

Universidad Rafael Landívar

Campus de Quetzaltenango

Facultad de Ingeniería

Análisis y diseño 2



## Reporte #1 Proyecto

Jose Daniel Az Rios - 1508620

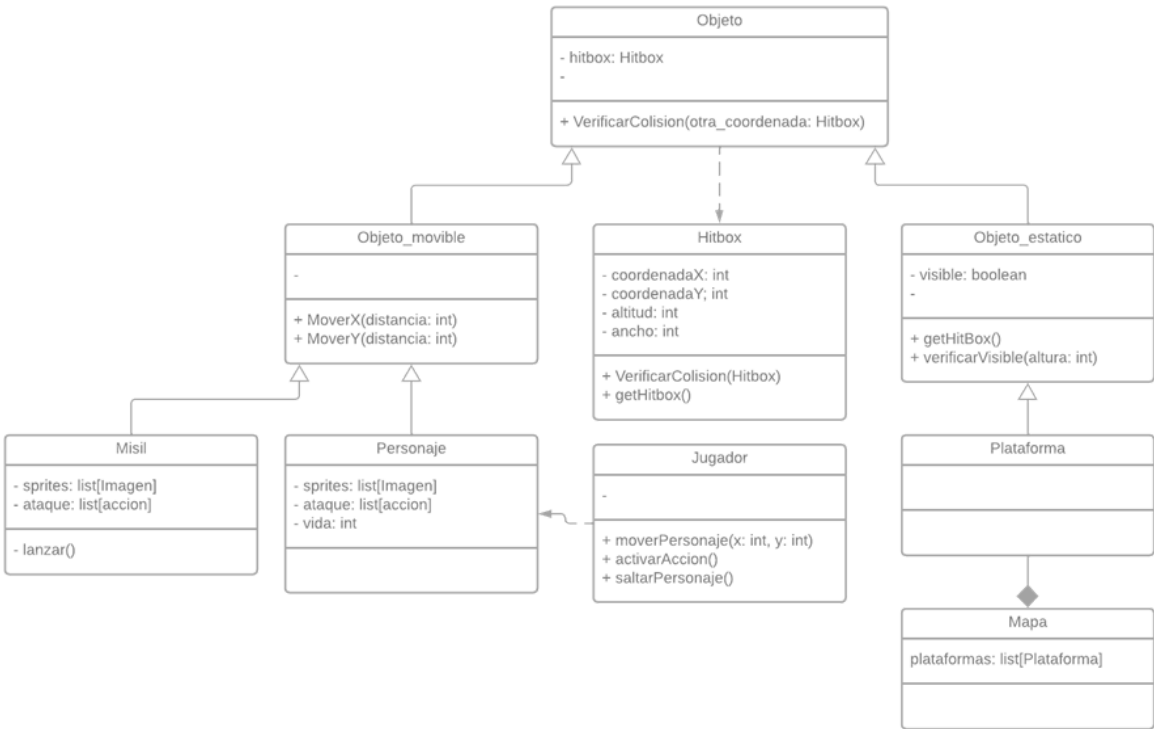
Carlos Estuardo Cancinos Vasquez - 1621620

Pablo David Puac García - 1518320

Rodolfo B. Chivalan de León - 1561817

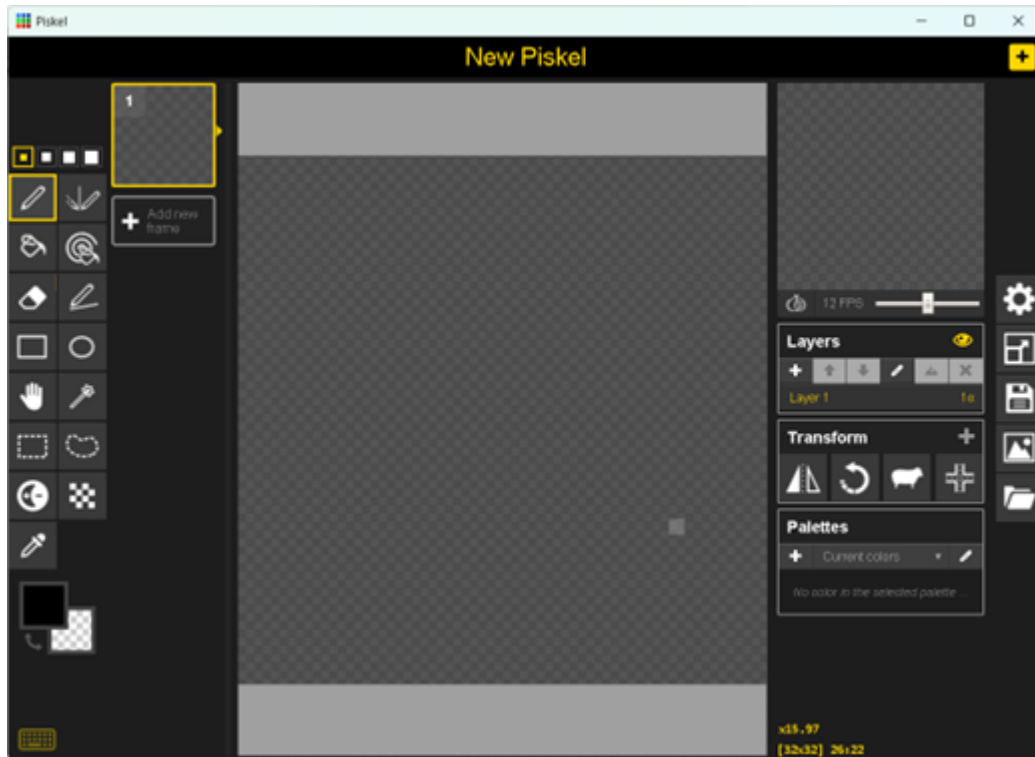
5 de Junio de 2023

Diagrama de clases inicial

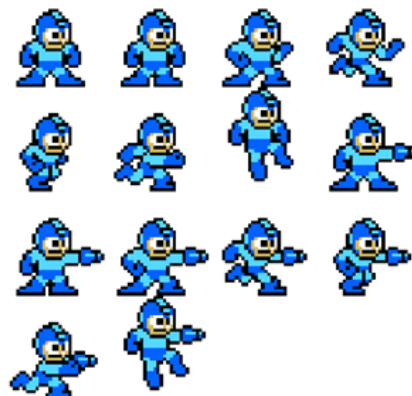


## Reporte Carlos

- Investigación de una herramienta de diseño para personajes, fondo, plataforma, etc. Que se acople al estilo definido para el videojuego, en este caso, píxel art. Para ello usamos varias opciones, y la más apropiada debido a su facilidad de uso principalmente fue piskel art.



- Con este programa podemos crear diseños estilo píxel art en diferentes dimensiones, además que facilita bastante el proceso de secuencia entre frames, y tiene una opción de exportación a sprites que agiliza bastante el proceso. Se hizo el diseño de los Sprites de los primeros personajes para el juego, tomando como referencias los frames del personaje "Megaman" y otros personajes conocidos:

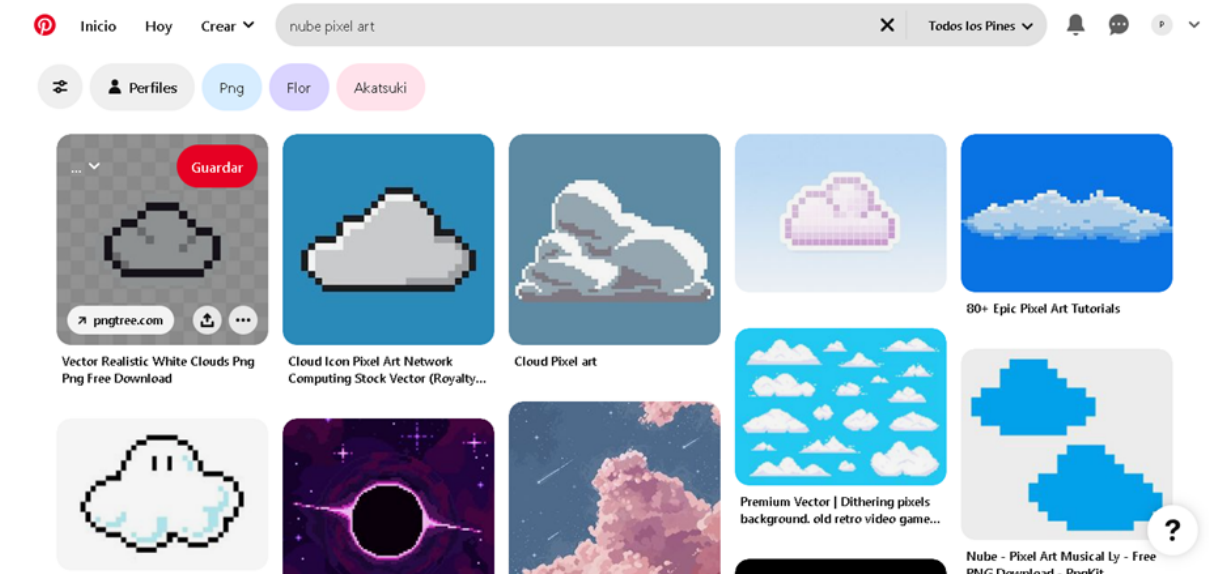
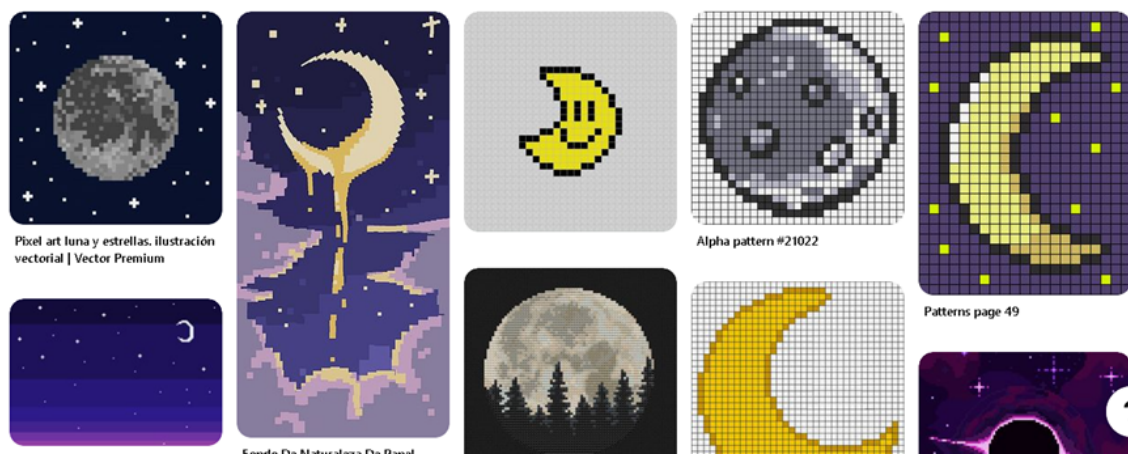
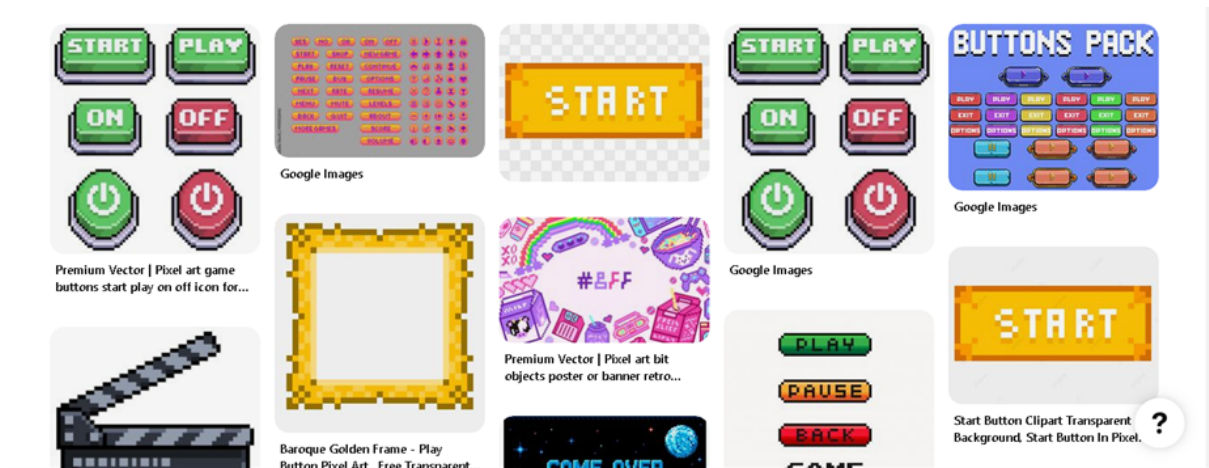


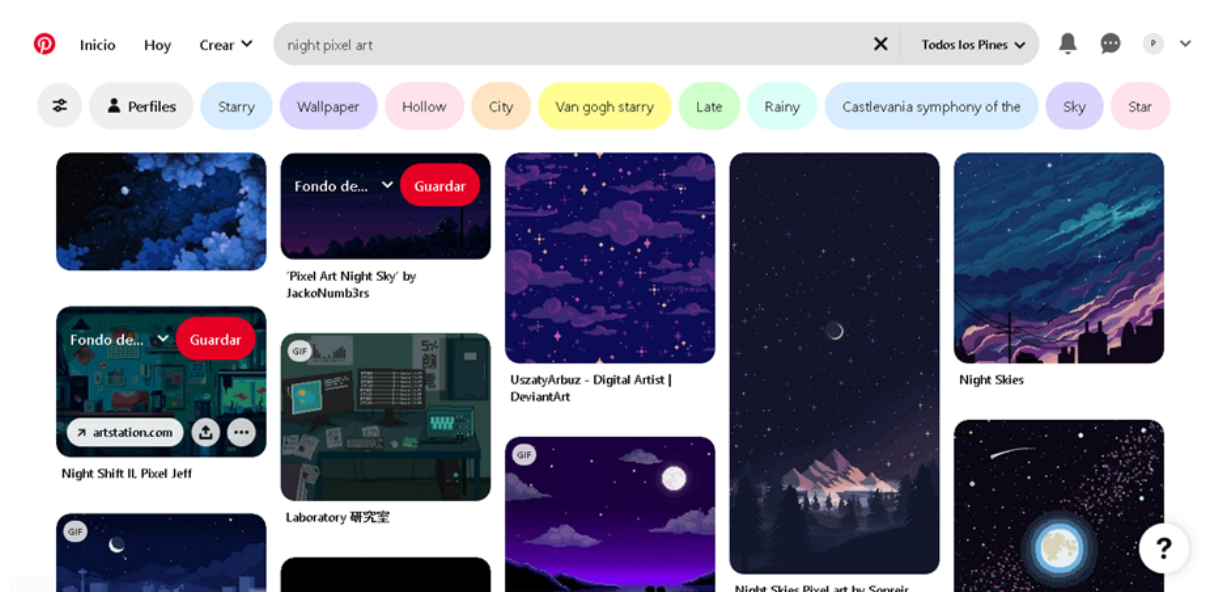


## Reporte Pablo:

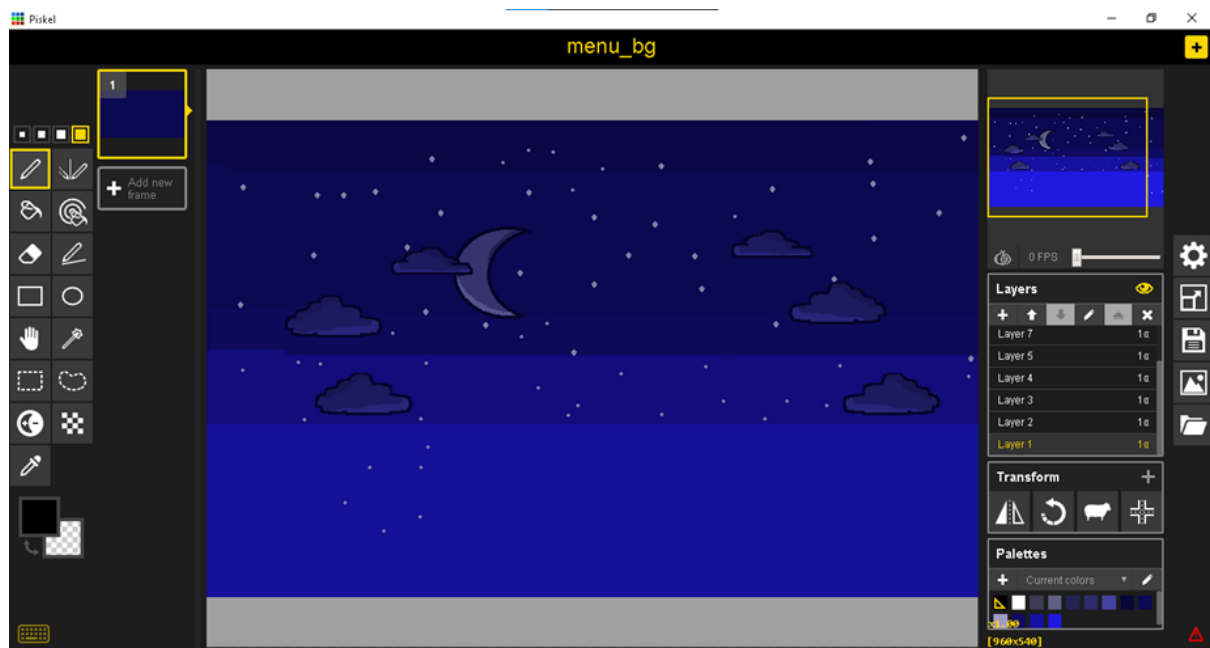
Diseño de Sprites de botones e investigación sobre creación de pixel art

Referencias:





Creación de menú de inicio:



Creación de botones:





Investigacion del motor grafico de pygame

[pygame.org/news](https://pygame.org/news)

revision de documentacion

Revisión de ejemplos para referencias

[pyGame: Realizando juegos con Python - Code INTEF](https://www.youtube.com/watch?v=wJMDh9QGRgs)

[\(1049\) Creating a platformer in Pygame with a camera, collisions, animation states and particle effects - YouTube](https://www.youtube.com/watch?v=wJMDh9QGRgs)

<https://www.youtube.com/watch?v=wJMDh9QGRgs>

## Informe: Motores Gráficos en 2D para Proyectos de Videojuegos en Python

### 1. Pygame:

- Descripción: Pygame es una biblioteca de juegos de código abierto y ampliamente utilizada en Python. Proporciona una API sencilla y potente para crear juegos en 2D.
- Características clave:
  - Gráficos en 2D: Pygame ofrece capacidades gráficas para dibujar formas, imágenes, sprites y fondos.
  - Sonido: Permite reproducir y manipular efectos de sonido y música de fondo.
  - Entrada de usuario: Proporciona manejo de eventos del teclado, mouse y joystick.
- Enlaces útiles:
  - Sitio web: <https://www.pygame.org/>

### 2. Arcade:

- Descripción: Arcade es un motor gráfico 2D específicamente diseñado para la creación de juegos en Python.
- Características clave:
  - Gráficos y animaciones: Arcade facilita el dibujo de formas geométricas, imágenes y animaciones en la pantalla.
  - Sonido: Proporciona soporte para efectos de sonido y música.
  - Física básica: Ofrece simulación básica de física, como gravedad y colisiones.
- Enlaces útiles:
  - Sitio web: <https://arcade.academy/>

### 3. Panda3D:

- Descripción: Aunque es principalmente un motor de juegos en 3D, Panda3D también se puede utilizar para crear juegos en 2D.
- Características clave:
  - Renderizado acelerado por hardware: Proporciona un rendimiento eficiente para gráficos en 2D.
  - Detección de colisiones: Incluye funciones para detectar colisiones entre objetos.
  - Shaders: Soporta shaders personalizados para efectos visuales avanzados.
- Enlaces útiles:
  - Sitio web: <https://www.panda3d.org/>

### 4. Pyglet:

- Descripción: Pyglet es una biblioteca de juegos en Python que permite crear juegos en 2D y 3D.
- Características clave:
  - Gráficos y animaciones: Pyglet proporciona herramientas para dibujar imágenes, sprites y animaciones.



- Sonido: Permite reproducir efectos de sonido y música.
- Manejo de eventos: Proporciona una interfaz para manejar eventos de teclado, mouse y ventana.
- Enlaces útiles:
  - Sitio web: <https://pyglet.readthedocs.io/en/latest/>

Recursos para utilizar el motor grafico Pygame:

1. Sitio web oficial de Pygame:
  - Descripción: El sitio web oficial de Pygame es un recurso invaluable que contiene documentación completa, tutoriales y una comunidad activa de desarrolladores. Proporciona una guía detallada para comenzar con Pygame y ofrece ejemplos de código útiles.
  - Enlace: [Pygame - Sitio web oficial](#)
2. Canal de YouTube de Pygame Tutorials por Sentdex:
  - Descripción: Sentdex es un popular creador de contenido de Python en YouTube. Su canal Pygame Tutorials ofrece una serie de tutoriales en video que cubren diversos aspectos de Pygame, desde los fundamentos hasta proyectos más avanzados.
  - Enlace: [Pygame Tutorials por Sentdex](#)
3. Libro "Making Games with Python & Pygame" de Al Sweigart:
  - Descripción: "Making Games with Python & Pygame" es un libro gratuito escrito por Al Sweigart. El libro proporciona una introducción práctica a la programación de juegos en Python utilizando el motor Pygame. Cubre desde los conceptos básicos hasta la creación de juegos completos.
  - Enlace: [Making Games with Python & Pygame](#)

## Manual de Instalación y Uso de Pygame

### Paso 1: Verificar los requisitos del sistema

- Asegúrate de tener Python instalado en tu sistema. Pygame es compatible con Python 2.7 y versiones posteriores, pero se recomienda utilizar Python 3.x.

### Paso 2: Instalar Pygame

- Abre una terminal o línea de comandos en tu sistema.
- Utiliza pip para instalar Pygame ejecutando el siguiente comando: **pip install pygame**.

### Paso 3: Verificar la instalación

- Para verificar si la instalación de Pygame fue exitosa, ejecuta el siguiente comando en la terminal o línea de comandos: **python -m pygame.examples.aliens**.
- Deberías ver una ventana emergente con un juego simple de naves espaciales. Esto indica que Pygame se ha instalado correctamente.

### Paso 4: Crear un proyecto de Pygame

- Abre tu editor de código preferido.
- Crea un nuevo archivo y guarda el archivo con una extensión ".py"
- Importa el módulo Pygame en tu archivo de la siguiente manera: `import pygame`

### Paso 5: Inicializar Pygame

- Antes de utilizar las funciones de Pygame, es necesario inicializar el módulo. Agrega el siguiente código al inicio de tu archivo: `pygame.init()`

### Paso 6: Crear una ventana

- Para mostrar gráficos en Pygame, necesitas crear una ventana. Agrega el siguiente código después de la inicialización:

```
screen_width = 800

screen_height = 600

screen = pygame.display.set_mode((screen_width, screen_height))

pygame.display.set_caption("Mi Juego")
```

### Paso 7: Evento de cierre

- Pygame utiliza un ciclo principal para actualizar y mostrar los gráficos. Es importante manejar el evento de cierre correctamente para evitar bloqueos. Agrega el siguiente código dentro del ciclo principal:

```
running = True

while running:

    for event in pygame.event.get():

        if event.type == pygame.QUIT:

            running = False
```

#### Paso 8: Actualizar la pantalla

- Para mostrar los cambios en la pantalla, es necesario actualizarla. Agrega el siguiente código al final del ciclo principal:

```
pygame.display.flip()
```

#### Paso 9: Finalizar el programa

- Una vez que hayas salido del ciclo principal, es recomendable finalizar Pygame correctamente. Agrega el siguiente código al final de tu archivo:

```
pygame.quit()
```

Reporte Rodolfo:

## Instalación de Pygame

Se instala Pygame desde la terminal de Pycharm.

```
1 | pip install pygame
```

Instala la última versión mostrando el siguiente mensaje.

### Resultado:

```
Successfully installed pygame-(version...)
```

Para empezar a utilizar el módulo pygame, es necesario importar junto con el módulo sys ya que contiene métodos del sistema en la que se van a utilizar para crear la primera ventana del juego.

```
1 | import pygame
2 | import sys
```

Se escribe init para empezar a ejecutar Pygame.

```
1 | pygame.init()
```

Una vez que se inicializa Pygame, se crea la ventana de la siguiente manera.

```
1 | pantalla = pygame.display.set_mode((500, 400))
```

Lo que hace es crear una ventana de 500 pixeles de ancho por 400 pixeles de alto.

Después se personaliza la ventana agregándole un título.

```
1 | pygame.display.set_caption('Mi primer juego :D')
```

Es importante agregar un bucle para que la ventana se mantenga activa, de lo contrario se cargará un instante y al finalizar el código desaparecerá.

```
1 | while True:  
2 |     for event in pygame.event.get():  
3 |         if event.type == pygame.QUIT:  
4 |             pygame.quit()  
5 |             sys.exit()
```

A continuación, se muestra cómo queda la ventana del juego.

