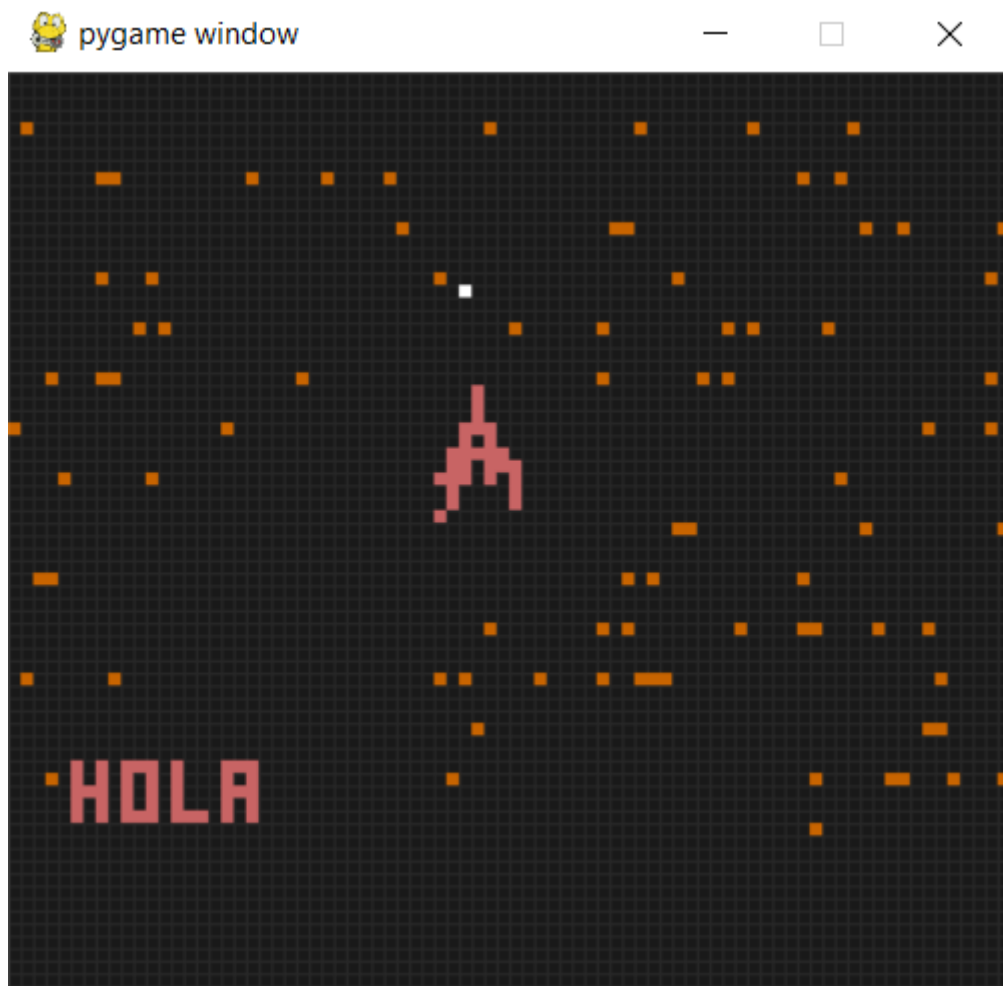


Informe NewBaseGame

Alumno: Ezequiel Dimnik

Video juego espacial donde debemos ir disparando los obstáculos y moviéndonos con el teclado para no salirnos del mapa.



Todo esto lo hacemos en la plataforma Spyder programando con Python.

El profesor nos mandó un código base en el cual teníamos que hacer modificaciones.

- 1) Reemplazar la figura (nave).
- 2) Cambiar el color de los disparos.
- 3) Cambiar la música de fondo.
- 4) Añadir sonido a los disparos.
- 5) Añadir mensaje
- 6) Modificación de distancia de disparo

1_ Copiamos un modelo de una nave que hicimos anteriormente y lo reemplazamos en nuestro nuevo código.

```
gameState[bxpos+5,59+bypos] = 0
    gameState[bxpos+5,58+bypos] = 0
    gameState[bxpos+5,57+bypos] = 0
    gameState[bxpos+5,56+bypos] = 0
    gameState[bxpos+5,55+bypos] = 0
    gameState[bxpos+6,54+bypos] = 0
    gameState[bxpos+6,53+bypos] = 0
    gameState[bxpos+6,52+bypos] = 0
    gameState[bxpos+6,57+bypos] = 0
    gameState[bxpos+6,55+bypos] = 0
    gameState[bxpos+7,59+bypos] = 0
    gameState[bxpos+7,58+bypos] = 0
    gameState[bxpos+7,57+bypos] = 0
    gameState[bxpos+7,56+bypos] = 0
    gameState[bxpos+7,55+bypos] = 0
#Alas
    gameState[bxpos+4,57+bypos] = 0
    gameState[bxpos+8,57+bypos] = 0
    gameState[bxpos+4,58+bypos] = 0
    gameState[bxpos+8,58+bypos] = 0
    gameState[bxpos+4,59+bypos] = 0
    gameState[bxpos+4,60+bypos] = 0
    gameState[bxpos+4,61+bypos] = 0
```

```
gameState[bxpos+3,62+bypos] = 0
gameState[bxpos+9,58+bypos] = 0
gameState[bxpos+3,59+bypos] = 0
gameState[bxpos+9,59+bypos] = 0
gameState[bxpos+9,60+bypos] = 0
gameState[bxpos+9,61+bypos] = 0
```

#H

```
gameState[xpos+5,59+ypos] = 1
gameState[xpos+5,58+ypos] = 1
gameState[xpos+5,57+ypos] = 1
gameState[xpos+5,56+ypos] = 1
gameState[xpos+5,55+ypos] = 1
gameState[xpos+6,54+ypos] = 1
gameState[xpos+6,53+ypos] = 1
gameState[xpos+6,52+ypos] = 1
gameState[xpos+6,57+ypos] = 1
gameState[xpos+6,55+ypos] = 1
gameState[xpos+7,59+ypos] = 1
gameState[xpos+7,58+ypos] = 1
gameState[xpos+7,57+ypos] = 1
gameState[xpos+7,56+ypos] = 1
gameState[xpos+7,55+ypos] = 1
```

#Alas

```
gameState[xpos+4,57+ypos] = 1
gameState[xpos+8,57+ypos] = 1
gameState[xpos+4,58+ypos] = 1
gameState[xpos+8,58+ypos] = 1
gameState[xpos+4,59+ypos] = 1
gameState[xpos+4,60+ypos] = 1
```

```
gameState[xpos+4,61+ypos] = 1
gameState[xpos+3,62+ypos] = 1
gameState[xpos+9,58+ypos] = 1
gameState[xpos+3,59+ypos] = 1
gameState[xpos+9,59+ypos] = 1
gameState[xpos+9,60+ypos] = 1
gameState[xpos+9,61+ypos] = 1
```

2) Cambiamos el color de disparo del código base que era de color rojo a blanco, cambiando los valores de la siguiente parte del código

line 270 : Color blanco

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

3) Para cambiar la música de fondo pusimos un enlace de youtube en un convertidor mp3, seguido de esto pusimos el audio en la carpeta de juegos y en el código. Así quedó la parte de la música de fondo, con el tema salgo a cazar de YSY A

```
#Instantiate mixer
mixer.init()
```

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

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

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

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

4) Usamos un audio mp3 el cual lo guardamos en la carpeta juegos y después lo llamamos desde el código, nos quedó de la siguiente forma.

```
mixer.music.load('shoot.mp3')  
    print("shoot started playing....")
```

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

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

5) Añadimos un mensaje que salta al principio del juego y nos sigue con la nave. Optamos por poner algo simple como un Hola

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

```
#mensaje
```

```
#H
```

```
gameState[5,59] = 1
```

```
gameState[5,58] = 1
```

```
gameState[5,57] = 1
```

```
gameState[5,56] = 1
```

```
gameState[5,55] = 1
```

```
gameState[6,57] = 1
```

```
gameState[7,59] = 1
```

```
gameState[7,58] = 1
```

```
gameState[7,57] = 1
```

```
gameState[7,56] = 1
gameState[7,55] = 1
#
#O
gameState[9,59] = 1
gameState[9,58] = 1
gameState[9,57] = 1
gameState[9,56] = 1
gameState[9,55] = 1
gameState[10,59] = 1
gameState[10,55] = 1
gameState[11,59] = 1
gameState[11,58] = 1
gameState[11,57] = 1
gameState[11,56] = 1
gameState[11,55] = 1
#
#L
gameState[13,59] = 1
gameState[13,58] = 1
gameState[13,57] = 1
gameState[13,56] = 1
gameState[13,55] = 1
gameState[14,59] = 1
gameState[15,59] = 1
#
#A
gameState[17,59] = 1
gameState[17,58] = 1
gameState[17,57] = 1
gameState[17,56] = 1
gameState[17,55] = 1
gameState[18,55] = 1
```

```
gameState[18,57] = 1
gameState[19,59] = 1
gameState[19,58] = 1
gameState[19,57] = 1
gameState[19,56] = 1
gameState[19,55] = 1
```

6) Tuvimos que modificar la distancia del disparo ya que generaba un error que era al mantener presionado la barra espaciadora se generaba una línea, lo que hicimos fue restarle el valor de la variable ypos_canon

line 90

```
xpos_canon = xpos + 6
ypos_canon = ypos + 49
```

Codigo Completo:

```
# -*- 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('salgo.mp3')
```

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

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

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

```
width, height = 400, 400
```

```
bg = 25, 25, 25
```

```
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 = 0  
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 + 49
```

```
pauseExect = True
```



```
if event.type == pygame.KEYDOWN:
```

```

if event.key == pygame.K_LEFT:
    xvel = xvel - 1
elif event.key == pygame.K_RIGHT:
    xvel = xvel + 1
elif event.key == pygame.K_UP:
    yvel = yvel - 1
elif event.key == pygame.K_DOWN:
    yvel = yvel + 1
elif event.key == pygame.K_SPACE:
    gameState[xpos_canon,ypos_canon] = 2
    mixer.music.load('shoot.mp3')

    print("shoot started playing....")

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

    #Play the music
    mixer.music.play()

else:
    pauseExect = not pauseExect
if event.type == pygame.QUIT:
    stay = False
    pygame.quit()

# Detectamos si se presiona el ratón.
mouseClick = pygame.mouse.get_pressed()

if sum(mouseClick) > 0:
    posX, posY = pygame.mouse.get_pos()
    celX, celY = int(np.floor(posX / dimCW)), int(np.floor(posY / dimCH))
    gameState[celX, celY] = 1

#Borrado y escritura
if (xpos != bxpos) or (ypos != bypos):

    #H

    gameState[bxpos+5,59+bypos] = 0
    gameState[bxpos+5,58+bypos] = 0
    gameState[bxpos+5,57+bypos] = 0

```

```
gameState[bxpos+5,56+bypos] = 0
gameState[bxpos+5,55+bypos] = 0
gameState[bxpos+6,54+bypos] = 0
gameState[bxpos+6,53+bypos] = 0
gameState[bxpos+6,52+bypos] = 0
gameState[bxpos+6,57+bypos] = 0
gameState[bxpos+6,55+bypos] = 0
gameState[bxpos+7,59+bypos] = 0
gameState[bxpos+7,58+bypos] = 0
gameState[bxpos+7,57+bypos] = 0
gameState[bxpos+7,56+bypos] = 0
gameState[bxpos+7,55+bypos] = 0
#Alas
```

```
gameState[bxpos+4,57+bypos] = 0
gameState[bxpos+8,57+bypos] = 0
gameState[bxpos+4,58+bypos] = 0
gameState[bxpos+8,58+bypos] = 0
gameState[bxpos+4,59+bypos] = 0
gameState[bxpos+4,60+bypos] = 0
gameState[bxpos+4,61+bypos] = 0
gameState[bxpos+3,62+bypos] = 0
gameState[bxpos+9,58+bypos] = 0
gameState[bxpos+3,59+bypos] = 0
gameState[bxpos+9,59+bypos] = 0
gameState[bxpos+9,60+bypos] = 0
gameState[bxpos+9,61+bypos] = 0
```

#H

```
gameState[xpos+5,59+ypos] = 1
gameState[xpos+5,58+ypos] = 1
gameState[xpos+5,57+ypos] = 1
gameState[xpos+5,56+ypos] = 1
gameState[xpos+5,55+ypos] = 1
gameState[xpos+6,54+ypos] = 1
gameState[xpos+6,53+ypos] = 1
gameState[xpos+6,52+ypos] = 1
gameState[xpos+6,57+ypos] = 1
gameState[xpos+6,55+ypos] = 1
gameState[xpos+7,59+ypos] = 1
gameState[xpos+7,58+ypos] = 1
gameState[xpos+7,57+ypos] = 1
```

```

gameState[xpos+7,56+ypos] = 1
gameState[xpos+7,55+ypos] = 1
#Alas
gameState[xpos+4,57+ypos] = 1
gameState[xpos+8,57+ypos] = 1
gameState[xpos+4,58+ypos] = 1
gameState[xpos+8,58+ypos] = 1
gameState[xpos+4,59+ypos] = 1
gameState[xpos+4,60+ypos] = 1
gameState[xpos+4,61+ypos] = 1
gameState[xpos+3,62+ypos] = 1
gameState[xpos+9,58+ypos] = 1
gameState[xpos+3,59+ypos] = 1
gameState[xpos+9,59+ypos] = 1
gameState[xpos+9,60+ypos] = 1
gameState[xpos+9,61+ypos] = 1

```

#Actualizamos la posicion de borrado

bxpos = xpos

bypos = ypos

if tiempo_global % 24 == 0:

```

gameState[:,0] = np.heaviside(np.random.rand(1,nxC)-0.9,1) * 3

```

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

Calculamos el polígono que forma la celda.

poly = [(x * dimCW, y * dimCH),

((x+1) * dimCW, y * dimCH),

((x+1) * dimCW, (y+1) * dimCH),

((x * dimCW, (y+1) * dimCH)]

Si la celda está "muerta" pintamos un recuadro con borde gris

if gameState[x, y] == 0:

pygame.draw.polygon(screen, (40, 40, 40), poly, 1)

Si la celda está "viva" pintamos un recuadro relleno de color

elif gameState[x, y] == 3:

pygame.draw.polygon(screen, (200, 100, 0), poly, 0)

elif gameState[x, y] == 2:

pygame.draw.polygon(screen, (255, 255, 255), poly, 0)

else:

pygame.draw.polygon(screen, (200, 100, 100), poly, 0)

if tiempo_global>30 and 1 not in gameState:

pygame.quit()

print('Tu puntuacion fue de', hit_count, ' puntos.')

username = input("Ingresa su nombre para guardar, o N para no hacerlo")

if username != 'N':

file = open("scores.txt", "w")

to_save = str([username, hit_count])

```
file.write(to_save + os.linesep)
file.close()
```

```
# Mostramos el resultado
pygame.display.flip()
```

```
#mensaje
```

```
#H
```

```
gameState[5,59] = 1
gameState[5,58] = 1
gameState[5,57] = 1
gameState[5,56] = 1
gameState[5,55] = 1
gameState[6,57] = 1
gameState[7,59] = 1
gameState[7,58] = 1
gameState[7,57] = 1
gameState[7,56] = 1
gameState[7,55] = 1
```

```
#
```

```
#O
```

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

```
#
```

```
#L
```

```
gameState[13,59] = 1
gameState[13,58] = 1
gameState[13,57] = 1
gameState[13,56] = 1
gameState[13,55] = 1
gameState[14,59] = 1
gameState[15,59] = 1
```

```

#
#A
gameState[17,59] = 1
gameState[17,58] = 1
gameState[17,57] = 1
gameState[17,56] = 1
gameState[17,55] = 1
gameState[18,55] = 1
gameState[18,57] = 1
gameState[19,59] = 1
gameState[19,58] = 1
gameState[19,57] = 1
gameState[19,56] = 1
gameState[19,55] = 1
#

```

#Lista de teclas y como llamarlas

pygame

Constant ASCII Description

K_BACKSPACE \b backspace

K_TAB \t tab

K_CLEAR clear

K_RETURN \r return

K_PAUSE pause

K_ESCAPE ^[escape

K_SPACE space

K_EXCLAIM ! exclaim

K_QUOTEDBL " quotedbl

K_HASH # hash

K_DOLLAR \$ dollar

K_AMPERSAND & ampersand

K_QUOTE quote

K_LEFTPAREN (left parenthesis

K_RIGHTPAREN) right parenthesis

K_ASTERISK * asterisk

K_PLUS + plus sign

K_COMMA , comma

K_MINUS - minus sign

K_PERIOD . period

K_SLASH / forward slash

K_0 0 0

K_1 1 1

K_2 2 2

# K_3	3	3
# K_4	4	4
# K_5	5	5
# K_6	6	6
# K_7	7	7
# K_8	8	8
# K_9	9	9
# K_COLON	:	colon
# K_SEMICOLON	;	semicolon
# K_LESS	<	less-than sign
# K_EQUALS	=	equals sign
# K_GREATER	>	greater-than sign
# K_QUESTION	?	question mark
# K_AT	@	at
# K_LEFTBRACKET	[left bracket
# K_BACKSLASH	\	backslash
# K_RIGHTBRACKET]	right bracket
# K_CARET	^	caret
# K_UNDERSCORE	_	underscore
# K_BACKQUOTE	`	grave
# K_a	a	a
# K_b	b	b
# K_c	c	c
# K_d	d	d
# K_e	e	e
# K_f	f	f
# K_g	g	g
# K_h	h	h
# K_i	i	i
# K_j	j	j
# K_k	k	k
# K_l	l	l
# K_m	m	m
# K_n	n	n
# K_o	o	o
# K_p	p	p
# K_q	q	q
# K_r	r	r
# K_s	s	s
# K_t	t	t
# K_u	u	u
# K_v	v	v
# K_w	w	w
# K_x	x	x


```
# K_y      y      y
# K_z      z      z
# K_DELETE      delete
# K_KP0      keypad 0
# K_KP1      keypad 1
# K_KP2      keypad 2
# K_KP3      keypad 3
# K_KP4      keypad 4
# K_KP5      keypad 5
# K_KP6      keypad 6
# K_KP7      keypad 7
# K_KP8      keypad 8
# K_KP9      keypad 9
# K_KP_PERIOD .    keypad period
# K_KP_DIVIDE /    keypad divide
# K_KP_MULTIPLY *  keypad multiply
# K_KP_MINUS  -    keypad minus
# K_KP_PLUS   +    keypad plus
# K_KP_ENTER  \r   keypad enter
# K_KP_EQUALS =    keypad equals
# K_UP        up arrow
# K_DOWN      down arrow
# K_RIGHT     right arrow
# K_LEFT      left arrow
# K_INSERT    insert
# K_HOME      home
# K_END       end
# K_PAGEUP    page up
# K_PAGEDOWN  page down
# K_F1        F1
# K_F2        F2
# K_F3        F3
# K_F4        F4
# K_F5        F5
# K_F6        F6
# K_F7        F7
# K_F8        F8
# K_F9        F9
# K_F10       F10
# K_F11       F11
# K_F12       F12
# K_F13       F13
# K_F14       F14
# K_F15       F15
```

# K_NUMLOCK	numlock
# K_CAPSLOCK	capslock
# K_SCROLLLOCK	scrollock
# K_RSHIFT	right shift
# K_LSHIFT	left shift
# K_RCTRL	right control
# K_LCTRL	left control
# K_RALT	right alt
# K_LALT	left alt
# K_RMETA	right meta
# K_LMETA	left meta
# K_LSUPER	left Windows key
# K_RSUPER	right Windows key
# K_MODE	mode shift
# K_HELP	help
# K_PRINT	print screen
# K_SYSREQ	sysrq
# K_BREAK	break
# K_MENU	menu
# K_POWER	power
# K_EURO	Euro
# K_AC_BACK	Android back button