

## Informe Videojuego – Lucas Gastón Isla Liberatore

### Explicación del código

Lo primero que se hace en el código es importar las librerías necesarias. Luego de ese se carga el archivo .mp3 de la canción de fondo e inicia la reproducción

Se define el tamaño de los pixeles del tablero y cuantos contendrá. También se especifica el color y se determina cuando una celda está muerta o viva.

Lo siguiente es declarar las variables de posición, las de velocidad y las de tiempo. Inicia un while que no para nunca, el cual corre las funciones necesarias para alterar la velocidad cada vez que se presiona una tecla

Posteriormente se inicia una serie de bucles for, los cuales definen cada una de las acciones al momento de presionar una tecla.

Luego de que sigue, es establecer y borrar las celdas que conforman la nave, de esta manera realizando el movimiento.

Ya casi a lo último se encuentra un bucle for, que realiza la lógica de los disparos, el color y su velocidad, además de la posición. Luego el movimiento de los escombros, determinando su velocidad y su aleatoriedad en su posición. Además se declara la interacción entre los disparos y los objetivos. Lo último en el código es contar las partículas destruidas y declara el score

### Cambios

Se decidió por la implementación de un cambio en la música, ya que era lo más fácil, una vez ubicada la parte donde se define esto en el código original, se busco alguna pista musical para intercambiarla. Se decidió por el leitmotiv del personaje Raiden Shogun, del videojuego Genshin Impact



```
#Load audio file  
mixer.music.load('Raiden Shogun.mp3') #musica  
print("music started playing...")
```

Lo mismo seguía con el sonido de los disparos, se ubico la parte del código donde se defina el archivo de música, y una vez ubicado se decidió por un sonido, se decidió uno proveniente del videojuego Arknights



Lo siguiente se decidió por cambiar el color de los disparos, a lo cual se le solicitó ayuda al profesor. Una vez con su ayuda se seleccionó un color, se decidió por un color azul.

```

pygame.draw.polygon(screen, (200, 100, 0), poly, 0)
elif gameState[x, y] == 2:
    pygame.draw.polygon(screen, (0, 147, 255), poly, 0)

```



Luego se solicitó un cambio en el dibujo de la nave. Se intentó por los medios propios, pero al no ser capaz se le volvió a pedir ayuda al profesor, una vez con su ayuda se cambiaron unos valores necesarios para el acomodamiento de la nave



```
gameState[bxpos+20,bypos+11]=-0
gameState[bxpos+21,bypos+11]=-0
gameState[bxpos+22,bypos+11]=-0
gameState[bxpos+23,bypos+11]=-0
gameState[bxpos+24,bypos+11]=-0
gameState[bxpos+25,bypos+11]=-0
gameState[bxpos+26,bypos+11]=-0
```

```
gameState[bxpos+21,bypos+12]=-0
gameState[bxpos+25,bypos+12]=-0
```

```
gameState[bxpos+21,bypos+13]=-0
gameState[bxpos+25,bypos+13]=-0
```

```
gameState[bxpos+20,bypos+14]=-0
gameState[bxpos+22,bypos+14]=-0
gameState[bxpos+24,bypos+14]=-0
gameState[bxpos+26,bypos+14]=-0
```

```
gameState[bxpos+21,bypos+10]=-0
gameState[bxpos+22,bypos+10]=-0
gameState[bxpos+23,bypos+10]=-0
gameState[bxpos+24,bypos+10]=-0
gameState[bxpos+25,bypos+10]=-0
```

```
gameState[bxpos+22,bypos+9]=-0
gameState[bxpos+23,bypos+9]=-0
gameState[bxpos+24,bypos+9]=-0
```

```
gameState[bxpos+22,bypos+8]=-0
```

```
gameState[bxpos+23,bypos+8]=-0
```

```
##
gameState[xpos+20,ypos+11]=-1
gameState[xpos+21,ypos+11]=-1
gameState[xpos+22,ypos+11]=-1
gameState[xpos+23,ypos+11]=-1
gameState[xpos+24,ypos+11]=-1
gameState[xpos+25,ypos+11]=-1
gameState[xpos+26,ypos+11]=-1
```

```
gameState[xpos+21,ypos+12]=-1
gameState[xpos+25,ypos+12]=-1
```

```
gameState[xpos+21,ypos+13]=-1
gameState[xpos+25,ypos+13]=-1
```

```
gameState[xpos+20,ypos+14]=-1
gameState[xpos+22,ypos+14]=-1
gameState[xpos+24,ypos+14]=-1
gameState[xpos+26,ypos+14]=-1
```