

UNIVERSIDAD NACIONAL DE INGENIERIA

Facultad de Ingeniería Eléctrica y Electrónica - FIEE



Curso: Programación Orientada a Objetos (BMA15)

SkyLine Legends: Desarrollo de Videojuego en Python

Integrantes:

- Elvis Orlando García Mendoza

Docente:

Mg. Ing. Washington Ernesto Loayza Mejía

Ciclo: 2025 - I

Lugar: Lima – Perú

Índice

1. Introducción
2. Objetivos del proyecto
 - a) Objetivo general
 - b) Objetivos específicos
3. Proyectos similares
4. Diagrama UML
5. Clases y métodos principales
6. Características Técnicas del Juego
7. Ventajas, Limitaciones y Mejoras Propuestas
8. Conclusiones
9. Bibliografía

1) Introducción

El presente documento describe el desarrollo del videojuego SkyLine Legends, una propuesta interactiva realizada en Python utilizando la librería Pygame. Este proyecto busca simular una experiencia de conducción tipo arcade con elementos como colisiones, sonidos realistas, vidas, sistema de puntuación y menús interactivos.

SkyLine Legends destaca por la implementación de Programación Orientada a Objetos (POO), el uso de recursos visuales y sonoros, y el enfoque modular que permite una escalabilidad futura. Su estructura permite que los elementos del juego (vehículos, obstáculos, entorno) se gestionen de forma independiente y reutilizable.

2) Objetivos del proyecto

a) Objetivo General:

Desarrollar un videojuego tipo arcade en Python utilizando Pygame, que aplique los principios de la programación orientada a objetos, fomentando habilidades en desarrollo de videojuegos y lógica de simulación.

b) Objetivos Específicos:

- Implementar clases independientes para cada entidad del juego.
- Utilizar herencia y encapsulamiento para mantener el código limpio y modular.
- Diseñar un sistema de vidas y colisiones.
- Incorporar sonidos de motor, colisiones y curación.
- Diseñar una interfaz inicial con menús funcionales y un sistema de puntuación.
- Integrar una intro animada con frames y audio.
- Incluir control de música con selección y pausa.
- Controlar la dificultad mediante velocidad y generación de obstáculos.

3) Proyectos similares

a) Road Fighter:

Clásico juego de carreras publicado por Konami en 1984. El jugador conduce un auto en una carretera con tráfico, evitando choques para llegar al final. Se basa en mecánicas simples pero desafiantes.

Características:

- Perspectiva en tercera persona.
- Obstáculos en movimiento lineal.
- Sistema de puntuación y vidas.

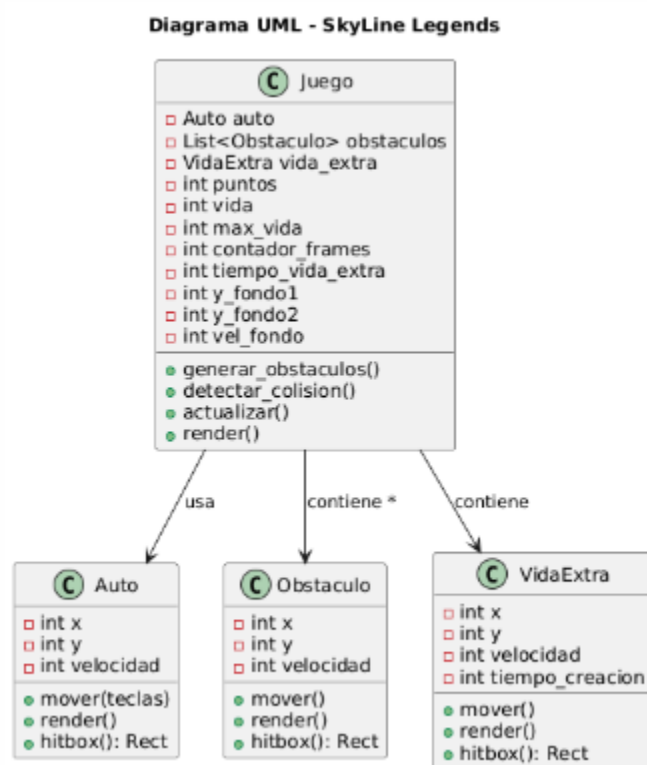
b) Highway Racer:

Juego de carreras para dispositivos móviles que presenta tráfico dinámico y escenarios cambiantes. El jugador obtiene puntos mientras más tiempo evite colisiones a alta velocidad.

Características:

- Diferentes modos de juego.
- Movimiento lateral y scroll vertical.
- Elementos coleccionables.

4) Diagrama uml



5) Clases y métodos principales

Clase Auto:

Representa al jugador. Se mueve lateralmente dentro de la carretera.

Métodos:

- mover(teclas): Controla movimiento con flechas.
- render(): Dibuja el auto.
- hitbox(): Retorna el área de colisión.

Clase Obstaculo:

Autos enemigos que descienden en pantalla.

Métodos:

- mover(): Baja el obstáculo.
- render(): Dibuja el enemigo.
- hitbox(): Detecta colisión.

Clase VidaExtra:

Objetos coleccionables para recuperar vidas.

Métodos: mover(), render(), hitbox()

Clase Juego:

Controla toda la lógica del juego.

Funciones internas:


- generar_obstaculos()
- detectar_colision()
- actualizar()
- render()

Variables: vidas, puntos, scroll de fondo, contador de frames, vida_extra.

Funciones externas:

- main(): Ciclo principal.
- mostrar_menu_inicio(): Menú principal interactivo.
- mostrar_game_over(): Pantalla de derrota.
- reproducir_intro(): Reproduce intro desde frames + audio.

6) Características Técnicas del Juego

- Control lateral por teclado con eventos `pygame.KEYDOWN` y `KEY.get_pressed()`.
- Vehículos enemigos generados con `random.randint()` y temporizador de frames.
- Scroll vertical infinito para simular movimiento real del fondo.
- Sistema de sonido completo: motor, choque, curación, y música de fondo.
- Sistema de vida visual y funcional, con barra de vida y objetos curativos.
- Control de música: cambio con teclas `n`, `b`, y pausa con `p`.
- Detección de colisiones con `pygame.Rect.colliderect()`.
- Puntaje acumulativo mostrado en tiempo real.
- Intro animada que recorre frames `.png` con un archivo `.wav` sincronizado.
-  Estructura modular del código y recursos externos.

7) Ventajas, Limitaciones y Mejoras Propuestas

Ventajas:

- Modularidad clara (cada clase con responsabilidades únicas).
- Recursos sonoros y visuales bien integrados.
- Interfaz de usuario atractiva y funcional.
- Código mantenible y ampliable.

Limitaciones:

- No guarda puntajes en archivos ni base de datos.
- Modo carrera externo no documentado.
- Sin niveles de dificultad progresiva.
- Colisiones solo con el jugador, no entre enemigos.

Mejoras Propuestas:

- Incluir sistema de niveles y enemigos más complejos.
- Añadir efectos visuales (partículas, explosiones).
- Guardar puntuaciones y crear tabla de rankings.
- Implementar sistema de logros y recompensas.
- Desarrollar modo multijugador local o en red.
- Mayor personalización de música y vehículos.

8) Conclusiones

El desarrollo de SkyLine Legends ha permitido reforzar los conceptos fundamentales de la Programación Orientada a Objetos mediante una aplicación práctica y motivadora. A través de la separación de clases, el uso de eventos de teclado, sonidos, menús y scroll, se logró un videojuego funcional y extensible.

La organización modular del proyecto permite futuras expansiones como:

- Más escenarios y niveles.
- Dificultades variables.
- Registro de puntuaciones.
- Integración en red para multijugador.

Este proyecto también promueve habilidades de depuración, trabajo en equipo y creatividad en el desarrollo de software.

9) Bibliografía

- Sweigart, A. (2015). Invent Your Own Computer Games with Python. No Starch Press.
- Van Rossum, G., & Drake Jr, F. L. (2009). The Python Language Reference Manual. Network Theory Ltd.
- Documentation - Pygame Library: <https://www.pygame.org/docs/>
- Real Python: <https://realpython.com/pygame-a-primer/>
- Python Software Foundation. <https://www.python.org/>