

Proyecto Final

Juego tipo Arcade

Duck Hunt

Amoguimba Jessica, Guallichico Nataly

1. OBJETIVO GENERAL

Desarrollo de juego tipo arcade en Python, aplicando los conocimientos adquiridos en Programación Avanzada.

2. TEMA DEL PROYECTO

Nuestro proyecto es desarrollar el conocido juego **Duck Hunt** que es uno de los videojuegos más recordados en los 80 desarrollado por Nintendo Research & Development.



Esta vez con la ayuda de Python y Pygame recrearemos este grandioso juego.

3. HERRAMIENTAS A USAR

Como primer paso para empezar a desarrollar este juego instalamos Pygame en la terminal del símbolo del Sistema usando:

```
pip install pygame
```

después de esto comprobaremos si se encuentra instalada correctamente en nuestra versión de python.

Escribimos:

```
import pygame
```

en la terminal de Python, una vez instalado correctamente trabajaremos en el código con la ayuda de la herramienta Sublime Text.

```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Intel)] on
type "help", "copyright", "credits" or "license" for more information.
>> import pygame
pygame 1.9.6
Hello from the pygame community. https://www.pygame.org/contribute.html
>> _
```

4. CÓDIGO

En el encabezado del código primero iré las librerías que usaremos para el desarrollo del juego.

```
import pygame
from pygame.locals import*
from sys import exit
#randmicos
from random import randint
```

Daremos colores de fondo para insertar la imagen de fondo, seguido de la creación de la ventana y del tamaño de la misma.

```
negro = (0, 0, 0)
blanco = (255, 255, 255)
screen = pygame.display.set_mode((890, 550), 0, 32)
```

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

Daremos posiciones al mouse y también a la posición de los patos que van a salir de forma randomica, para ello usamos la librería **random**.

```
# titulo a la ventana
pygame.display.set_caption("Duck Hunt")

#Variables para la posicion del mouse.
x_pos = 0
y_pos = 0

# variables para la posicion cuando hagamos clic con el mouse
x_click = 0
y_click = 0

# variable para la posicion del pato (randomico)
x_duck = 0
y_duck = randint(0, 450) #tamaño restringido para los aleatorios en Y
```

Seguido de esto pondremos la música al juego, código que podemos ver su formato en el link que comentamos. Además, podemos configurar el volumen el cual también nos ayudamos del link que esta comentado.

```
# init (frecuencia = 22050, tamaño = -16, canales = 2, búfer = 1024)
#https://www.pygame.org/docs/ref/mixer.html#pygame.mixer.init
pygame.mixer.init(44100, -16, 2, 1024)

# configuracion del volume de la musica entre 0.0/1.0
#https://www.pygame.org/docs/ref/music.html#pygame.mixer.music.set_volume
pygame.mixer.music.set_volume(0.5)
#musica de fondo al inicial el juego
pygame.mixer.music.load("fondoMusica.mp3")
#numero de veces q se repetira la musica
pygame.mixer.music.play(2)
```

Después, creamos un bucle while el cual llevara como inicio un evento para el mouse y la posición de este junto con los clics, `get_pos()` nos devuelve la posición X y Y del cursor del mouse.

```
while True:

    for event in pygame.event.get():
        #
        if event.type == QUIT:
            exit()
        elif event.type == MOUSEMOTION:
            x_pos, y_pos = pygame.mouse.get_pos()
        elif event.type == MOUSEBUTTONDOWN:
            x_click, y_click = pygame.mouse.get_pos()

    posicion = (x_pos - 50, y_pos - 50)

    x_duck += 1

    if x_duck * velocidad > 890 and not pierde:
        x_duck = 0
        y_duck = randint(0, 450)

        # error, toca la musica de game over
        pygame.mixer.music.load("duckHurtGameOver.ogg")
        pygame.mixer.music.play()

        pierde = True
```

Condicionamos la velocidad del pato si no pierde seguimos enviando patos de forma aleatoria.

Y si el caso es que pierde (si pierde es verdadero), la música Game Over sonará.

Dentro del mismo bucle ya pondremos la imagen de fondo y agregamos las letras del marcador en la parte frontal del fondo con ayuda del código `pygame.front.SysFront` y damos tipo de letra, tamaño, color, el lugar donde estará y también le indicamos que coloque el contador de los patos que va matando.

```
screen.blit(pygame.image.load("fondo.png"), (0, 0))
#muestra en screen el puntaje de patos muertos
# .from... muestra al frente del fondo principal
screen.blit(pygame.font.SysFont("comicsansms", 60).render("Score: " + str(puntos), True, blanco), (500, 450))
```

También condicionamos para la colisión del pato y la mira.

Y decimos que si la posición del pato con la posición del mouse al momento que hagamos clic sean iguales suene la música que indica que suma un punto al Score además la variable sumara uno y la velocidad aumentara cada vez que sumemos puntos y los patos seguirán saliendo aleatoriamente.

En la ventana enviamos la imagen del pato y las posiciones en X y en Y las cuales serán que en X ira

```
# COLISION PATO-MIRA
if x_click in range(x_duck * velocidad - 30, x_duck * velocidad + 30) and y_click in range(y_duck - 30, y_duck + 30):
    # agremos musica cuando mata al pato
    pygame.mixer.music.load("punto.ogg")
    pygame.mixer.music.play()

    #va sumando los puntos
    puntos += 1
    #y aumenta la velocidad
    velocidad += 1
    x_duck = 0
    y_duck = randint(50, 500)

screen.blit(pygame.image.load("pato.gif"), (x_duck * velocidad, y_duck))
```

junto con la velocidad y en Y la forma aleatoria.

Ya, por último, si el caso es que pierde es decir sale del rango X y Y, aparecerá la imagen del perrito y un GAME OVER en la pantalla.

Para finalizar, cargamos la imagen de la mira con la posición para todo el tiempo que dure el juego.

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

```
if pierde:
    x_duck = -50
    y_duck = -50
    #imagen en la mitad del fondo
    screen.blit(pygame.image.load("Sprite_del_Perro_de_Duck_Hunt.gif"), (670, 180))
    screen.blit(pygame.image.load("gameOver.jpg"), (75, 150))
    #screen.blit(pygame.image.load("salir.gif"), (650, 75))

screen.blit(pygame.image.load("mira.gif").convert(), posicion)
pygame.display.update()
```

Para la creación del menú principal se hizo lo siguiente.

```
from khinsider import *
import os
from sys import exit
ventana = 10

ventana.title("PLAY DUNCKHUNT!!!")
ventana.frame((width-400,height-400))
ventana.config(bg="pink")
ventana.config(cursor="hand2")
def JUGAR():
    print("JUGAR")
def SALIR():
    print("SALIR")
anchoboton=25
altoboton=2

boton1=Button(ventana,text="JUGADOR 1",width=anchoboton,height=altoboton,bg="pink",command=lambda:system("python proyecto_duckhunt.py"),place=(20,20))
boton2=Button(ventana,text="JUGADOR 2",width=anchoboton,height=altoboton,bg="pink",command=lambda:system("python proyecto_duckhunt.py"),place=(20,120))
boton3=Button(ventana,text="Salir",width=anchoboton,height=altoboton,bg="pink",command=quit),place=(20,220)
pack()
```

Se creó una ventana, le dimos tamaño, y un color de fondo. Creamos tres botones los cuales los dos primeros nos traslada al juego usando la librería **import os** y el tercer botón nos sacara del menú.

5. ANEXOS

Ventana de inicio del juego.



Pantalla de inicio del juego.



Captura cuando se pierde el juego.



6. BIBLIOGRAFIA

Sonidos:

<https://downloads.khinsider.com/game-soundtracks/album/duck-hunt>

Imágenes:

Samos imágenes de google las cuales después fueron modificadas en Paint.

Códigos de ayuda:

<https://www.pygame.org/docs/ref/mixer.html#pygame.mixer.init>

https://www.pygame.org/docs/ref/music.html#pygame.mixer.music.set_volume