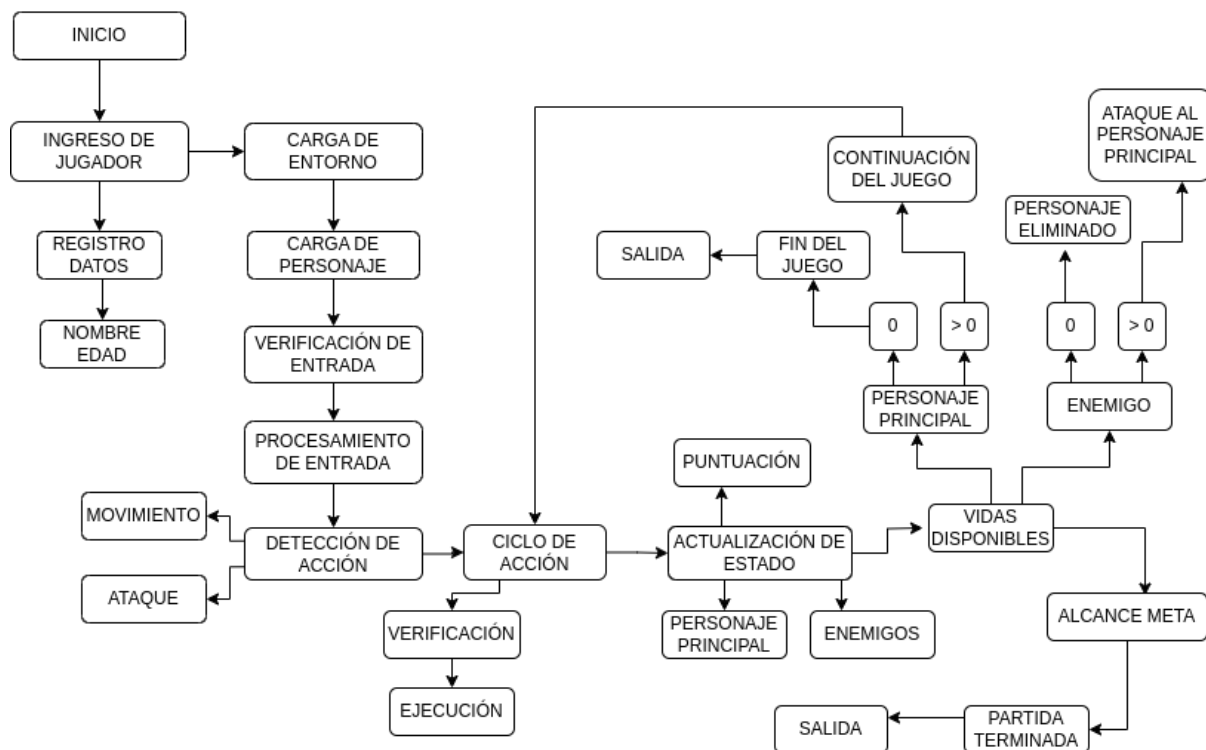


Informática II

Miguel Angel Botero Gonzalez
Yuliana Betancur Saldarriaga

Diseño

Diagrama de funcionamiento



Estructura de datos

Personaje principal	<ul style="list-style-type: none">• Rick
Enemigos	<ul style="list-style-type: none">• Atacante #1• Atacante #2• Líder principal
Armamento	<ul style="list-style-type: none">• Pistola• Pistola 2.0
Entorno	<ul style="list-style-type: none">• Background• Obstáculos• Plataformas• Mejoras armamento
Registros	<ul style="list-style-type: none">• Estadísticas• Puntuaciones

Descripción de clases, atributos y métodos

- **Clase Game:** Esta será la clase principal que controlará la lógica del juego. Debe gestionar la inicialización y la representación gráfica.

Métodos:

startGame(): Inicializa el juego.

renderGame(): Dibuja los elementos del juego en la pantalla.

- **Clase Player** (hereda de QGraphicsItem): Esta clase representa al personaje principal del juego, que es Rick. Debe gestionar su movimiento, disparo y colisiones.

Atributos:

Nombre

Salud

Armamento
Velocidad
Posición

Métodos:

moveLeft(): Mueve al personaje principal a la izquierda.

moveRight(): Mueve al personaje principal a la derecha.

jump(): Permite al personaje principal saltar.

shoot(): Disparo ejecutado por personaje principal.

handleCollision(): Maneja las colisiones con enemigos u objetos del juego.

die(): Gestiona la eliminación del personaje principal cuando un enemigo le dispara.

- **Clase Enemy** (hereda de QGraphicsItem): Representa a los enemigos del juego. Puedes crear subclases para diferentes tipos de enemigos.

Atributos:

Tipo
Salud
Armamento
Velocidad
Posición

Métodos:

move(): Controla el movimiento de los enemigos.

die(): Gestiona la eliminación de un enemigo cuando el personaje principal le dispara.

shooten(): Gestión de disparos por parte de los enemigos

- **Clase Bullet** (hereda de QGraphicsItem): Representa los proyectiles disparados por el personaje principal.

Atributos:

Velocidad
Posición
Cantidad
Dirección
Daño

Métodos:

move(): Controla el movimiento de las balas.

hit(): Gestiona el impacto de las balas en los diferentes personajes (principal y enemigos).

apharm(): Determinar y aplicar el daño generado al personaje.

imprv(): Aplicación de mejoras al proyectil.

- **Clase Gun** (hereda de **QGraphicsItem**): Representa las armas poseídas por el personaje principal.

Atributos:

Posición
Dirección
Color

Métodos:

shot(): Generar disparo de arma.

load(): Carga de arma obtenida en el nivel.

change(): Gestionar cambio de arma.

- **Clase GameScene (hereda de QGraphicsScene)**: Esta clase controla la escena del juego y gestiona la interacción entre los objetos del juego.

Atributos:

Personajes
Objetos
Velocidad

Posición
Cantidad

Métodos:

keyPressEvent(QKeyEvent* event): Maneja eventos de teclado para controlar al personaje principal.

advance(): Actualiza el juego en cada fotograma.

- **Clase MainWidget** (hereda de QWidget): Esta clase es la ventana principal del juego y contiene la escena del juego.

Atributos:

Ancho
Alto
Fondo

Métodos:

initGame(): Inicializa la escena y comienza el juego.

- **Clase MainWindow** (hereda de QMainWindow): La ventana principal de la aplicación que contiene MainWidget.
- **Clase Scoreboard:** Registro de la puntuación del jugador.

Atributos:

Nombre
Edad
Puntuación

Métodos:

increaseScore(int points): Incrementa la puntuación del jugador.

displayScore(): Muestra la puntuación en la pantalla.

