
VIDEOJUEGO 3D: QOMP

VIDEOJUEGOS
2020-2021 Q1

Kenny Alejandro
Tanvir Hossain



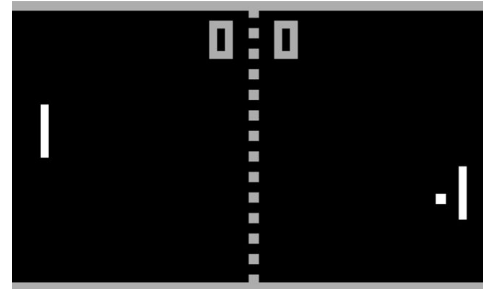
UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Índice

1	Introducción	2
2	Descripción del proyecto	4
2.1	Juego QOMP	4
2.2	Instrucciones del juego	4
2.3	Diagramas	5
2.3.1	Diagrama de flowchart	
2.3.2	Diagrama de objetos	
2.4	Partes comunas y funcionalidades del juego	6
2.4.1	Escena principal	
2.4.2	Niveles	
2.4.2.1	Nivel 1	
2.4.2.2	Nivel 2	
2.4.2.3	Nivel 3	
2.4.2.4	Nivel 4	
2.4.2.5	Nivel 5	
2.4.3	Escena de rieles	
2.4.4	Funcionalidades	
2.4.4.1	Jugador	
2.4.4.2	Enemigos	
2.4.4.3	Objeto final del nivel	
2.4.4.4	Funcionalidades del juego QOMP	
2.4.4.5	Sonido y música	
2.4.5	Objetos del juego	
3	Metodología	14
3.1	Product backlog	16
3.2	Diagrama de Gantt	19
4	Conclusiones	20
5	Bibliografía	21

1 Introducción

PONG es el juego que se ha tomado como referencia para desarrollar el juego QOMP (mirar subcapítulo 2.1). Pong es uno de los primeros videojuegos de la plataforma arcade y es un juego que está basado en el deporte de *ping-pong* con gráficos simples en dos dimensiones. El juego consiste en la intervención de



un jugador que controla una paleta moviéndola verticalmente de la pantalla, y que puede competir contra un oponente controlado por la IA u otro jugador, que controla una segunda paleta en el lado opuesto. Los jugadores usan las paletas para golpear una pelota de un lado a otro. El objetivo consiste en que uno de los jugadores consiga más puntos que el oponente al finalizar el juego y estos puntos se obtienen cuando el jugador adversario falla al devolver o responder la pelota

Pong fue un videojuego de la primera generación de videoconsolas publicado por la compañía estadounidense *Atari Inc.*, creado por *Nolan Bushnell* y *Ted Dabney*. El juego fue lanzado el 29 de noviembre de 1972. *Atari*, que era una pequeña empresa en ese momento, comenzó a fabricar los juegos y el hecho de ser un juego sencillo y entretenido lo que resultó en un aumento en el número de pedidos. En 1973, la empresa había cumplido 2.500 pedidos y, a finales de 1974, vendió más de 35.000 unidades. Aunque ya existía el primer microprocesador, no se incorporó esta tecnología a las máquinas recreativas Pong debido al alto costo y se utilizaron dispositivos discretos y lógica discreta.

El éxito de Pong hizo crear nuevas versiones de este producto. En 1974 se propone una versión casera de Pong que se conectaría a un televisor, **Home Pong**. Los desarrolladores y diseñadores como Darlene, Alcorn y Lee empezaron a crear los diseños y el prototipo, y los basó en la misma tecnología digital utilizada en sus juegos de arcade. En 1975, *Atari* convirtió a Pong en un juego de consola en el que se incorpora una cpu con la tecnología MOS 6507. Después de llegar a un acuerdo exclusivo con *Sears*, Home Pong fue un éxito instantáneo. Alrededor de 150 000 unidades se vendieron durante la temporada 1975. Pong pronto estuvo en los hogares de muchas familias estadounidenses y en algunos países se comercializaron réplicas del mismo bajo el nombre de **Telematch**. El

juego se convirtió en el producto más exitoso de Sears en ese momento, lo que le valió a *Atari* un premio Sears Quality Excellence Award.

Posteriormente se crearon muchas variantes, p.e *Pong* en 3D, nuevas versiones que contenían algunas innovaciones, aunque el juego básico sigue siendo el mismo. Varias compañías lanzaron clones para capitalizar el éxito de la consola doméstica, muchas de las cuales continuaron produciendo nuevas consolas y videojuegos. La popularidad de *Pong* declinó en la década de 1980 cuando los videojuegos pasaron de moda temporalmente, pero ya había asegurado su lugar en la historia como el juego de arcade más popular hasta ese momento.

Actualmente existen diversas versiones remasterizadas de *Pong*, hay versiones flash jugables desde el navegador y otras descargables, a la mayoría de estos juegos se le han añadido funciones y optimizado su control.

2 Descripción del proyecto

2.1 Juego QOMP

QOMP es una evolución de PONG y POMG. Qomp es un juego de pala y pelota pero hay una pequeña diferencia. El jugador controla la pelota y el espacio en lugar de la pala. Dicha pala es controlada por la IA. El objetivo es ir desde un punto origen a un punto destino pasando por diferentes mecánicas sorprendentes e intrigantes obstáculos de las pantallas, p.e: las palas, que impiden el paso del jugador desde una pantalla hacia otra. El eslogan de este juego sería “si eres pelota, escapa”.

2.2 Instrucciones del juego

Al inicio tenemos una pantalla principal del juego que consta de un menú navegable mediante el *mouse*. Se puede elegir las siguientes opciones:

- play: para iniciar el juego
- instructions: para controlar el juego, la tecla “space” para mover la pelota y “s” para ir rápido
- credits: para ver información acerca de los desarrolladores

Una vez dentro del juego (previamente hemos tenido que seleccionar la opción play) se muestra una visión global de los niveles que este dispone (ver figura 1). El jugador empezará desde el nivel 1 y tendrá que atravesar todos los obstáculos o enemigos para pasar de un nivel a otro mediante los rieles. Esto se consigue si previamente se ha obtenido la llave, que estará colocada en algún lugar del mapa, para desbloquear dicho nivel. Se puede usar el god mode pausa con la tecla “p” en que los obstáculos no tendrán efectos en el juego y el jugador podrá atravesar los niveles fácilmente. Además, hemos agregado para poder desbloquear y acceder a los niveles de forma directa solo apretando las teclas del nivel correspondiente del 1 al 5, y con la tecla “c” se puede tener una visión general de los niveles.

El jugador tendrá el objetivo de superar todas las fases del juego y llegar al final para superar al *boss*. No se descuentan las vidas del jugador sino que dispondrá de vidas infinitas. Sin embargo, si algún enemigo o obstáculo mata al jugador entonces este volverá a empezar desde un punto anterior que denominamos

checkpoint, que registra la última posición por donde ha pasado el jugador. En cada nivel se añade un grado más de dificultad respecto al anterior en que el jugador se verá enfrentado a nuevos retos y enemigos del juego.

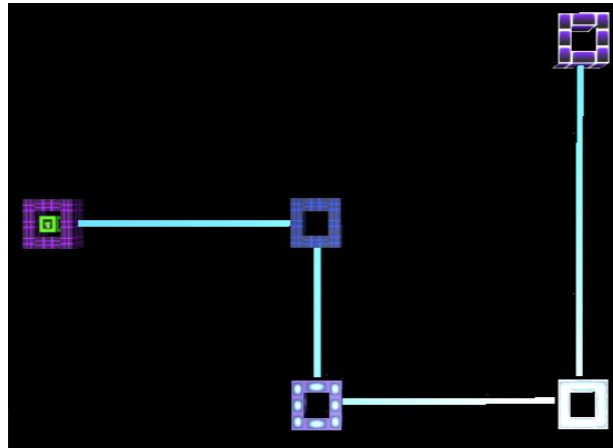


Figura 1. Visión general de los niveles del juego

En estos niveles se pueden destacar varios componentes o objetos, mirar el subapartado 2.3.2 diagrama de objetos.

2.3 Diagramas

2.3.1 Diagrama de flowchart

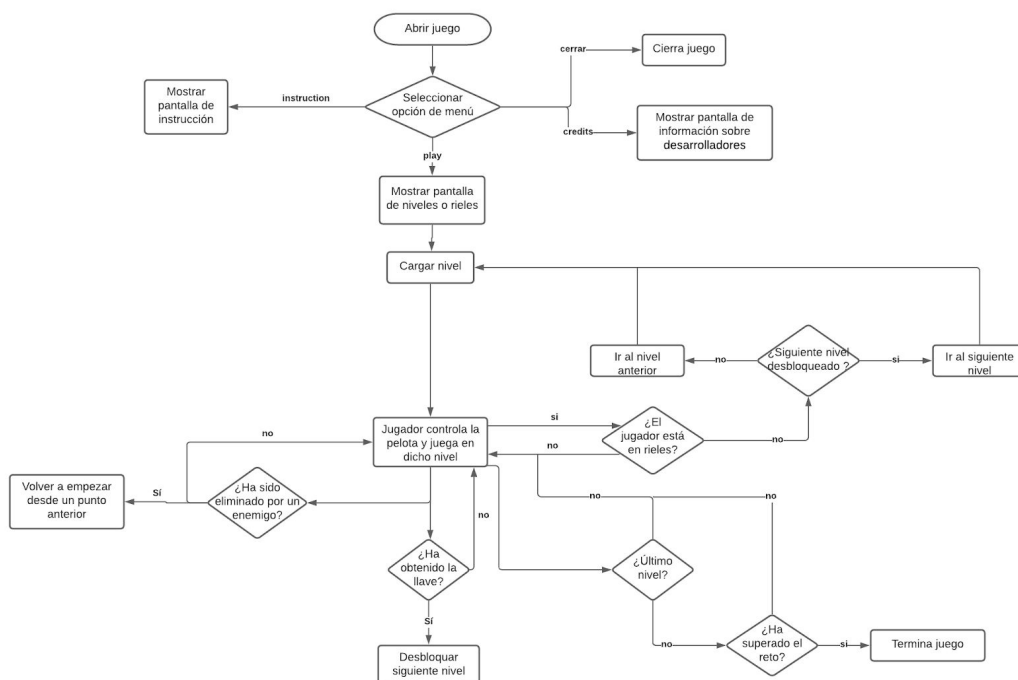


Figura 2. Diagrama flowchart del funcionamiento del juego

2.3.2 Diagrama de objetos

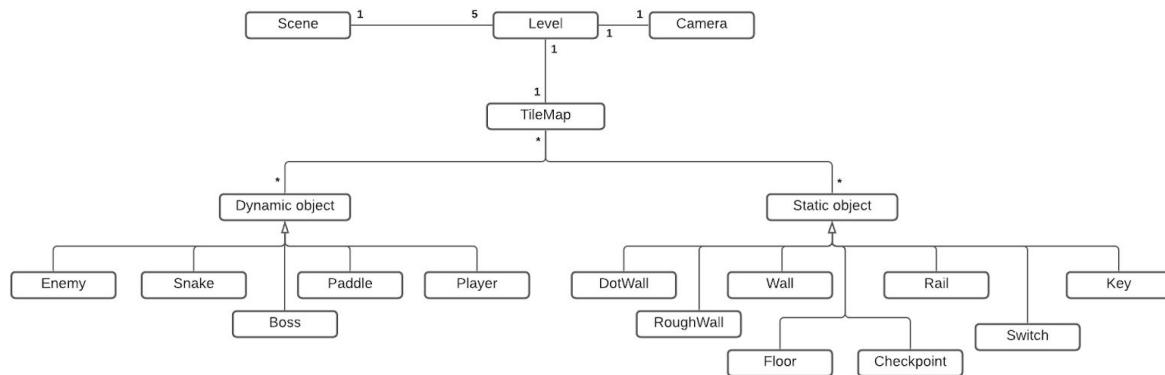


Figura 3. Diagrama de objetos del juego

2.4 Partes comunas y funcionalidades del juego

Hemos desarrollado un videojuego de la plataforma arcade que está basado en los juegos PONG, POMG y QOMP. La mecánica del juego consiste superar los diferentes niveles para llegar a la última fase en que el jugador tendrá que atravesar diferentes obstáculos en diferentes pantallas. Para ello, el jugador puede controlar la pelota para cambiar la dirección vertical.

En la parte arte y temática hemos seguido el modelo del juego de *Geometry dash*. En primer lugar hicimos un análisis para saber si era idóneo seguir este modelo para nuestro juego. Después hicimos una recopilación de imágenes, texturas, colores, etc. para poder diseñar nuestros mapas del juego. Posteriormente los sonidos. Cabe mencionar que aparte de este modelo también hemos seguido los diferentes aspectos o características de los juegos POMG y QOMP.

2.4.1 Escena principal

La estructura básica consta de 4 pantallas: menú, play, instrucciones y créditos. La pantalla principal o menú se abre cuando inicializamos el juego. Se ofrecen las opciones de jugar, ver las instrucciones y los créditos del juego.



Figura 4. Menú principal

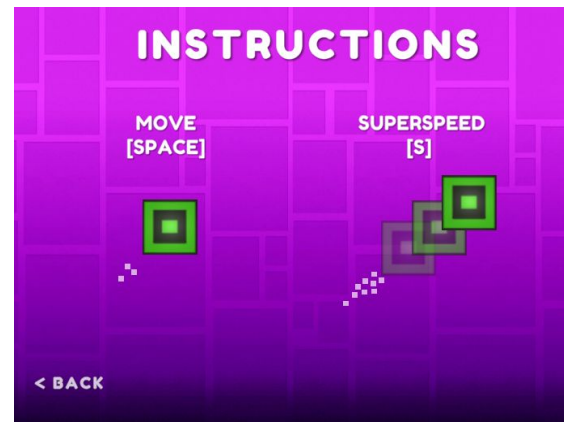


Figura 5. Instrucciones

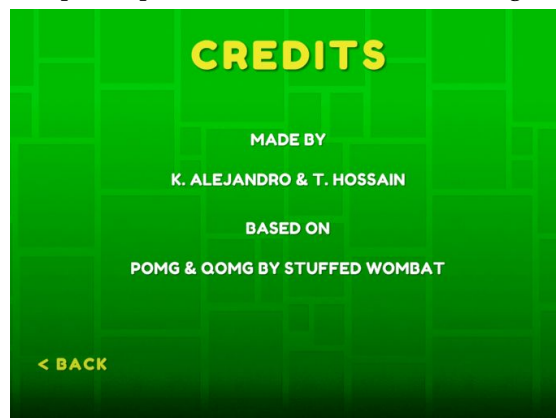


Figura 6. Créditos

2.4.2 Niveles

Disponemos de 5 niveles. En cada nivel la dificultad aumenta gradualmente y la estética cambia. Cuando empezamos a crear los niveles en unity pensamos crearlos de manera dinámica, es decir construir nuestro protocolo en el fichero y posteriormente leerlo y generar el mapa de tiles. Sin embargo, nos dimos cuenta que era más fácil hacer el uso de la técnica drag & drop que nos ofrece el motor unity. Para ello, antes de construirlo en unity hicimos un diseño provisional (que es nuestro protocolo) en el excel para poder coordinar fácilmente. A continuación mostramos un ejemplo.



Figura 7. Diseño del mapa del nivel 3

El diseño completo de los mapas se encuentra en el siguiente enlace:

https://docs.google.com/spreadsheets/d/17HhOkuRgGO9gmXhhPk-_E88ZJszAxqjwdc7h_EtHWc4/edit#gid=1113813793

2.4.2.1 Nivel 1

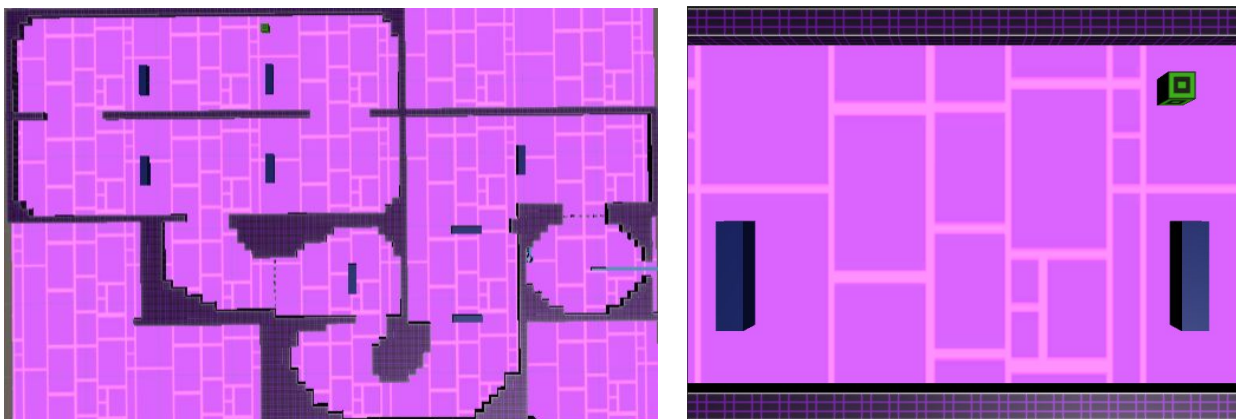


Figura 8. Nivel 1

Este nivel está basado al juego POMG. Se presenta al iniciar el juego y consideramos un nivel bastante fácil de superar. La única dificultad de esta fase es atravesar las palas IA.

2.4.2.2 Nivel 2

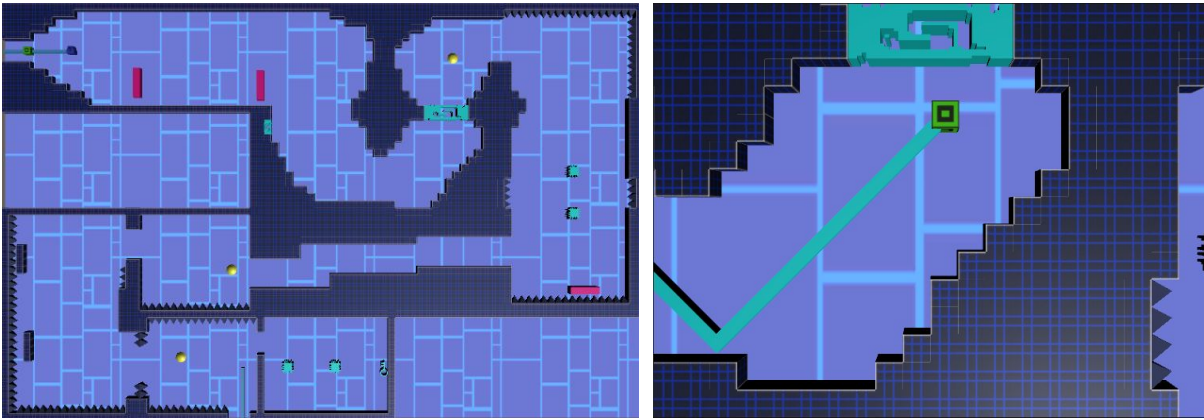


Figura 9. Nivel 2

En este nivel hemos incrementado un poco la dificultad. Hemos incorporado más obstáculos o enemigos. También algunos *checkpoints*. Se ha implementado la funcionalidad de la serpiente en que el jugador ha de hacer el movimiento zig-zag para destruir la pared u obstáculo de la serpiente.

2.4.2.3 Nivel 3

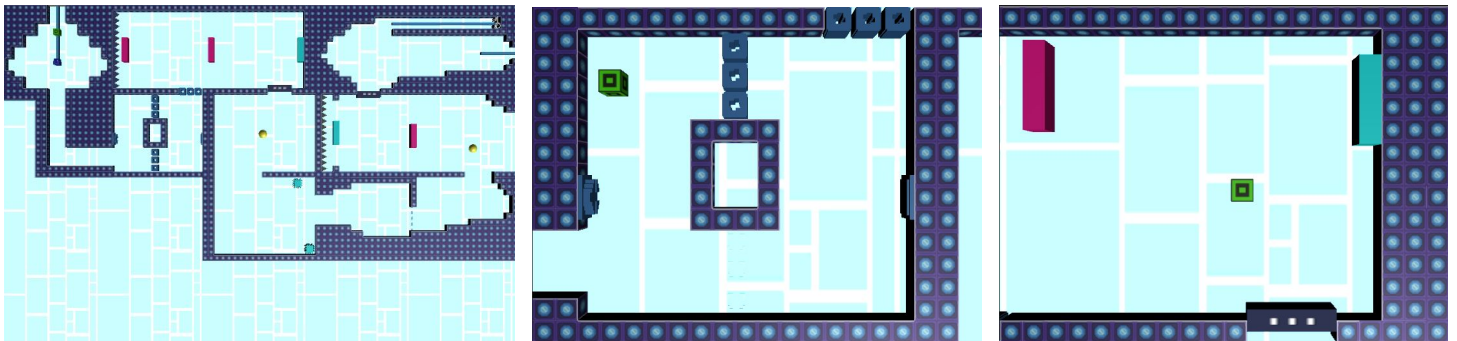


Figura 10. Nivel 3

En este nivel hemos añadido más funcionalidades para incrementar el grado de dificultad. Por una parte se observa que hay un obstáculo o pared para pasar de una pantalla a otra y esta pared se consigue abrir mediante un botón de encendido. Por otra parte, se observa otro obstáculo que se abre si el jugador consigue chocar contra la pala IA (en la imagen está representada en color cian). En este caso el objetivo de la pala es huir del jugador.

2.4.2.4 Nivel 4

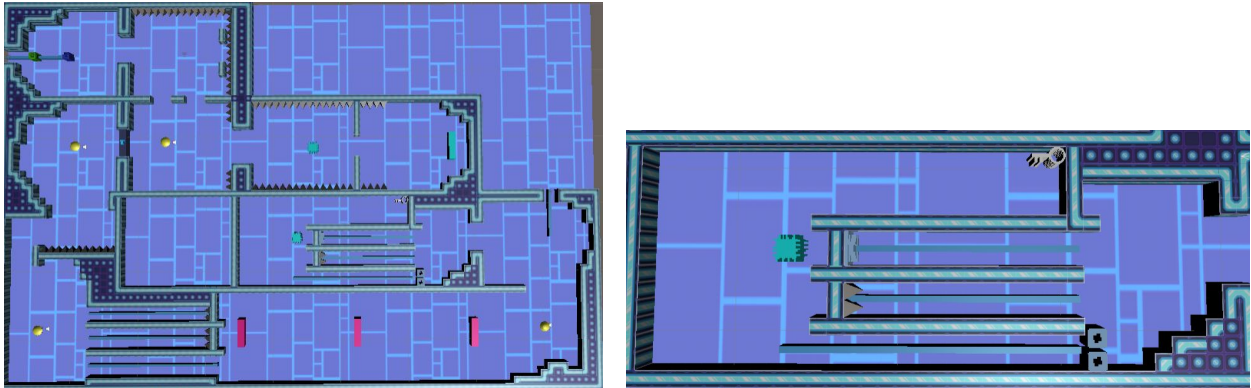


Figura 11. Nivel 4

Siguiendo el mismo criterio de antes hemos incorporado más elementos y obstáculos.

2.4.2.5 Nivel 5

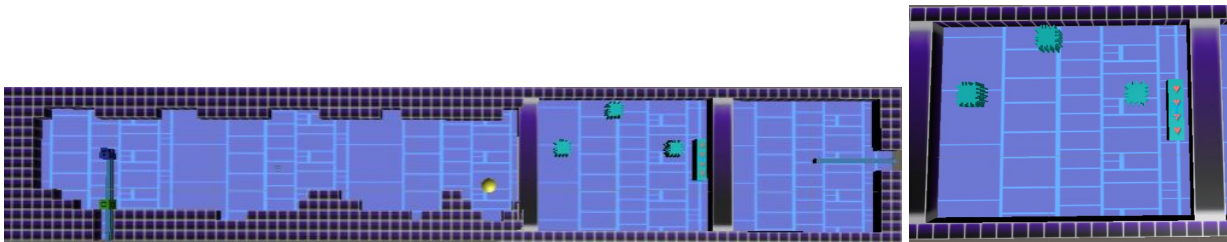


Figura 12. Nivel 5

Este es el último nivel del juego en que el jugador tendrá que enfrentarse al *boss* para superar el reto. Tiene una cierta complejidad ya que el jugador estará encerrado en una pantalla rodeados de enemigos. El objetivo es eliminar las vidas del boss mediante colisiones y evitar no tocar los obstáculos.

2.4.3 Escena de rieles

Para poder pasar de un nivel a otro es necesario desbloquearlo mediante la obtención de la llave. Y esta transición de pasar de un mapa al otro la hacer mediante los rieles. De esta manera el jugador sabrá en qué nivel del mapa se encuentra y hacia qué nivel se dirige (mirar figura 1). Pero hay que tener en cuenta que si el jugador no consigue obtener la llave entonces no podrá controlar la pelota en el siguiente nivel (mirar figura x), es decir se denegará el acceso al siguiente nivel.

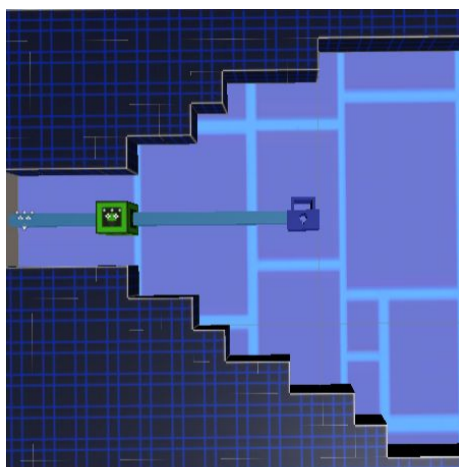


Figura 13. Acceso denegado a un nivel concreto

En la figura 13 observamos que el jugador quiere acceder en este nivel pero se encuentra cerrado ya que no ha obtenido la llave en el nivel anterior.

2.4.4 Funcionalidades

2.4.4.1 Jugador

El jugador puede controlar la dirección de la pelota, exclusivamente en el eje vertical. También puede aumentar la velocidad de la pelota que denominamos *superspeed*, en el juego se consigue mediante la tecla “s”.

2.4.4.2 Enemigos

Las palas actúan de forma que hacen de obstáculos al jugador. Estas tienen aplicadas las IA en que comprueba unas determinadas condiciones de la matriz lookup. Tenemos 3 palas diferentes. La primera se mueve constantemente y se le aplica una IA. La segunda hace el obstáculo cuando el jugador se encuentra cerca de su zona. La tercera huye del jugador para evitar el choque contra el jugador (mirar figura 10).

También tenemos otros obstáculos como paredes o cubos con o sin puntas, algunos ejemplos se pueden observar en las figuras 10 y 11.

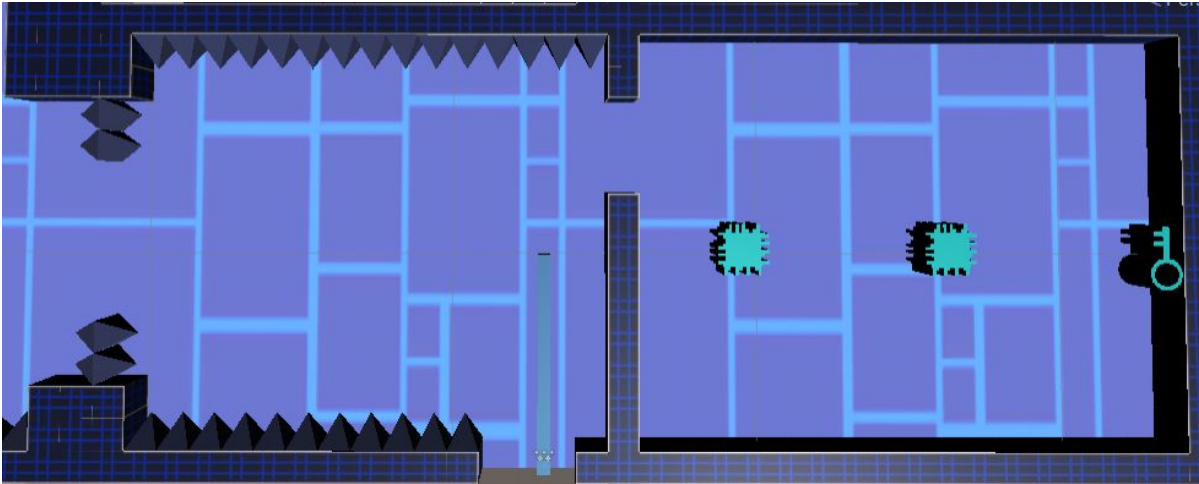


Figura 14. Obstáculos

2.4.4.3 Objeto final del nivel

Al final de cada nivel tenemos una llave para terminar dicho nivel y mediante una *rail* se puede pasar de este nivel al siguiente.

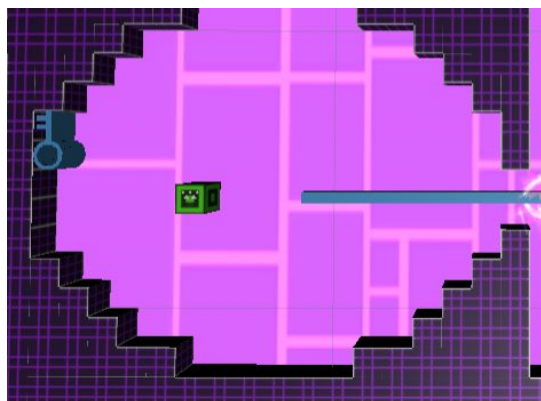


Figura 15. Parte final de un nivel

2.4.4.4 Funcionalidades del juego QOMP

Hemos implementado las siguientes funciones:

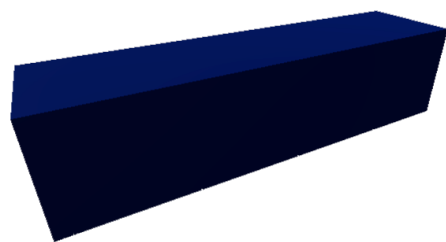
- los rieles para para de un nivel a otro
- movimiento de *snake*
- obstáculos que involucran *switch-on* y *switch-off*
- diferentes tipos de palas

2.4.4.5 Sonido y música

Los sonidos están recogidos del juego QOMP, Geometry dash y otros juegos.

2.4.5 Objetos del juego

A continuación mostramos los objetos que componen el juego.



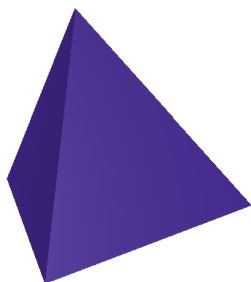
Paddle



Player



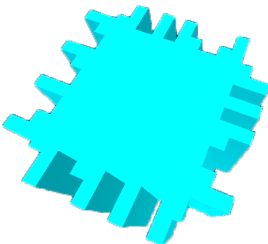
Wall shape



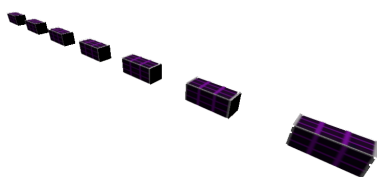
Rough wall



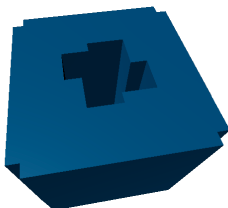
Snake



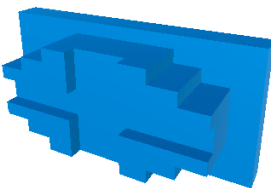
Rough wall



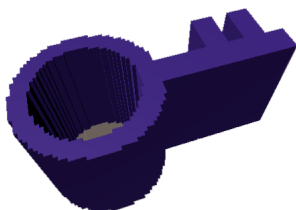
Dot wall



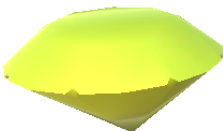
Rough wall



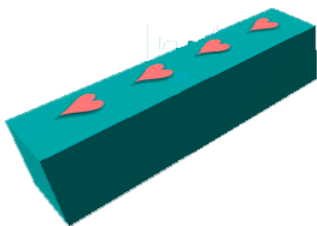
Switch



Key



Checkpoint



Boss

3 Metodología

La metodología de trabajo que hemos decidido usar es **Scrum**. Usaremos distintas herramientas para desarrollar nuestro proyecto de manera ágil. Para hacer la gestión del proyecto usaremos la herramienta *Notion*. Es una herramienta que nos permite planificar el proyecto, definir tareas, asignar roles, etc. Hemos elegido esta herramienta porque ya la usamos en los cursos anteriores y queríamos seguir usándola por el conocimiento y su facilidad de uso.

Para seguir la tradición de la metodología scrum nos reuníamos 3 a 4 veces en cada sprint (daily scrum), ya que es muy importante para el desarrollo completo del proyecto. En esta reunión respondíamos las siguientes preguntas:

- ¿Qué hice ayer?
- ¿He tenido algún tipo de problema durante el desarrollo o si tengo algún problema?
- ¿Qué es lo que voy a hacer hoy?

Daily Scrum meeting: 4/12/2020

- ▼ Tanvir
 - ▶ Hice las funciones de detectar colisiones, gestionar las palas y el jugador
 - ▶ Tuve problemas con las configuraciones de physics, lo cual las respuestas de colisiones no son adecuadas
 - ▶ Voy a solucionar bugs de collision detection y avanzar en la documentacion de memoria
- ▼ Kenny
 - ▶ Hice el diseño de los 2 primeros niveles y he plasmado la parte inicial de diseño del nivel 1 en unity
 - ▶ Tuve problemas con los tamaños de los cubos, que no encajaban con el tamaño del mapa
 - ▶ Voy ha terminar de plasmar el nivel 1 y tratar de diseñar los niveles 3, 4

Figura 16. Ejemplo de daily scrum meeting

En el siguiente link se encuentran todos los detalles:

<https://www.notion.so/VJ-3D-6c0c4fdb5c794a4a8c019196a178bb2a>

Por otra parte, también hemos aplicado la metodología **Extreme programing** y hemos seguido los fundamentos de pair programing para familiarizarnos con el entorno unity, solucionar bugs, discutir, etc. Para ello hemos usado la aplicación *anyDesk* que nos permite conectar de manera remota y trabajar conjuntamente.

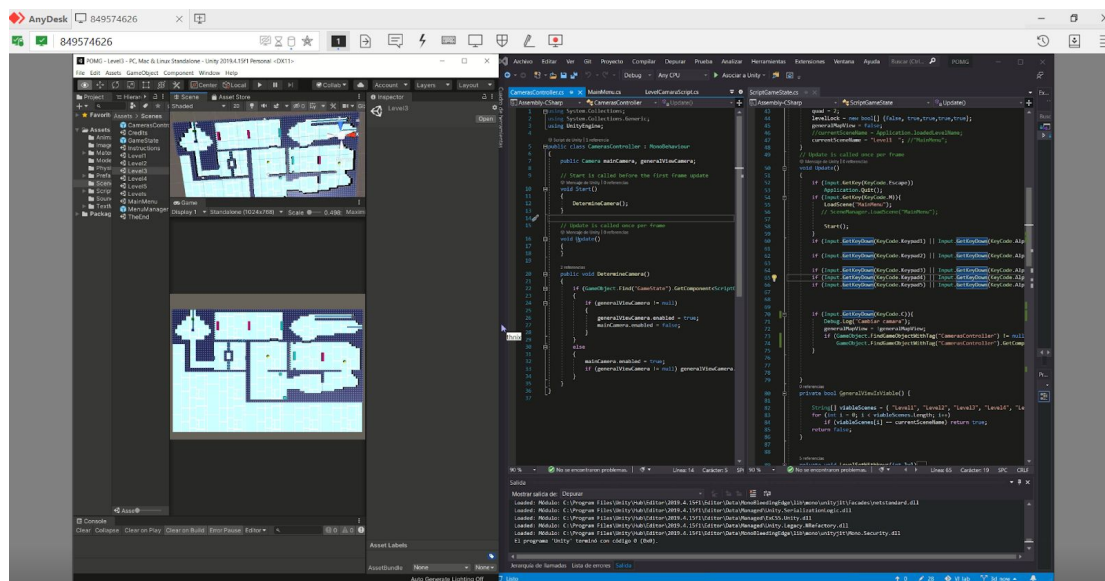


Figura 17. Pair programming con la herramienta anyDesk

En la figura 17 se puede observar que los dos estamos trabajando conjuntamente, por ejemplo, uno se ocupa de Unity y otro de los scripts.

Respecto al entorno de trabajo hemos creado un repositorio de Github para gestionar diferentes versiones de código. Hemos seleccionado *Unity 3d* como el motor de juego, y el lenguaje de programación a usar es C# (csharp).



Figura 18. Ejemplo del timeline de los commits del repositorio de Github

3.1 Product backlog

El juego que vamos a desarrollar tiene bastantes funcionalidades y puede resultar un poco difícil de gestionarlas. Es por esta razón que las hemos dividido en distintos temas e historias de usuarios.

TEMA	Historia de usuario
Menú	<ul style="list-style-type: none">● Pantallas principales menú, play, instrucciones, créditos
Mapa	<ul style="list-style-type: none">● Crear y dibujar nivel 1, 2, 3, 4, 5● Añadir temática por cada nivel● Añadir checkpoint● Mostrar mapa● Hacer scroll de mapa● Gestionar cambios de niveles● Conectar rieles entre niveles
Jugador	<ul style="list-style-type: none">● Detectar colisiones● Gestionar movimiento● Gestionar superspeed
Pala	<ul style="list-style-type: none">● Aplicar IA en los movimientos● Gestionar tres tipos de palas
Cámara	<ul style="list-style-type: none">● Traslaciones entre pantallas
Sonido	<ul style="list-style-type: none">● Música de fondo y efectos especiales
Obstáculo	<ul style="list-style-type: none">● Gestionar rough wall● Gestionar rough box● Gestionar Switch● Dot wall
Funcionalidades adicionales	<ul style="list-style-type: none">● Snake style● Keys and locks

Tabla 1. Temas y historias de usuarios

Toda esta información del product backlog se puede encontrar en el siguiente enlace:

<https://www.notion.so/db9ce6a9b83746fe9cbcefe2909949bb?v=9a0184c555f342d58799a74a984fe686>

Nuestro *product backlog* está dividido en varias columnas, cada una de ellas representa un tema. Cada columna también está dividida en múltiples filas, siendo cada una de ellas una historia de usuario, que a la vez, están nombradas por unas etiquetas. Si se clicla en cada una, se podrá ver la descripción de cada tarea, sus story points, bugs, etc.

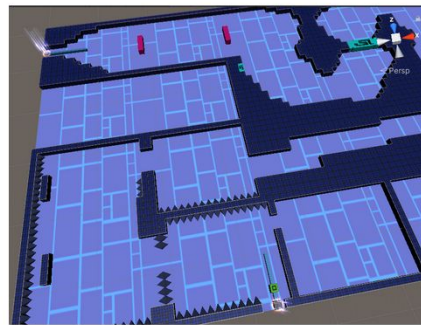
Description

- En algunos casos al cambiar de un nivel el player no esta en la posición que debería.

Steps to reproduce

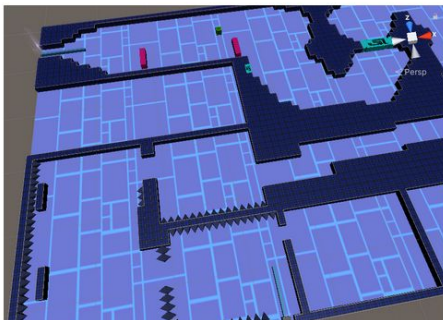
1. Reproducir en nivel 3
2. Ir a la base de retorno hacia nivel 2.
3. Esperar hasta que se reproduca nivel 2.
4. El player no aparece en la posición que debería.

2. Donde debería aparecer:



Screenshots

1. Donde aparece:



Solution:

El error fue debido a que solo se inicializaba la **position** del **Transform** del **Player** pero no la **position** del componente **RigidBody** asociado al **Player**.

Figura 19. Ejemplo de reportar bugs

El objetivo de la primera sprint fue familiarizarnos con el entorno unity 3d, definir el ciclo del juego y las funcionalidades básicas del jugador y pala.

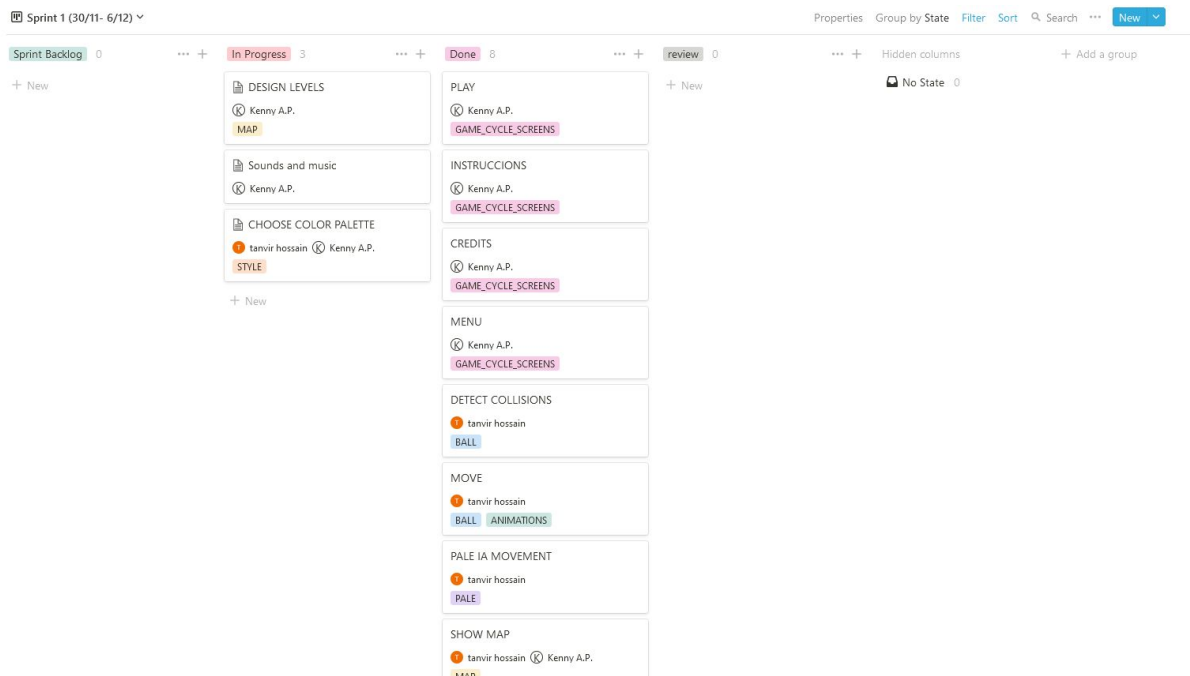


Figura 20. Sprint 1 (30/11/20 - 6/12/20)

En la segunda iteración quisimos terminar el nivel 1 incorporando todas las funciones necesarias o involucradas del juego. Una vez terminada, empezar con los niveles 2 y 3. Hay que mencionar que en esta semana tuvimos varios bugs, p.e: en la traslación de la cámara, en los rails, pasar de un nivel a otro, etc. Pudimos solucionarlos a tiempo y conseguimos nuestros objetivos.

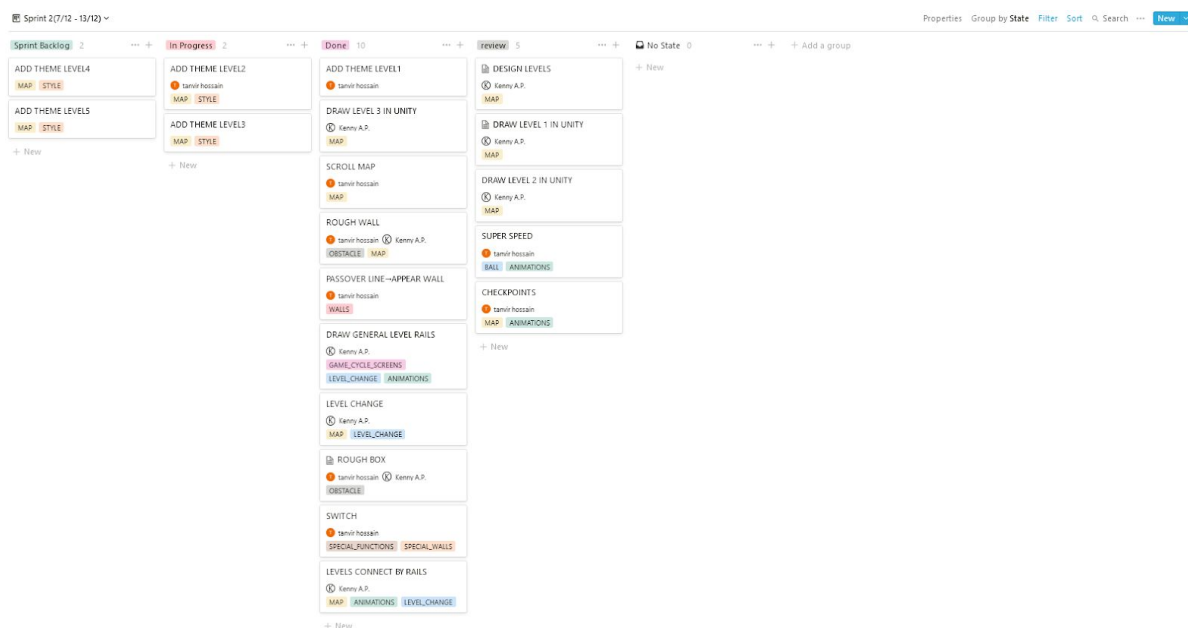


Figura 21. Sprint 2 (7/12/20 - 13/12/20)

En la última semana los objetivos fueron acabar los niveles 3, 4 y 5, solucionar algunos bugs anteriores e incorporar funcionalidades de QOMP.

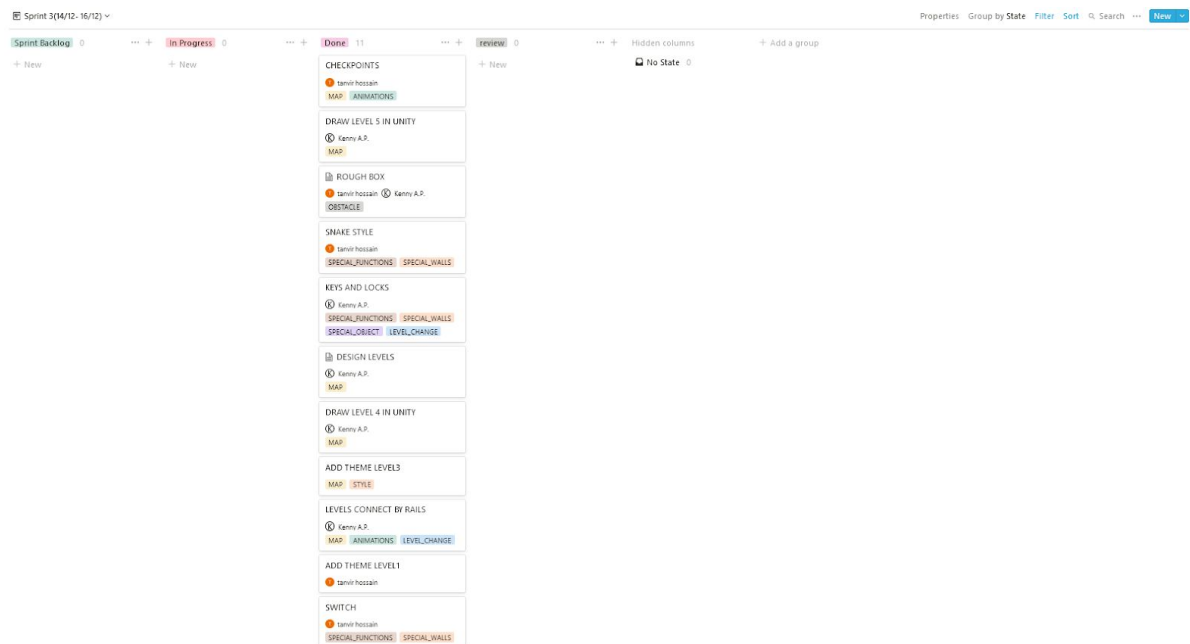


Figura 22. Sprint 3 (14/12/20 - 16/12/20)

3.2 Diagrama de Gantt

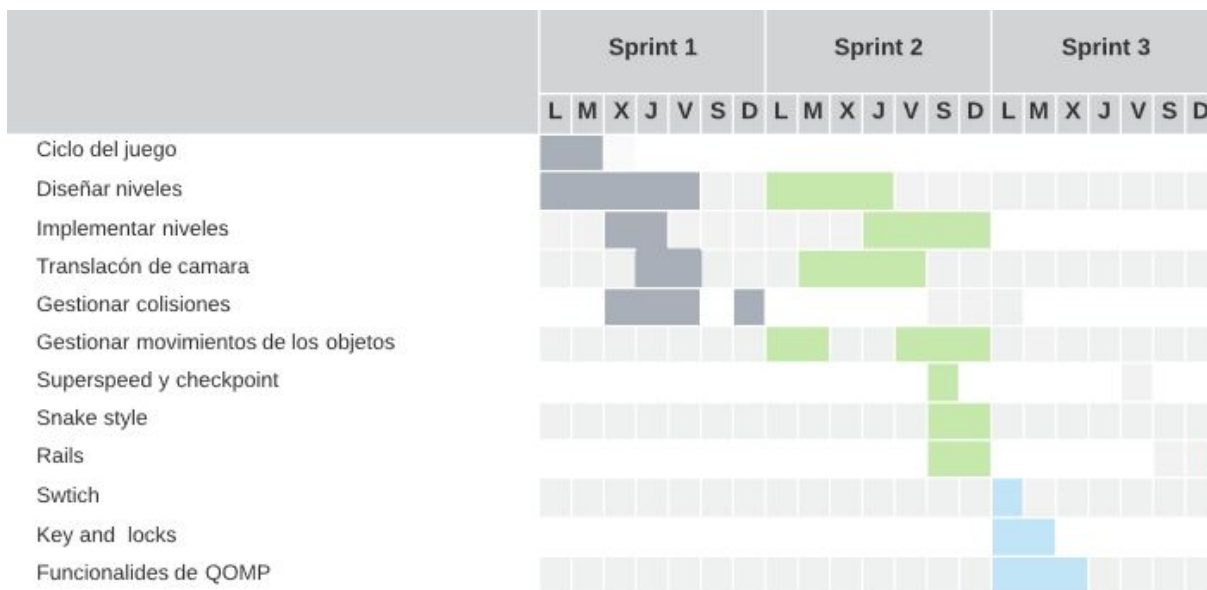


Figura 23. Diagrama de Gantt

4 Conclusiones

La idea de implementar el juego clásico PONG en 3D e incorporar nuevas funcionalidades era un reto. Al inicio del proyecto tuvimos ciertas dudas en desarrollar el juego con la API OpenGL o con un motor de juego como *Unity*. Pensábamos que sería mejor seguir con OpenGL ya que el juego tiene la lógica 2D y por tanto todo lo que vimos en el juego 2D se podría aplicar en esta práctica. La diferencia estaba en la gestión de modelos y sistemas de partículas. Estuvimos viendo unos tutoriales acerca del entorno unity y vimos que nos proporciona herramientas útiles como drag & drop, sistemas de partículas, animaciones, etc. Finalmente, elegimos unity.

Antes de iniciar la fase de implementación tuvimos que familiarizarnos con Unity y con el lenguaje de programación *csharp*. Dedicamos unos tres días para ver algunos tutoriales de YouTube. Después empezamos con la fase de elaboración en que definimos nuestro product backlog para empezar la fase de implementación. Al inicio del sprint íbamos un poco atrasado con el cumplimiento de las tareas ya que teníamos carga de trabajos de otras asignaturas. Sin embargo, en cada sprint fuimos sumando más historias de usuarios y esto se ha notado bastante en las iteraciones 2 y 3.

Creemos que hemos cumplido con la mayoría de las cosas que se piden en el enunciado y estamos satisfechos por el resultado obtenido del proyecto. Pensamos que ha valido la pena dedicar tanto esfuerzo y dedicación para desarrollar el juego 3D en el que hemos aprendido nuevos conceptos de la programación de videojuegos.

5 Bibliografía

“Pong”

<https://es.wikipedia.org/wiki/Pong>

Página de Wikipedia del juego Pong

“Atari PONG”

<http://www.computinghistory.org.uk/det/4007/Atari-PONG>

Artículo sobre la evolución del juego Pong

“Qomp”

<https://www.taminggaming.com/game/Qomp>

Review sobre el juego Qomp

“Documentación oficial de Unity”

<https://learn.unity.com/tutorials>

Tutoriales de unity

“Unity3d game development tutorial”

<https://www.youtube.com>

Tutoriales de unity 3d del canal Sebastian Lague

“Example of a game”

[https://www.youtube.com/watch?v=SvileTt2_Lc&list=PLFt_AvWsXl0ctd4dgE1](https://www.youtube.com/watch?v=SvileTt2_Lc&list=PLFt_AvWsXl0ctd4dgE1F8g3uec4zKNRV0)

F8g3uec4zKNRV0

Tutoriales de un game series

“Diseño de los modelos”

<https://ephtracy.github.io/>

Herramienta para modelos 3D

“Temática de geometry dash”

<https://www.google.com>

Búsqueda de texturas, colores y sonidos