



EXAMEN FINAL PYTHON

Autor

Julio De Jesús Rojas Martínez

Docente

Ing. Ricardo Alberto Maidana Giménez

**Caacupé – Paraguay
2024**

Índice

| | |
|-------------------------------------|----|
| RESUMEN..... | 3 |
| INTRODUCCIÓN | 4 |
| OBJETIVOS | 5 |
| 1.1. Objetivo General | 5 |
| 1.2. Objetivos Específicos | 5 |
| 1.3. JUSTIFICACIÓN | 6 |
| CAPITULO II – MARCO TEÓRICO | 7 |
| 2. Historia del Juego | 7 |
| 3. Uso del Juego | 7 |
| 4. GitHub y Su Repositorio | 8 |
| 5. Observaciones y Mejoras | 8 |
| CONCLUSIÓN | 10 |
| Referencias | 11 |

RESUMEN

El "Juego de Carreras" es un proyecto de desarrollo de videojuegos creado con Python y la biblioteca Pygame. Este proyecto tiene como objetivo principal permitir a los desarrolladores principiantes adquirir experiencia práctica en la creación de videojuegos, abarcando aspectos esenciales como gráficos, detección de colisiones, gestión de eventos y sonido. El juego simula una carrera en la que el jugador controla un auto, evita obstáculos y trata de obtener la mayor puntuación posible en un tiempo limitado. El desarrollo del juego incluye la configuración de elementos visuales y sonoros, la implementación de la lógica del juego, y la gestión de la interfaz de usuario para ofrecer una experiencia interactiva y entretenida. A través de este proyecto, se exploran y aplican técnicas fundamentales del desarrollo de videojuegos, con un enfoque en la práctica y la optimización del código.

INTRODUCCIÓN

El Juego de Carreras se creó como un proyecto de aprendizaje para permitirme practicar el desarrollo de videojuegos utilizan Do Python y la librería Pygame. El objetivo de este proyecto era poner en práctica los conocimientos adquiridos en la creación de gráficos, la colisión, gestión de eventos y sonido, que son algunos de los conceptos fundamentales detrás del desarrollo de videojuegos. Python fue elegido como el lenguaje de programación debido a su simplicidad y su extensa gama de bibliotecas y herramientas, mientras que Pygame se seleccionó por ser una biblioteca específica para la creación de videojuegos que facilita estas tareas.

OBJETIVOS

1.1.Objetivo General

1.1.1. Desarrollar un videojuego de carreras utilizando Python y la biblioteca Pygame para aplicar conocimientos adquiridos en programación y diseño de videojuegos, creando una experiencia interactiva y divertida que integre gráficos, sonidos y mecánicas de juego básicas.

1.2.Objetivos Específicos

1.2.1. Implementar Mecánicas de Juego: Diseñar e implementar las mecánicas básicas del juego, como el movimiento del automóvil, la detección de colisiones con obstáculos y el sistema de puntuación, para ofrecer una experiencia de juego fluida y desafiante.

1.2.2. Desarrollar Gráficos y Sonidos: Crear gráficos simples y agregar efectos de sonido que enriquezcan la experiencia del jugador, asegurando que el juego sea visualmente atractivo y auditivamente agradable.

1.2.3. Realizar Pruebas y Optimización: Ejecutar pruebas exhaustivas del juego para identificar y corregir errores, optimizar el rendimiento y garantizar que el juego funcione de manera estable en diferentes plataformas.

1.3.JUSTIFICACIÓN

El desarrollo de videojuegos es una disciplina que combina creatividad con habilidades técnicas, ofreciendo una experiencia educativa integral. En este proyecto, la creación del Juego de Carreras con Python y Pygame se justifica por varias razones:

Desarrollo de Habilidades Técnicas: Trabajar con Pygame permite a los desarrolladores novatos aprender y aplicar conceptos clave en el desarrollo de videojuegos, como la gestión de eventos, la detección de colisiones y la creación de gráficos. Python, siendo un lenguaje fácil de aprender, proporciona una base sólida para adquirir habilidades en programación orientada a objetos, manejo de bibliotecas y desarrollo de algoritmos.

Creatividad y Diseño: La creación de un juego fomenta la creatividad al permitir el diseño de mecánicas de juego, la implementación de gráficos y la integración de sonido. Este proyecto ofrece una oportunidad para experimentar con ideas y conceptos en un entorno controlado, lo que es fundamental para el crecimiento como desarrollador.

Prototipado Rápido y Iteración: Pygame es ideal para prototipar ideas de juegos rápidamente y realizar iteraciones. La facilidad de uso de Pygame y la rapidez con la que se pueden implementar cambios facilitan la experimentación y mejora continua del juego, permitiendo a los desarrolladores aprender de manera iterativa.

CAPITULO II – MARCO TEÓRICO

Pygame es una biblioteca de Python que facilita el desarrollo de videojuegos 2D al proporcionar herramientas para la gestión de gráficos, sonidos y eventos. En tu código, utilizas Pygame para:

Inicialización y Configuración: Configuras la pantalla del juego, defines colores y cargas imágenes para el auto, los obstáculos y el fondo. La función `pygame.display.set_mode()` establece el tamaño de la ventana del juego, y `pygame.display.set_caption()` define el título de la ventana.

Manejo de Imágenes: Usas `pygame.image.load()` para cargar imágenes y `pygame.transform.scale()` para ajustar su tamaño. Esto te permite tener gráficos apropiados para el juego y asegurar que se ajusten al diseño deseado.

Control de Eventos: Implementas un bucle de eventos para manejar las acciones del usuario, como el movimiento del auto y la interacción con los botones. La detección de eventos como `pygame.QUIT` y `pygame.MOUSEBUTTONDOWN` permite responder a las acciones del jugador y gestionar el flujo del juego.

Actualización de Pantalla: Usas `pygame.display.flip()` para actualizar la pantalla con los gráficos más recientes. Esto es esencial para que los cambios en el juego se reflejen en la interfaz de usuario.

Gestión del Tiempo y Velocidad: El reloj (`pygame.time.Clock()`) controla la velocidad de actualización del juego. Ajustas la velocidad de movimiento de los obstáculos y el auto, así como la densidad de los obstáculos, para que el juego se vuelva progresivamente más desafiante.

Sonidos: Cargas y reproduces sonidos usando `pygame.mixer.Sound()`. Esto enriquece la experiencia del jugador al añadir efectos sonoros para colisiones y el motor del auto.

2. Historia del Juego

Inspirado en los juegos de carreras de arcade de los años 80 y 90, el juego busca capturar la emoción de las carreras con un auto en una carretera llena de obstáculos.

3. Uso del Juego

El jugador mueve el auto con las teclas de flecha izquierda y derecha, evitando obstáculos y tratando de obtener la mayor puntuación posible antes de que se agote el tiempo.

4. GitHub y Su Repositorio

Tu código puede beneficiarse enormemente del uso de GitHub para gestionar el desarrollo. Aquí hay algunas prácticas recomendadas relacionadas con GitHub:

Control de Versiones: Usa Git para gestionar y registrar cambios en tu código fuente. Esto te permite volver a versiones anteriores si es necesario, colaborar con otros desarrolladores y mantener un historial detallado de las modificaciones.

Repositorios: Crea un repositorio en GitHub para almacenar tu código fuente, recursos del juego (como imágenes y sonidos) y documentación. Esto facilita la organización y el acceso al proyecto.

Branching y Merging: Utiliza ramas para desarrollar nuevas características o hacer cambios importantes sin afectar la versión principal del juego. Una vez que los cambios están listos, puedes fusionarlos con la rama principal mediante pull requests.

Documentación: Incluye un archivo README en tu repositorio para proporcionar una descripción general del juego, instrucciones para ejecutarlo y cualquier otra información relevante. Esto es útil tanto para ti como para otros posibles colaboradores.

Colaboración: Si trabajas en equipo, GitHub facilita la colaboración mediante pull requests, revisiones de código y la gestión de issues. Esto asegura que todos los miembros del equipo estén alineados y puedan contribuir eficazmente al desarrollo del juego.

5. Observaciones y Mejoras

Aquí hay algunas observaciones y sugerencias para mejorar tu código:

Optimización del Movimiento del Auto: En lugar de usar valores fijos para el movimiento del auto (`auto_x -= 5` y `auto_x += 5`), considera usar una variable que represente la velocidad del auto. Esto te permitirá ajustar la velocidad de movimiento de manera más flexible y dinámica.

Gestión de Recursos: Carga las imágenes y sonidos solo una vez en lugar de hacerlo repetidamente en el bucle principal. Esto mejorará la eficiencia del juego.

Manejo de Colisiones: Puedes agregar una función específica para manejar las colisiones que incluya lógica para reducir la puntuación, detener el juego, etc.

Multijugador: La implementación del modo multijugador es una buena adición. Asegúrate de que el cambio de turno y la gestión de puntuaciones estén bien definidos para evitar errores durante el juego.

Comentarios y Documentación: Asegúrate de documentar bien tu código con comentarios que expliquen la funcionalidad de cada función y bloque de código. Esto ayudará a ti y a otros a comprender el código más fácilmente.

Interfaz de Usuario: Considera mejorar la interfaz de usuario con gráficos y botones más atractivos. Esto puede incluir un menú de inicio más detallado y opciones adicionales de configuración.

CONCLUSIÓN

En general, el desarrollo del Juego de Carreras no solo permitió familiarizarse con técnicas básicas de desarrollo de videojuegos, sino que también añadió las ventajas de utilizar Pygame para tales proyectos. Pygame no solo es fácil de entender y tiene documentación extensiva y una comunidad de soporte, sino que también es compatible con muchas plataformas y ha facilitado el rápido desarrollo de prototipos. A lo largo de todo el proceso, hubo una serie de etapas esenciales, desde la planificación y el diseño hasta la implementación, las pruebas, algunas fases de optimización y esfuerzos de remodelación. Al final, el juego Juego de Carreras se inspiró en varios de los clásicos del género. Este proyecto buscó capturar la emoción de las carreras y presentar un desafío atractivo para los jugadores, consolidando así una sólida base en el desarrollo de juegos con Python y Pygame.

Referencias

- [1] [En línea]. Available: <https://www.python.org/downloads/>.
 - [2] [En línea]. Available: <https://www.pygame.org/docs/>.
 - [3] [En línea]. Available: <https://github.com/>.
 - [4] [En línea]. Available: <https://codigonautas.com/que-es-python/#:>.
-