

GAME DESIGN DOCUMENT

Eduardo Montaña Gómez
Luis Antonio Bello González

1. Design History

1.1 Version 1.0

2. Vision statement

2.1 Logline del Juego

Shooter en primera persona en donde el jugador tiene el control de una nave espacial y busca eliminar al resto de jugadores en un campo de juego que permite libertad de movimiento en tres dimensiones.

2.2 Gameplay synopsis

En este título, el jugador se adentra en un mundo tridimensional y toma el control de un vehículo aeroespacial, dicho vehículo cuenta con un arma al igual que otros jugadores dentro del mismo entorno. El jugador puede desplazarse con total libertad por el mundo de juego. Hacia arriba, hacia abajo, hacia adelante, atrás y a los costados.

3. Audience, Platform, and Marketing

3.1 Publico Objetivo

Según la guía de clasificaciones ESRB para videojuegos, este título se clasifica como Everyone 10+, lo que significa que se recomienda para jugadores a partir de los 10 años, debido a su contenido (violencia de caricatura, fantasía o leve).

El público objetivo principal se encuentra en un rango de edad aproximado de 10 a 30 años, con una inclinación hacia el género masculino, aunque no se limita exclusivamente a este grupo. El objetivo es que cualquier persona, sin importar su edad o género, pueda disfrutar del juego con total libertad y comodidad.

3.2 Plataforma

Debido a que el juego fue desarrollado en Unity, y con el fin de facilitar las pruebas prácticas dentro de la misma plataforma de desarrollo, se decidió que el juego se ejecutará en PC, en sistemas operativos compatibles con Steam (Windows, macOS y Linux).

Los movimientos del jugador están diseñados para ejecutarse mediante teclado, mientras que la navegación en los menús de inicio, pausa y configuración puede realizarse tanto con teclado como con mouse.

3.3 Requerimientos del Sistema

Requisitos mínimos y recomendados para ejecutar el juego de manera óptima en PC. Estos valores pueden variar ligeramente dependiendo de la configuración del juego.

Requisitos Mínimos:

- Sistema Operativo: Windows 10/11.
- Procesador: Intel Core i3 (4ª generación) o equivalente
- Memoria RAM: 8 GB
- Almacenamiento: 2 GB de espacio disponible
- Periféricos: Mandos de Xbox 360 o Xbox one.

Requisitos Recomendados:

- Sistema Operativo: Windows 10/11.
- Procesador: Intel Core i5 o AMD Ryzen 5
- Memoria RAM: 8 GB
- Almacenamiento: 8 GB o más de espacio disponible
- Periféricos: Mandos de xbox 360 o xbox one.

3.4 Top performers

Entre los juegos más reconocidos con mecánicas similares a nuestro título, podemos destacar los siguientes:

- En cuanto al movimiento controlado por el jugador, se ha tomado inspiración de títulos como Rocket League y Minecraft, especialmente por su fluidez y respuesta en el control del personaje o vehículo.
- En cuanto al diseño del entorno, se han considerado referencias como LEGO Star Wars, particularmente en los niveles donde se controlan naves, por su estilo visual y dinámica de juego.

3.5 Comparación de Características

Lo que intentamos hacer con este juego y la razón por la que creemos que podría competir con otros juegos del género es la simplicidad del juego, mecánicas agregadas y el toque humorístico.

Con simplicidad del juego nos referimos a que no tiene demasiadas mecánicas agregadas como sobreuso de botones del teclado, objetos del juego, ni reglas innecesariamente complicadas. No intentamos hacer un juego que requiera un esfuerzo excepcional sino que sea un juego que ofrezca al jugador pasar un rato agradable.

Con mecánicas agregadas queremos hacer notar los tipos de movimiento parecidos a una nave espacial alienígena típica de cine y televisión con total libertad de movimientos en el plano tridimensional.

El toque humorístico del juego viene desde la parte visual absurda del mismo en el que en lugar de un shooter “normal” de naves sea un shooter de naves con armas de fuego que pueden agarrar como si fueran humanos.

3.6 Expectativa de Ventas

Dado que se trata de un proyecto escolar desarrollado con tiempo y recursos limitados, no consideramos su comercialización. Aún así consideramos que el juego tiene el potencial de competir con muchos de los títulos pequeños, independientes del género juegos casuales y para pasar el rato que se lanzan año con año.

4. Análisis Legal

Este proyecto está siendo desarrollado con fines estrictamente académicos y sin fines de lucro, como parte de un trabajo escolar. En su construcción se han utilizado modelos 3D creados por artistas independientes, los cuales fueron obtenidos de plataformas que permiten su descarga gratuita y uso bajo licencias abiertas.

El análisis legal del título toma en cuenta los siguientes puntos:

- Los modelos 3D empleados cuentan con una licencia que permite su uso no comercial.
- En caso de requerirse, se ha incluido el crédito correspondiente al autor o creador del modelo.
- No se ha modificado ningún contenido que tenga restricciones de modificación en su licencia.
- El proyecto no será distribuido comercialmente ni publicado con fines lucrativos.

En caso de que el proyecto evolucione hacia una versión pública o comercial, se revisará nuevamente el origen y la licencia de todos los recursos utilizados para garantizar el cumplimiento legal correspondiente.

5. Gameplay

El juego es multijugador en pantalla dividida. Cada jugador controla una nave tipo UFO que tiene la capacidad de disparar.

Los jugadores deberán eliminarse unos a otros hasta un máximo de tres veces. Una vez que se alcance esta cantidad de muertes, se muestran los resultados y se muestra el menú principal, por si se quiere jugar otra vez o se quiere terminar el juego.

5.1 Controls

joystick axis X - Movimiento hacia la izquierda

joystick axis Y - Movimiento hacia atrás

joystick axis X- Movimiento hacia la derecha
joystick axis Y - Movimiento hacia adelante
joystick axis 4 - Movimiento horizontal de la pantalla
joystick axis 5 - Movimiento vertical de la pantalla
Right trigger - Disparo
Right Bumper - Movimiento hacia arriba
Left Bumper - Movimiento hacia abajo
Button A - Aceptar
Button B - Volver

5.1.1 Interfaces

Contiene dos menús con los que el jugador puede interactuar. El primero es el que aparece al inicio del juego (menú principal) y el menú de pausa que permite al jugador detener el tiempo y tiene la posibilidad de reanudar.

5.1.2 Rules

Proyectiles: el jugador puede disparar proyectiles a su adversario mediante su arma. Si estos proyectiles impactan el sphere collider del otro jugador, realizarán daño y esto puede hacer que el otro jugador pierda una vida o que pierda el juego si es que ya ha perdido dos vidas previamente.

Si el jugador muere, éste reaparecerá en uno de los spawnpoints designados previamente. Actualmente sólo hay 4 spawnpoints.

Cuando el jugador recibe daño, su barra de vida se ve decrementada y si llega a cero, el jugador perderá una vida.

5.1.3 Scoring/winning conditions

Para ganar solo es necesario acabar con las 3 vidas del rival, esto se logra bajando su vida a cero tres veces a través de disparos.

5.2 Modes and other features

Los modelos utilizados para ambos jugadores fueron las UFO Battleship que se encuentran en la unity asset store.

<https://assetstore.unity.com/packages/3d/vehicles/space/ufo-battleship-289193>

Estas naves espaciales son un modelo simple de los típicos UFOS ovnis que se pueden encontrar en películas y videos de internet.

5.3 Levels

No se cuenta con niveles, esto debido a que los jugadores serán los que se enfrenten unos a otros en un mapa establecido. Sin embargo, se podrían pensar en varios mapas a seleccionar en una siguiente versión del juego.

6. Personajes

Como tal no existen personajes con personalidad o vida propia. Cada jugador otorga su forma de jugar a su player (su modelo de ufo).

En el juego el jugador 1 tomará el control del UFO de color azul mientras que el jugador 2 tomará el control del UFO de color naranja.

7. Historia

No existe una historia o narrativa ya que se trata solamente de un juego 1v1 sin contexto.

8. El mundo del juego

En nuestro mundo jugable nos encontramos en el escenario del espacio exterior, en un sistema de planetas enanos en medio del universo. En este espacio podemos encontrar un número finito de objetos, cubos espaciales para darle orientación al jugador y 5 planetas enanos (Egypt, Ice, Forest, Red y Jungle). Y por último 2 UFO's que son los objetos controlados por los jugadores.

9. Media List

9.1 Interface assets

En el juego tenemos varias assets de interfaz:

- Menús:
 - Principal: Este contiene a su vez botones para jugar y de salida.
 - De pausa: contiene botones, para resumir, para regresar al menú principal y termina el juego.
- Barra de vida: El jugador tiene una barra de vida que le indica cuando daño le han hecho hasta el momento.
- Número de vidas: El jugador puede ver en pantalla el número de vidas que le restan.

9.2 Entornos

- La escena principal es un lugar del espacio, donde hay planetas y unos cubos verdes que de momento no se sabe si representan algo o no. Este sería el único nivel por el momento.
- Los planetas que dan ambientación tienen un estilo de polígonos "low poly". De esta forma el juego tiene un cariz más "cartoon".
- La escena cuenta con tres luces direccionales ubicadas de tal forma que son perpendiculares unas con otras.

9.3 Personajes

Sólo se cuenta con dos personajes, cada uno es controlado por un jugador.

Estos personajes están representados mediante un modelo 3D de un UFO. El jugador uno tiene un UFO azul y el segundo jugador tendrá un UFO naranja.

Estos personajes cuentan con la habilidad de disparar mediante un arma que tienen a un costado, así como que también cuentan con una barra de vida y un indicador de cuántas vidas le restan.

9.4 Animación

El juego de momento no cuenta con animaciones.

9.5 Music and sound effects

El juego contiene varios efectos de sonido:

- Daño: Al recibir daño el jugador producirá un ligero sonido de choque metálico.
- Disparo: El jugador puede disparar con LT en el mando y este producirá un efecto de sonido por cada disparo realizado.
- Muerte: Al morir el modelo del jugador se convertirá en una animación de explosión acompañada de sonido de explosión

10. Technical Spec

10.1 Technical analysis

10.1.1 New technology

No se usa tecnología nueva para el desarrollo del juego.

10.1.2 Major software development tasks

Estamos usando una licencia de Unity para uso personal que nos permite desarrollar juegos con ciertas restricciones.

10.1.3 Alternatives

Una buena alternativa a usar Unity y desarrollar el videojuego ahí sería Unreal Engine, sin embargo, éste game engine pide más requerimientos de hardware que un universitario podría no tener.

10.1.4 Estimated resources required

En cuanto a hardware, un procesador de 2.5Gz servirá, al igual que una memoria RAM de 8 GB. Y en cuanto a software, MacOS, Linux y Windows sirven para esta labor.

10.2 Development platform and tools

10.2.1 Software

El software utilizado para el desarrollo del videojuego fue Unity en su versión 2021.3.41f1.

10.2.2 Hardware

Dos computadoras portátiles con al menos 8 de RAM.

10.3 Delivery

10.3.1 Required hardware and software

El software requerido para poder correr el juego es Windows 10 u 11 de 64 bits.

El hardware requerido para poder ejecutar el juego es tener al menos 8 de RAM y 1 GB de almacenamiento.

10.4 Game Engine

10.4.1 Technical Specs

Las especificaciones técnicas del game engine usado (Unity), en su versión 2021.3.41f1 son las siguientes:

- Windows, de preferencia 10 y 11. Para MacOS, High Sierra 10.13+ (Intel), Big Sur 11+ (Apple Silicon o Intel con Rosetta 2) y para linux, Ubuntu 18.04/20.04, CentOS 7 (X64 con OpenGL 3.2+ o Vulkan).
- RAM recomendada: al menos 8 GB.
- CPU: arquitectura x64 con soporte SSE2.
- GPU mínima: OpenGL 3.2+ (también soporta Vulkan); compatibilidad con GPUs Nvidia y AMD.

10.4.2 Design

Unity está diseñado de tal forma que permite identificar las siguientes características:

Component-based / Entity-Component: los GameObjects actúan como contenedores vacíos a los que añades Componentes (Transform, Rigidbody, scripts, etc.) para definir comportamientos

Arquitectura modular y extensible: las partes vitales (renderizado, física, UI, scripting) funcionan como módulos intercambiables .

Escena como grafo: la escena se gestiona como un árbol jerárquico donde cada nodo es un GameObject con su Transform.

Render pipelines separados: Unity ofrece múltiples sistemas de renderizado (Built-in, URP, HDRP) adaptables según objetivos y rendimiento

Scripting en C# + backend en C++: internamente el motor está en C++, mientras el editor y scripts usan C# (con Mono o IL2CPP).

10.4.3 Collision detection

El juego en cuestión contiene “collision detection” en todos los objetos que son mostrados en pantalla. Esta detección de colisiones es lograda usando “sphere collider” (y en algunos pocos, cube collider) en todos los objetos, incluyendo naves, cubos, balas y planetas.

10.5 Interface technical specs

10.5.1 Features

Las interfaces principales para el usuario serán dos menús. Uno de ellos de inicio y otro de pausa.

El jugador verá en pantalla una barra de vida y un indicador en la esquina inferior izquierda de cuántas vidas le restan.

El jugador también cuenta con una mira o “crosshair” para que se de una idea de a dónde apuntan sus proyectiles, aunque esto es más una guía que el punto exacto donde los proyectiles impactarán puesto que ellos están formados por rigid body.

10.5.2 Details

El menú principal contiene dos botones. Uno que dice “jugar” que si el jugador hace click invocará la escena principal. Mientras que el otro que dice “salir” termina el programa.

El primer menú puede ser controlado mediante el mando o gamepad, pero el segundo menú aún está en espera de ser modificado para tener un comportamiento similar.

La barra de vida usa un método en el script [PlayerHealth.cs](#) que hace que cada vez que el jugador reciba un impacto ésta se vea reducida. Haciendo que el jugador se de una idea de su vida actual.

10.6 Controls' technical specs

10.6.1 Features

El tipo de input aceptado para este juego es de mandos o “gamepads”. Éstos pueden ser de Xbox One o de Xbox 360. Sin embargo, es recomendable manipular los menús usando el mouse.

El jugador puede mirar hacia arriba, hacia abajo, hacia los lados, pero sin restricción, de manera que la nave puede hacer un giro de 360 grados en cualquier dirección.

El jugador se puede mover hacia arriba, hacia abajo, hacia el frente, atrás y hacia los lados. Esto con el fin de simular que la nave se puede mover con libertad en el espacio.

El jugador puede disparar proyectiles.

10.6.2 Details

Para mover la cámara, el jugador deberá usar los ejes 4 y 5 para un movimiento horizontal y vertical, respectivamente.

Para avanzar o retroceder, el jugador deberá usar los ejes Y y X, mientras que para moverse en vertical, los botones LB y RB.

Si el jugador quiere disparar, entonces se debe de usar el Right Trigger.

10.7 Lighting models

10.7.1 Modes

10.7.1.1 Features

Sólo se usan tres puntos de luz direccional mixta, el tipo de luz que viene por default en Unity. Se admiten sombras suaves
No hay ciclo de día o noche.

10.7.1.2 Details

La fuerza de las sombras es de 1.
La resolución es la que viene por defecto, y el tipo de renderizado es automático.

10.8 Rendering system

En nuestro caso usamos el Built-in o standard Render Pipeline.

10.8.1 Technical specs

Compatible con Windows, macOS, Linux, iOS, Android y consolas. Sombras en tiempo real: requieren GPU con soporte de shadow mapping. En PC/Direct3D necesitan textura D16/D24X8 o DF16/DF24; en OpenGL necesitan la extensión GL_ARB_depth_texture.
Forward Rendering: ruta por defecto; cada objeto se dibuja en uno o varios pases según luces activas.

Deferred Rendering: soportado en PC y consolas con Shader Model 3.0 o superior; no disponible en móviles.

Vertex Lit: modo ligero que solo considera iluminación por vértice — útil para dispositivos de gama baja.

Soporta tanto shaders programables (Shader Model 2.0 en escritorio, OpenGL ES 3.0 en móviles) como fijos en casi todas las plataformas (excepto consolas).

Desde Unity 2021.2, cuenta con Shader Graph limitado para compatibilidad, pero no se actualiza activamente

10.8.3 Camera

La cámara siempre sigue al jugador desde la parte trasera de la nave de forma que pueda ver el escenario y a su contrincante. Esta cámara está pensado para que el juego se asemeje a un shooter en tercera persona.

10.8.3.1 Operation

La cámara está pensada para que sea en tercera persona. El jugador no tiene control sobre ella.

La cámara sigue al jugador en tiempo real, sin embargo no tiene algún sistema que evite que la cámara choque o atraviese objetos (que inicialmente se pensó hacerse con ray casting).

10.8.3.2 Features

El jugador, y por lo tanto la cámara, se puede controlar mediante el uso de los ejes 4 y 5 del joystick.

10.9 Internet/network spec

No es necesario contar con internet para jugar el juego, ya que es local en pantalla dividida.

10.10 System parameters

10.10.1 Máximo de Jugadores

Como máximo (y como mínimo) tenemos un total de 2 jugadores. En una siguiente versión se intentará implementar el juego en red para que sea posible que haya más jugadores conectados.

10.10.4 Sitios Web

Por el momento no se cuenta con ningún sitio web dedicado al videojuego.

10.10.6 Persistencia

El juego no cuenta con persistencia en esta versión.