

# Universidad de Costa Rica

Proyecto personal Flappy Bird

Herramientas de ciencia de datos II

CA-0305

Felix Madrigal Mora – C24459

Profesor: Luis Alberto Juárez Potoy

## Introducción

Para el presente proyecto hice uso de la herramienta de programación Python, y sus librerías, empezando por pygame la cual fue la principal librería que me permitió desarrollar mi juego y será abarcada más adelante, ahora bien, voy a dar inicio explicando un poco de la historia y en que consiste Flappy Bird.

Flappy Bird es un juego que fue creado en el año 2013, el cual fue desarrollado por Dong Nguyen por medio de su propia compañía Gears, a pesar de ser lanzado en mayo del 2013 el juego empezó a tomar mayor popularidad a principios del 2014, consiguiendo mas de 50 millones de descargas y convirtiéndose en el juego numero uno en la app store, el juego estaba generando un flujo de entradas de efectivo de aproximadamente 50 mil dólares por día, sobre todo por publicidad.

A pesar de que el juego se había vuelto de los mas consumidos en el mundo, su creador decidió eliminarlo ya que dijo que era peligroso para los usuarios, esto debido a que se volvió sumamente adictivo, y esta no era la meta de su creador, el cual solo quería que fuera un juego que proporcionase distracción y una forma de relajarse, pero por un tiempo relativamente corto, sin embargo, los usuarios se pasaban horas tratando de cada vez mejorar su puntaje en el juego.

Por esta razón el juego fue eliminado de todas las plataformas en las cuales se encontraba en el 2014, a pesar de esto algunos usuarios de apple se les quedo el juego en su almacenamiento al ya tenerlo descargado, por lo que esto genero que por medio de plataformas de venta de artículos usados como Ebay, se empezaran a generar subastas absurdamente altas, llegando a los 100 mil dólares por un iphone 5.

Ahora bien en que consiste plenamente Flappy Birds, este juego es relativamente básico, tanto así que se afirma que fue creado en 3 días, el juego consiste en que se tiene un pájaro, en medio de la pantalla y apenas uno lo toca, el juego da inicio, uno tiene que esta tocando constantemente la pantalla para que el pájaro se mantenga en el aire, sin embargo se le agrega dificultad ya que se tienen dos tuberías(pipes) las cuales si el pájaro pega en ellas se muere, por lo que se debe intentar mantener el pájaro en el aire y además que pase por en medio de ambas tuberías, el juego sin embargo no aumentaba su velocidad pero la dificultad consistía en mantener el pájaro constantemente en el aire, tomando en cuenta que había que subir y bajar muchas veces de un solo golpe o no le iba a dar el tiempo de reacción; por lo que un juego muy simple se volvió bastante desafiante y adictivo al siempre querer tener mayor puntaje.

Ya que es un juego que fue eliminado y ya no se consigue me pareció interesante tratar de replicarlo, realizando algunos cambios que a mi parecer lo vuelven más competitivo, además conllevan a trabajar en equipo para poder sobrepasar el puntaje, ya que el juego ahora no depende de un solo jugador si no que, de dos, si un solo jugador se muere el juego se detiene por lo que ya no va a poder sumando puntos, esto desarrolla un trabajo en equipo que permite mejorar la coordinación. De igual forma agregue un puntaje el cual le va a permitir a la persona saber como va y que tan difícil se va poniendo el juego, esto debido a que mi juego incrementa de velocidad cada cierta cantidad de puntos, cabe destacar que esto es 100% modificable y se puede poner cuanta velocidad de aumento se quiere y cada cuanta cantidad de puntos.

## Explicación del código y librerías

El código se puede encontrar en el repositorio:

[https://github.com/FelixMadrigal/Proyecto\\_personal\\_herramientas](https://github.com/FelixMadrigal/Proyecto_personal_herramientas)

Pygame es una librería que permiten una mayor facilidad a la hora crear videojuegos bidimensionales, esta se basa en SDL la cual a su vez también es una librería que permite el acceso al audio, imágenes, teclado, ratón y a lo que es toda la parte de la interfaz gráfica, sin embargo, SDL se encuentran en lenguaje C, solo que fue la inspiración el creador de pygame para desarrollarla.

Pygame permite acceso a algunos aspectos primordiales como lo son:

- Sonido: permite implementar y reproducir sonidos.
- Eventos: permite manejar eventos como entradas del teclado
- Colisiones: permite detectar colisiones con objetos
- Sprites: se puede implementar figuras sobre todo rectángulos, esto para dibujar y realizar el juego de forma correcta
- Gráficos: permite manipular y alterar superficies en 2D

Ahora ya que conocemos cuales son los principales aspectos que nos permite desarrollar pygame, podemos abarcar cual debería ser la estructura correcta para hacer un buen uso.

Bueno, primeramente se debe iniciar por la preparación del entorno, es decir descargar el paquete de pygame, esto haciendo `pip install pygame`, seguidamente se debe importar usando el siguiente comando `import pygame`, sin embargo luego de esto faltaría un paso mas para poder usar las funciones que ya trae pygame por dentro, es necesario hacer `from pygame.locals import *`, el asterisco no es necesario y se podrían colocar en vez de este los

nombres de únicamente se van a utilizar, ya que importar todas las funciones aunque no se vayan a usar puede no ser buena práctica.

Luego de hacer dichos pasos se debería de definir todo el bucle principal, en el que se va abarcar tanto las clases que se deben realizar, como los métodos que se incluyen para el juego y las funciones necesarias para que este de inicio y se muestre en la consola.

Entonces para entender que debería ir en el bucle principal debemos abarcar cual es el esqueleto primordial que se debe de seguir para poder implementar un juego.

Entonces primero se debe tener la clase main o Game(), además hacer uso de sprites y sonidos, en el caso de mi código implemente varias clases en cuales a lo largo de todas estas hago uso de sprites, tenemos lo que es la clase Bird la cual se encarga de implementar todas las imágenes del pájaro y a su vez sus funciones, como lo son que el pájaro salte, que sus alas se muevan, y es importante que al inicio se define `pygame.sprite.Sprite`, esto para que sea más fácil a la hora de manejar los objetos y bucles como las colisiones, de crear grupos de sprites y manejarlos más fácilmente, en esta misma clase se hace uso `pygame.mask.from_surface`, esta función permite sincronizar el pájaro para manejar las colisiones perfectas por pixel, haciéndolo mucho mas preciso, el funcionamiento se genera creando una mascara donde cada pixel no transparente se considera como parte de este tipo mascara de colisión.

Luego tengo la clase Pipe la cual funciona para implementar las imágenes de las tuberías, para colocar los tubos tanto arriba como debajo de la pantalla y considerando dejar un espacio en medio que permita el paso del pájaro para el juego, a su vez permite detectar también las colisiones creando de nuevo una máscara pero ahora con la imagen cargada, se usan comandos propios de pygame como `pygame.image.load` que nos ayuda a cargar las imágenes desde nuestras carpetas y también tenemos `pygame.transform.scale` que funciona para poder rotar o invertir las imágenes y de esta forma colocar las tuberías arriba.

Y por ultima esta la clase Ground, en esta se hace básicamente lo mismo que con la clase Pipe pero ahora se cargan las imágenes del suelo, se detectan las colisiones con estas mismas y se acomodan en la posición correcta.

Es importante que dentro de la clase Pipe y Ground se implementa la función `update` la cual permite que el suelo y las tuberías se mueva hacia la izquierda permitiendo un efecto visual cuando se inicia el juego de que el pájaro se está moviendo.

A su vez es importante mencionar que se realizan dos funciones por fuera de las clases esto porque era más fácil a la hora de generarlo tenerlo fuera de clases ya que estas dos funciones se encargan de verificar si las tuberías que se mueven a la izquierda se salen de la pantalla se borren y se creen nuevas, la primera es como general para saber cuando se sale y la segunda se encarga de crear nuevas tuberías del tamaño correcto.

Y ya por ultimo pero fuera de todas las clases tenemos lo que es la inicialización propia del juego donde se usan los principales comandos de pygame como lo son: `pygame.init()` el cual permite dar inicio a todas las demás funciones, `pygame.display.set_mode` con esta parte creamos la consola, `pygame.image.load` se cargan todas las imágenes necesarias, `pygame.transform.scale` se ajustan las dimensiones a la pantalla, `pygame.sprite.Group` crea un grupo de ciertos sprites para poder implementarlos de manera más rápida y fácil, `pygame.time.Clock()` este comando permite controlar la velocidad a la cual se actualiza el juego, `pygame.font.Font()` este lo implemente para poder tener un marcador, `pygame.event.get()` que nos permite acceder a los comandos recibidos por el teclado, y por ultimo `pygame.mixer.music.load()`, `pygame.mixer.music.play()` los cuales permiten cargar y reproducir la música.

## Conclusiones

Bueno los resultados obtenidos por su parte fueron los deseados, el código funciono y se pudo jugar de forma correcta, a su vez siento que el trabajo no solo me permitió conocer la implementación de pygame y videojuegos en Python, sino que también esto es un indicador de las muchas funciones y librerías que tiene Python y no conocemos, las cuales pueden ser muy valiosas para adquirir nuevos conocimientos o bien pasarla bien un rato y aprender a hacer cosas nuevas que tal vez no tengan un significado para la carrera pero sirven como distracción y a la vez se puede mejorar aspectos de la programación.

Como recomendación seria dedicar u poco mas de tiempo para estudiar mas a fondo la librería y entender mas a fondo su funcionamiento y que otras funciones se podrían implementar al juego para que sea más divertido, además mejorar en aspectos como el main ya que yo hice todo luego de las clases, y a pesar de que intente pasar esto a otro .py por aparte que fuera propiamente el main me empezó a dar muchos problemas los cuales no logre resolver.

## Referencias

Maldonado, R. (2024, 10 abril). ¿Qué es Pygame? | KeepCoding Bootcamps. *KeepCoding Bootcamps*. <https://keepcoding.io/blog/que-es-pygame/#:~:text=Pygame%20es%20una%20librer%C3%ADa%20para,hardware%20gr%C3%A1fico%20de%20nuestro%20ordenador>

*pygame.sprite* — *pygame v2.6.0 documentation*. (s. f.).  
<https://www.pygame.org/docs/ref/sprite.html>

Russ, & Russ. (2023, 28 febrero). *How to use PyGame Masks for Pixel Perfect Collision - coding with Russ*. Coding With Russ.  
<http://www.codingwithruss.com/pygame/how-to-use-pygame-masks-for-pixel-perfect-collision/>

Infobae. (2017, 22 noviembre). El creador de Flappy Bird explicó por qué eliminó su juego. *Infobae*. <https://www.infobae.com/2014/02/11/1542979-el-creador-flappy-bird-explico-que-elimino-su-juego/>