**Introducción - COMECOCOS**

Este documento describe en detalle el proceso de desarrollo de un juego inspirado en Pac-Man, creado en Python con las bibliotecas Pygame y Tkinter. La idea principal es ofrecer una experiencia de juego clásica con mejoras visuales y de jugabilidad. El jugador controla un personaje que recorre un laberinto, recolectando puntos mientras evita a los enemigos. Para ello, hemos desarrollado una inteligencia artificial básica para los enemigos y una interfaz gráfica interactiva que mejora la experiencia del usuario.

**Requisitos del Sistema**

Para ejecutar el juego sin problemas, necesitas:

* **Python (versión 3.x)** como base del desarrollo.
* **Pygame**, para la gestión de gráficos, animaciones y lógica del juego.
* **Tkinter**, para los menús y configuraciones de la interfaz gráfica.
* **Otras bibliotecas necesarias:**
  + random: Para generar movimientos impredecibles en los enemigos.
  + time: Para gestionar pausas y tiempos del juego.
  + sys: Para manejar eventos del sistema y mejorar el rendimiento.

**Estructura del Código**

El código está organizado en varios archivos para facilitar su mantenimiento y escalabilidad:

* main.py: Inicia el juego, maneja eventos y ejecuta el bucle principal.
* laberinto.py: Se encarga de la generación y diseño del laberinto.
* jugador.py: Contiene la clase del personaje principal y sus interacciones.
* enemigos.py: Define la inteligencia artificial y los movimientos de los enemigos.
* interfaz.py: Gestiona los menús y configuraciones usando Tkinter.
* config.py: Almacena configuraciones generales, colores y ajustes de velocidad.

**Diagrama de Clases y Flujo del Juego**

Para entender cómo se relacionan las clases principales, aquí tienes un esquema básico:

+--------------+ +------------+ +-----------+

| Jugador | ----> | Laberinto | ----> | Enemigos |

+--------------+ +------------+ +-----------+

| | |

v v v

Movimiento Colisiones IA de enemigos

**Pasos del Juego**

1. Se carga el laberinto y se inicializan los elementos.
2. Aparece el menú de inicio, donde el jugador puede configurar opciones como la dificultad y el sonido.
3. El jugador se mueve con las teclas de dirección.
4. Los enemigos patrullan el laberinto siguiendo un algoritmo de IA.
5. Se detectan colisiones entre el jugador, los enemigos y los límites del laberinto.
6. Si el jugador pierde todas sus vidas, aparece una pantalla de "Game Over".
7. Si recolecta todos los puntos, avanza al siguiente nivel o gana la partida.

**Funciones Clave y Lógica del Juego**

Algunas de las funciones más importantes en el código son:

* mover\_jugador(): Controla el movimiento del personaje y evita que atraviese paredes.
* detectar\_colisiones(): Comprueba si el jugador choca con un enemigo o un muro.
* mover\_enemigos(): Define la estrategia de movimiento de los enemigos.
* actualizar\_pantalla(): Redibuja todos los elementos del juego en cada fotograma.
* mostrar\_menu(): Muestra las opciones de configuración antes de empezar.
* reiniciar\_juego(): Permite reiniciar la partida si el jugador pierde.
* manejar\_eventos(): Captura las acciones del usuario (teclas, cierre de ventana, etc.).
* dibujar\_laberinto(): Dibuja el laberinto en pantalla.

**Pruebas y Validación del Juego**

Para asegurar que todo funcione correctamente, se realizaron diferentes pruebas:

* **Movimiento:** Se verificó que el personaje se desplace con fluidez.
* **Colisiones:** Se probó que los choques con muros y enemigos se detecten correctamente.
* **Inteligencia Artificial:** Se ajustó el comportamiento de los enemigos.
* **Interfaz Gráfica:** Se revisó la funcionalidad de los menús y configuraciones.
* **Rendimiento:** Se optimizó la velocidad del juego para evitar ralentizaciones.
* **Escalabilidad:** Se probó el juego en distintas resoluciones de pantalla.
* **Sonido:** Se revisó que los efectos sonoros estén sincronizados con las acciones.

**Mejoras Futuras y Expansión**

Para seguir mejorando la experiencia de juego, se tienen planeadas varias actualizaciones:

* **Nuevos niveles:** Laberintos con mayor dificultad y nuevas mecánicas.
* **Mejora en la IA de los enemigos:** Un algoritmo más avanzado para hacerlos más estratégicos.
* **Gráficos y sonido mejorados:** Animaciones más fluidas y efectos envolventes.
* **Modo multijugador:** Posibilidad de jugar en línea o en cooperativo local.
* **Power-ups:** Objetos que otorguen ventajas temporales al jugador.
* **Sistema de puntuación:** Registro de récords para fomentar la competencia.
* **Compatibilidad con móviles:** Adaptación del juego para pantallas táctiles.