



# UNIVERSIDAD DE ANTIOQUIA

---

## Facultad de Ingeniería

Etapa 2 y 3: Análisis y Diseño

Informática II

Mario Esteban Estrada Gonzalez

CC: 1233191679

Juan José Díaz Zuluaga

CC: 1001456540

Docente

Aníbal José Guerra Soler

2023-2

UNIVERSIDAD DE ANTIOQUIA

FACULTAD DE INGENIERÍA

SEDE MEDELLÍN

Octubre 2023

Medellín, Colombia

## **Diseño**

### ***Descripción de Clases***

Main\_window:

Esta clase se utilizará para crear la ventana de la aplicación, donde gestionará todo el marco de la interfaz de usuario, incluyendo las barras de estado y el manejo de eventos de ventana.

Juego:

En esta clase se gestionará toda la dinámica de creación y destrucción de objetos, el inicio del juego, su finalización junto con todos los elementos que interactúan dentro del mismo, en cada uno de los niveles administra los diferentes comportamientos de cada objeto.

Personajes:

Es una clase abstracta puesto que no se creará una instancia como tal pero sí se presta como una clase base para las clases enemigos y personaje principal, además contiene los atributos necesarios para la dinámica del juego.

Enemigos:

Tal como indica el identificador de esta clase los objetos que permite crear serán los enemigos a los cuales el personaje principal debe derrotar, como se mostró en la primera etapa existen 3 tipos de objetos diferentes los cuales son novato, intermedio y difícil, donde su divergencia consiste en el valor de los atributos heredados de la clase personaje.

Personaje\_Principal:

Esta clase permite crear 2 objetos los cuales son los dos tipos de personajes principales con los que cuenta el juego, además permite cambiar el valor de los atributos Health y Speed que son heredados de la clase personajes.

Armas:

Esta clase tiene anidado dentro de sus atributos otra clase ammunition, puesto que aunque existen 3 tipos diferentes de armas estas no interactúan directamente con el enemigo si no que

disparan las balas las cuales si afectarán la vida de los personajes, sin embargo es necesario mostrarle al usuario las características visuales de cada arma.

Ammunation:

La clase permite crear las instancias (balas) las cuales serán ilimitadas e interactúan directamente con los enemigos al restarles vida, además contiene como método el tipo de movimiento que presentan estas puesto que se disparan hacia la ubicación del puntero del mouse.

Power\_Ups:

Aquí se crean los modificadores de habilidad que interactúan con el personaje principal y le permiten ganar el juego más fácil, existen dos tipos de objetos, los que cambian el valor del atributo Health y los que cambian el valor del atributo Speed.

### ***Estructura de datos***

Para manipular y controlar todos los objetos del juego se plantea el uso de contenedores, esto con el objetivo de saber por ejemplo cuantos enemigos están en la escena y a su vez observar saber cómo cambian sus atributos como la vida.

Entonces, en general toda la estructura de datos se define a través de dos partes: El manejo y la interacción entre clases, y el almacenamiento en contenedores de los diferentes objetos como enemigos, armas y municiones.

### ***Tablas de Atributos y Métodos***

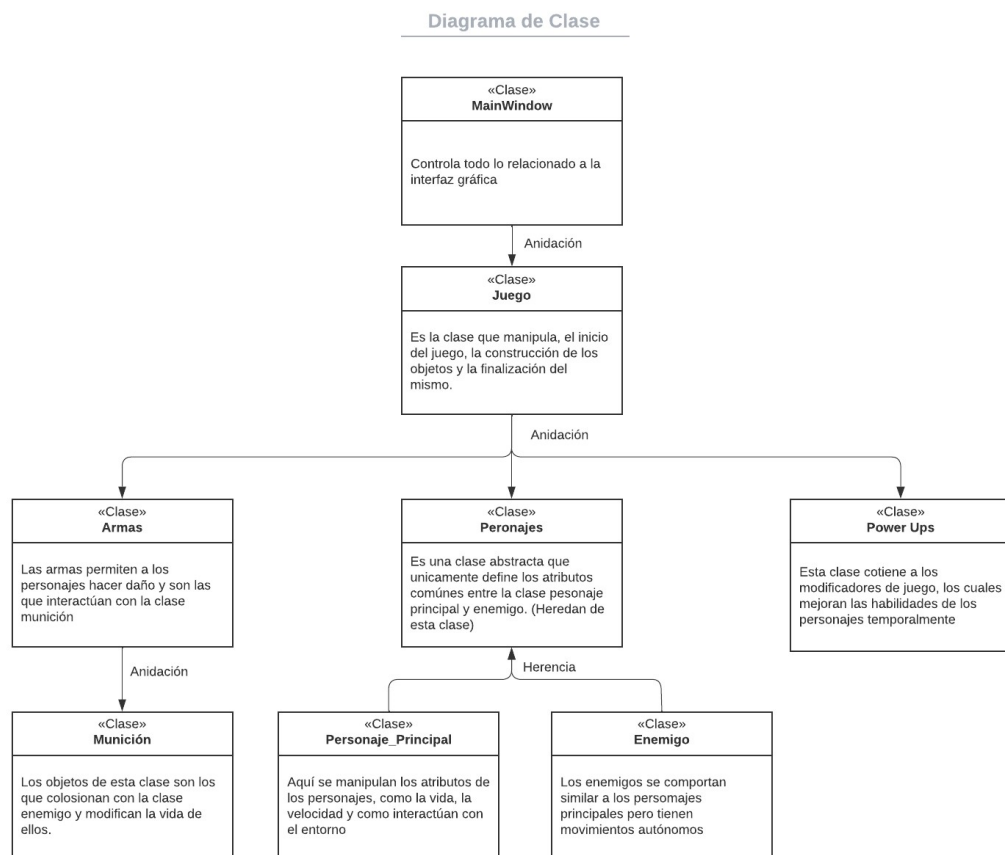
<b><i>Clase: Juego</i></b>	
<b><i>Atributo</i></b>	<b><i>Definición</i></b>
escena-public	Lugar donde se encuentran todas los objetos del juego
<b><i>Métodos</i></b>	<b><i>Definición</i></b>
set_graphics-public	permite insertar todos los personajes, grow ups, enemigos y demás elementos del juego

<b>Clase: Personajes</b>	
<b>Atributo</b>	<b>Definición</b>
Health-public	Vida que posee cada personaje
Speed-public	Velocidad para moverse en la escena
Weapon-public	Objeto de otra clase que indica cuanto daño (damage) puede ocasionar el personaje.
Damage-public	Daño puede ocasionar el personaje a otros.
<b>Métodos</b>	<b>Definición</b>
set_health-public	Permite cambiar el valor de health
set_speed-public	Permite cambiar el valor de velocidad
change_weapon-public	Permite asignarle un objeto de la clase arma al personaje.
set_damage-public	Permite cambiar el nivel de daño del personaje.
move-private	Describe el movimiento con el que rige el personaje.
cut_sprite-private	Permite escoger la imagen del personaje dentro de los sprites

<b>Clase: Armas</b>	
<b>Atributo</b>	<b>Definición</b>
ammunition -public	Es un objeto de otra clase que será el que impacte contra los enemigos.
<b>Métodos</b>	<b>Definición</b>
cut_sprite- private	Permite escoger la imagen del arma dentro de los sprites
move-private	Describe el movimiento con el que se rige el arma.

<i>Clase: Ammunition</i>	
<i>Atributo</i>	<i>Definición</i>
Damage-public	Daño puede ocasionar el personaje a otros.
<i>Métodos</i>	<i>Definición</i>
cut-sprite- private	Permite escoger la imagen de la munición dentro de los sprites
move-private	Describe el movimiento con el que se rige la munición.

## Diagrama de Clases



## Física del Juego

Las principales físicas del juego se definen a través de movimientos conocidos de la cinemática; en resumen se utilizan 3 movimientos diferentes.

- MRU: Tanto el personaje principal, como los enemigos en el primer nivel se mueven con un movimiento rectilíneo uniforme sin aceleraciones. El único cambio es cuando los enemigos colisionan con los límites del juego
- MRUA: Las balas en ambos niveles se mueven con aceleración y de manera rectilínea, partiendo con una velocidad inicial definida
- Espiral de Arquímedes: Finalmente los enemigos en el segundo nivel se mueven en una espiral alrededor de toda la pantalla de juego. Este movimiento se mueve utilizando un parámetro de tiempo. Y se descompone en coordenadas cartesianas  $x,y$ .

### ***Métodos de Interacción***

Para que el juego se pueda desarrollar correctamente deben haber múltiples interacciones entre los objetos de las diferentes clases; esto se debe dar a través de métodos en los cuales los objetos interpreten señales como choques con los límites de la zona de juego, con objetos que hacen daño o con elementos modificadores de juego como “Power Ups”.

Los principales métodos de interacción son:

- Colisiones: Las colisiones son el núcleo de las interacciones de todos los objetos entre sí, ya que a través de esto, los diferentes elementos del juego logran definir sus acciones y métodos a ejecutar.
- Perder\_vida(): El cuál es un método tanto de la clase Personaje\_Principal como de la clase Enemigo. Ya que cada uno de ellos debe reducir el valor del objeto vida, el personaje principal cuando colisiona con un enemigo y el enemigo cuando colisiona con el objeto munición.
- Disparo: El objeto disparo se puede modelar como un método de la clase arma pero debe estar relacionado también con la clase personaje principal y la clase munición.
- Power\_Up: Los power ups son elementos que mejoran el desempeño del personaje principal, pueden darle más vida o más velocidad al jugador y para obtenerlos se realiza mediante una colisión, en la que el personaje atrapa el power\_up