

En esta parte del código se realiza la carga del archivo de la música que vamos a usar de fondo del juego. Además de importar pygame, numpy, time y os

```
# -*- coding: utf-8 -*-
import pygame
import numpy as np #alias de numpy
import time
import os

from pygame import mixer

pygame.init()

#Instantiate mixer
mixer.init()

#Load audio file
mixer.music.load('musica.mp3')

print("music started playing....")

#Set preferred volume
mixer.music.set_volume(0.2)

#Play the music
mixer.music.play()
```

En esta otra parte del código se ingresan al programa las variables nuevas y sus valores correspondientes, como el tamaño de la matriz, y las variables de posición → xpos, ypos, bxpos, bypos.

```
width, height = 400, 400

bg = 35, 35, 70

screen = pygame.display.set_mode((height, width))
screen.fill(bg)

# Tamaño de nuestra matriz
nxC, nyC = 80, 80

# Estado de las celdas. Viva = 1 / Muerta = 0
gameState = np.zeros((nxC, nyC))

#dimensiones de cada celda individual
dimCW = width / nxC
dimCH = height / nyC

xpos = 35
ypos = 0
bxpos = xpos+1
bypos = ypos

xvel = 0
yvel = 0
xtiempo = 0
ytiempo = 0
tiempo_global = 0

xpos_canon = xpos + 6
ypos_canon = ypos + 52
```

comienzo del bucle de ejecución en donde estarán todos los comandos del juego

[illegible]

en esta otra parte nos encargamos de dibujar la nave y de borrar la misma a medida que se va moviendo por la pantalla

```
gameState[bxpos+8,58+bypos] = 0
gameState[bxpos+12,58+bypos] = 0
gameState[bxpos+7,58+bypos] = 0
gameState[bxpos+13,58+bypos] = 0
gameState[bxpos+6,59+bypos] = 0
gameState[bxpos+14,59+bypos] = 0
```

```
#punta
```

```
gameState[xpos+10,54+ypos] = 1
gameState[xpos+10,53+ypos] = 1
gameState[xpos+10,52+ypos] = 1
```

```
#H
```

```
gameState[xpos+9,59+ypos] = 1
gameState[xpos+9,58+ypos] = 1
gameState[xpos+9,57+ypos] = 1
gameState[xpos+9,56+ypos] = 1
gameState[xpos+9,55+ypos] = 1
gameState[xpos+10,59+ypos] = 1
gameState[xpos+10,55+ypos] = 1
gameState[xpos+11,59+ypos] = 1
gameState[xpos+11,58+ypos] = 1
gameState[xpos+11,57+ypos] = 1
gameState[xpos+11,56+ypos] = 1
gameState[xpos+11,55+ypos] = 1
```

```
#Alas
```

```
gameState[xpos+8,57+ypos] = 1
gameState[xpos+12,57+ypos] = 1
gameState[xpos+8,58+ypos] = 1
gameState[xpos+12,58+ypos] = 1
```

por ultimo esta la seccion en donde se ejecuta todas las condiciones de la nave como de los objetos

```
for y in range(0, nxC):
    for x in range(0, nyC):

        #Movimiento de escombros
        if tiempo_global % 6 == 0:
            if (y in range(79)) and (gameState[x,y] == 3):
                gameState[x,y] = 0
                gameState[x,y+1] = 4
            if (y in range(79)) and (gameState[x,y] == 4):
                gameState[x,y] = 3

        #fisica del disparo
        if (y in range(79)) and (gameState[x,y+1] == 2):
            if gameState[x,y] == 3:
                gameState[x,y] = 0
                gameState[x,y-1] = 0
                gameState[x+1,y] = 0
                gameState[x-1,y] = 0
                gameState[x+1,y-1] = 0
                gameState[x-1,y-1] = 0
                hit_count += 1
            else:
                gameState[x,y] = 2

        if (y in range(79)) and (gameState[x,y] == 2) and (gameState[x,y+1] == 0):
            gameState[x,y] = 0
```