

**ANALISIS PREVIO DEL PROYECTO FINAL**

JUAN PABLO RENDÓN JIMÉNEZ

INFORMATICA II

PROFESOR: AUGUSTO SALAZAR

FACULTAD DE INGENIERIA

UNIVERSIDAD DE ANTIOQUIA

MEDELLIN

2023

Contenido

[**Descripción** 3](#_Toc135294218)

[**Detalles** 3](#_Toc135294219)

[ **Personajes** 3](#_Toc135294220)

[ **Niveles** 5](#_Toc135294221)

[ **Menú** 5](#_Toc135294222)

[ **Caligrafía** 5](#_Toc135294223)

[**Físicas** 5](#_Toc135294224)

[**Requerimientos** 6](#_Toc135294225)

[**Clases Necesarias** 6](#_Toc135294226)

World War III: The Mission

# **Descripción**

En plena Tercera Guerra Mundial, un avión militar alemán se encuentra en una misión crítica: llevar suministros a las tropas alemanas que se encuentran en el frente de batalla. El objetivo es “simple”, el avión debe volar 10 kilómetros para entregar los suministros y salvar la vida de muchos soldados.

Sin embargo, la tarea no será sencilla. El camino está plagado de peligros. Bolas de fuego, obstáculos que pueden surgir en cualquier momento, aves en el cielo y otros peligros se interpondrán en el camino del avión alemán.

El avión tiene que esquivar y sortear todos estos peligros para poder cumplir su misión. Si logra llegar a su destino y entregar los suministros, se salvarán muchas vidas de soldados alemanes que luchan en el frente de batalla.

La suerte de muchos depende de este avión y de la habilidad de su piloto para esquivar todos los obstáculos y peligros que se presenten en su camino. ¿Logrará el avión alemán cumplir su misión y salvar vidas?

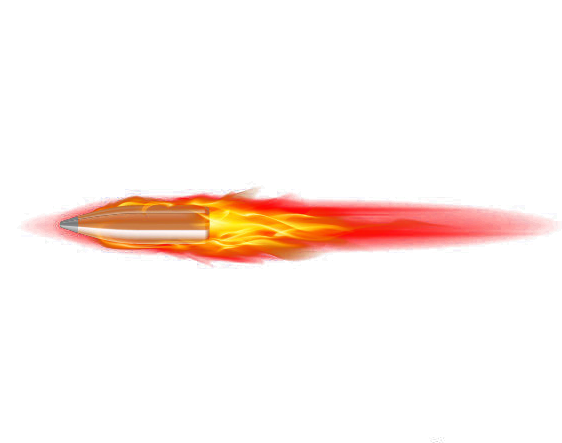
# **Detalles**

## **Personajes**

* **Avión:**



* **Bala:**



* **Bola de fuego:**

****

* **Misil:**

****

* **Aves:**

****

## **Niveles**

* **Nivel 1:** El avión no podrá disparar y deberá esquivar todos los obstáculos que vendrán con una velocidad de acuerdo con la dificultad indicada por el usuario. Uno de esos obstáculos será una vida que podrá usar a partir de ese momento y en otros niveles. El puntaje valdrá el doble.
* **Nivel 2:** se activará la habilidad del disparo. No caerán vidas. El puntaje solo valdrá una unidad.

## **Menú**

El menú constará de tres secciones: una para elegir la **dificultad**, otra para escoger el **nivel** y por último una sección de **ayuda** donde se explica el funcionamiento del juego y las teclas disponibles (Figura 1.1).



**Figura 1.1:** Previsualización del menú del juego.

## **Caligrafía**

Showcard Gothic.

# **Físicas**

* Fórmulas para el desplazamiento de Avión, Bola de fuego, aves y bala:

**Xf = Xo + V**

Xf: Posición final.

Xo: Posición Inicial.

V: Velocidad.

* ~~Fórmulas para recrear el movimiento semiparabolico de la bomba.~~

**~~Posy = 9.8\*(tiempo\*tiempo)~~**

**~~Posx = Vo\*tiempo~~**

**~~Velocidad = 9.8\*tiempo~~**

~~Posy: Posición en Y.~~

~~Posx: Posición en X.~~

~~Vo: Velocidad inicial.~~

# **Requerimientos**

* Avión con la capacidad de subir y bajar. Por defecto, tendrá 3 vidas inicialmente.
* El avión debe tener la capacidad de defenderse con una bomba disparada por el eje X.

~~El avión debe tener la capacidad de defenderse de los obstáculos que se encuentren por debajo de este activando una bomba. (usar la fórmula de movimiento semiparabolico)~~

* Bola de fuego que aparecerá de manera aleatoria en el eje Y. Si impacta el avión reducirá dos vidas.
* Aves que reducen una vida.
* Estas dos últimas deben aparecer mínimo 6 en pantalla, para darle emoción al juego.
* Medidas de Escena: 642 x 414.
* Vida que otorgue una vida extra al avión.

# **Clases Necesarias**

* **Avión:** esta clase contendría la lógica del avión y su movimiento. Incluiría la capacidad de mover el avión hacia arriba y abajo para evitar obstáculos, y también de disparar a peligros que puedan aparecer.

**Atributos:**

Double posx, posy, radio;

Int velocidad, vida;

**Metodos:**

**Avion**(double x,double y,int r);

**Void MovArriba;**

**Void MovAbajo;**

void ***Paint*(…);**

* **Bala:** Manejaría la velocidad de disparo del avión, su recorrido en X y su daño en el obstáculo.

**Atributos:**

Double posx, posy, radio;

Int velocidad;

**Métodos:**

**Bala**(double x,double y,int r);

void **left**();

void spawn();

void ***paint*(…);**

* **Bolas de fuego:** controla la aparición de las bolas de fuego y su movimiento a través de la pantalla.

**Atributos:**

Int posx, posy, radio, velocidad;

**Métodos:**

**Bala**(double x,double y,int r);

void **left**();

void spawn();

void ***paint(…);***

* **Aves:** controla la aparición de las bolas de fuego y su movimiento a través de la pantalla.

**Atributos:**

Int posx, posy, radio, velocidad;

**Métodos:**

**Bala**(double x,double y,int r);

void **left**();

void spawn();

void ***paint (…);***

* **Misil:** controla la aparición de las bolas de fuego y su movimiento a través de la pantalla.

**Atributos:**

Int posx, posy, radio, velocidad;

**Métodos:**

**Bala**(double x,double y,int r);

void **left**();

void spawn();

void ***paint*(…);**

* **~~Meta~~**~~: Cuando el avión llegue a la pista se entregarán los suministros y se incrementará el puntaje del jugador.~~
* **~~Puntaje~~**~~: controla el puntaje del jugador.~~
* **~~Interfaz de usuario:~~** ~~Controlaría la pantalla del juego y los controles que el usuario necesitaría para jugar, como el menú.~~

Otros métodos que se incluyen, pero al mainwindow:

bool **EvaluarColision**();

bool **Col\_AvionBola**();

int cont=0;

int vidas=0;

void **msg\_win**(int cont);