

Instituto Privado de la Cámara Argentina de Comercio  
Desarrollo de Videojuegos

Materia: Desarrollo de Videojuegos I – Práctica Profesional I

Segunda Prueba Parcial

Agustín Delger

Año 2016

# 1 CONTENIDO

---

2	Introducción .....	3
3	Propuesta de juego .....	4
3.1	Nivel 1 .....	4
3.2	Nivel 2 .....	4
3.3	Nivel 3 .....	5
3.4	Nivel 4 .....	5
4	Propuesta de arte .....	7
4.1	Menu de opciones .....	7
4.2	Escenario .....	8
4.3	Aeronaves .....	9
5	Análisis técnico.....	10
5.1	Interfaz del juego .....	10
5.2	Personaje principal.....	10
5.3	Enemigos, obstáculos y pickables .....	11
5.4	Proyectiles.....	12
5.5	Fondo de escena .....	12
5.6	Singleton .....	12
6	Bibliografía y referencias .....	13

## 2 INTRODUCCIÓN

---

El juego que se propone en este documento se inspira principalmente en los primeros juegos *shoot 'em up* que podían ser encontrados en las máquinas recreativas **arcade**, tales como *Space Invaders* y *Asteroids*. Utilizando una mecánica de juego muy similar a la de estos en algunos casos, y mecánicas que aparecen recién en juegos más modernos en otros, se busca ambientar el juego a la idea de enormes aparatos recreativos para nada modernos donde la cantidad de botones y/o *sticks* que se tenían no era abundante, y en consecuencia los juegos tenían una limitante con respecto a esto. En este proyecto se jugará con estas limitaciones también, y por ello se limitará al uso de tan sólo 5 botones: 4 de desplazamiento y 1 de acción, lo que podría ser reemplazado por un *stick* de movimiento y 1 botón de acción si quisiera ser llevado e implementado en una máquina *arcade*.

Se decidió inspirar el presente trabajo en este tipo de juegos ya que fueron icónicos en la historia de los videojuegos a nivel mundial, y posiblemente gran parte de la masiva popularización de los videojuegos en la actualidad sea mérito de estos.

### 3 PROPUESTA DE JUEGO

---

Se trata de un juego de corta duración que contiene 4 niveles con jugabilidad y controles distintos, los cuales luego de ser ganados se los deberá volver a jugar en una segunda etapa aumentando el nivel de dificultad. El jugador controlará un avión de combate de la segunda guerra mundial y se enfrentará a distintos escenarios de supervivencia y combate en los cuales el único objetivo es mantenerse con vida hasta llegar a destino. Se proponen dos mecánicas de juego diferentes que variarán en cada nivel, otorgándole un dinamismo muy particular al juego.

#### 3.1 NIVEL 1

En el primer nivel del juego, el jugador volará el avión desde una cámara lateral, teniendo únicamente control sobre la potencia del motor. Esto quiere decir que lo único que el jugador podrá realizar es variar su altitud, desplazándose verticalmente en la pantalla pero quedando fija su posición horizontal. Esto lo podrá hacer utilizando tan solo la flecha superior del teclado para ascender, o soltándola y dejando actuar la gravedad para descender.

El objetivo de este nivel es introductorio. Dado que el juego no contendrá ningún tipo de tutorial, se busca comenzar con un nivel sencillo, sin ningún tipo de obstáculo o enemigo en los primeros segundos para que el jugador pueda comprender y adaptarse a la mecánica de juego ya que será la misma mecánica que se utilizará en el siguiente nivel. Pasados los segundos iniciales, el jugador deberá comenzar a adquirir habilidades de pilotaje para esquivar distintos obstáculos con los que se enfrentará, como montañas o tormentas eléctricas. Sin ningún tipo de vidas o puntaje, el objetivo del jugador es sobrevivir por unos pocos minutos sin colisionar contra ninguno de los obstáculos o salirse de la pantalla, ya que cualquiera de estas acciones ocasionará la destrucción de la aeronave teniendo que volver a comenzar el nivel.

Como se ha mencionado previamente, hay una segunda etapa del juego en la cual se deberá volver a jugar todos los niveles, pero con la dificultad aumentada. En el caso de este nivel, siendo ya conocida la mecánica de juego, esto se realizará aumentando la velocidad del avión, reduciendo el tiempo de aparición de los obstáculos y acortando las distancias entre éstos. Estos cambios harán de este nivel ya, no un nivel introductorio, sino un gran desafío.

#### 3.2 NIVEL 2

Continuando con la mecánica de juego del nivel anterior, los controles del jugador serán los mismos agregando ahora la barra espaciadora para disparar proyectiles. Dejando de lado los obstáculos, el objetivo del nivel será nuevamente llegar a destino con vida destruyendo o esquivando todas las aeronaves del ejército enemigo y sus proyectiles. Estos enemigos volarán en dirección opuesta al jugador, manteniendo una altitud constante y disparando en línea recta.

Considerando que las únicas posibilidades del jugador son ascender y descender, los proyectiles serán lanzados en línea recta en la dirección en la que se está yendo en el momento. Es decir, si se está

ascendiendo, el disparo saldrá levemente hacia arriba y si se está descendiendo, saldrá levemente hacia abajo. Lo que esto genera, por un lado, es que apuntar sea algo mucho más complejo que en los juegos en los que solo se puede disparar en dirección horizontal, siendo esquivar a los enemigos una opción muy recomendable por momento. Por otro lado, esto permite esquivar balas enemigas y contraatacar al mismo tiempo, ya que esto se puede hacer desde una altitud diferente al enemigo, permitiéndole al jugador adquirir diferentes estrategias de combate y convertirse en un piloto mucho más hábil.

Con el objetivo de desalentar el esquivar a la totalidad de los enemigos, al destruirlos se podrán obtener diferentes *power-ups*, como un escudo, que protegerá a la aeronave contra un proyectil o una colisión, o un arma mejorada, para facilitar la destrucción de más enemigos. Estos elementos quedarán en el aire una vez destruido el enemigo, y deberán ser agarrados para activarlos.

Para la segunda etapa del juego, se aumentará la dificultad al reducir el tiempo entre enemigo y enemigo y aumentar la frecuencia de disparos de éstos.

### 3.3 NIVEL 3

El tercer nivel del juego introduce una nueva mecánica. A diferencia de los niveles anteriores, la cámara es cenital, viendo el avión desde arriba, y el desplazamiento de la pantalla es vertical, es decir, volando con rumbo norte. En este escenario, el jugador se encuentra volando sobre territorio enemigo, por lo que vendrán proyectiles de todas direcciones. Con total libertad de movimiento en la pantalla, el objetivo de este nivel es llegar a destino sin ser impactado.

Al igual que el nivel 1, este nivel funciona como introducción a esta nueva mecánica, ya que será utilizada también en el siguiente nivel. Con este propósito, el avión no estará aprovisionado con ningún tipo de armamento, y se deberá llevar a cabo la misión únicamente mediante maniobras evasivas. Se busca que el jugador se adapte a la nueva mecánica y adquiera buenas habilidades de desplazamiento.

Para la segunda etapa del juego el nivel dejará de ser introductorio. Para esto, se aumentará la velocidad de desplazamiento de avión, pero al mismo tiempo, también la velocidad y frecuencia de aparición de los proyectiles.

### 3.4 NIVEL 4

El cuarto y último nivel continúa con la mecánica del nivel previo, asumiendo que el jugador ya tiene un buen entendimiento de la misma. En este nuevo escenario el jugador se enfrenta a aeronaves enemigas que se aproximan en dirección opuesta decididas a acabar con su vida, ya sea disparando proyectiles o embistiéndolo en forma suicida. Agregando nuevamente la barra espaciadora al controlador, el jugador deberá contraatacar o utilizar sus conocimientos de evasión para sobrevivir.

A diferencia del nivel 2, el jugador podrá disparar únicamente en sentido vertical, al igual que los enemigos. De todas formas, estos últimos ya no volarán en línea recta, sino que algunos tendrán movimientos aleatorios, y otros volarán en dirección al jugador con intenciones de estrellarse contra éste.

Nuevamente, se podrán obtener *power-ups* como un escudo o mejores armas a partir de destruir aeronaves enemigas.

Para la segunda etapa del juego, esta vez no solo se aumentará la velocidad y frecuencia de los enemigos, sino que, además, la cantidad de proyectiles con los que el jugador contará serán finitos, teniendo la posibilidad de obtener más a partir de las naves enemigas. Siendo éste el último nivel del juego, se busca que el jugador aplique todos los conocimientos obtenidos en los niveles anteriores, y que lo haga con gran precisión. Por un lado, el jugador deberá intentar utilizar la menor cantidad posible de proyectiles mediante maniobras evasivas para no acabarlos, pero por el otro, la cantidad de enemigos será mucho mayor, por lo que será prácticamente indispensable el uso de éstos. Por este motivo, cada proyectil deberá ser disparado con gran determinación.

## 4 PROPUESTA DE ARTE

---

Si bien el juego está inspirado en las antiguas máquinas *arcade*, no se busca lo mismo en cuanto a los gráficos. Utilizando imágenes realistas en el menú inicial del juego, e imágenes levemente más animadas en el juego, se busca ambientar el juego en la década de 1940, en la segunda guerra mundial. Se muestran a continuación diferentes referencias en las que se inspira el arte del juego.

### 4.1 MENU DE OPCIONES

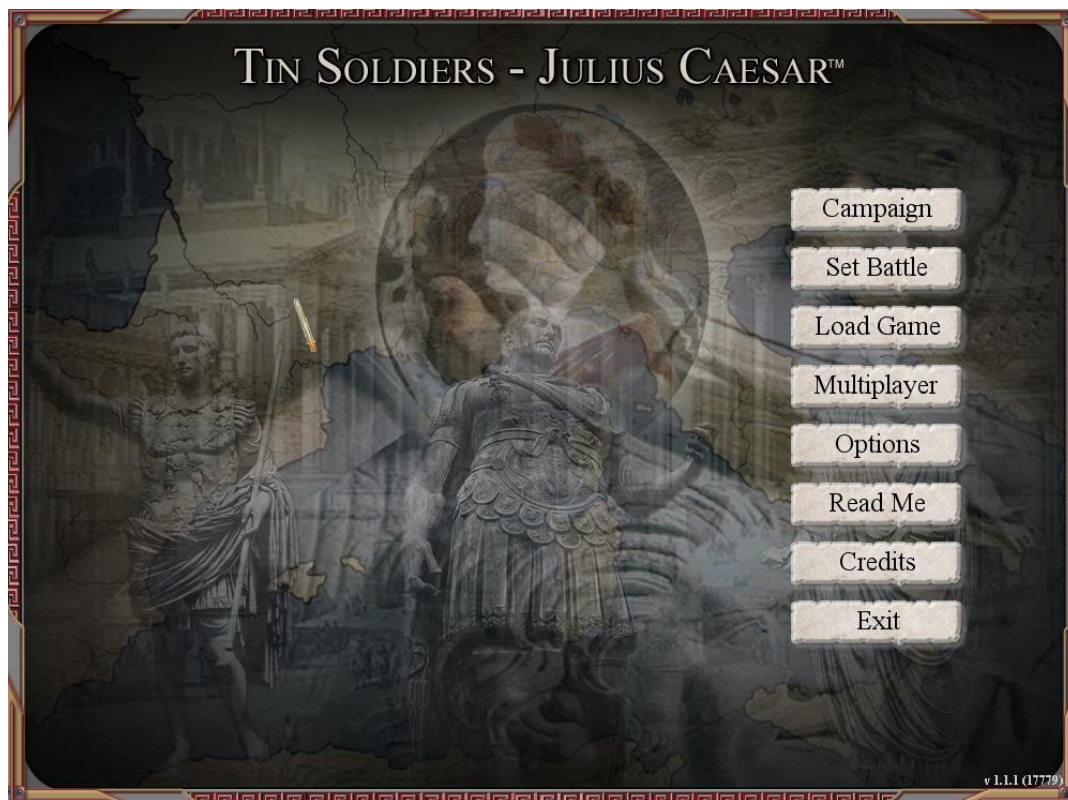


Imagen 1: Referencia de menú de opciones.

Se utilizarán imágenes reales en el fondo del menú de inicio para ambientar al usuario en la época del juego. En este caso, se utilizará la idea de imágenes superpuestas para lograr mostrar una gran cantidad de ellas en un espacio reducido. Podrán verse distintos aviones y pilotos de la época.

## 4.2 ESCENARIO

Si bien para los primeros dos niveles el escenario es tan solo cielo de fondo, para los otros que utilizan cámara cenital, se busca mostrar un ambiente de guerra de fondo. Se podrán ver soldados, tanques y lanzamisiles. La siguiente imagen muestra una referencia de lo buscado.



Imagen 2: Referencia de escenario.

Si bien esta imagen muestra lo que se busca, como el juego trata únicamente de aviones, estos volarán a mayor altitud, por lo que la imagen de fondo deberá ser más lejana. Un ejemplo más concreto de esto se puede ver en la siguiente escena, mostrando de lejos una escena de guerra y caos total.





Imagen 3: Referencia de imagen de fondo.

### 4.3 AERONAVES

La siguiente imagen muestra una gran variedad de aviones de combate del tamaño y estilo de gráfico que se encontrarán en el juego en los últimos niveles. Para los primeros se verán desde una cámara lateral y el tamaño de los gráficos será levemente más grande.



Imagen 4: Referencia de aeronaves.

## 5 ANÁLISIS TÉCNICO

---

Se propone como plataforma de desarrollo el motor Unity, con el fin de aprovechar todos los elementos que éste provee para el desarrollo de videojuegos. Para una descripción más detallada y clara de esta sección, se analizarán por separado todos los elementos del engine requeridos para los *GameObjects* principales del juego, y cómo serán utilizados.

### 5.1 INTERFAZ DEL JUEGO

Con el objetivo de lograr una interfaz clara y amigable para el usuario y a su vez sencilla de implementar, se utilizarán las herramientas **UI** (*User Interface*) provistas por Unity. La interfaz se descompondrá en diferentes paneles **UI** donde cada uno de estos albergará los botones, textos e imágenes necesarios. El usuario podrá desplazarse muy fácilmente por las distintas opciones con el uso del mouse, o bien con el uso de las flechas del teclado y la barra espaciadora.

Para esta parte, se incluirán, además, scripts y componentes de animación para controlar el desplazamiento y efectos de los distintos cuadros de texto y opciones de la interfaz, acompañados por sonido de movimiento o al hacer clic para darle un *feedback* al usuario al seleccionar un opción.

### 5.2 PERSONAJE PRINCIPAL

El personaje principal, o en este caso la aeronave del jugador, es probablemente uno de los objetos más sencillos de implementar en cuanto a desplazamiento, pero así mismo el más complejo cuando se trata de animaciones y disparar proyectiles, principalmente en los dos primeros niveles. Teniendo en cuenta el cambio de mecánica al llegar al tercer nivel, dividiremos el análisis en dos partes: los dos primeros niveles y los últimos dos.

De todas formas, si bien gran parte de los componentes utilizados en ambas mecánicas difieren mucho, algunos se mantienen invariantes. Dentro de estos últimos, los principales son **Box Collider 2D** y **Rigid Body 2D**. Estos componentes son de suma importancia ya que en conjunto se encargarán de detectar las colisiones, lo que dará condición de derrota al jugador.

#### Primeros dos niveles

Recordando rápidamente la mecánica de estos niveles, el jugador puede únicamente ascender o dejarse caer por la gravedad y en el caso del segundo nivel, también puede disparar. El movimiento del jugador es muy básico en este caso, y como mencionado previamente, puede ser implementado con gran facilidad si se tiene acceso al sistema de física de Unity. Esto se lleva a cabo activando la gravedad que trae el componente **Rigid Body 2D** ya mencionado, y aplicando fuerzas de ascenso desde un script cuando se presiona la tecla correspondiente.

A pesar de ser prácticamente nula la animación del avión mientras se encuentra en vuelo, el objeto deberá tener un componente de animación para que en el momento en el que se detecte una colisión se genere la explosión de la nave. Los efectos de ascenso y descenso, en cambio, se controlaran directamente desde la rotación del **Transform** del objeto que viene por defecto, ya que siendo una cámara lateral, la imagen de la aeronave deberá ser rotada dependiendo de las fuerzas ejercidas sobre ella en cada momento. Esto no es nada fácil, ya que tratándose de fuerzas y ángulos, el grado de rotación deberá ser calculado aplicando funciones trigonométricas y otras operaciones matemáticas. Estos mismos cálculos se utilizarán para marcar la trayectoria de los proyectiles en el momento del disparo.

### Últimos dos niveles

A diferencia de los dos primeros niveles, en estos la cámara es cenital y el desplazamiento es libre. Manteniéndose una altitud fija, ya no es necesario utilizar la gravedad provista por la física de Unity, aunque se seguirá utilizando el componente **Rigid Body 2D** para controlar el movimiento desde un script, utilizando los comandos `GetAxis("Horizontal")` y `GetAxis("Vertical")` que Unity ofrece.

En cuanto a las animaciones, en estos niveles es mucho más sencillo realizarlo ya que se requiere únicamente un *Sprite Sheet* con las animaciones de desplazamiento horizontal del avión y la de vuelo en línea recta. Nuevamente será necesario incluir la animación de explosión.

## 5.3 ENEMIGOS, OBSTÁCULOS Y PICKABLES

Al igual que la aeronave principal, todo enemigo, obstáculo y pickable deberá tener los componentes **Box Collider 2D**, **Rigid Body 2D**, **Sprite Renderer**, y un script de movimiento particular para cada uno, el cual tomará como parámetro la posición inicial del objeto, y en caso de aplicar, también controlará la frecuencia y sentido en que disparará. Deberán estar acompañados, además, de las animaciones correspondientes.

Si bien todos estos objetos se desplazarán a través del **Transform**, algunos tendrán comportamientos mucho más complejos que otros. Los obstáculos y pickables, por un lado, y algunos enemigos tendrán desplazamientos lineales. Por otro lado, habrá otros enemigos que tendrán desplazamientos zigzagueantes de forma aleatoria y/o velocidad no constante, y finalmente se encuentran las anteriormente descritas aeronaves suicidas que deberán desplazarse en dirección al jugador. Para lograr esto último, el jugador tendrá un método público que devolverá su posición actual y que podrá ser accedido por este tipo de enemigos, y así saber en qué dirección movilizarse.

Dado que se requerirán una gran cantidad de obstáculos y enemigos en cada nivel, se utilizará un **Object Pool** para evitar la creación y destrucción descontrolada de objetos en tiempo de juego. Esto se implementará mediante un script que tomará el **prefab** de algún objeto y la cantidad de instancias que se desean crear. Deberá tener un método público para permitir la activación o desactivación de éstas. Para la activación también deberá tomar la posición inicial de los objetos, que en la mayoría de los casos tendrá una componente aleatoria, para que aparezcan en diferentes posiciones de la pantalla.

## 5.4 PROYECTILES

Si bien serán necesarios diferentes tipos de proyectiles a lo largo del juego (para el jugador, los enemigos y los *power-ups*), la forma en que estos se implementan es la misma, ya que se manejarán todos desde un mismo script. Este script recibirá el **prefab** de cada tipo de bala o misil a utilizar, y tendrá un método público que toma como parámetro la bala que se quiere disparar en ese instante, justo con los datos iniciales del proyectil como la posición y la dirección en que se desea disparar.

Cada **prefab** de proyectil que se desee implementar deberá tener un **Sprite Renderer**, un **Box Collider 2D** y un **Rigid Body 2D** para el gráfico y para detectar las colisiones. El movimiento de cada uno deberá ser controlado, a su vez, desde otro script genérico que utilizará cada proyectil, el cual fijará la velocidad inicial, aceleración (si la tuviera) y demás datos, y llevará a cabo el desplazamiento a partir de la posición del componente **Transform**. También deberá tener la animación de explosión y el efecto de sonido correspondiente.

Considerando que se utilizarán una gran cantidad de proyectiles, éstos se manejarán también con un **Object Pool**, utilizando el mismo script descripto previamente.

## 5.5 FONDO DE ESCENA

Dado que en todos los escenarios el avión se encuentra en vuelo, el fondo de escena consistirá simplemente en una imagen de fondo y un script de movimiento. Dependiendo del nivel, la imagen de fondo será simplemente de un cielo despejado con algunas nubes, o territorio enemigo en caso de los últimos niveles. De todas formas, ambas imágenes compartirán la propiedad de ser *loops*, es decir, imágenes que se repiten una y otra vez, ya que es necesario simular un escenario infinito.

Para dar una sensación de lejanía, el fondo se desplazará a una velocidad muy inferior a la de los obstáculos y enemigos. Esto se llevará a cabo modificando la posición del componente **Transform** desde el script.

## 5.6 SINGLETON

Todos los *Game Objects* previamente mencionados y descriptos podrán acceder y a su vez ser accedidos por esta clase estática, la cual tendrá como finalidad el controlar los aspectos generales del juego. Esto permite evitar el uso de eventos y en consecuencia facilitar el desarrollo del juego, considerando que se trata de un proyecto pequeño.

## 6 BIBLIOGRAFÍA Y REFERENCIAS

---

Documentación oficial de Unity: <https://docs.unity3d.com/Manual/index.html>

Imagen 1: Referencia de menú de opciones, [http://3.bp.blogspot.com/-1xRUTSAK0Kc/TtaDfmZM\\_eI/AAAAAAAAAUg/pQZZWge36P8/s1600/menu.jpg](http://3.bp.blogspot.com/-1xRUTSAK0Kc/TtaDfmZM_eI/AAAAAAAAAUg/pQZZWge36P8/s1600/menu.jpg)

Imagen 2: Referencia de escenario, [https://cdn.tutsplus.com/gamedev/authors/michael-james-williams/GraphicRiver\\_War\\_Game\\_Sprite\\_Sheet.jpg](https://cdn.tutsplus.com/gamedev/authors/michael-james-williams/GraphicRiver_War_Game_Sprite_Sheet.jpg)

Imagen 3: Referencia de imagen de fondo,  
[http://cf.shacknews.com/images/20120313/slider4\\_21378.jpg](http://cf.shacknews.com/images/20120313/slider4_21378.jpg)

Imagen 4: Referencia de aeronaves, <https://s-media-cache-ak0.pinimg.com/236x/d4/a4/09/d4a409833e629ea2dba0afa380cdaa0c.jpg>