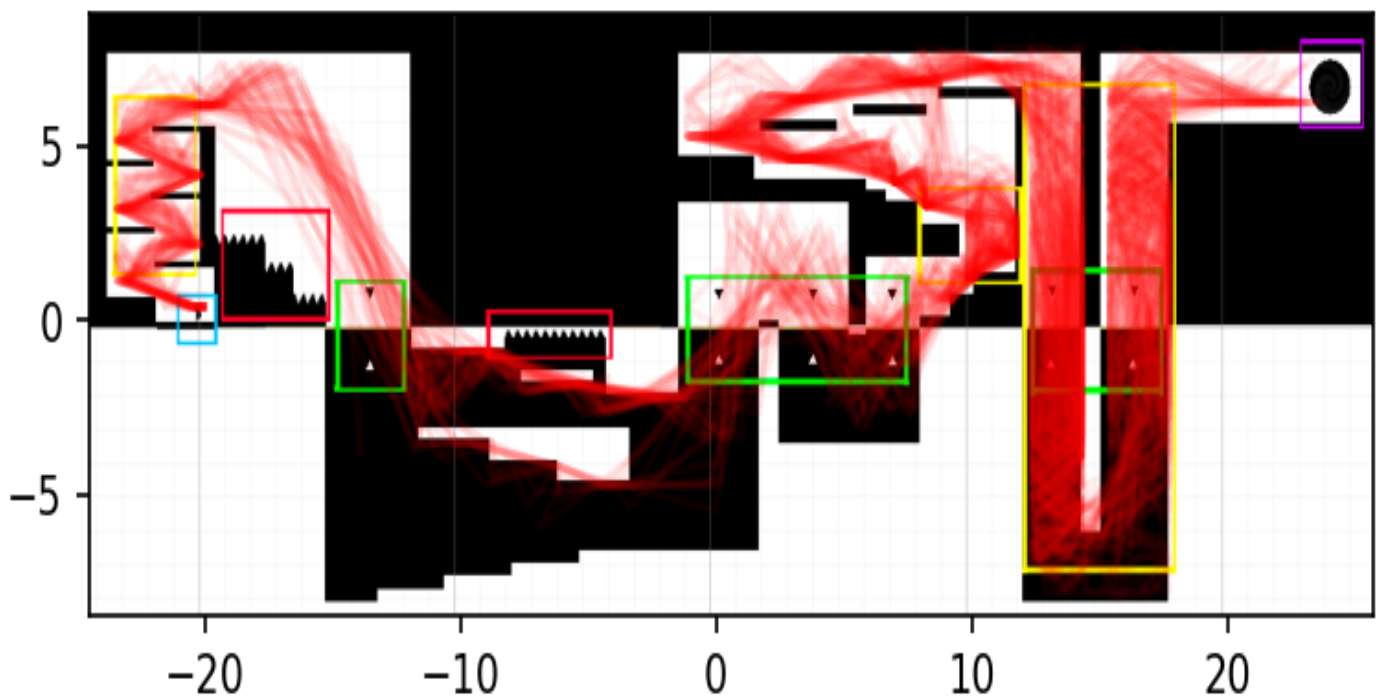
('Caminos en el nivel', 2)



('Caminos en el nivel', 3)



('Caminos en el nivel', 4)



ANÁLISIS DE LOS RESULTADOS

A la hora de analizar los resultados, hay que destacar varios factores importantes. El primero de todos es, obviamente, el amplio número de pruebas realizadas con usuarios de entre 20 y 29 años en comparación al resto de rangos de edad. Dado que nos movimos por nuestros círculos y redes sociales y no se realizó una búsqueda pormenorizada de niveles de población equitativos en cada grupo, el mayor volumen de pruebas se ajusta de esta manera. Además, también se tratan en su mayoría de jugadores experimentados (o al menos, se autodenominan como tal), lo cual también tiene que ver con los círculos de difusión de la prueba de usuario y con la naturaleza digital y costumbre de jugar a videojuegos que suelen tener las personas que cubren este rango de edad.

Esto influye ampliamente en el número de respuestas recibidas que percibían los primeros niveles del videojuego como fácil o muy fácil, ya que tienen un carácter más relacionado con el tutorial y aprender a manejar las mecánicas que con el reto en sí. Al ser dirigidos a quienes menos experimentados estén, tiene sentido que la gran mayoría de usuarios hayan elegido que la dificultad era muy baja. También deducimos que esto influye en que no se encontrasen apenas respuestas en el extremo contrario, reduciéndose a 1 o 2 el número de usuarios que perciben como muy difícil la dificultad de los niveles. Con todo esto, aún así vemos que la dificultad en los niveles donde se producen más retos se percibe de una manera equilibrada.

El segundo factor importante a destacar era la existencia de algún *bug* en la build del videojuego que impedía la correcta realización de algunas pruebas y de su posterior análisis. Algunos usuarios experimentaron un bug en el movimiento que impedía que se pudiera avanzar en el nivel, por lo que algunas pruebas acabaron ahí, y otras tuvieron que reiniciarse, pudiendo influir en el siguiente resultado. No obstante, sí que pudimos seguir la pista a los primeros intentos de usuarios que tuvieron que reiniciar la partida, teniendo pese a todo resultados más realistas de las partidas reales (considerando como reales aquellas que se realizaron sin perturbaciones como estas). Al final, estos errores fueron más complicados a la hora de analizar los resultados (puesto que tuvimos que hacer muchas más comprobaciones y cálculos para ajustar los resultados de manera realista) que para la realización de las pruebas en sí por parte de los usuarios.

Este factor en la diferencia de edad y experiencia se puede ver también en que el número de muertes de los grupos con más edad y de los grupos con menor experiencia es mucho mayor que los más jóvenes. Con esto podemos analizar que el diseño de niveles y de las mecánicas (lo que más nos interesa) de nuestro juego no es del todo accesible para los jugadores menos experimentados.

Para seguir respondiendo a la pregunta de “¿Cómo de difícil es cada nivel?” podemos entonces ver que en el nivel 2 claramente ha habido un pico de dificultad, pese a que las respuestas del formulario no lo reflejan como tal. Hemos detectado una mayor cantidad de muertes y de tiempo en el que se tarda en completar el nivel, y viendo las zonas en las que estas muertes sucedieron y sabiendo que no nos han indicado una dificultad superior en este nivel, podemos concluir que el final del mismo es una zona en la que los usuarios morían más de la cuenta, ya que, pese a que debe ser retante, tampoco la intención en la

dificultad es que lo fuera más que, por ejemplo, el último nivel (que, aunque tenga menos zonas en las que se pueda morir, vemos como aún así la gente tarda más en completar el segundo nivel que el último).

Siguiendo la línea comentada anteriormente, podemos observar que en el nivel 4 la gente con ninguna experiencia percibe mucha dificultad en el nivel. Esto significa que la curva de dificultad de los niveles anteriores no está bien ejecutada del todo, porque aunque el juego no esté dirigido principalmente a quienes no tienen nada de experiencia, quizás las zonas en las que se acumulaban los jugadores (en los cambios de gravedad del final) resultan demasiado retantes para este sector de usuarios. Cabe destacar de nuevo que el volumen de usuarios con nada de experiencia ha sido mucho menor, por lo que este resultado podría variar con una muestra mayor. Por último, destacar que el nivel 1 y el nivel 3 han dado resultados positivos, con percepción y resultados de dificultad deseados.

Pasando al apartado de las mecánicas, en primer lugar podemos observar que (respondiendo a la primera pregunta planteada), la mecánica de *wall jump* es más difícil de lo que deseamos que fuera. Podemos observar que incluso gente que está experimentada y muy experimentada ha valorado que el *wall jump* era muy difícil, y el total de la gente experimentada indicó que era difícil. Además, todos los usuarios tardaron un tiempo muy similar en completar el primer *wall jump*, superior a la marca que marcamos como ideal que rondaría los 3-4 segundos. En segundo lugar, podemos observar que los cambios de gravedad han dado resultados esperados en cuanto a nivel de dificultad, pero sí que es verdad que el número de cambios de gravedad en el último nivel es bastante superior debido a que la zona final ha sido un lugar en el que muchos jugadores se han quedado estancados (también puede observarse en las gráficas de caminos).

Por último, en cuanto a la última pregunta de investigación, podemos observar que la gran mayoría de jugadores en el nivel 3 han decidido tomar primeramente el camino inferior, y no ha habido ninguna muerte en este nivel. Esto significa que las zonas de peligro de este nivel no están del todo bien colocadas puesto que no suponen ningún peligro. Relacionando los caminos que toma con la dificultad, podemos ver también lo que comentamos previamente del nivel 4: en la zona final se acumulan muchos caminos puesto que la gente no consigue realizar los *wall jump* correctamente, quedándose demasiado enfrascados en esa zona. Los niveles 1 y 2 son totalmente lineales, así que los resultados son los esperados.

CONCLUSIONES FINALES

Con este estudio hemos podido responder de manera práctica las preguntas de investigación planteadas inicialmente, incluso con un volumen de pruebas que podría haber sido mayor y más variado y con algunos errores en la entrega de formularios. Esto nos da a entender que la metodología que hemos utilizado ha sido una buena decisión, y que las métricas que medimos también reflejaron los resultados que necesitábamos. Concluimos que la mecánica del wall jump ha resultado ser más difícil de lo esperado, la de cambio de gravedad ha resultado tener una dificultad adecuada, que en algunos niveles (sobre todo el 2 y el 4) hay zonas que generan muchas muertes acumuladas y necesitan un rediseño, y, como última conclusión, los jugadores identifican bien los caminos que deben tomar y no son influyentes totalmente en la dificultad de los niveles ni las mecánicas.

Nos quedamos satisfechos con los resultados obtenidos y las herramientas utilizadas para el proyecto, pese a unas posibles mejoras en cuanto a la utilización de un servidor propio para no depender de los usuarios a la hora de recibir las pruebas, además de las posibles mejoras en cuanto a los bugs que se encontraba en la build del juego.

Como pequeño punto negativo nos hemos dado cuenta que algunas formas de plantear cuestiones a los usuarios no eran del todo efectivas. Consideramos que los datos que obteníamos no reflejaban totalmente la realidad pues nuestra pregunta era demasiado subjetiva y cada persona respondía en función de su propio criterio dando lugar a una dispersión muy grande de las respuestas y una relación muy débil con las consecuencias en la prueba.

ADENDA

ADENDA AL TRABAJO FINAL DE LA ASIGNATURA USABILIDAD Y ANÁLISIS DE JUEGOS
GDV-UCM 2021-22.

En Madrid, a 16 de Mayo de 2022.

REUNIDOS:

Los miembros del grupo 6: Eva Lucas, David Carmona, David Godoy, Jesús Ramos, Jorge Bello y Tomás López.

ACUERDAN:

Que el trabajo presentado ha sido realizado por los 6 miembros del grupo. Todos hemos contribuido en todas las partes del trabajo de una manera similar:

- Eva Lucas: Gráficas, formulario, difusión para la prueba y búsqueda de probadores, memoria, pair programming.
- David Carmona: lectura de los archivos de trazas, gráficas, interpretación de las trazas generales, representación de los caminos de los jugadores.
- David Godoy: estructura inicial del archivo de interpretación de los datos, interpretación de las trazas: cambios de gravedad y "zonas especiales", gráficas, bug fixing
- Jesús Ramos: Scripts de lectura de archivo, colaborar en pair programming durante el desarrollo.
- Jorge Bello: Gráficas varias, formulario, difusión y búsqueda de probadores, interpretación de las trazas de los jugadores (de los .json) y memoria.
- Tomás López: Lectura de datos del formulario, código base de las gráficas, gráficas de barras, memoria.

POR

Eva Lucas
David Carmona
David Godoy
Jesús Ramos
Jorge Bello
Tomás López