

DISEÑO DE UN VIDEOJUEGO DE COMBATE DE NAVES UTILIZANDO LENGUAJE PYTHON

Rodrigo López Soto

Joselyn Barquero Castillo

Noel Blandón Saborio

INTRODUCCIÓN

Se realiza un juego de naves espaciales en el lenguaje de programación Python mediante el uso de la librería Pygame, donde el jugador se enfrenta a naves enemigas. Mediante una interfaz gráfica el jugador tiene control de la nave, la cual se puede desplazar horizontalmente. El usuario cuenta con un arma (láser), que dispara para enfrentarse a las naves enemigas, el usuario pierde cuando las naves contrarias lo impactan directamente.

OBJETIVO GENERAL

Realizar un juego de naves espaciales en el lenguaje de programación Python mediante el uso de la librería pygame, el juego consiste en una serie de niveles en los cuales el jugador se enfrenta a naves enemigas, mediante una interfaz gráfica el jugador tiene control de la nave la cual se puede desplazar horizontal y verticalmente mediante el uso del teclado.

OBJETIVO ESPECÍFICO 1

- Investigar características y herramientas de la librería Pygame de Python, instalación de Pygame.

DISEÑO Y VALIDACIÓN DEL OBJETIVO ESPECÍFICO 1

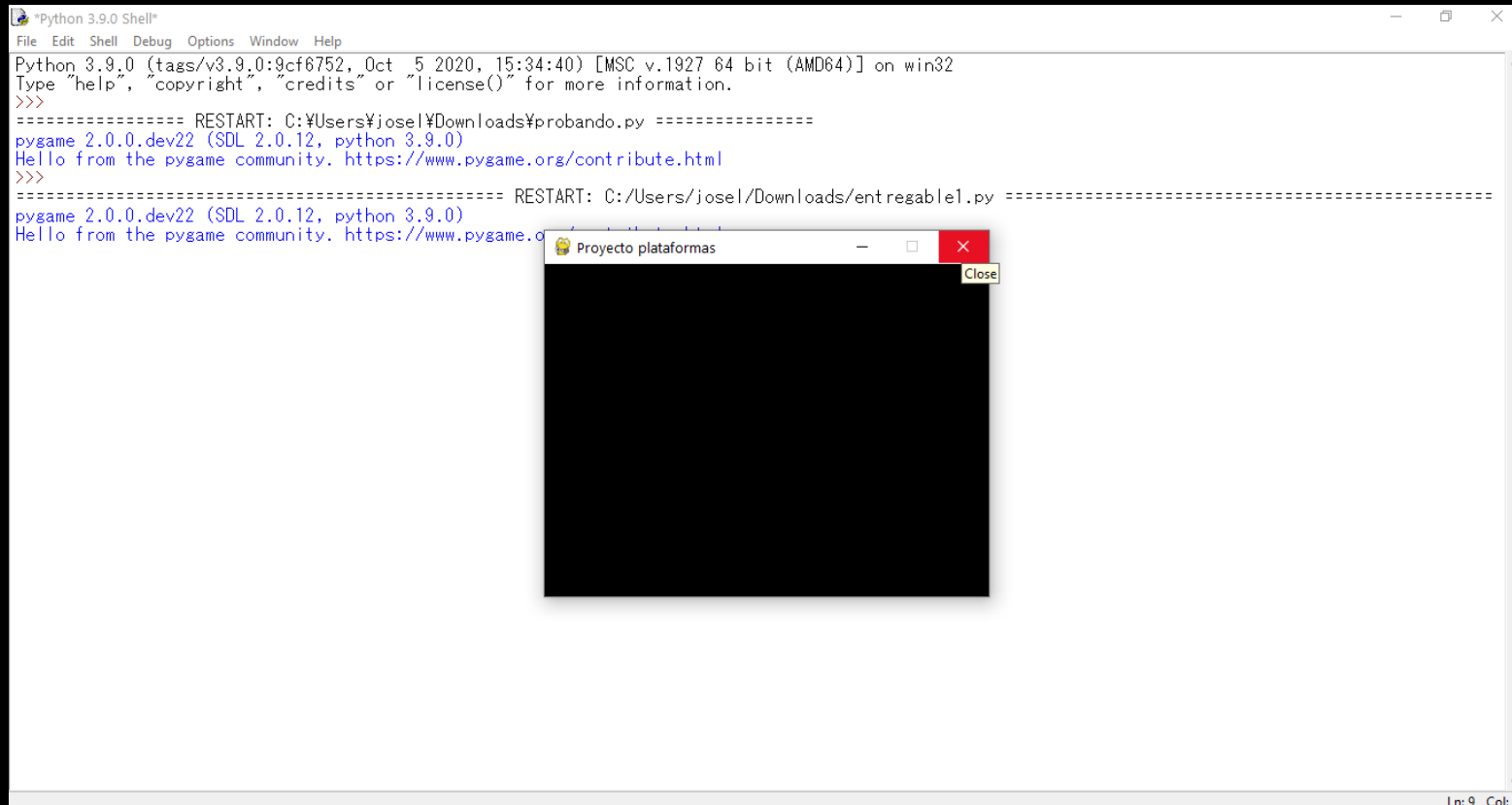
- Para la instalación de Pygame en Linux, basta usar el comando en terminal `sudo apt-get install python-pygame`. Para la instalación en Windows, desde el command prompt se utiliza el comando `pip install pygame`.
- Algunos módulos de relevancia en Pygame:
 - ☐ `display`: controlar la ventana de visualización o la pantalla.
 - ☐ `draw`: dibujar formas simples en una superficie.
 - ☐ `image`: guardar y cargar imágenes.
 - ☐ `joystick`: administrar dispositivos de joystick.
 - ☐ `key`: administrar el teclado.
 - ☐ `mouse`: administrar el mouse.

OBJETIVO ESPECÍFICO 2

- Programar una función principal del juego naves especial.

DISEÑO Y VALIDACIÓN DEL OBJETIVO ESPECÍFICO 2

- Crear una ventana de visualización, con ayuda del módulo display.



OBJETIVO ESPECÍFICO 3

- Programar una función que permita crear el objeto (nave espacial) además de implementar el movimiento de la misma.

DISEÑO Y VALIDACIÓN DEL OBJETIVO ESPECÍFICO 3

- Clases para la creación de objetos.
- Módulo .key para manejar los movimientos y disparos de la nave.

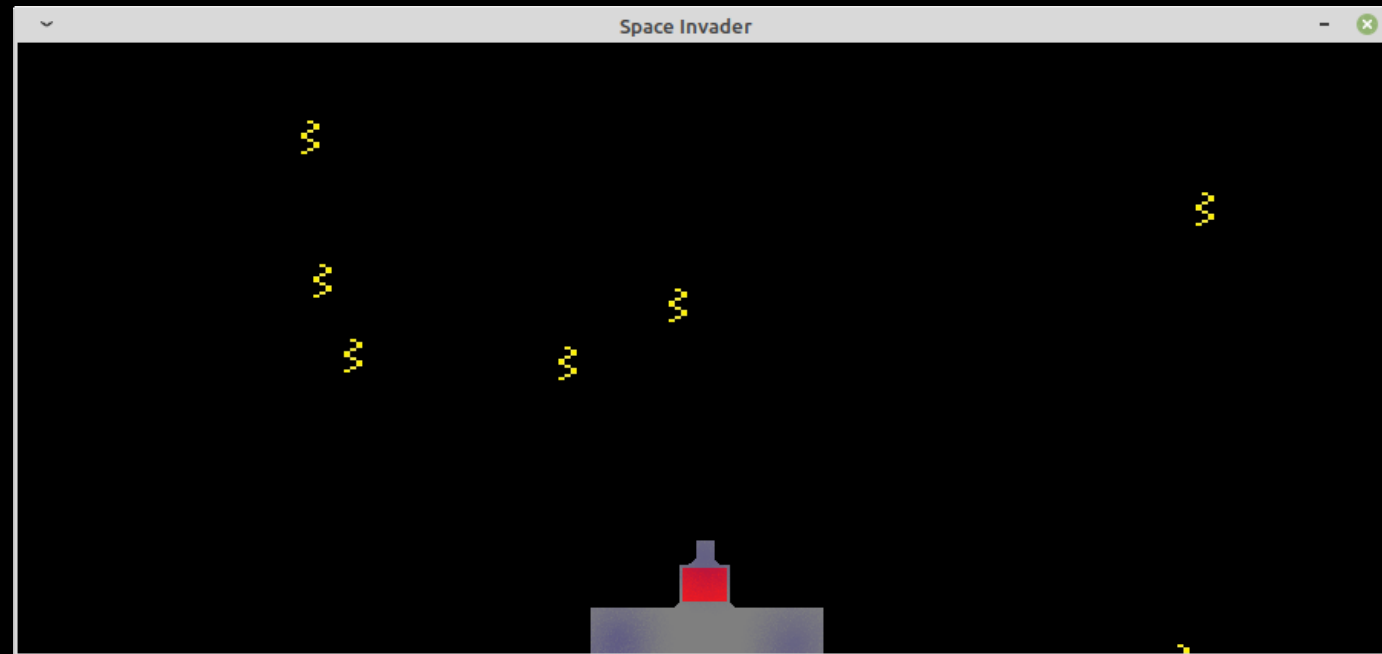


OBJETIVO ESPECÍFICO 4

- Programar una función que adhiera armas (láser) al objeto además de aplicar restricciones del movimiento al objeto.

DISEÑO Y VALIDACIÓN DEL OBJETIVO ESPECÍFICO 4

- Se definen los atributos del objeto láser, similar a la clase del objeto nave.



OBJETIVO ESPECÍFICO 5

- Programar función que permita incluir distintos niveles al juego con distintas configuraciones de enemigos.

DISEÑO Y VALIDACIÓN DEL OBJETIVO ESPECÍFICO 5

- Frecuencia a la que disparan las naves enemigas
- Cantidad de naves enemigas
- A mayor dificultad hay más naves y disparan más rápidamente
- Se implementaron 3 niveles de dificultad

OBJETIVO ESPECÍFICO 6

- Programar función que permita incluir distintas pistas musicales a cada uno de los niveles del juego.

DISEÑO Y VALIDACIÓN DEL OBJETIVO ESPECÍFICO 6

- Una pista musical por cada nivel
- Sonido de fondo, explosiones y disparos
- Pygame.mixer
- Mp3 y wav

OBJETIVO ESPECÍFICO 7

- Diseñar y programar una función que adhiera un menú interactivo para el usuario.

DISEÑO Y VALIDACIÓN DEL OBJETIVO ESPECÍFICO 7

- Un "boton " o sprite para cada seccion del menu
- Sonido de fondo para la introduccion al menu y sus secciones
- Previsualizacion de las diferentes naves a controlar

OBJETIVO ESPECÍFICO 8

- Diseñar y programar una función que permita guardar el avance de las distintas partidas realizadas en el juego.

DISEÑO Y VALIDACIÓN DEL OBJETIVO ESPECÍFICO 8

- Guardado en archivo txt
- Anexado al puntaje esta el tipo de nave usada en cada partida

CONCLUSIONES

- Se logró programar un videojuego de combate de naves en 2D en lenguaje de programación Python utilizando Pygame.
- Se pudieron añadir pistas musicales de fondo al videojuego utilizando funciones de Pygame, particularmente `pygame.mixer.music`.
- Seleccionó el lenguaje de programación Python para el videojuego.
- Se añadieron sonidos de disparo utilizando formato `.wav` con `pygame.mixer.sound`. Se utilizó `.wav` para que no interfiriera con la música de fondo que está en formato `mp3`.
- Uso de distintos módulos de Pygame, tales como `sys`, `display`.
- Implementación de otros niveles cambiando la dificultad según la nave elegida.
- Se crearon funciones y clases con ayuda de los módulos que ofrece Pygame. Para la funcionalidad de las mismas se tomaron en cuenta la cantidad de enemigos y el tiempo entre disparos.
- Implementación de archivos multimedia, por ejemplo, con ayuda del módulo `pygame.mixer.music`, permitiendo controlar el audio que se le pasa como parámetro.