Nombre Apellido: David Cusi   
Legajo: 42286775   
Email: davidcusi.estudiante@gmail.com

Profesor:Gonzalo Rodrigo  
Com: 03

Programacion 1  
trabajo practico

****

**Introducción**

En este trabajo práctico, desarrollamos un juego que simula el movimiento de un personaje, pep, en un entorno de plataformas, representado por varias islas. El objetivo del juego es mantener al personaje dentro de las plataformas; si el personaje cae del mundo, el juego termina. El proyecto involucra la implementación de clases orientadas a objetos que controlan el movimiento del personaje y la interacción con el entorno, proporcionando una experiencia interactiva al usuario.

**Descripción**

Clases Implementadas

Juego:

Controla la lógica principal del juego.

Variables principales:

Entorno entorno: Administrador de la ventana gráfica.

Isla[] islas: Conjunto de islas en el juego.

Pep pep: Personaje principal.

Gnomo[] gnomos, Tortuga[] tortugas: Enemigos y aliados respectivamente.

ArrayList<BolaDeFuego>: Lista de bolas de fuego activas.

Métodos destacados:

tick(): Actualiza el estado del juego en cada ciclo.

moverPep(): Maneja el movimiento y las acciones de Pep.

crearNuevoGnomo(), crearNuevaTortuga(): Genera nuevas instancias de gnomos y tortugas.

Pep:

Representa al personaje controlado por el jugador.

Métodos principales:

iniciarSalto(), actualizarSalto(Isla[]): Permiten a Pep saltar y respetar la gravedad.

colisionaConTortuga(), colisionaConIsla(): Verifican interacciones con otros elementos.

Gnomo:

Aliados que deben ser rescatados.

Métodos principales:

moverGnomo(), moverGnomoAbajo(): Mueve a los gnomos horizontal y verticalmente.

cambiarDireccionAleatoria(): Modifica su dirección al caer en una isla.

colisionaConTortuga(), colisionaConPep(): Verifica colisiones con tortugas y Pep.

Tortuga:

Enemigos que atacan a los gnomos y a Pep.

Métodos destacados:

apareceTortuga(), mover(): Controlan su aparición y movimiento.

colisionaConBolaDeFuego(): Verifica si una tortuga es golpeada por una bola de fuego.

BolaDeFuego:

Proyectiles lanzados por Pep.

Métodos principales:

moverBolaDeFuego(): Mueve la bola en la dirección indicada.

fueraDePantalla(int anchoPantalla): Verifica si sale del área visible.

Isla:

Elementos fijos del juego donde los personajes pueden posicionarse.

Métodos destacados:

moverIzquierda(), moverDerecha(): Permiten movimiento lateral opcional.

dibujar(Entorno): Dibuja la isla en pantalla.

**Problemas Encontrados y Soluciones:**

Problema de límites de colisión: Uno de los desafíos fue calcular correctamente los límites de las colisiones entre el personaje y la isla, ya que pep debe considerarse “dentro” de una isla si sus límite están dentro de los de la isla.

Solución: Se calcularon los límites izquierdo y derecho tanto del personaje como de la isla usando su posición x y ancho, lo que permitió simplificar la condición de colisión.

Solución: Simplificamos las condiciones intermedias y usamos variables temporales, lo que mejoró la claridad y la eficiencia.

Gestión de imágenes faltantes: Se implementaron rectángulos de colores para representar elementos en caso de errores en la carga de imágenes.

**Implementación**

codigo

**Conclusiones**

En este proyecto, aprendimos cómo estructurar las clases para representar entidades de juego como personajes y plataformas, utilizando técnicas de programación orientada a objetos. Trabajamos en la lógica de colisiones. Además se vio errores comunes en el diseño de juegos, como los problemas de colisiones.