



***FACULTAD DE INGENIERÍA CARRERA DE INGENIERÍA
INFORMÁTICA FACULTAD COMUNITARIA DE
CAACUPÉ***

OPTATIVA I

Tema: Desarrollo de un juego con Python

Alumno:

- Ignacio David Franco Espínola

Profesor: Ing. Ricardo Maidana.

Curso:

- Quinto año, Noveno semestre

Turno:

- Noche

Caacupé – Paraguay Año:2024

Introducción

El desarrollo de juegos es una disciplina que combina creatividad, programación y diseño para crear experiencias interactivas que entretienen y desafían a los jugadores. Pygame, una biblioteca de Python diseñada específicamente para el desarrollo de juegos, ofrece una plataforma accesible y flexible para principiantes y desarrolladores experimentados. En este proyecto, crearemos un juego básico de Buscaminas utilizando Pygame.

El objetivo de este proyecto es desarrollar un juego en el que el jugador pueda descubrir todas las celdas sin minas en un tablero. Aunque es un proyecto sencillo, proporciona una excelente introducción a los conceptos fundamentales del desarrollo de juegos.

¿Por qué elegir un juego de Buscaminas?

Elegí este juego para implementar los conocimientos aprendidos durante las clases y ponerlos en práctica de una manera efectiva. Un juego de Buscaminas es ideal porque, a pesar de su simplicidad, involucra una lógica de juego interesante y desafiante que es perfecta para principiantes.

¿Qué herramientas elegiste y por qué?

La herramienta utilizada es Pygame, una biblioteca muy popular que proporciona una amplia gama de funcionalidades para gráficos, sonido y manejo de eventos, lo que la hace fácil de aprender y ejecutar. Estas características la convierten en una excelente opción para el desarrollo de este proyecto.

Ventajas de las herramientas

Algunas ventajas de Pygame son:

- Fácil de usar y aprender.
- Accesible a varios lenguajes de programación.
- Eficaz en la carga y manipulación de imágenes.
- Amplio soporte de la comunidad.

Descripción y representación gráfica del juego

En este juego, los jugadores deben descubrir todas las celdas sin minas en un tablero. El tablero se representa con una cuadrícula de celdas que pueden ser descubiertas al hacer clic en ellas. Algunas características del juego incluyen:

1. **Tablero Dinámico:** El juego contará con un tablero dinámico donde las minas se colocan aleatoriamente en cada partida.
2. **Modos de Dificultad:** Se ofrecerán diferentes modos de dificultad (fácil, medio, difícil) para mantener el desafío.
3. **Interfaz Intuitiva:** La interfaz gráfica será sencilla e intuitiva, mostrando el número de minas adyacentes a cada celda descubierta.
4. **Reinicio Rápido:** Los jugadores podrán reiniciar el juego rápidamente después de una derrota o victoria.

Objetivos del Juego

1. Descubrir todas las celdas sin minas para ganar el juego.
2. Evitar hacer clic en las celdas con minas.
3. Utilizar la lógica para deducir la ubicación de las minas basándose en los números mostrados en las celdas adyacentes.

Historia

Pygame es una biblioteca que permite el desarrollo de videojuegos en Python. Su historia es interesante y destaca su evolución desde un proyecto personal hasta convertirse en una herramienta popular en la comunidad de desarrollo de juegos. Pygame fue creado por Pete Shinnars y lanzado por primera vez el 28 de octubre de 2000. Shinnars quería crear una biblioteca que facilitara la creación de juegos en Python, inspirándose en la popular biblioteca SDL (Simple DirectMedia Layer).

Historia del Buscaminas

El juego de Buscaminas, conocido en inglés como Minesweeper, es un clásico de los videojuegos de lógica y estrategia. Su origen se remonta a la década de 1960, pero ganó una popularidad masiva a partir de los años 1990 con su inclusión en los sistemas operativos de Microsoft Windows.

Documento explicativo del código

Requisitos

- Python 3.x
- Juego Py

Estructura del código

Importación de bibliotecas

- import pygame
- import random
- import sys

Variables Globales

- **primera_celda_pulsada**: Indica si la primera celda ha sido pulsada.
- **calidades_casilla**: Diccionario que almacena información sobre cada celda.
- **bombas_restantes**: Número de bombas que quedan por marcar.
- **bombas_correctas**: Número de bombas marcadas correctamente.
- **puntos**: Puntuación del jugador.

Funciones

crear_cuadrícula(numero, tamaño_cuadrícula)

Crea la cuadrícula del juego.

- **Args:**
 - numero (int): Número de filas y columnas de la cuadrícula.
 - tamaño_cuadrícula (int): Tamaño de cada celda de la cuadrícula.

crear_bombas_random(numero, primer_click, dificultad)

Coloca bombas de forma aleatoria en la cuadrícula, asegurando que no se coloquen alrededor del primer clic del jugador.

- **Args:**
 - numero (int): Número de filas y columnas de la cuadrícula.
 - primer_click (tuple): Índice (fila, columna) del primer clic del jugador.
 - dificultad (float): Factor de dificultad que determina el número de bombas.

donde_ahy_bombas()

Calcula el número de bombas adyacentes a cada celda de la cuadrícula.

color_numero(element, color)

Dibuja el número de bombas adyacentes en una celda específica.

- **Args:**
 - element (int): Número de celda.
 - color (tuple): Color del texto.

click_casilla(numero, event, dificultad)

Maneja los clics del ratón en las celdas del juego.

- **Args:**
 - numero (int): Número de filas y columnas de la cuadrícula.
 - event (pygame.event): Evento del clic del ratón.
 - dificultad (float): Factor de dificultad que determina el número de bombas.
- **Returns:**
 - tuple: Indicadores de victoria y derrota.

menu()

Muestra el menú principal del juego y permite seleccionar la dificultad.

- **Returns:**
 - tuple: Número de filas/columnas, dificultad y tamaño de celdas.

mostrar_bombas(numero, casillas)

Muestra el número de bombas restantes en la interfaz.

- **Args:**
 - numero (int): Número de filas y columnas de la cuadrícula.
 - casillas (int): Tamaño de cada celda.

mostrar_puntos()

Muestra los puntos actuales en la interfaz.

game(numero, dificultad, casillas)

Maneja el ciclo principal del juego.

- **Args:**
 - numero (int): Número de filas y columnas de la cuadrícula.
 - dificultad (float): Factor de dificultad que determina el número de bombas.
 - casillas (int): Tamaño de cada celda.

run()

Inicia el juego y gestiona el menú y la cuadrícula inicial.

Estructura del Juego

1. **Inicio:** El juego inicia llamando a la función run(), que muestra el menú y permite seleccionar la dificultad.
2. **Menú:** La función menu() permite al jugador seleccionar la dificultad del juego.
3. **Creación de la Cuadrícula:** Basado en la selección del jugador, se llama a crear_cuadrícula().
4. **Juego Principal:** El ciclo principal del juego se maneja en la función game(), que llama a click_casilla() para manejar los clics del ratón.
5. **Mostrar Bombas y Puntos:** Durante el juego, mostrar_bombas() y mostrar_puntos() actualizan la interfaz con el número de bombas restantes y los puntos del jugador.
6. **Final del Juego:** El juego termina cuando el jugador gana o pierde, y se reinicia llamando a run() nuevamente.

Consideraciones de Dificultad

La dificultad del juego se ajusta mediante el factor dificultad, que determina el número de bombas en la cuadrícula:

- **Fácil:** Menos bombas.
- **Normal:** Número moderado de bombas.
- **Difícil:** Más bombas.

Ejecución del Código

Para ejecutar el código, se debe tener instalada la biblioteca Pygame. El script se puede ejecutar directamente desde la línea de comandos o un entorno de desarrollo Python.

Conclusión

El desarrollo del juego de Buscaminas con Python utilizando Pygame trata de establecer una base sólida y funcional que puede servir como punto de partida para un proyecto más complejo y detallado. Este trabajo proporciona una base robusta para un juego específico de Buscaminas con Pygame, cubriendo conceptos fundamentales de la programación de juegos, incluyendo el manejo de eventos, la actualización de gráficos y el control del jugador. Este proyecto, aunque básico, ofrece un excelente punto de partida para desarrollar un juego más complejo y completo. Con futuras mejoras y expansiones, este juego puede evolucionar significativamente, proporcionando una experiencia de juego rica y entretenida.

Anexo

