

PROYECTO TALLER DE PROGRAMACIÓN

JUEGO CON INTERFAZ GRÁFICA TKINTER

“CRAZY ROAD”

Profesor: Ing. Luis Barboza Artavia

Estudiante: Alessandro Hidalgo Prendas

Carné: 2020099003

I Semestre 2020

Instituto Tecnológico de Costa Rica

Ingeniería en Computadores

Tabla de Contenidos

Introducción	3
Marco Teórico	4
Propuestas de diseño de solución	6
Ejemplos de funcionamiento	7
Ventana de Menú.	7
Ventana de Juego.	7
Ventana de Instrucciones.	8
Ventana de Puntajes.	8
Ventana de Créditos.	9
Ventana de Game-Over.	9
Conclusiones	9

Introducción

Este proyecto consiste en realizar un juego similar al clásico de los años 80 “Road Fighter”, un juego de carreras que tuvo mucho éxito en su tiempo, pero este debe ser hecho con un diseño nuevo o temática propia. Para este proyecto se debe de usar el lenguaje de programación Python, que es con el lenguaje que se inicia la enseñanza en primer curso de programación (CE-1102). Este proyecto tiene como objetivo que aprendamos a buscar fuentes con las que guiarnos para poder realizar el proyecto de manera individual y obtener conocimiento por cuenta propia ya sea probando o utilizando como referencia códigos de distintas fuentes como de tutoriales o páginas de programación.

Este proyecto consiste en un juego de carreras que básicamente lo que hay que hacer es tratar de esquivar vehículos enemigos, para no perder vidas e ir acumulando puntaje conforme avanza el tiempo, el objetivo es alcanzar el mejor puntaje posible, para iniciar el jugador deberá ingresar su nombre de usuario en una entrada de texto que tendrá disponible en la interfaz del menú del juego, en esta interfaz también encontrará botones que lo llevarán a distintas ventanas como la de instrucciones, la de los créditos y la de los puntajes más altos, en la ventana de las instrucciones se muestra una breve explicación de cómo ganar en el juego y también aparecerán los controles asignados para mover el carro del jugador, en la ventana de los puntajes saldrán los mejores 7 puntajes obtenidos y por ultimo está el botón de la ventana que muestra los créditos del juego.

Este juego ha sido desarrollado usando el lenguaje de programación Python en su versión (3.8.2), este lenguaje tiene la ventaja que para los principiantes puede que sea un poco más sencillo de aprender que otros lenguajes de programación ya que este es de cierta forma más simplificado y rápido que otros, también tiene la ventaja de tener muchas librerías incluidas. Para este proyecto las librerías de interfaz que podíamos utilizar eran Pygame y Tkinter, en mi caso escogí la librería de interfaz Tkinter, ya que la había usado anteriormente y asumí que sería más fácil de emplearla para realizar mi proyecto.

Marco Teórico

En este proyecto fue necesario importar varias librerías para que el juego pudiera funcionar correctamente, las librerías son funciones que vienen predeterminadas dentro de un lenguaje que se encargan de facilitar la manera de programar y logran dar una mejor funcionalidad al programa diseñado.

Estas fueron las bibliotecas que se utilizaron para la realización de este proyecto.

```
from tkinter import *
import time
import winsound
import threading
from threading import Thread
from tkinter import messagebox
```

Explicación de las bibliotecas importadas para poder realizar el juego.

1. Biblioteca tkinter:

Esta es la principal para este proyecto ya que es una de las bibliotecas del lenguaje de programación Python que se encarga de la interfaz gráfica de usuario (GUI).

¿Qué nos permite hacer esta librería en el proyecto?

Permite lo que es hacer toda la interfaz grafica de usuario como por ejemplo crear las diferentes ventanas necesarias para el funcionamiento correcto del juego, también incluye funciones dentro de ella como la función de crear botones en los que se puede configurar un comando para que realice una función, además de esto tiene la función de Canvas que es el apartado orientado a objetos, se pueden crear imágenes o formas geométricas que se pueden posicionar en distintas coordenadas de la ventana y también se pueden mover o modificar, otra función integrada en la librería tkinter es la función de Label que se usa para crear por ejemplo textos en la interfaz de la ventana, esta es muy similar a la función Entry que viene con esta biblioteca, esta sirve para crear entradas de texto para que el usuario ingrese datos.

Ejemplos de código de funciones integradas en tkinter.

```
self.gameName = Label(frame, text="CRAZY ROAD", font=("Berlin Sans FB Demi", 50), fg="black", bg="#9fc9ea")
self.gameName.place(x=10, y=10, width=500, height=120)

self.entryname = Entry(frame, textvariable = playerName)
self.entryname.place(x=170, y=170, width=200, height=30)

self.creditsButton = Button(frame, text=("CREDITS"), fg="black", bg="DeepSkyBlue2",
                             font=("Berlin Sans FB Demi", 14), highlightthickness=0,
                             command = self.openCreditsScreen )
self.creditsButton.place(x=200, y=380, width=140, height=30)

self.c= Canvas(self.gameScreen,width=700,height=800)
self.c.place(x=0,y=0)
self.background = self.c.create_image(350,400,image=mapGAME)
```

2. Biblioteca time:

Esta biblioteca es la encargada de cargar todas las funciones integradas relacionadas a el tiempo como la hora y la fecha.

De esta biblioteca se utilizó en este proyecto el módulo (time.sleep()),este sirve para determinar el tiempo en el que se quiere detener una función y/o sirve para establecer con la velocidad a la que se quiere ejecutar el programa, este módulo es muy importante para que el juego se vea más dinámico y no vaya todo a una misma velocidad.

`time.sleep(0.02)` Este es un ejemplo de uso en el código del proyecto.

3. Biblioteca winsound:

Esta se encarga específicamente de dar funciones para cargar, reproducir o detener sonidos, los formatos deben de estar en formato de audio .WAV y es importante para que el programa tenga más emoción y hayan efectos de sonido en el juego, para este proyecto se usó winsound.PlaySound() que es el encargado de los sonidos.

`winsound.PlaySound("CRASH.wav",winsound.SND_ASYNC)`

4. Biblioteca threading:

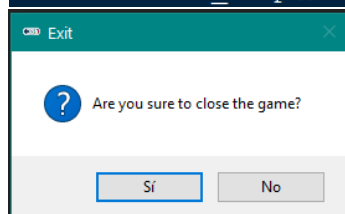
Esta biblioteca se encarga de crear hilos, los hilos son funciones que hacen que alguna función deseada se repita tantas veces o tanto tiempo se quiera hasta detenerla, en el juego se uso para crear los hilos de reproducir la música y para los hilos de movimiento que llamaban a funciones recursivas para poder repetir las llamadas de la función y obtener un movimiento fluido, al igual que llamar a la función que movía el fondo.

```
def startTHREAD():
    thread = Thread(target=moveENEMY,args=(self,))
    thread.daemon = True
    thread.start()
```

5. MessageBox de tkinter:

Este apartado de la biblioteca tkinter sirve para llamar cuadros de mensaje que tienen 2 opciones (si y no) sirve como botones, pero en forma de cuadro de mensaje hay que validar si el valor es true o false y depende de eso que ejecute una función o no.

```
def exitGame():
    #ask if you wanna close the window
    value = messagebox.askquestion("Exit","Are you sure to close the game?")
    if value == "yes":
        window.destroy()
        music_stop()
```



Este es un ejemplo de lo que muestra en interfaz.

Propuestas de diseño de solución**Tkinter en vez de Pygame:**

Use tkinter en vez de Pygame ya que anteriormente había realizado un trabajo con la biblioteca de GUI tkinter y había investigado bastante sobre esta biblioteca entonces tenía al menos las bases, en cambio nunca he utilizado Pygame y si lo hubiera hecho tendría que haber empezado desde cero.

Clases en vez de definición:

En este proyecto para definir las ventanas utilice las clases en vez de las funciones para abrir una ventana normal, tal vez era más sencillo usar solo funciones, pero me gustaba más el orden o la estructura que tenía el código, me parecía que era más fácil buscar todo por sección.

Recursividad simple en vez de Iteración

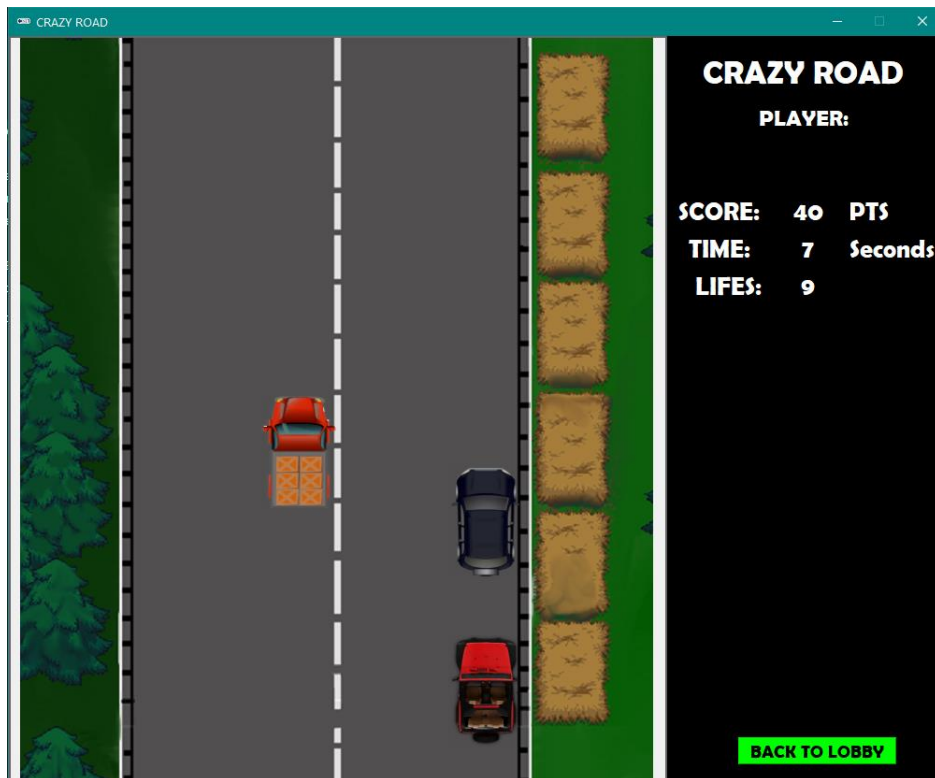
Otra posible solución que se pudo haber implementado fue utilizar iteraciones utilizando el for y el while, en vez de tener que hacerlo con recursividad simple, pero no se podía implementar por no haberlo visto hasta el momento en el curso.

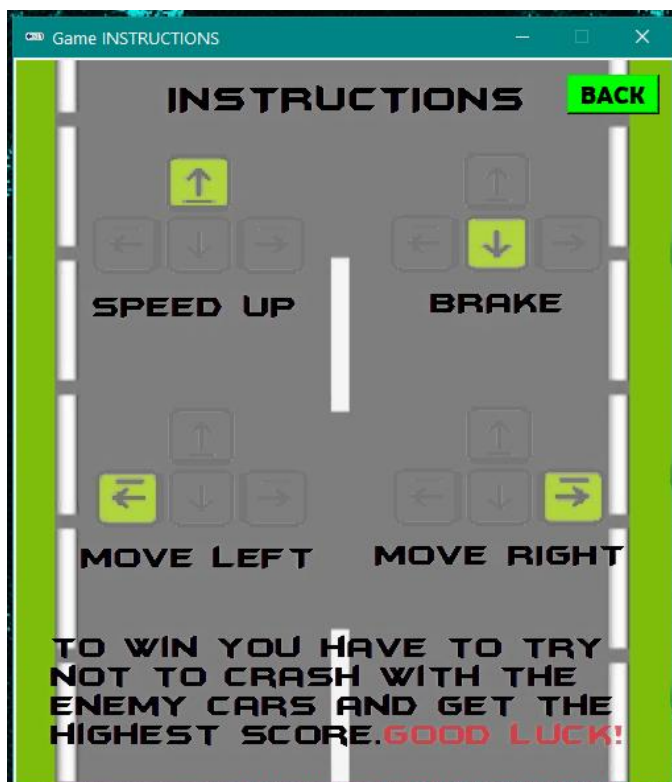
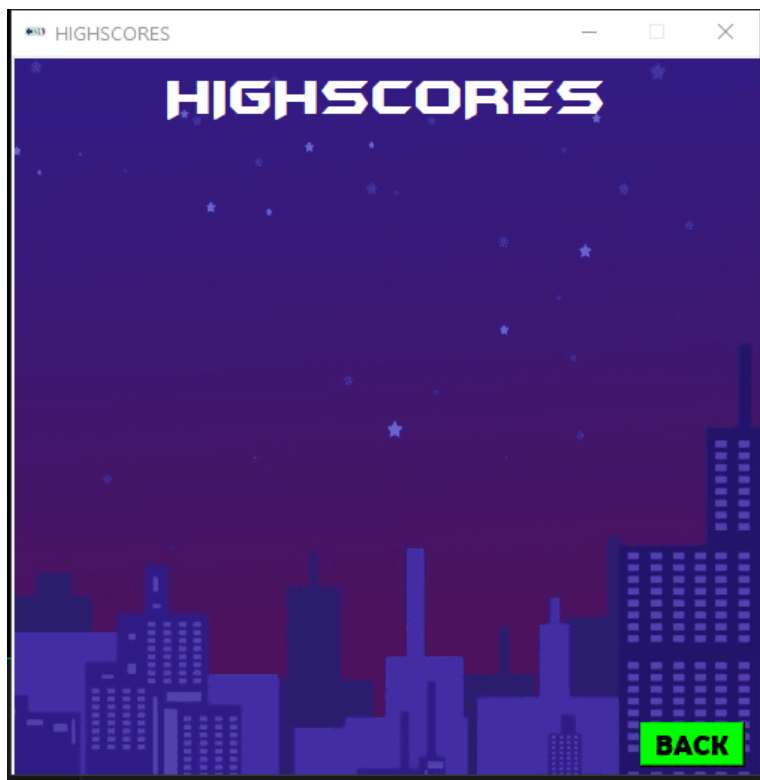
Ejemplos de funcionamiento

Ventana de Menú.



Ventana de Juego.



Ventana de Instrucciones.**Ventana de Puntajes.**

Ventana de Créditos.**Ventana de Game-Over.****Conclusiones**

- Se logró crear un juego de reflejos con una interfaz simple desarrollada con la librería de Python Tkinter y otras librerías de Python.
- El proyecto promovió la investigación necesaria para aprender como implementar las diferentes bibliotecas de Python de la mejor manera y que funcionara el juego y las funciones creadas.
- Con este proyecto aprendí muchas cosas nuevas como los hilos, aprender a manejar mejor las clases, también a detectar colisiones y modificar variables. En fin, logre obtener un gran conocimiento gracias a la investigación que tuve que hacer para comprender mejor las cosas.