

Star Fox: Quarantine

Videojuegos - Proyecto 3D
Curso 2019/2020 (Q2)

Arnau Ruana
Víctor Barroso

Índice

Índice	2
1. Introducción	4
2. Juego original	5
3. Descripción del proyecto	7
3.1. Instrucciones	7
Movimiento	7
Apuntado	8
Disparo	8
Evasión (roll)	8
Silencio	8
Utilidades	9
3.2. Diagrama de ventanas	9
Menú	11
Controles	12
Opciones	12
Créditos	13
Salir	14
Level #1	14
Level #2	15
Pausa	16
Fin del juego (Victoria)	16
Fin del juego (Derrota)	17
3.3 Diagrama de flujo	18
3.4 Elementos del juego	20
3.4.1 Personajes	20
Player	20
Torreta	22
Tanque	23
Droide	24
Space Fighter	24

3.4.2 Obstáculos	25
3.4.3 Disparos y explosiones	26
4. Metodología	30
5. Conclusiones	33
6. Bibliografía	34

1. Introducción

En el siguiente documento expondremos cómo se ha llevado a cabo la realización de un proyecto consistente en la programación de un videojuego 3D mediante un motor gráfico llamado *Unity*. En este caso, el videojuego al que se hará un homenaje y en el cual nos basaremos es el exitoso Star Fox 64.

Cómo fue recibido el juego original, el funcionamiento del nuestro, cómo lo hemos organizado, y qué fechas hemos usado para controlar el avance del mismo, son algunas de las preguntas que esta memoria tratará de clarificar.

Para mostrar ciertas características de nuestro proyecto, añadiremos diversas fotografías que ayudarán a entender mejor las dudas, pese a que éstas sean tomadas durante la fase de desarrollo del videojuego y, por ende, puedan llegar a variar pequeños detalles una vez el juego quede terminado.

Por último comentaremos nuestra valoración personal del trabajo realizado frente a los retos que se planteaban en este proyecto y, en general, en la asignatura de videojuegos.

2. Juego original

Star Fox 64 (conocido en Europa como *Lylat Wars*) es un *scrolling shooter* en 3D de combates aéreos diseñado para la consola [Nintendo 64](#). El juego fue lanzado por primera vez en 1997. Fue desarrollado por *Nintendo Entertainment Analysis and Development* ([Nintendo EAD](#)) y distribuido por **Nintendo** y Nintendo eShop (servicio en línea ofrecido por Nintendo Network).

Debido a su gran éxito, el juego fue relanzado en varias ocasiones para distintas plataformas:

- En 2003 para la consola [iQue Player](#).
- En 2007 para Consola Virtual (servicio de descarga de videojuegos clásicos ofrecido por Nintendo para [Wii](#), [Nintendo 3DS](#) y [Wii U](#)).
- En 2011 sale a la venta un remake llamado [Star Fox 64 3D](#) para la Nintendo 3DS.

Este juego, destinado a casi todos los públicos (en Europa PEGI 7 y en Alemania para mayores de 12), tuvo un gran número de ventas. Fue el segundo juego con más ventas en 1997, sólo superado por Mario Kart 64. Vendió más de 300,000 copias en los 5 primeros días en EEUU, y más de 1 millón a lo largo de ese mismo año. El juego, desde su fecha de lanzamiento, ha vendido más de 4 millones de copias (para la Nintendo 64).

Además de su gran éxito entre el público, fue también aclamado por la crítica. Destacan su sistema de ramificación de niveles así como la gran ejecución del giro del jugador. Otro de sus grandes logros son los distintos modos multijugador, lo que le da un amplio valor de *replay*. Un punto muy común en la mayoría de reviews es la cantidad de voces añadidas en la lectura de los diálogos, sin precedentes si hablamos de juegos de cartucho.

Otros de los puntos que también se aplaudieron ampliamente fueron la precisión de los controles analógicos, diseños de jefes, la implementación del rumble pak y las escenas cinemáticas. Sus notas, tanto en reseñas como en críticas, hablan por sí solas:

- Metacritic: **88/100**
- Edge: **9/10**
- GamePro: **5/5**
- IGN: **8.7/10**

Star Fox 64 fue nombrado, según *Guinness World Records Gamer's Edition*, el 45º mayor juego de todos los tiempos, y en 1997 *Electronic Gaming Monthly* lo considero el 39º.

En cuanto al desarrollo, cabe destacar que la idea original fue el desarrollo del Star Fox 2, el que sería el segundo juego de la serie, y que saldría para la Super Nintendo Entertainment System. Sin embargo, finalmente el productor de la serie, Shigeru Miyamoto, decidió retrasar el lanzamiento para poder aprovechar la potencia de la Nintendo 64.

El equipo de programación fue liderado por Kazuaki Morita (programador de Nintendo que, a día de hoy, es conocido por sus trabajos en gran cantidad de juegos de la serie [The Legend of Zelda](#)). Por otro lado contó con Mitsuhiro Takano (escritor) y con Kōji Kondō y Hajime Wakai (compositores).

El juego contiene 4 MB de información de voces, muestreados a 8 kHz, comprimidos en un ratio aproximado de 1:3 o 1:4. Por otro lado el juego salió al mercado con un *bundle* con el dispositivo Rumble Pak, como se mencionó anteriormente. Este dispositivo extraíble, creado por Nintendo, proporciona retroalimentación háptica, es decir, recrea el sentido del tacto. Esta tecnología aplicada al Star Fox 64 aumenta la sensación de realismo, provocando que el mando vibre ante ciertas situaciones, como recibir daño.

Para más información sobre el juego, se puede ver un trailer que incluye gameplay, o incluso la propia compra de esta entrega, en la página [web oficial de Nintendo](#).

3. Descripción del proyecto

Star Fox: Quarantine es un juego basado en el **Star Fox 64**. Su principal objetivo es avanzar por el territorio enemigo, evitando ser destruido e intentando acabar con el mayor número de enemigos posibles, con el fin de dejarlos lo más desarmados posible para que no te maten. Para ello, nuestro protagonista pilotará un caza espacial equipado con un cañón, que le permitirá acabar con sus enemigos. Además, el caza le permitirá moverse con facilidad por el entorno hostil, con la posibilidad de realizar una maniobra de esquite (*roll*) para evitar ser golpeado. Para sobrevivir no solo deberá evitar los disparos de la tropa rival, sino también chocarse con ellos o con el propio entorno del planeta donde se encuentra.

El juego consta de cuatro escenas realizadas en *Unity* que se distribuyen como se muestra a continuación (figura 1).

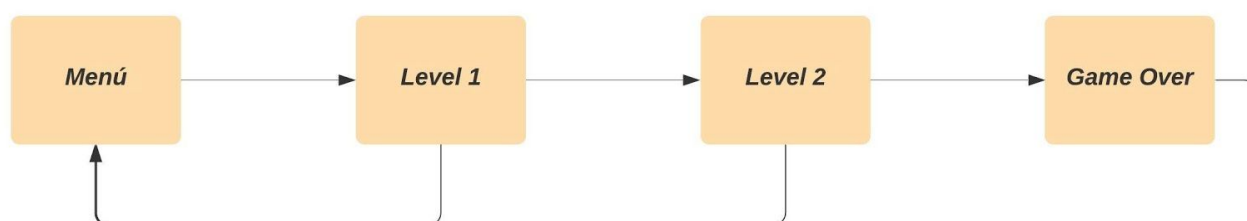


Figura 1. Distribución de escenas en Unity.

3.1. Instrucciones

En esta sección detallaremos las instrucciones del videojuego separándolas en seis principales subapartados.

Movimiento

Como hemos especificado anteriormente en este documento, el jugador controla una nave espacial. Su movimiento a lo largo del mapa es controlado mediante las **flechas del teclado** o las tradicionales **WASD** como se muestra a continuación:

- ↑ | **W** : movimiento hacia arriba
- ↓ | **S** : movimiento hacia abajo

- → | **D** : movimiento hacia la derecha
- ← | **A** : movimiento hacia la izquierda

El movimiento hacia delante es efectuado en el juego por defecto sin la posibilidad de volver hacia atrás. No obstante, el movimiento de dicha nave no es totalmente libre, sólo puede moverse a lo largo de una cierta área rectangular definida por el propio videojuego.

Apuntado

El apuntado de la nave es completamente independiente de su movimiento, facilitando así la evasión de los disparos enemigos mientras se les apunta y dispara, con el objetivo de enriquecer la experiencia de juego.

El propio **cursor del ratón** funciona como punto de mira hacia el cual el proyectil saldrá disparado impactando contra todo aquello que se encuentre en su camino.

Disparo

Los proyectiles de la nave se disparan con el **botón izquierdo del ratón**. El jugador puede elegir si disparar uno a uno clicando varias veces o disparar múltiples proyectiles manteniendo dicho botón del ratón apretado.

Evasión (*roll*)

Con la finalidad de evadir los disparos de los enemigos o los diferentes obstáculos que se presentan en el mapa, el jugador debe mover la nave para esquivarlos (como hemos explicado en uno de los apartados anteriores).

No obstante, el jugador tiene la posibilidad de usar, durante un periodo de tiempo relativamente corto, un movimiento especial llamado *roll* el cual le permite moverse más rápido y evadir mejor las diferentes hostilidades que se le presenten.

Esta acción se lleva a cabo pulsando la **tecla 'espacio'** del teclado. Cabe destacar que tiene un tiempo de *cooldown* o enfriamiento hasta poder ser utilizado de nuevo con el fin de no poder abusar en exceso de esta habilidad.

Silencio

El volumen general del juego puede ser silenciado o activado en cualquier momento pulsando la tecla **'M'**.

Cuando el juego se encuentre silenciado la música ambiental y los efectos especiales dejarán de ser audibles, así como las interacciones con los botones del menú.

Utilidades

A continuación veremos las teclas que activan funcionalidades especiales del juego para mejorar la experiencia del mismo o para facilitar la navegación entre ventanas para debugar y/o corregir con mayor facilidad:

- **ESC** → pausa el juego en las escenas #1 y #2 mostrando el menú de pausa.
- **F1** → carga directamente la escena #1.
- **F2** → carga directamente la escena #2.
- **F5** → carga directamente el menú principal del juego (el punto de partida).
- **F6** → carga directamente la pantalla de agradecimientos (sin importancia de que no se hayan superado los dos niveles).
- **F9** → activa/desactiva el modo dios (godmode), el cual permite al jugador no recibir ningún tipo de daño.

3.2. Diagrama de ventanas

El siguiente diagrama (figura 2) muestra la distribución de ventanas o pantallas de las que se compone nuestro juego. Cabe destacar la gran funcionalidad del menú, sirviendo de puente, ya que se puede acceder a muchas otras pantallas, así como ser accedido desde estas.

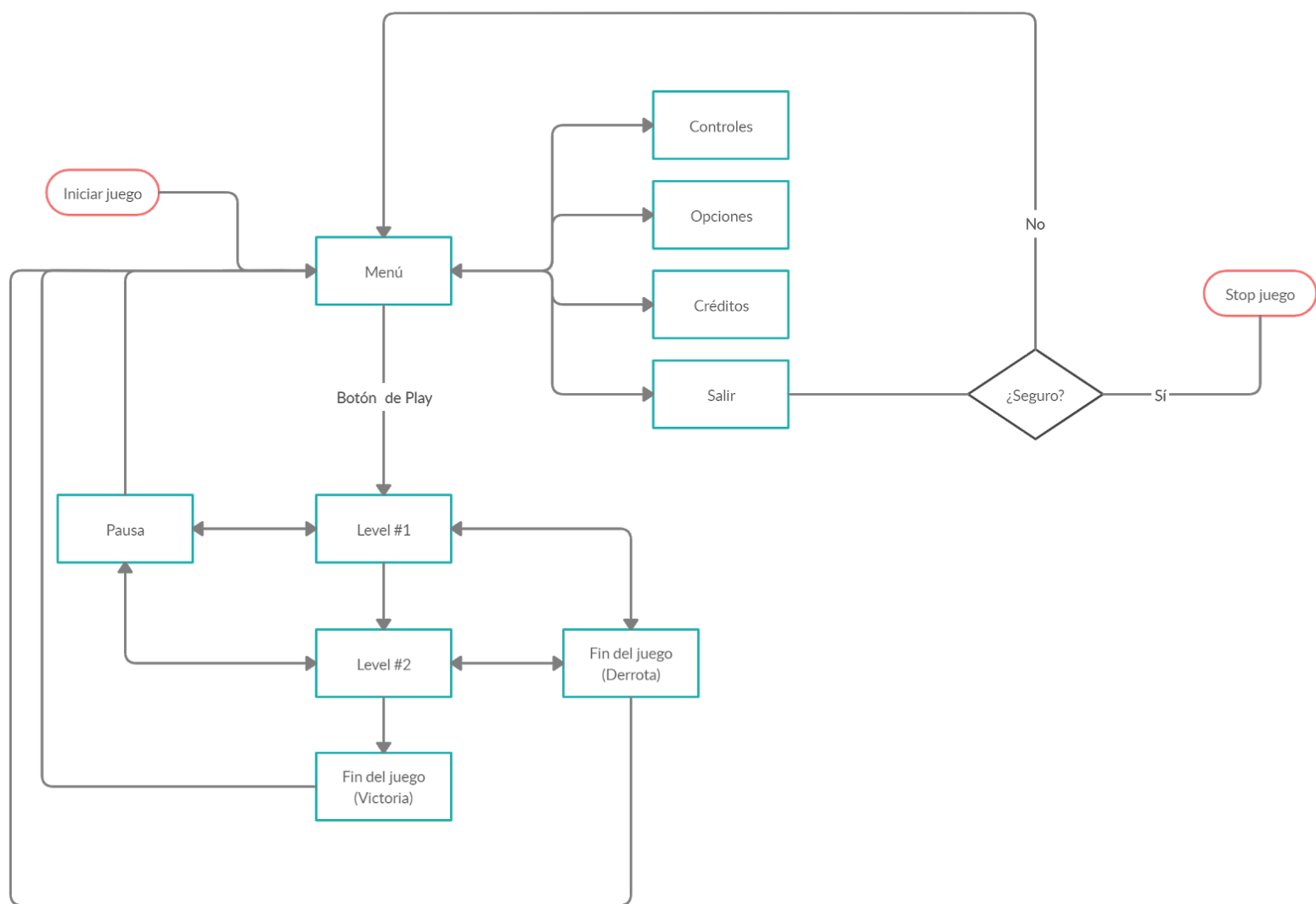


Figura 2. Diagrama de ventanas del juego.

A continuación, haremos un breve resumen de cada una de estas pantallas.

Menú

El menú es la ventana principal del videojuego. Sirve de puente entre los niveles y las distintas opciones que se muestran o para empezar a jugar. Muestra el título del juego seguido de un panel con botones (figura 3) que nos llevarán a las distintas ventanas a las que podremos acceder.

Estos botones son:

- **Jugar** → Nos llevará al primer nivel del juego
- **Opciones** → Nos llevará al menú de opciones
- **Controles** → Nos llevará al menú de controles
- **Créditos** → Nos llevará al menú de créditos
- **Salir** → Nos llevará a la pantalla de salir

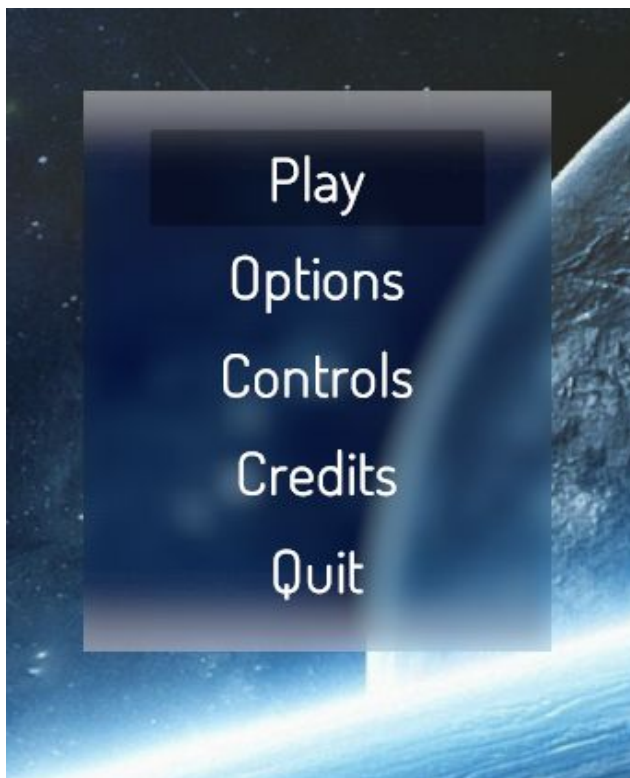


Figura 3. Panel del menú principal

Controles

Se accede desde el Menú principal. Nos mostrará un panel (figura 4) con los distintos controles necesarios para jugar.

Cuenta con un botón Back para volver al Menú.



Figura 4. Panel de controles

Opciones

Se accede desde el Menú principal. Este panel (figura 5) permite modificar el sonido del juego.

Cuenta con un botón Back para volver al Menú.

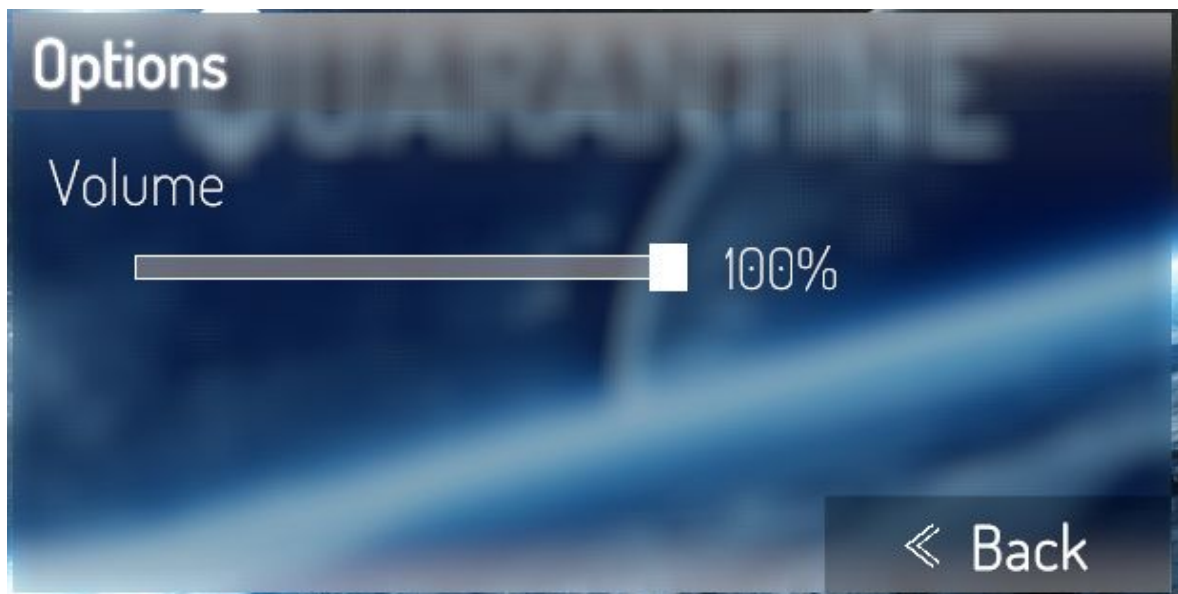


Figura 5. Panel de opciones

Créditos

Se accede desde el Menú principal. Muestra los desarrolladores del juego (figura 6).

Cuenta con un botón Back para volver al Menú.

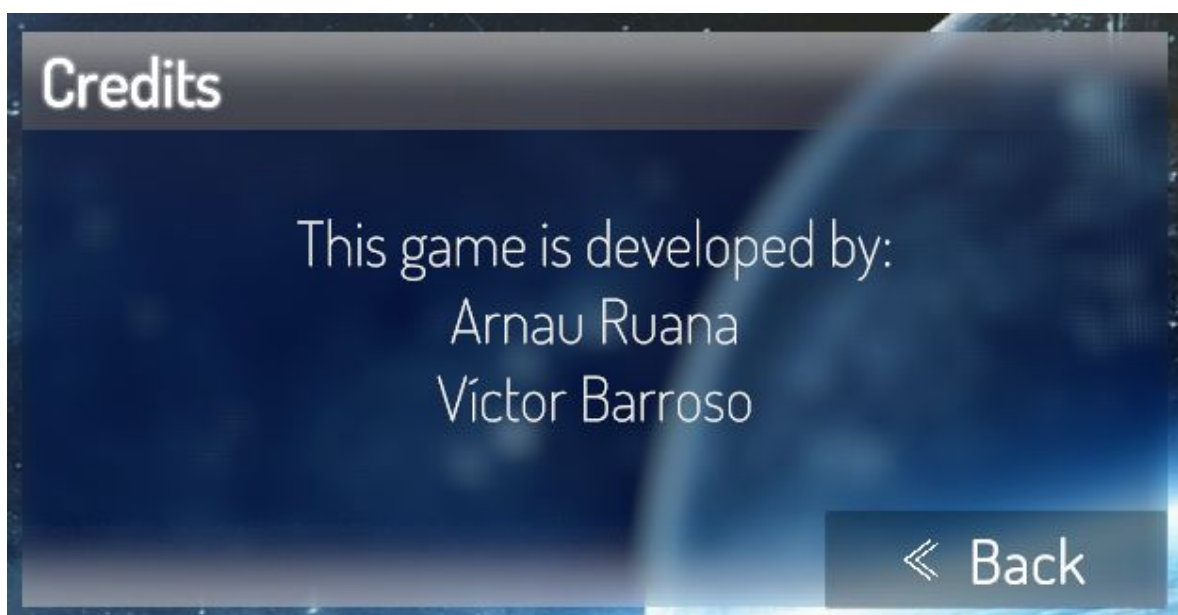


Figura 6. Panel de créditos

Salir

Se accede desde el Menú principal. Un nuevo panel pedirá una confirmación de salida (figura 7).

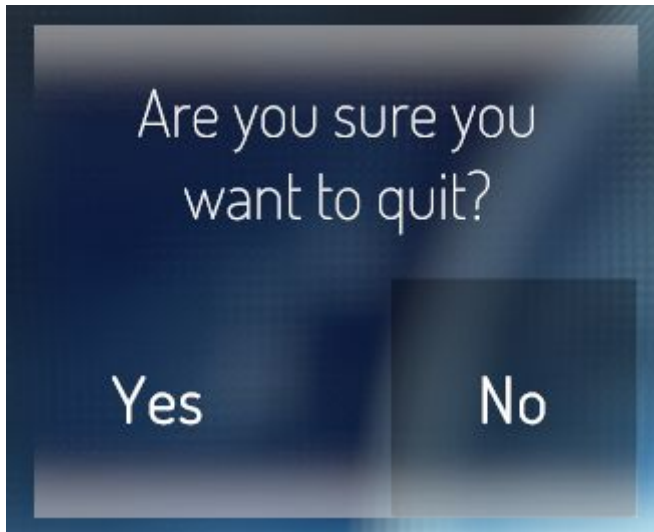


Figura 7. Confirmación de salida

En caso positivo, cerrará la aplicación del juego.
En caso negativo, devolverá al jugador al Menú.

Level #1

El nivel 1 del juego representa el espacio exterior. En dicho nivel el jugador deberá esquivar tanto los asteroides que se le interpongan en su camino como destruir las defensas enemigas, compuestas por Láseres, Droides y Space Fighters (explicados más adelante).

Una vez pasados todos los enemigos y obstáculos exitosamente, el jugador habrá ganado este nivel y será trasladado automáticamente al siguiente.

A continuación, en la siguiente fotografía (figura 8) se puede apreciar la panorámica del nivel.

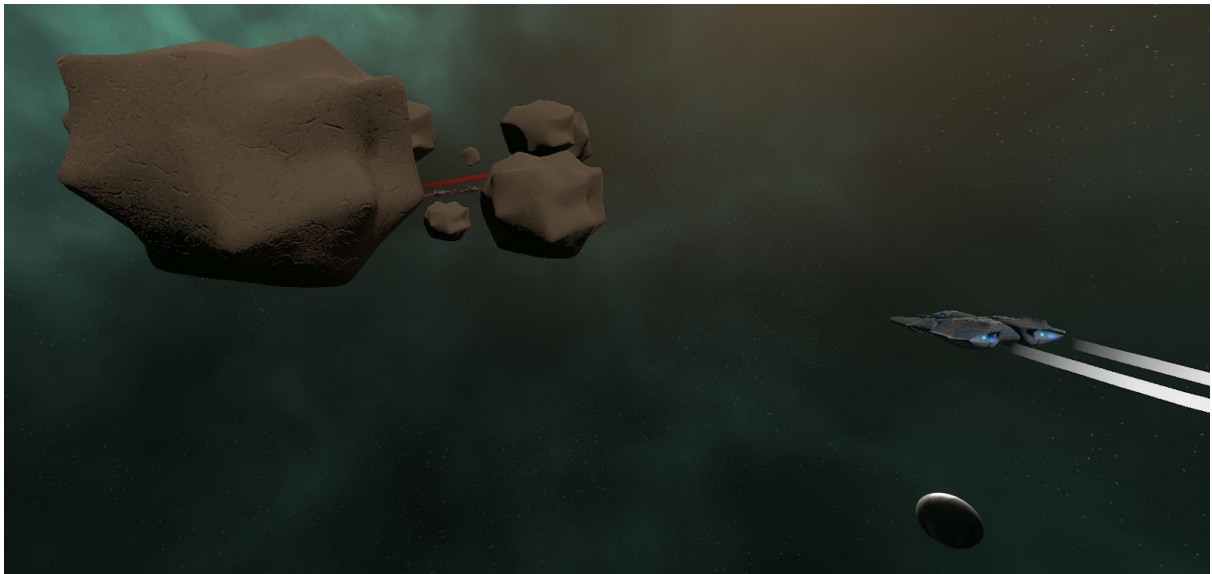


Figura 8. Panorámica del primer nivel en el espacio

Level #2

El nivel 2 transcurre en un planeta enemigo con montañas y vegetación. Ahora el equipo enemigo estará compuesto por Torretas fijas a torres, Tanques y Space Fighters. Además tendrás que evitar colisionar tanto con ellos como con Edificios, Árboles y el propio Terreno.

La siguiente figura (figura 9) representa una vista previa del nivel.



Figura 9. Panorámica del segundo nivel terrestre

Pausa

Pantalla que aparece al pausar el juego durante un nivel con la tecla '**ESC**'.
Se muestran 3 opciones (figura 10):



Figura 10. Menú de pausa durante los niveles.

1. **Continuar** → Quita la pausa y el jugador sigue donde lo dejó.
2. **Reintentar** → Vuelves a empezar el nivel que estabas jugando.
3. **Salir** → Devuelve al jugador al Menú Principal.

Fin del juego (Victoria)

Pantalla que aparece cuando superas los dos niveles diseñados.
Aparece un mensaje de agradecimiento junto a los desarrolladores del juego como se muestra en la fotografía siguiente (figura 11).

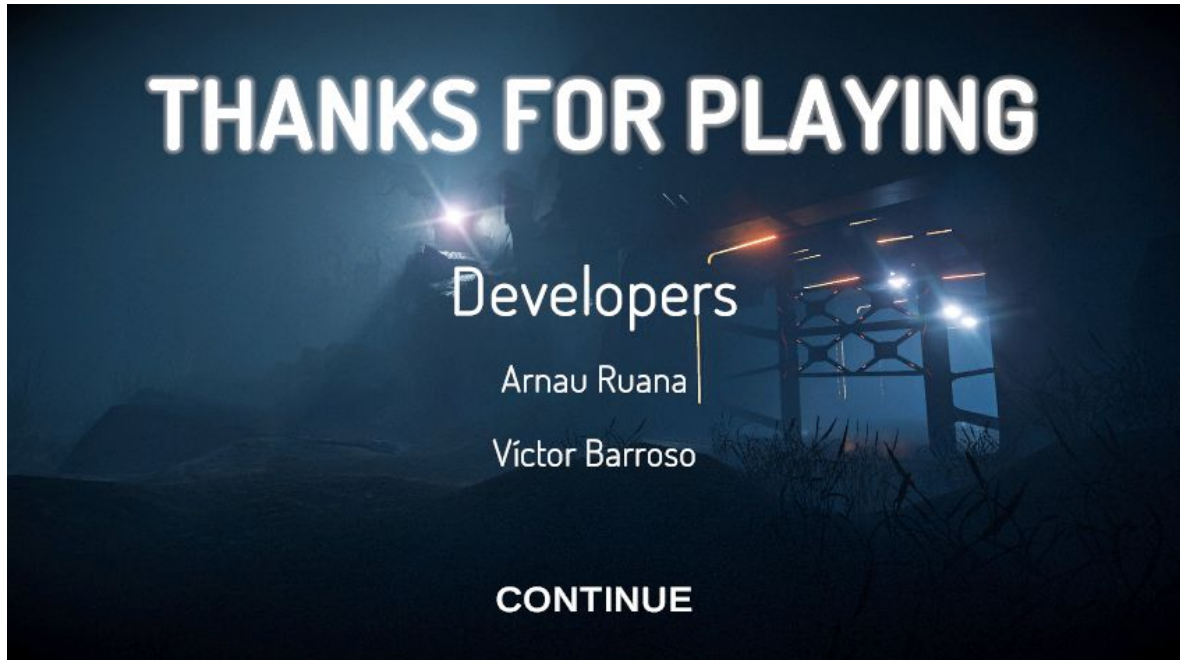


Figura 11. Mensaje de agradecimiento por terminar el juego.

1. **Continuar** → Devuelve al jugador al Menú Principal.

Fin del juego (Derrota)

Pantalla que aparece cuando la vida del jugador llega a 0.

Se muestran 2 opciones (figura 12):



Figura 12. Menú de derrota durante los niveles.

1. **Reintentar** → Vuelves a empezar el nivel que estabas jugando.
2. **Salir** → Devuelve al jugador al Menú Principal.

3.3 Diagrama de flujo

Para dejar un poco más claro el flujo de eventos hemos decidido realizar 2 diagramas de flujo. El primero de ellos (figura 13) hace referencia al menú principal y como de éste se puede llegar a los otros menús, salir de la aplicación, o cargar el nivel 1, que nos llevaría al segundo diagrama. Nótese que ambos diagramas de flujo se explican por sí solos.

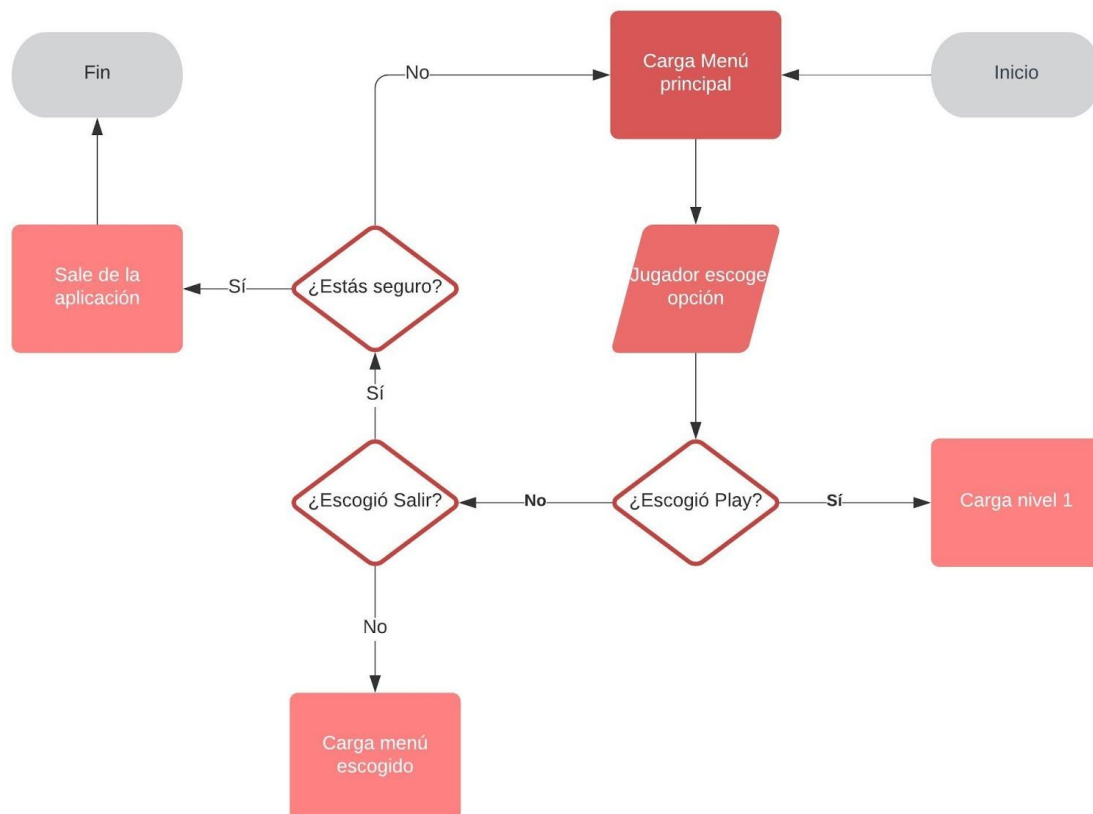


Figura 13. Diagrama de flujo del Menú Principal

Una vez cargas el nivel, entra en juego el segundo diagrama (figura 14), que muestra los eventos principales que pueden suceder *ingame*.

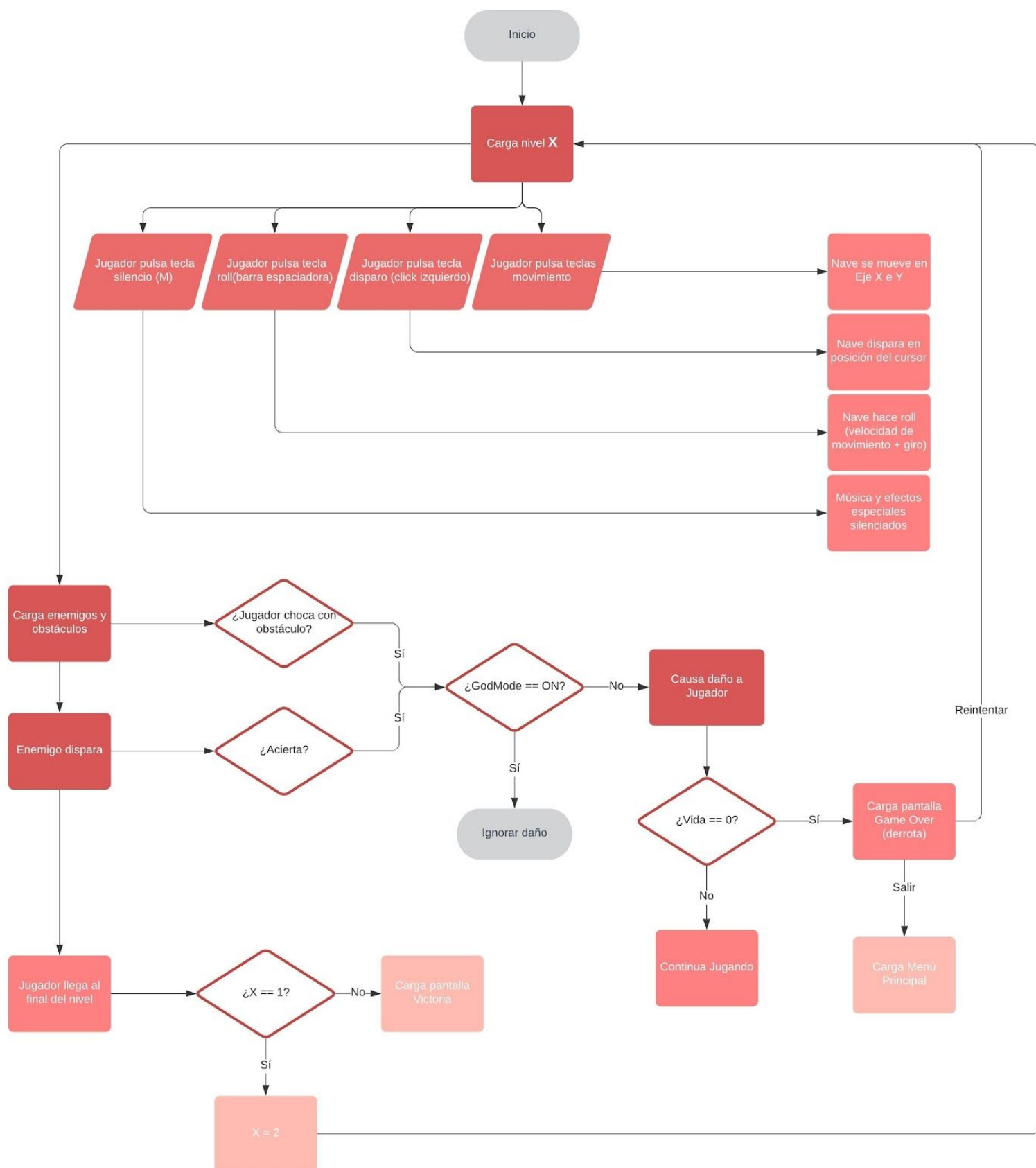


Figura 14. Diagrama de flujo de Niveles

3.4 Elementos del juego

Antes de comenzar a hablar sobre cada uno de los elementos y entidades que forman nuestro juego cabe mencionar que todos ellos son colisionables de algún modo, lo que implica que disparan un evento siempre que entran en colisión con otro objeto. Esto permite a nuestro jugador principal chocarse contra todos los objetos del mapa. Así mismo, todas estas entidades tienen sus propios efectos de sonido para ser totalmente distinguidas en función del mismo.

A continuación desglosaremos y agruparemos cada uno de estos objetos en tres grandes grupos.

3.4.1 Personajes

Player

Es la nave de la que el jugador tomará los controles.

El modelo de esta nave (figura 15) tiene un diseño futurista, con una gama de colores basadas principalmente en el gris y el azul.



Figura 15. Modelo de la nave Player.

Los colores de esta nave están escogidos para que se pueda distinguir claramente de los de las naves enemigas, cuyo color más característico es el rojo.

Además, esta nave al avanzar deja tras de sí 2 rastros (figura 16). En primer lugar una pequeñas llamas azules que salen de sus motores, creadas a través de un sistema de partículas. En segundo lugar, dos estelas blancas creadas con trails.



Figura 16. Rastros de la nave Player.

Por otro lado, en cuanto al movimiento, cabe destacar la utilización de dos objetos adicionales. Estos objetos son dos esferas. Una llamada Move Ball, que será la que se mueva de posición, y la nave se encargará de dirigirse hacia ella a una velocidad estipulada. Gracias a ella conseguimos el movimiento de la nave en el eje X e Y de la pantalla.

La otra esfera es Look Ball, que seguirá a la Move Ball, pero estará más lejos de la nave que ésta, y será la que la nave mire para rotar en los ejes X e Y a la hora de girar.

Para que la nave avance en la posición Z de la pantalla, utilizamos la utilidad de *Unity* llamada *Cinemachine*. Cinemachine es una suite de herramientas para cámaras dinámicas. Con ella podemos diseñar un rail (figura 17) y añadir la cámara que queremos mover (junto con la nave Player) en un vagón para que avance por este.

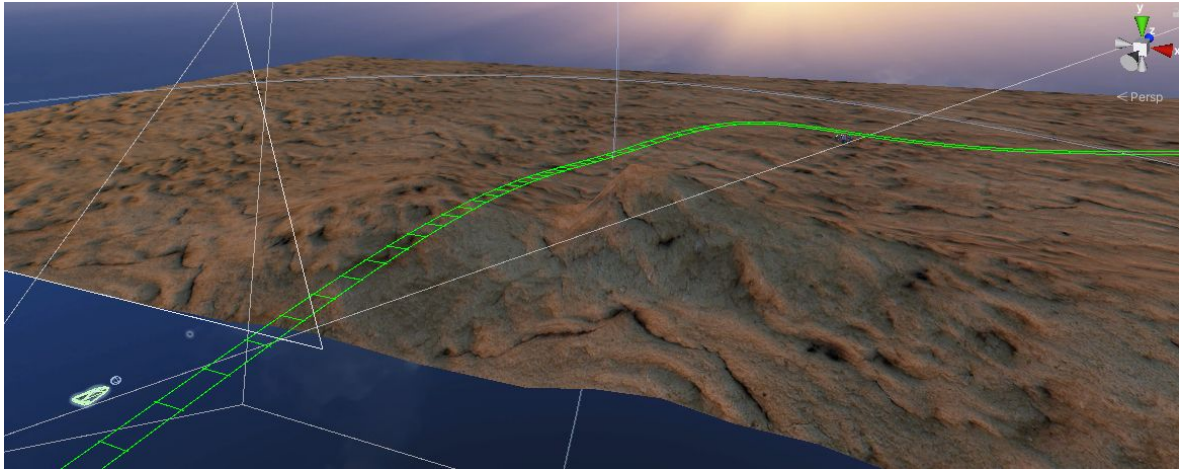


Figura 17. Rail de movimiento diseñado con Cinemachine.

Además del movimiento básico, tenemos una habilidad (roll) que nos permite girar sobre el eje Z y aumentar la velocidad de movimiento en el eje X e Y. Más información en su respectivo apartado de instrucciones del juego (3.1).

Por último, respecto a la nave del Player, hablaremos del sistema de disparos. El jugador podrá apuntar con la posición del ratón, y disparar haciendo click izquierdo. Este disparo colisionará con los enemigos causandoles una cierta cantidad de daño.

Más adelante hablaremos en detalle sobre los disparos, tanto del jugador como de los enemigos.

Torreta

Las torretas son uno de los enemigos fijos que aparecen en nuestro juego. Aparecen generalmente en posiciones más elevadas (encima de torres) y disparan a nuestro jugador.

El modelo de estas torretas (figura 18) es algo más parecido a la tecnología actual que el del resto de enemigos.

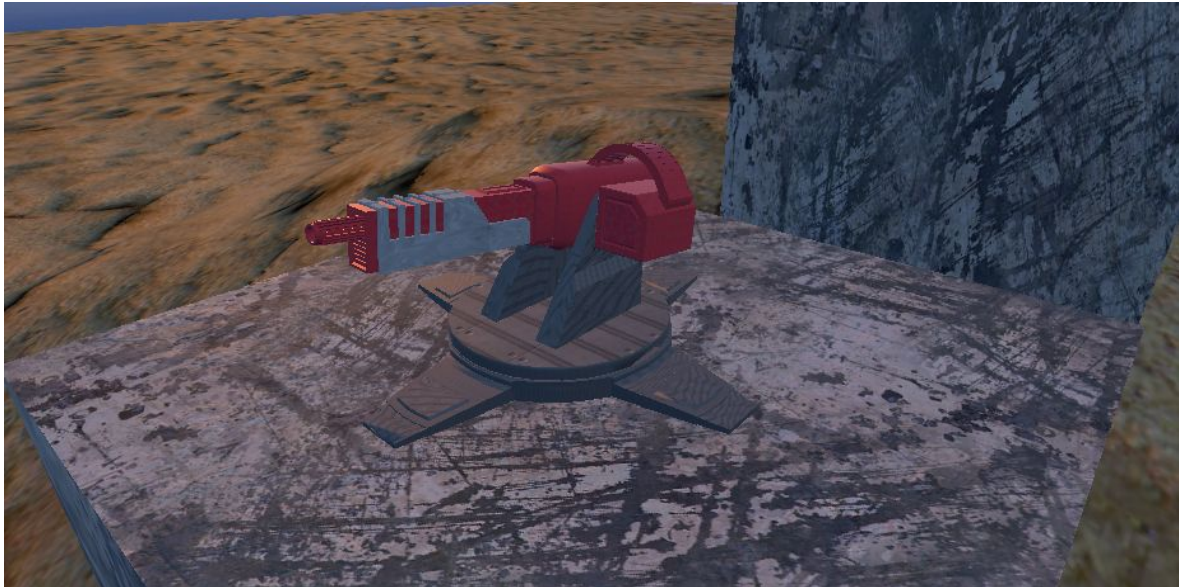


Figura 18. Modelo de enemigo Torreta

Como se puede ver, los colores predominantes son el rojo y el gris (color característico de los enemigos).

Tanque

Es un enemigo móvil, y en este caso es terrestre, por lo que aparecen siempre al nivel del suelo. Un modelo de tanque bastante similar a lo que usualmente vemos (figura 19). Obviamente sigue la gama de colores del resto de los enemigos.



Figura 19. Modelo de enemigo Tanque.

Droide

El droide es un enemigo espacial que se apoya en superficies (como asteroides). En este caso, dada su difícil capacidad de movimiento a la hora de luchar, actúa como un enemigo fijo. Su modelo es más futurista que los mencionados anteriormente (figura 20).

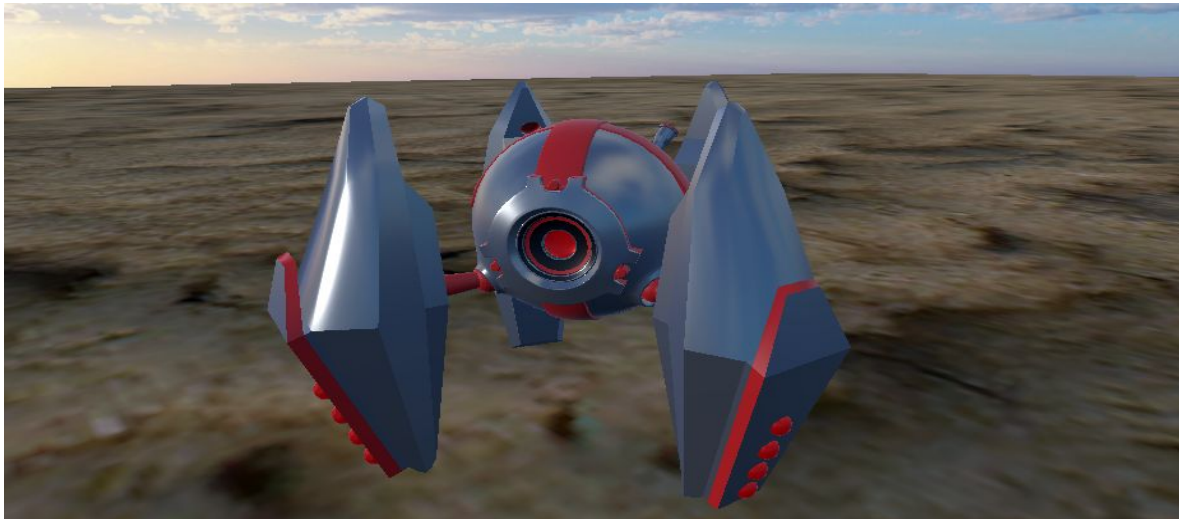


Figura 20. Modelo de enemigo Droide.

Space Fighter

Es la nave aérea del enemigo. Su modelo, pese a ser lowpoly, es futurista, y sigue los colores del bando enemigo (rojo y gris) (figura 21).



Figura 21. Modelo de Space Fighter

3.4.2 Obstáculos

En el juego, a parte de enemigos, encontramos otros obstáculos que también nos pueden causar daño. El que más puede llamar la atención a primera vista sería la Esfera Láser (figura 22), la cual dispara un láser que al colisionar con la nave nos causará daño. Dado su carácter estático (tanto en el movimiento, como en el disparo del láser), y su carencia por tanto de una IA, hemos decidido considerarla un obstáculo en lugar de un enemigo como tal.

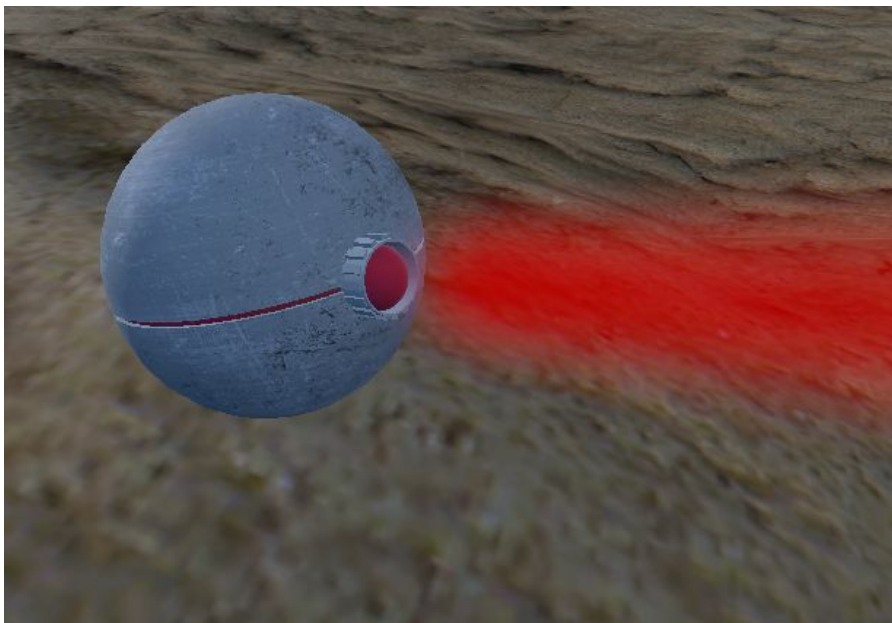


Figura 22. Modelo de Esfera Láser

A parte de este tipo de obstáculo, en el juego podemos encontrar otros que a su vez harán de elementos decorativos, como los asteroides (figura 23) del nivel 1 o como pueden ser edificios (figura 24) o árboles (figura 25) del nivel 2.

Cabe destacar también que el propio terreno donde se disponen todos los elementos del nivel 2 también es colisionable.

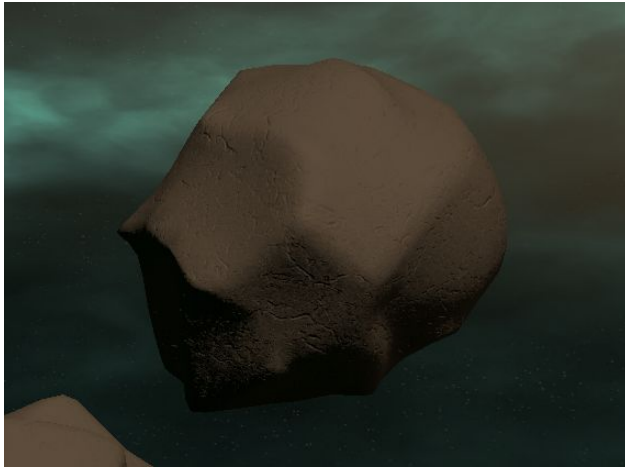


Figura 23. Asteroide



Figura 24. Ejemplo de edificio usado



Figura 25. Disposición aleatoria de árboles colisionables

3.4.3 Disparos y explosiones

En este videojuego abundan los objetos del tipo disparo y del tipo explosión o efecto especial. Todos los personajes anteriormente explicados disponen de sus propios disparos, tanto el jugador que se encarga de pilotar la nave, como las entidades controladas por la IA.

Estos disparos, a su vez, disponen de colisionadores que les permiten detectar en qué momento han entrado en contacto con otro objeto del mapa para interactuar con éste consecuentemente.

Es aquí donde entran en juego las explosiones. Efectos especiales que simulan el contacto de un proyectil con otro objeto o la destrucción total del mismo. Por ejemplo, cuando nosotros disparamos a los enemigos y los disparos colisionan con ellos, a parte de quitarles un cierto porcentaje de vida, también se destruyen los disparos generando una explosión para simular que ha chocado.

A continuación mostraremos los dos tipos de proyectiles que existen, la imagen de la izquierda (figura 26) representa el del jugador aliado y la de la derecha (figura 27) el disparo de los enemigos.

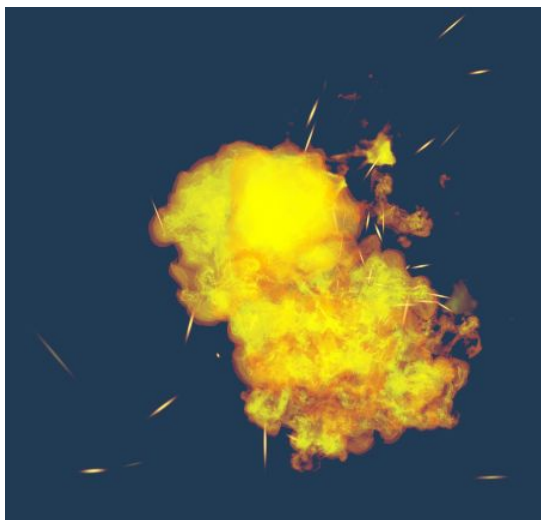


Figura 26. Bola de fuego amarilla que representa el disparo aliado.

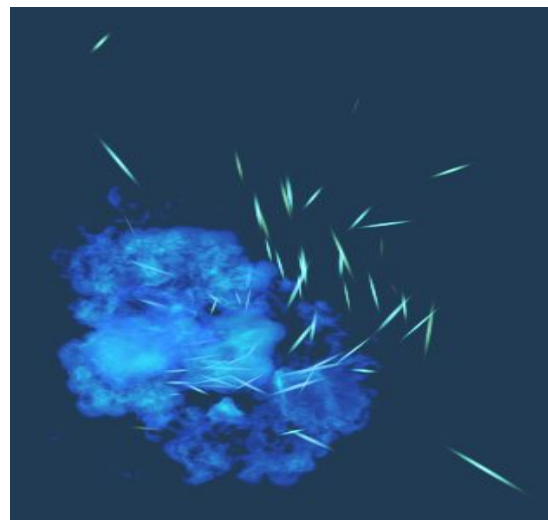


Figura 27. Bola de fuego azul que representa el disparo enemigo.

Como podemos observar en las fotos ambos disparos difieren visualmente en el color, para clarificar rápidamente *ingame* cuales son aliados y cuáles no. Internamente, cada uno de estos disparos se comporta ligeramente diferente para facilitar la puntería del jugador o hacer más asequibles los niveles.

Cada uno de los disparos mostrados con anterioridad crea su propia explosión (figura 28 y figura 29) al impactar con otros objetos del mapa o con el terreno. Ahora los cambios entre uno y el otro son simplemente visuales.

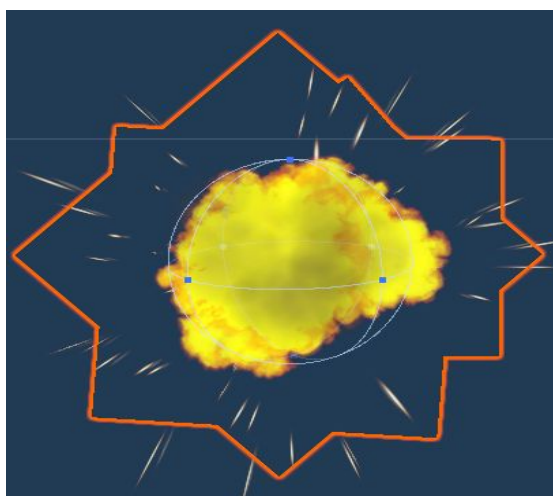


Figura 28. Explosión del disparo aliado

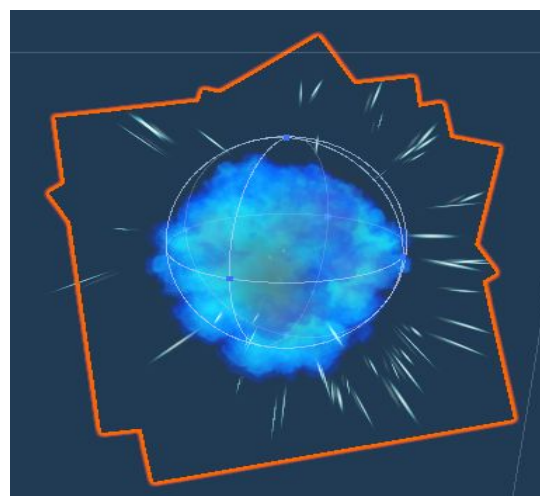


Figura 29. Explosión enemiga

Adicionalmente, hemos considerado oportuno añadir un efecto de llamas para incendiar, durante un breve periodo de tiempo, la hierba del segundo mapa en caso que un disparo impacte contra ella. Simplemente es estético pero creemos que le da más realismo y riqueza a la interacción con el mapa.

Estos dos efectos (figura 30 y 31) mantienen los mismos colores que los anteriores para diferenciarlos.



Figura 30. Llamas en hierba provocadas por disparo amigo

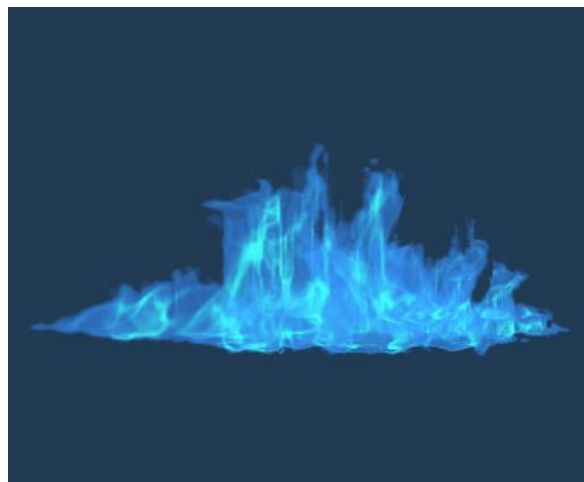


Figura 31. Llamas en hierba provocadas por disparo enemigo

Antes de comenzar a hablar acerca de las explosiones, consideramos oportuno mencionar que el *firerate* de los enemigos, así como la cantidad de vida que nos

quitan y les quitamos puede variar en función del nivel para ajustar correctamente la dificultad del mismo.

Por otro lado tenemos las explosiones, las cuales representan que la vida de el jugador o de una entidad hostil ha llegado a cero. Esto provoca la muerte instantánea del objeto representado por una gran explosión y su posterior desaparición y eliminación del mapa.

Seguidamente, en las figuras 32 y 33, se puede ver gráficamente cómo son estas explosiones y observar que el código de colores seguido permanece intacto.

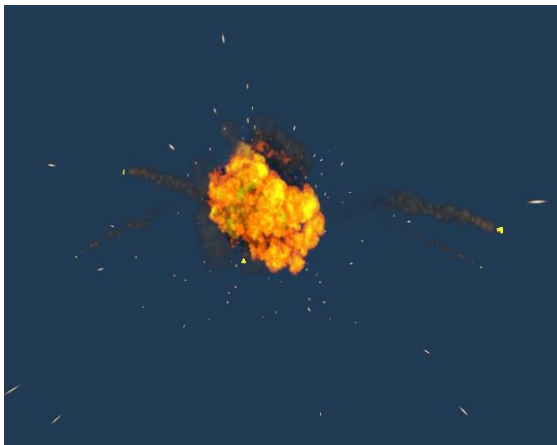


Figura 32. Explosión de la nave controlada por el jugador cuando su vida llega a 0

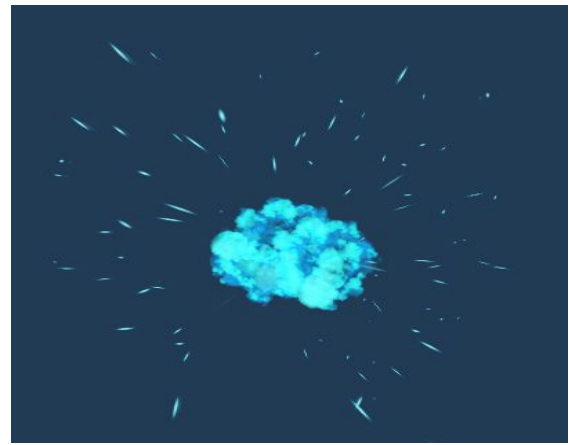


Figura 33. Explosión de la nave enemiga cuando su vida llega a 0

Aunque visualmente sean muy parecidas a las explosiones de los disparos, su efecto es significativamente más grande y notorio.

Finalmente, cabe destacar que cada uno de estos disparos y efectos especiales expuestos en esta sección está dotado de su respectivo efecto de sonido que lo identifica.

4. Metodología

Uno de los puntos claves a la hora de realizar un trabajo es la organización de las tareas, tanto en el tiempo, como en el equipo. Para ello una de las primeras cosas que debe hacerse es plantearse unas metas en ciertas fechas e intentar ajustarse a éstas. Una forma muy común de realizarlo es mediante un Diagrama de Gantt. Nosotros realizamos uno (figura 34) que hemos intentado seguir al máximo, pese a que, como era de esperar, hubo ligeros cambios en él.

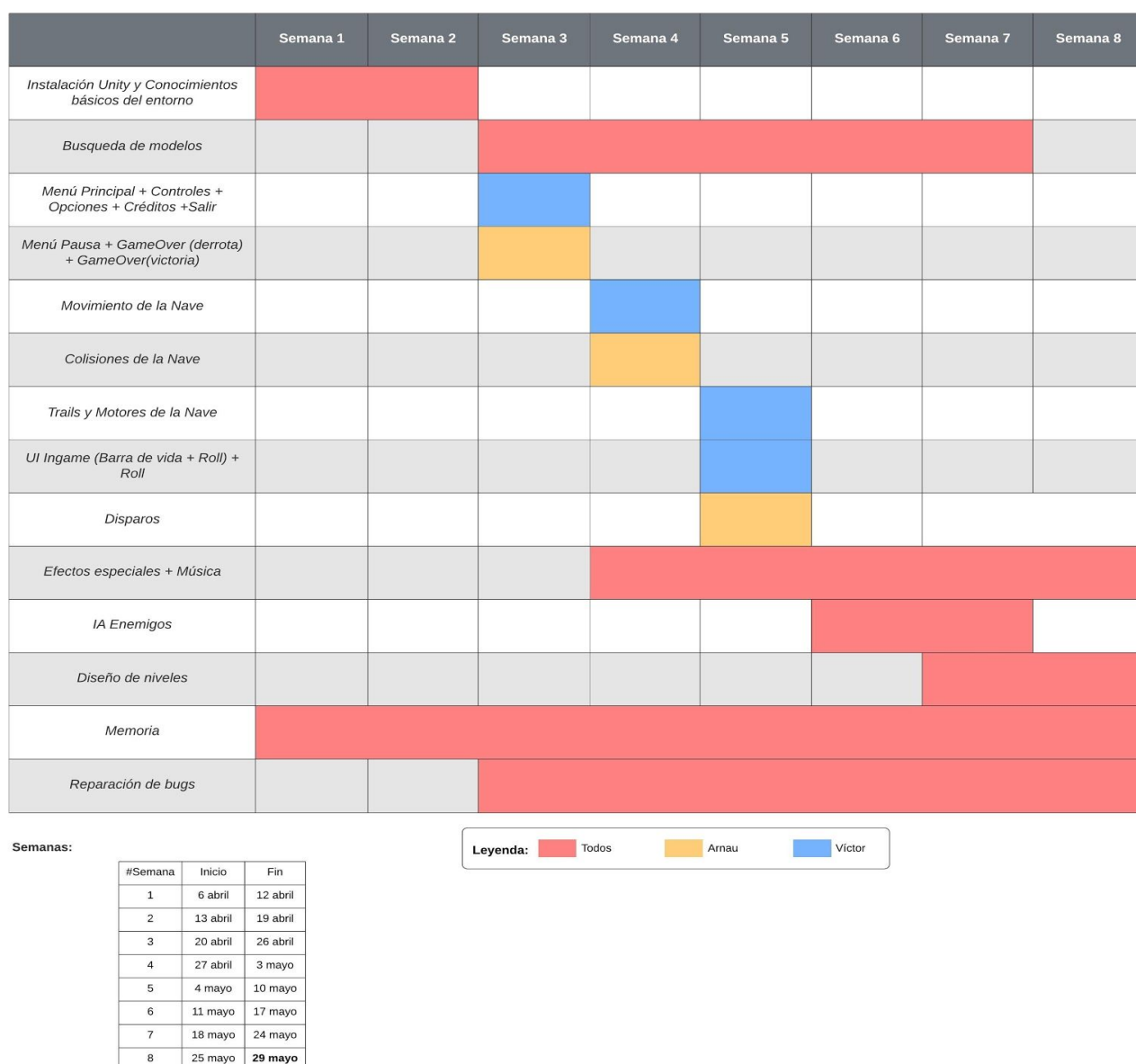


Figura 34. Diagrama de Gantt del proyecto.

Como se puede ver, decidimos dedicar las dos primeras semanas a informarnos sobre el juego original, así como a preparar el entorno *Unity* donde trabajaríamos y familiarizarnos con él.

Una vez llegada la tercera semana comenzamos el desarrollo del juego propiamente dicho. Empezamos a diseñar los distintos menús y buscar los modelos 3D que usamos en el juego, tarea que se extendería durante prácticamente todo el desarrollo. Seguiríamos con tareas divididas, centrándonos ahora en la nave del jugador (su movimiento, roll, disparos, colisiones, etc).

Aunque las tareas estuvieran divididas, nuestro contacto casi diario a través de Internet para consultarnos decisiones o ayudarnos en dudas o bugs que fueran surgiendo, ha hecho que ambos hayamos trabajado en mayor o menor medida en todos los aspectos del videojuego.

La parte final del desarrollo consistió en la programación de la IA enemiga y el diseño de niveles, el cual estaba altamente ligado a lo anterior, pues tendríamos que ir variando las variables de la IA para ajustar la dificultad (para que no fuese ni muy sencillo ni casi imposible pasarse un nivel).

Dado la situación actual generada por el COVID-19, nuestra comunicación ha sido exclusivamente telemática. Principalmente hemos utilizado los servicios de Google de **Gmail** (para la comunicación diaria) y videollamadas en **Discord** para reuniones semanales donde comentar cuál era el estado actual del proyecto y cuáles eran los siguientes pasos que debíamos dar.

Para mantener una versión consistente de nuestro código, y poder tener un registro de las versiones junto con sus cambios, hemos utilizado **Github** (figura 35). De forma análoga hemos utilizado **Google Drive** para el desarrollo de la memoria de forma colaborativa.

















Commits on May 25, 2020		
Add script for the enemy fireball arnauruana committed 1 hour ago	 a53b8b4	
Fixed Physics FireballEnemy victorbarroso1 committed 13 hours ago	 a63eed1	
Some fixes Enemy Fireball & Explosion victorbarroso1 committed 13 hours ago	 c28fa1f	
Commits on May 24, 2020		
Color fireball changed. ... victorbarroso1 committed 14 hours ago	 1bac227	
Lots of changes in turrets AI and interaction arnauruana committed 16 hours ago	 e7bfc1b	
Change enemy fiball color arnauruana committed 18 hours ago	 a2e82aa	
Add ship destruction by turrets arnauruana committed 19 hours ago	 1f76fda	
Menu Updated victorbarroso1 committed 20 hours ago	 1f88497	

Figura 35. Registro de Commits de Github

5. Conclusiones

Como valoración personal podemos remarcar lo mucho que nos ha aportado la realización de este proyecto en competencias tan importantes, a día de hoy, como el trabajo en equipo o la organización y planificación a corto y largo plazo de proyectos de programación avanzados como éste. No obstante, la realización de este videojuego se ha llevado a cabo durante unos días un tanto difíciles dadas las circunstancias de esta emergencia sanitaria provocada por la pandemia de la COVID-19. Por ende, y como ya se ha comentado a lo largo de la memoria, ha sido fundamental mantener canales de comunicación abiertos para intercambiar ideas entre ambos, así como posibles dudas o errores que pudiesen surgir a lo largo de todo el desarrollo de la práctica.

También nos gustaría comentar que, si bien nos llevó su tiempo el aprendizaje del motor gráfico utilizado (*Unity*), una vez aprendido nos ha ayudado a entender la gran importancia que adquieren los motores gráficos en los videojuegos, debido a la gran facilidad que nos brindan a la hora de crear modelos y manipularlos mediante *scripts* absteniéndonos de todas las llamadas a sistema y a los respectivos controladores gráficos mediante OpenGL o DirectX.

Otro importante punto a destacar es la visión que nos ha otorgado, como jugadores habituales, este proyecto sobre la industria de los videojuegos. En concreto nos ha ayudado a ver cómo están hechos y planteados los videojuegos 3D y cómo se comunican las diferentes partes y objetos de ellos para crear una experiencia de juego en la cual todos los objetos interactúan al unísono. También, nos ha ayudado a darnos cuenta de lo complicado que llega a ser diseñar y desarrollar un gran videojuego debido a la sustancial cantidad de horas que necesitan para ser debugados y probados para poder lanzarse con garantías al mercado.

Finalmente, como valoración global de la asignatura, nos gustaría añadir que ha sido muy gratificante ver cómo, de la nada, creábamos dos videojuegos mientras poníamos en práctica gran parte de las cosas aprendidas a lo largo de toda la carrera de ingeniería informática. Apreciamos, también, haber podido desarrollar en equipo dos videojuegos, el primero en 2D y éste en 3D, que nos aportan una sólida base por si el día de mañana la necesitamos para trabajar en esta industria.

6. Bibliografía

“Star Fox 64”

https://en.wikipedia.org/wiki/Star_Fox_64

Información general acerca del videojuego Star Fox 64.

“Tutoriales de Unity de YouTube”

<https://www.youtube.com/user/Brackeys/videos>

<https://www.youtube.com/watch?v=JAb1NwVeLBw>

Canales de Youtube con tutoriales tanto de funcionalidades como de animaciones y diseño.

"Unity en general"

<https://docs.unity3d.com/Manual/index.html>

Documentación oficial de Unity.

“Diagramas y gráficas”

<https://app.lucidchart.com/documents#/dashboard>

Página para diseñar todo tipo de diagramas online gratis.

“Dudas diversas en Unity”

<https://answers.unity.com/>

<https://forum.unity.com/>

Foros para resolver dudas.

“Modelos en 3D”

<https://www.turbosquid.com/>

<https://free3d.com/>

Descarga de modelos 3D gratuita.

“Música y sonidos”

<https://www.freesfx.co.uk/>

Descarga de música y sonido gratuita.

“Sonidos y efectos especiales”

<https://freesound.org/>

Descarga de sonidos y efectos especiales de forma gratuita.