

Star Clone 46

VJ Q2 2019-2020

Joseba Sierra Montes
Àlex Aguilera Martínez

Índice

1. Juego de referencia: Star Fox 64	3
1.1 Descripción	3
1.2 Desarrollo	3
1.3 Datos económicos y recepción del juego	3
2. Descripción del proyecto	5
2.1 Objetivo del juego	5
2.2 Entidades	6
2.3 Instrucciones	11
2.4 Flow chart	13
2.5 Decisiones de implementación	13
2.6 Decisiones de arte	14
3. Metodología	15
3.1 Gantt Chart	15
3.2 Seguimiento de tareas	15
3.3 Reuniones semanales	17
3.4 Github source code	17
4. Conclusiones	17
5. Bibliografía	17

1. Juego de referencia: Star Fox 64

1.1 Descripción

Star Fox 64 es un **rail shooter 3D de acción** de combates aéreos, aunque también es descrito como un scrolling shooter[3] . Para hacerse una idea de en qué consiste el juego lo mejor es mirar un gameplay p.ej <https://www.youtube.com/watch?v=GhQp8le67X>.

El juego fue estrenado en **1997** para la consola **Nintendo 64** como un reboot del juego Star Fox de Super Nintendo. Fue publicado por Nintendo y desarrollado por su estudio interno Nintendo EAD[2] .

El formato del juego era 1 cartucho de 96 MB[1] . En la imagen 1, se puede ver la portada del videojuego en el cual aparece la consola en el que corría y que incluía el Rumble Pak, un dispositivo que se conectaba con el mando del Nintendo 64 para que este pudiera vibrar.

A parte de Nintendo 64, el juego también fue estrenado para iQue Player en 2003 y más tarde para Wii, Nintendo 3ds y Wii U mediante la consola virtual. Nintendo 3ds también recibió una versión especial que sacaba provecho de su pantalla táctil y giroscopio para los controles[1] .

1.2 Desarrollo

Inicialmente Star Fox 64 fue desarrollado casi por completo para la consola Super Nintendo, con el nombre de Star Fox 2, pero Miyamoto lo canceló y decidió trasladar el juego a Nintendo 64 que era más potente, y así combinar elementos de los juegos anteriores.

En el desarrollo del juego se vieron involucradas **más de 60 personas**, como se puede ver en los créditos[4] de este. Como figuras relevantes tenemos a Shigeru Miyamoto (Producer), Takao Shimizu (Director), Takaya Imamura (Art director) y Kazuaki Morita (Chief Programmer). A parte, y no por ello menos importante se vieron involucrados en el desarrollo todo tipo de perfiles como compositores, diseñadores, programadores, escritores, voice actors, supervisors, progress managers, etc.

1.3 Datos económicos y recepción del juego

Star Fox 64 fue uno de los juegos más vendidos de 1997, llegando hasta el millón de copias vendidas en Estados Unidos a lo largo del año. Solo en sus primeros días ya se vendieron más de 300.000 copias, sin embargo, las copias que se vendieron en Japón fueron algo más reducidas, llegando a las 75.595 copias durante la primera semana de ventas. Las ventas

totales acumuladas hasta ahora globalmente llegan hasta los **4 millones** si tenemos en cuenta la versión 3DS[1][5] .

Fue un juego con muy buena recepción, consiguiendo notas en todos los medios superiores al 8 o 9. Actualmente, tiene una puntuación de **88 en metacritic**[6] tanto de reviews de usuarios como de críticos. El juego llegó al puesto 73 del Top 200 de los juegos de Nintendo jamás hechos. Está, además, catalogado como el cuadragésimo quinto mejor juego de todos los tiempos por Guinness World Records Gamer's Edition en 2009.

El videojuego es conocido como Lylat Wars en algunos países porque Nintendo consideró que "Star Fox" sonaba muy parecido al nombre de una compañía alemana llamada "Star Vox", y se preocuparon por una posible batalla legal debido al nombre del videojuego.



Imagen 1 - Portada de Star Fox 64

2. Descripción del proyecto

2.1 Objetivo del juego

El principal objetivo del juego es **completar los niveles con la mayor puntuación posible**.

1. Completar un nivel es posible si se llega al portal del final sin morir.
2. Se consiguen puntos destruyendo enemigos, sus construcciones o recogiendo suministros de energía.
3. Se pierden puntos cada segundo.

Estos 3 hechos crean un balance que presiona al jugador a ir rápido para no ser penalizado, pero a la vez esto le puede pasar factura por tener más riesgo de morir o saltarse potenciales puntos. A esto se le suma la gestión de la energía. Gastarás toda la energía para acelerar o usar el arma especial láser, o serás cauto y guardarás algo para poder esquivar?.

Resumiendo, estos 2 componentes (score y gestión de energía), aunque sencillos, añaden profundidad suficiente al objetivo del juego y a las diferentes maneras de conseguirlo, influyendo en la posible toma de decisiones del jugador.

Si se falla al cumplir el objetivo del juego (al morir), es ‘game over’ y te da la posibilidad de volver a intentarlo con la pantalla de la imagen 2. En caso contrario sale una pantalla equivalente ‘Level Completed’. Ambas muestran la puntuación obtenida.

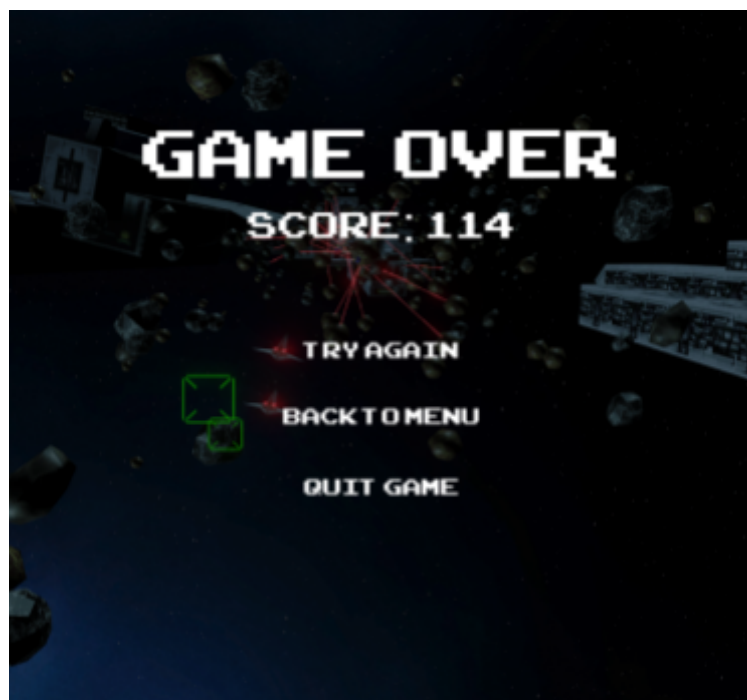


Imagen 2: Menú de derrota del jugador

2.2 Entidades

Player Ship

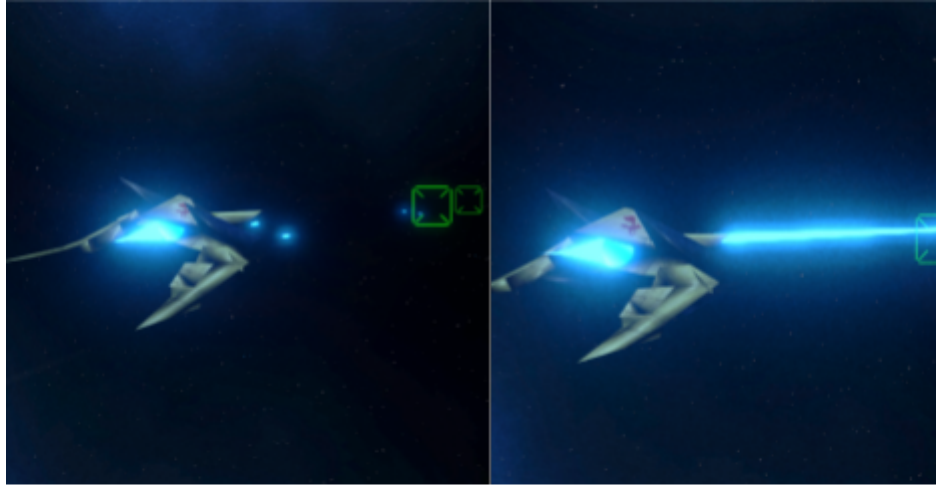


Imagen 3: Ataque básico y especial del player

Nave controlada por el jugador, permite ejecutar las siguientes acciones:

- **Ataque básico:** 2 rifles automáticos para infligir daño a los enemigos, no consumen energía.
- **Ataque especial:** Ataque láser (se puede observar a la derecha de la imagen 3), proporciona gran precisión y daño constante a costa de consumir una gran cantidad de energía por segundo.
- **Tilt:** Permite inclinar la nave en una dirección u otra, ofreciendo mayor control para esquivar obstáculos, balas, etc.
- **Roll:** Maniobra de giro que desplaza rápidamente la nave en una dirección y le proporciona inmunidad a las balas. Consume energía.
- **Boost:** Consume energía para aumentar la velocidad de la nave.
- **Frenar:** Disminuye la velocidad de la nave y permite apuntar mejor.

Los controles del jugador se explican en detalle en el apartado de instrucciones.

Como se puede ver la nave **consume energía** en ciertas acciones, esta se va regenerando con el tiempo, si se usa más de la cuenta hasta que llega a 0, la nave entra en un estado de ‘overheat’, en el que se penaliza sin poder usar energía hasta que esta se regenera hasta la mitad de su máximo. Esto se avisa al jugador mediante un sonido y mostrando la barra de energía roja.

Butterfly (Enemigo volador)

En cuanto una butterfly detecta al jugador, esta lo seguirá y se mantendrá delante de él mientras lo ataca. Por si sola una butterfly no es peligrosa, la principal característica de este enemigo es la capacidad de organizarse para mantener una **formación** con otras butterflies y **atacar sincronizadamente**.

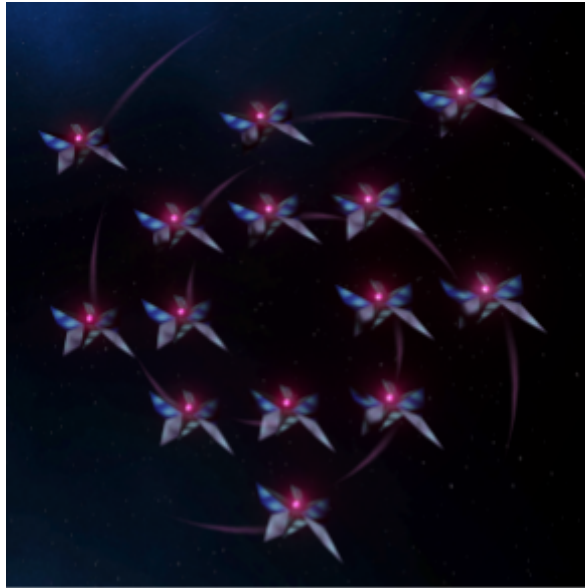


Imagen 4: Conjunto de butterflies en formación

Bolse (Enemigo volador)

Los bolses se mueven rápidamente en cierta dirección mientras disparan al jugador con una **alta cadencia** de fuego. Son bastante **escurridizos**, pero pueden marcar la diferencia en puntuación si el jugador es capaz de matarlos antes de que se vayan.



Imagen 5: Enemigo Bolse

Asteroids (Obstáculo)

Hay diferentes tipos de asteroides, estos son destruidos e infligen daño al jugador si este choca con ellos. Para facilitar la creación de los niveles, hemos creado un script '**Asteroid Spawner**', que instancia de **manera aleatoria** en una zona definida cierta cantidad de asteroides, esto en modo editor, para así permitirnos hacer cambios manuales antes del juego.

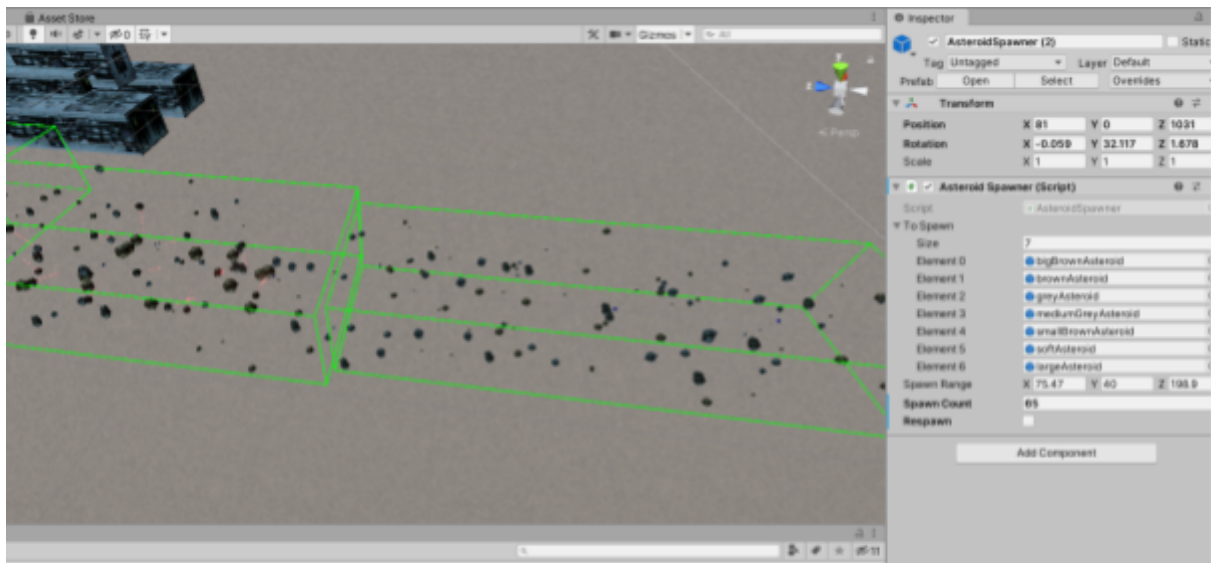


Imagen 6: Visualización del área de instanciación de los asteroid spawners

En la imagen 6, podemos ver un ejemplo de estos, el rectángulo verde indica el área en que el spawner puede instanciar los objetos. Algo tan simple como esto nos ahorra mucho tiempo y permite crear niveles como los de la imagen 7, con una gran variedad y número de asteroides añadiendo inmersión con solo 1 min configurando los parámetros del spawner.



Imagen 7: Vista del campo de asteroides ingame

Mining Asteroid (Obstáculo)

Asteroide algo diferente que tiene un láser que apunta a un asteroide normal. La nave controlada por el jugador debe **evitar el láser**, ya que de lo contrario, recibirá una gran cantidad de daño por segundo. Suelen **estar en medio de muchos asteroides** y se pueden ver con facilidad a lo lejos, pero no hay que confiarse, ya que pueden complicar la batalla contra los enemigos voladores. Si se destruye el asteroide al que impacta el láser, puede complicar la situación ya que hará que el láser llegue más lejos.

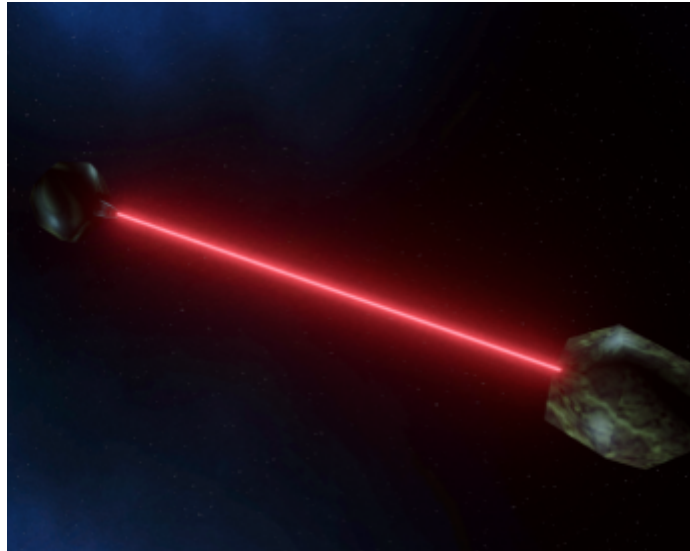


Imagen 8: Obstáculo Mining Asteroid

Fake Asteroid (Enemigo fijo)

Asteroide similar al anterior, con la principal diferencia que está en **constante movimiento** (rotación) y formado por **4 láseres**. Pueden estar colocados en cualquier sitio del nivel y al igual que el Mining Asteroid se ven con facilidad a lo lejos. La gracia de este asteroide es que el jugador tiene que pasar a tiempo entre un láser y otro sin colisionar.



Imagen 9: Enemigo fijo Fake Asteroid

Tanto el 'Mining Asteroid' como el 'Fake Asteroid' pueden ser destruidos para obtener puntos, aunque puede llevar de 2 a 5 segundos en función del arma usada.

Flagship Turret (Enemigo fijo)

Torreta fija en flagship que **gira y dispara** hacia la posición de la nave controlada por el jugador. La cadencia de fuego de estas torretas no es muy alta, pero el daño de cada disparo es bastante alto. Pueden estar **colocadas en diferentes posiciones** del flagship, algunas difíciles de llegar al disparar.



Imagen 11: Enemigo Flagship Turret

Asteroid Turret (Enemigo fijo)

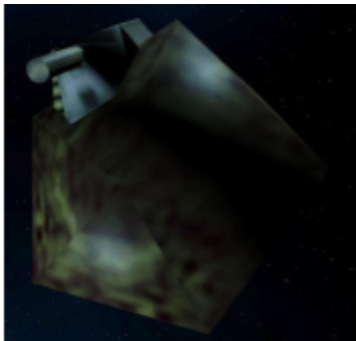


Imagen 12: Enemigo Asteroid Turret

Torreta fija en asteroide que gira y dispara hacia la posición del jugador. Es similar a la Flagship Turret, pero es más **pequeña**, con una **cadencia de fuego mayor** pero **daño reducido**. Se camufla fácilmente con el entorno. Con disparar al asteroide, ya sirve para destruir la torreta.

Supplies

Los supplies son objetos distribuidos a lo largo del nivel, normalmente en lugares estratégicos, que sirven para **recuperar la vida o la energía** de la nave. Como se puede ver en la imagen 13, hay de dos tipos; el de la izquierda para recuperar la energía y el otro para la vida. Tienen un efecto de estrellas brillantes y rotan para que sean más fáciles de identificar.



Imagen 13: Supplies de energía y vida.

WarpRing

El WarpRing es el anillo que se debe atravesar para poder **completar el nivel satisfactoriamente**. Al pasar a través de él, aparece la pantalla de Win Menu, comentada en el apartado 2.4, con la puntuación conseguida en el nivel.

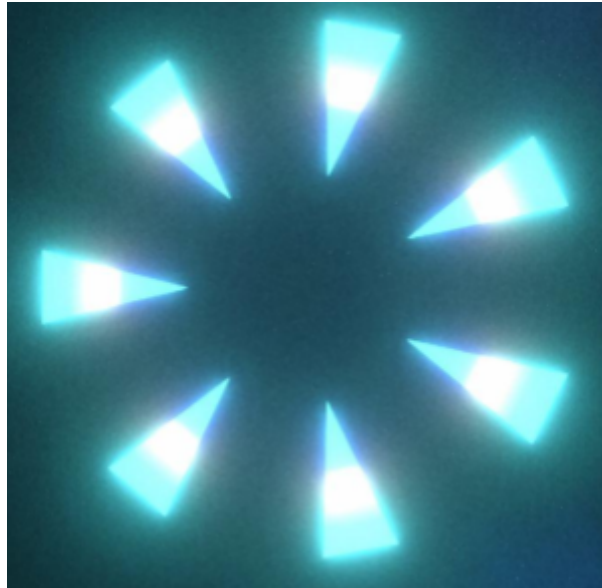


Imagen 14: WarpRing

2.3 Instrucciones

El juego, principalmente, se juega con teclado y ratón. Tiene varias funcionalidades similares al juego original, como ejemplo el boost y el dodge (este último, requisito para el proyecto), pero tiene algunas otras nuevas.

Los controles principales del juego son los siguientes:

- El movimiento dentro del juego, tanto para controlar la nave y el apuntado de la misma como para moverte por el menú, se debe hacer con el ratón. El ratón se corresponde al punto de mira, al cual dispara y se desplaza la nave.
- Para disparar el arma normal, click izquierdo y para hacer uso del láser incorporado, click derecho.
- Se permite inclinar la nave hacia el lado derecho o izquierdo usando las teclas A o D dependiendo el lado que se quiera inclinar.
- Para esquivar utilizando la maniobra de roll/dodge, se deben usar las teclas W,S,D,A dos veces dependiendo hacia donde se quiera hacer el roll.

- En cuanto al boost de la nave, se hace con la tecla SHIFT y para frenar y apuntar mejor, con la tecla SPACE.
- Finalmente, se tiene acceso a un menú de pausa mientras se está jugando usando la tecla ESC que permite volver al menú inicial, salir del juego, activar o desactivar el God Mode y, en caso de molestar, también la música de fondo.

Estas instrucciones comentadas aquí, también se pueden encontrar explicadas de una manera más resumida dentro del juego en el menú “controls”, ver imagen 15.

Como se puede ver, hemos hecho que el **menú inicial** sea **interactivo, ingame**, controlando la nave de fondo, eso nos permite enseñar los controles a la vez que el usuario los puede ir probando.

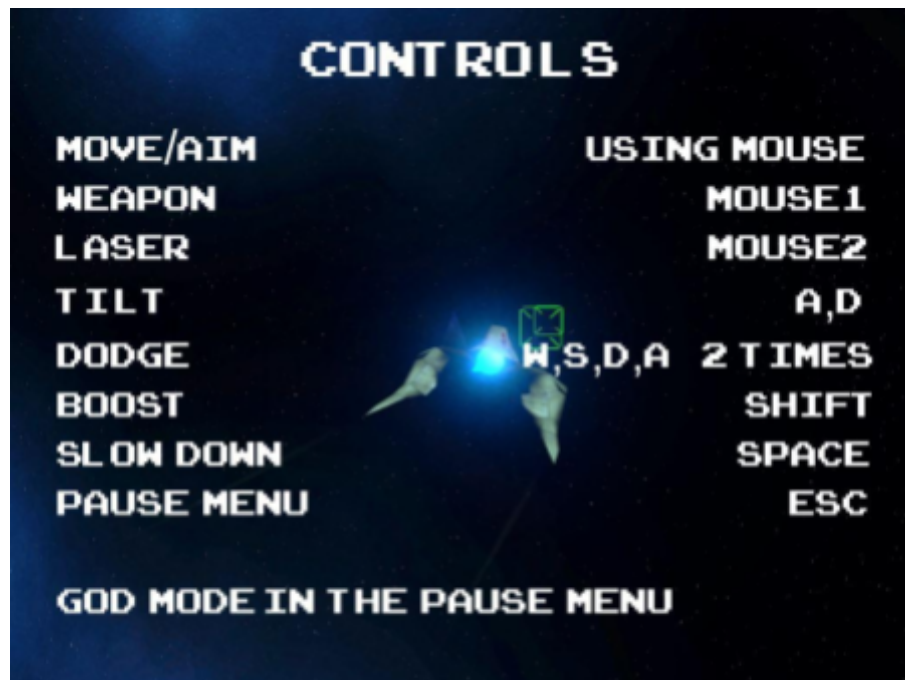


Imagen 15 - Menú “Controls” del juego

2.4 Flow chart

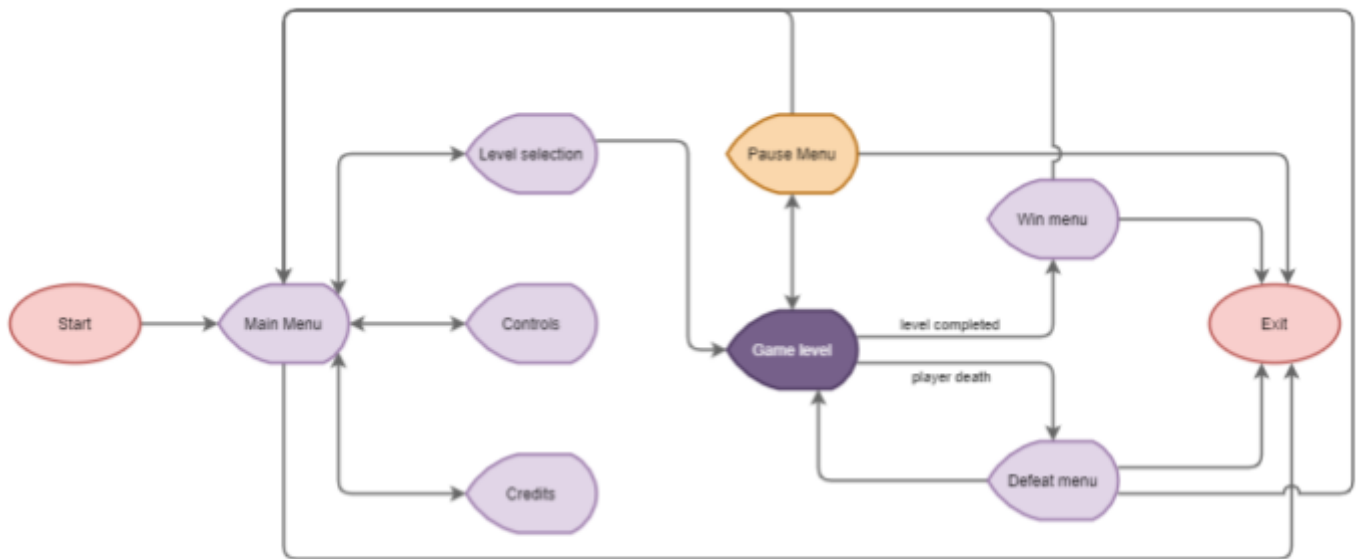


Imagen 16 - Flow chart de las diferentes ventanas del juego

El flujo de las ventanas de nuestro juego es el que se puede ver en la imagen 16 en el cual se empieza en un menú principal y te da la opción de jugar seleccionando uno de los dos niveles creados, o la opción de ver los controles o créditos. Tanto de level selection, controls y credits se puede volver al main menu con la tecla ESC.

Una vez seleccionado el nivel a jugar, el jugador, usando la tecla ESC, puede ir al menú de pausa. En este, se le da las opciones de volver al menú principal, de salir del juego o de seguir jugando, además puede activar/desactivar la música y godmode.

Finalmente, en caso de perder o ganar, le saldrá una pantalla diferente dependiendo del resultado al jugar que permite volver al menú principal o salir del juego. En caso de perder, el jugador puede intentar de nuevo el nivel, sin la necesidad de volver al menú principal.

2.5 Decisiones de implementación

Composition over inheritance:

Desde un principio hemos tratado de añadir comportamiento en base a componentes, haciendo que sea más fácil crear cosas nuevas haciendo combinaciones diferentes de componentes y ahorrando tiempo debido a la reutilización de estos. Por ejemplo, un caso sencillo: En vez de definir un script “Asteroid”, para hacer su movimiento de rotación, definimos un script “Rotate”, esto nos permite reutilizarlo para los supplies, el warp ring, fake asteroid , etc.

Todo parámetro expuesto en el editor:

Hemos intentado exponer todo parámetro que influya en el comportamiento del juego, para facilitar la etapa de balanceo, permitiendo experimentar con diferentes valores mientras se juega fácilmente, sin tener que entrar al código, recompilar, etc.

Controles de la nave:

Primero implementamos los controles de manera que el movimiento de la nave se hacía con W,A,S,D y el ratón sólo para el apuntado. Lo descartamos e hicimos algo más fiel al Star Fox original, donde la nave se mueve en la dirección en la que apunta, esto a la vez libera las teclas W,A,S,D que pueden ser usadas intuitivamente para el roll.

Interfaz IWeapon:

Usamos una interfaz para permitir que cualquier entidad con un componente de ataque pueda usar cualquier tipo de arma, en nuestro caso implementamos un rifle automático y un láser, con sus parámetros modificables para poder crear distintas variaciones de cada uno.

2.6 Decisiones de arte

Por lo que al aspecto visual se refiere, hemos decidido usar los modelos[7] de Star Fox 64. Los efectos consisten en el uso de partículas (mayoritariamente para las explosiones), trail renderers (para el rastro que dejan las naves) y postprocesado bloom, para reproducir el efecto de las balas, láseres, etc. Hemos decidido separar claramente el color del jugador, azul, con el de los enemigos y obstáculos del escenario.

En cuanto al tema de sonido (obtenido de un humble bundle[11] hace años), tanto menú y los 2 niveles a jugar tienen músicas distintas para marcar sus diferencias. Tanto disparos, colisiones, muerte/explosión, roll, boost, etc, tienen su efecto correspondiente para dar feedback al jugador.

3. Metodología

3.1 Gantt Chart

En el Gantt Chart de la imagen 17, podemos observar la planificación básica de nuestro proyecto. Desde un inicio, teníamos claro que queríamos dar más prioridad a las mecánicas básicas, a las piezas que compondrían el juego, antes que empezar con el diseño de niveles y efectos, ya que estos dependen enormemente de lo desarrollado en la 1a etapa.

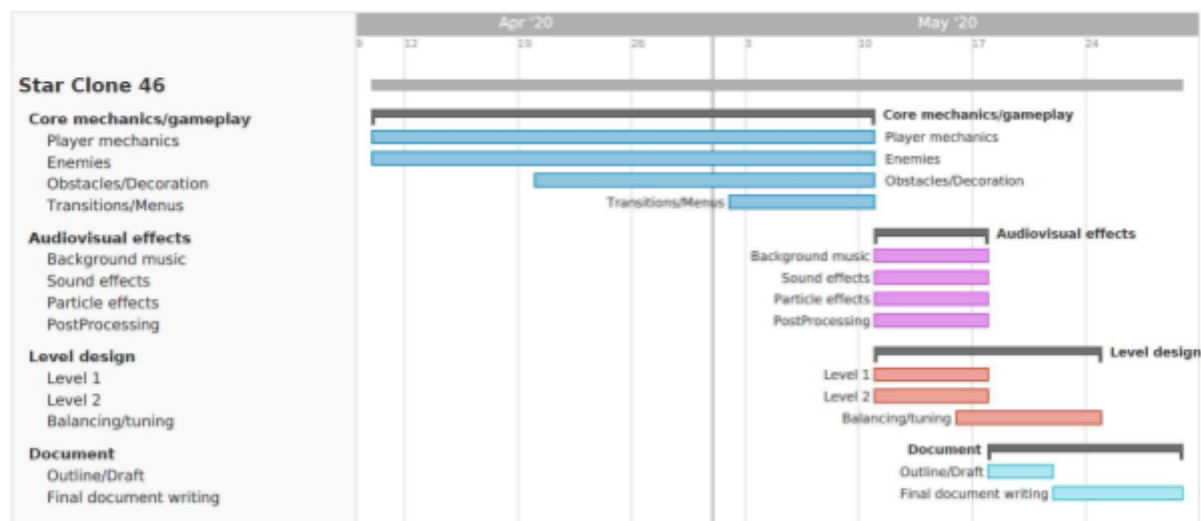


Imagen 17 - Gantt Chart del proyecto

También, quisimos reservar unos días para poder balancear y experimentar con diferentes parámetros del juego.

3.2 Seguimiento de tareas

Decidimos utilizar la herramienta <https://hacknplan.com/> comentada en la clase virtual sobre el juego 3D para hacer todo el seguimiento y organización de las tareas del juego.

En la imagen 18 y 19, se puede observar el estado de las tareas en unos días concretos, una corresponde a las tareas de la base del juego, y la otra a tareas de efectos audiovisuales. Era bastante dinámico, resultando en tareas descartadas u otras ideas añadidas cerca de la fecha límite planificada.

Core Gameplay 14 days left + Create + Info				
ALL ITEMS 15	Planned 7	In Progress 11	Testing 5	
USER STORY PROGRAMMING 21 ART DESIGN 2 WRITING MARKETING SOUND IDEAS BUG	#20 Dying behaviour in Programming	#4 Lateral movement (spaceship) in Programming 2 / 3	#9 Aiming mechanic (player spaceship) in Programming 10 / 12	
	#13 Menu GUI in Programming	#21 Special weapon in Programming	#8 Weapon class in Programming 10 / 12	
	#18 Scene level 1 in Design	#5 Roll (spaceship) in Programming 1 / 3	#6 Camera follow behaviour in Programming 10 / 12	
	#19 Scene level 2 in Design	#22 Laser cannon asteroid in Programming	#14 Defense stats, take damage in Programming	
	#24 God Mode in Programming	#15 Brain enemy in Programming	#7 Boost (spaceship) in Programming	
	#25 Game controls in Programming	#30 Boost ring in Programming 10 / 11		
	#27 Obstacle2 in Programming	#26 Asteroids in Programming		
		#21 Tower in Programming		
		#11 Boba ship in Programming		
		#29 Supply container in Programming		
		#23 player HUD in Programming 2 / 3		

Imagen 18: Estado de las tareas 1/5/20

Personal Hall Shooter game Sound and effects + Create + Info				
ALL ITEMS 1	Planned 1	In Progress 1	Testing 3	Completed 9
USER STORY PROGRAMMING 1 ART DESIGN 2 WRITING MARKETING SOUND IDEAS BUG	#52 Bullet explosion in Design	#44 PostProcessing in Programming	#55 Laser sword effect in Design	#43 Add enemy music in Programming
			#44 Add level music in Programming	#46 soundtrack in Design
			#45 Add level music in Programming	#48 Supply effect in Design 10 / 11
				#53 Death explosion in Design
				#51 Ship engine effect in Design
				#49 Ship wings rail in Programming
				#42 Boba and butterfly effects in Programming
				#46 title sound effect in Design
				#44 Boost sound in Programming

Imagen 19: Estado de las tareas de efecto audiovisuales, 16/5/20

3.3 Reuniones semanales

Cada cierto tiempo, normalmente una o dos veces por semana, hacíamos reuniones en llamada y compartiendo pantalla, haciendo uso de la herramienta Discord, el objetivo principal de estas reuniones era estudiar el avance del proyecto, planificar las tareas para la siguiente semana y resolver dudas. Por ejemplo, el 29/04/20, la reunión consistió en planificar la estructura del código para los siguientes elementos a implementar, mientras que la del 9/5/20, estuvimos decidiendo que efectos, sonidos y últimos modelos escoger para el apartado audiovisual del juego. El 16/5/20 la reunión consistió en jugar cada uno el nivel diseñado por el otro, para recomendar cambios, balanceos, etc.

3.4 Github source code

Se puede encontrar el código original del juego en Github, con una breve explicación y fotos del mismo, en el siguiente link: <https://github.com/josebasierra/starclone46>

4. Conclusiones

Estamos bastante satisfechos con el resultado obtenido, teniendo en cuenta el tiempo del que disponíamos para ello. Pero lo más importante ha sido el aprendizaje y experiencia ganada sobre el motor Unity y el proceso de desarrollar un pequeño juego completo en general. Por otra parte, aunque no hayamos creado nuestros propios modelos 3D, creemos que nos ha quedado un buen juego visualmente hablando.

Puede que lo más difícil del proyecto haya sido decidir cuando parar, que añadir y que quitar para mejorar el juego, saber aquello que conlleva menos tiempo de desarrollo y que puede crear un mayor impacto en la jugabilidad, etc. Siempre se pueden añadir nuevas características al juego o refinarlo más para mejorarlo y que sea más divertido.

5. Bibliografía

- [1] Es.wikipedia.org. 2020. *Star Fox 64*. [online] Disponible en: <https://es.wikipedia.org/wiki/Star_Fox_64> [Accedido el 20 de mayo 2020].
- [2] Star Fox Wiki. 2020. *Star Fox 64*. [online] Disponible en: <https://starfox.fandom.com/es/wiki/Star_Fox_64> [Accedido el 20 de mayo 2020].
- [3] En.wikipedia.org. 2020. *Scrolling Shooters*. [online] Disponible en: <https://en.wikipedia.org/wiki/Category:Scrolling_shooters> [Accedido el 20 de mayo 2020].

[4] Nintendo. *Star Fox 64/Credits*. [online] Se pueden ver todos los componentes de su equipo de desarrollo. Disponible en: <https://nintendo.fandom.com/wiki/Star_Fox_64/credits> [Accedido el 22 de mayo 2020].

[5] Video Game Sales Wiki. 2020. *Star Fox*. [online] Listado de las ventas totales de la saga Star Fox. Disponible en: <https://vgsales.fandom.com/wiki/Star_Fox#cite_note-:0-0> [Accedido el 22 de mayo 2020].

[6] Metacritic. 2020. *Star Fox 64*. [online] Score y reviews tanto de críticos como usuarios de Star Fox 64. Disponible en: <<https://www.metacritic.com/game/nintendo-64/star-fox-64>> [Accedido el 22 de mayo 2020].

[7] Models-resource.com. 2020. *Nintendo 64 - Star Fox 64 - The Models Resource*. [online] Disponible en: <https://www.models-resource.com/nintendo_64/starfox64/> [Accedido el 28 de mayo 2020].

[8] Hyman, T., 2020. *Basic Explosion Particle Effects In Unity | Cloudy Water Games*. [online] Cloudywatergames.com. Disponible en: <<http://www.cloudywatergames.com/2018/07/25/basic-explosion-particle-effects-in-unity/>> [Accedido el 28 de mayo 2020].

[9] Youtube.com. Star Effect In Unity3d [online] Disponible en: <<https://www.youtube.com/watch?v=gO-asCdrAPQ>> [Accedido el 28 de mayo 2020].

[10] Youtube.com. Everything to know about the PARTICLE SYSTEM. [online] Disponible en: <<https://www.youtube.com/watch?v=FEA1wTMJAR0&t=330s>> [Accedido el 28 de mayo 2020].

[11] Humble Bundle. 2020. *Humble Bundle*. Pack de sonidos usados. [online] Disponible en: <<https://www.humblebundle.com/>> [Accedido el 28 de mayo 2020].