

PYNG PONG - Sprint 1

Muñoz Bejarano, Joan Alejandro - Ing Electrónica

Quintero Ovalle, Jose Santiago - Ing Electrónica

Universidad Nacional de Colombia

Sede Bogotá

joamunozbe@unal.edu.co

jquinteroov@unal.edu.co

Resumen - Se definieron unas historias de usuario del desarrollo del proyecto, cada historia necesita de unos requerimientos para ejecutarse que a su vez requiere en algunos casos de librerías específicas.

Abstract - Some user stories for the development of the project were defined; each story needs requirements to be executed, which in turn requires specific libraries in some cases.

Palabras Clave - Librería, Requerimiento, Interfaz, Físicas, Datos, Ventanas

Keywords - Library, Requirements, interface, physics, data, windows

I. HISTORIAS DE USUARIO

1. **Como** usuario **quiero** que el juego se ejecute en una ventana **para** utilizar el programa de una forma óptima y agil

- Agregar una ventana como ejecutable del programa
- Agrupar toda la ejecución del código dentro de la ventana definida

2. **Como** desarrollador **quiero** modificar la ventana de ejecución (raíz) en tamaño, título, logotipo, color, entre otros **para**

estipular las características necesarias de la ventana

- Definir un título para la ventana del ejecutable
- Asignar un tamaño no modificable a la ventana
- Colocar un archivo.ico como logotipo de la ventana y del juego
- Dar color a la ventana

3. **Como** usuario **quiero** crear un perfil de usuario en el juego **para** loguearme con mi perfil cada vez que inicie el juego y tener mi registro en el juego

- Establecer usuario y contraseña
- Implementar la lectura de datos
- Almacenar el usuario en un archivo.txt
- Asociar los datos del juego al perfil

4. **Como** usuario **quiero** tener un logueo al inicio del programa **para** identificar el usuario que va a jugar

- Digitar usuario y contraseña
- Leer los perfiles creados
- Brindar acceso al coincidir (condicional)
- Denegar acceso al no coincidir

5. Como usuario quiero iniciar sin necesidad de un perfil para no guardar mi actividad e historial en el juego
- No se guarda información en la base de datos
 - Acceso limitado a funciones como guardar marcadores

6. Como usuario quiero ver un menú con todas las opciones del juego para saber que opciones ofrece el programa y elegir una de ellas
- Implementar una interfaz gráfica que liste las opciones
 - organizar botones en lista

7. Como usuario quiero música de fondo en el menú y efectos de sonido al interactuar con el mismo para darle mayor ambiente al juego
- reproducir loop continuo de fondo
 - implementar sonido al darle click
 - efecto de paso por las opciones

8. Como usuario quiero que cada opción del menú sea un botón para hacer más interactivo el mismo
- Interfaz grafica para el menú
 - representar cada opción con un botón
 - acción específica para cada botón

9. Como usuario quiero la opción de salir del juego en el menú para terminar la ejecución del programa
- Implementar cierre del programa
 - asociar botón con el cierre del programa

10. Como usuarios **quiero** configurar las

partidas antes de comenzar **para** definir las reglas

- Establecer a cuántos puntos el partido
- Definir número de sets
- Estipular la velocidad del juego

11. Como usuario **quiero** personalizar la apariencia de la partida **para** darle mayor personalización al juego

- Importar diseños 2D de skins de raquetas y pelotas
- Realizar diseños 2D de escenarios
- Elegir skin y escenario de juego previo a la partida

12. Como usuario **quiero** que la interfaz de las partidas tenga las raquetas, pelota y mesa con diseño 2D **para** simular las partidas de ping pong

- Imprimir raquetas, escenario y pelota con su diseño
- Asignarles físicas para interactuar entre sí

13. Como usuario **quiero** la opción de partida contra la CPU **para** jugar un solo jugador

- Asignar un botón para la opción
- Programar un código autónomo de movimiento para la CPU
- Programar movimiento del jugador por medio de inputs por teclado

14. Como usuario **quiero** la opción de partida contra otro jugador local **para** jugar en multijugador

- Asignar nuevos inputs para controlar al jugador 2
- Simultanear la actividad de ambos jugadores

15. **Como** usuario **quiero** la opción de partida 2vs2 **para** jugar multijugador en equipo

- Duplicar el comportamiento autónomo de la CPU
- Utilizar los inputs de los jugadores para un mismo equipo

16. **Como** usuario **quiero** un marcador en la **parte** superior para contabilizar los puntos del partido

- Establecer los dos contadores
- Delimitar los contadores siguiendo la condición de partida
- Imprimir los puntajes con diseño y tamaño
- Guardar el puntaje en la database

17. **Como** usuario **quiero** efectos de sonido en colisiones, anotación de puntos y fin del partido **para** darle un sonido característico a cada acción

- Importar efectos de sonido
- Asignar un efecto a cada evento (colisión, anotación de puntos, fin de partidas)

18. Como desarrollador quiero implementar físicas para las colisiones de la pelota en las partidas

- sistema para calcular el rebote
- detección de colisiones
- ajustar velocidad y dirección de la pelota
- fuerza aplicada de la raqueta
- actualizar posición de la pelota cada frame

19. Como usuario quiero un modo de entrenamiento para practicar la mecánica del juego

- implementar pared para el modo
- velocidad aumenta gradualmente
- límite de velocidad

20. Como usuario quiero guardar mis marcadores para tener un registro de los puntajes al iniciar mi perfil

- guardar en base de datos
- asociar marcadores con perfiles
- agregar resultados al registro

21. Como usuario quiero una opción en el menú que me permita ver mis marcadores

- agregar botón con la etiqueta ver marcadores
- diseñar tabla
- ordenar marcadores de mayor a menor

II. LIBRERÍAS

1. **Tkinter:** Es una biblioteca estándar de Python que se utiliza para crear interfaces gráficas de usuario (GUIs). Permite a los desarrolladores diseñar ventanas, botones, cuadros de texto, menús y otros elementos interactivos en aplicaciones de escritorio. Tkinter proporciona una forma sencilla de crear aplicaciones con interfaces visuales sin necesidad de depender de bibliotecas externas, facilitando la creación de programas con una interacción más amigable para el usuario [1].

Ejemplo:

```
from tkinter import *  
Vent=Tk()  
Vent.title("PRUEBA")  
Vent.geometry("700x400")  
Vent.mainloop()
```

Ejecuta una ventana de titulo "PRUEBA" de 700x400

2. **Pygame:** Es una biblioteca de Python diseñada para la creación de videojuegos y aplicaciones multimedia. Proporciona herramientas para trabajar con gráficos 2D, sonido, música y entrada del usuario,

permitiendo a los desarrolladores crear juegos, simulaciones y otras aplicaciones interactivas. Con Pygame, es posible gestionar la animación, detectar colisiones, manejar eventos y controlar el flujo del juego de manera sencilla, lo que la hace ideal para proyectos de desarrollo de videojuegos en 2D [2].

3. **Pyrebase:** Es una opción más sencilla para integrarse con Firebase desde aplicaciones cliente. Ambos permiten a los desarrolladores interactuar con la base de datos de Firebase, realizar consultas, subir archivos y manejar usuarios, todo dentro de una aplicación [3] Python.

Ejemplo:

```
import pyrebase
```

```
# Configuración de Firebase
```

```
config = {  
    "apiKey": "tu_api_key",  
    "authDomain": "tu_auth_domain",  
    "databaseURL": "tu_database_url",  
    "projectId": "tu_project_id",  
    "storageBucket": "tu_storage_bucket",  
    "messagingSenderId":  
    "tu_messaging_sender_id",  
    "appId": "tu_app_id",  
    "measurementId": "tu_measurement_id"  
}
```

```
# Inicializar Firebase
```

```
firebase = pyrebase.initialize_app(config)
```

```
# Obtener la referencia de la base de datos  
db = firebase.database()
```

```
# Escribir datos en la base de datos
```

```
data = {"nombre": "Juan", "edad": 30}  
db.child("usuarios").push(data)
```

```
# Leer datos de la base de datos
```

```
usuarios = db.child("usuarios").get()  
for usuario in usuarios.each():  
    print(usuario.val())
```

4. **Pillow:** Es la biblioteca más popular y fácil

de usar para abrir, manipular y guardar imágenes. Se trata de un fork de la antigua Python Imaging Library (PIL), y es muy adecuada para realizar tareas simples como abrir una imagen, redimensionarla, aplicar filtros, o guardarla en otro formato [4].

5. **NumPy:** Es la biblioteca fundamental para la computación científica en Python. Proporciona soporte para matrices y operaciones matriciales, lo que lo hace esencial para la programación numérica y científica [5]

6. **Panda:** Es una biblioteca de análisis de datos que proporciona estructuras de datos flexibles y herramientas de manipulación y análisis de datos. Es ampliamente utilizado en la limpieza y análisis de datos [4].

REFERENCIAS

[1] Amos, D. (2020). Python gui programming with tkinter. Tersedia: <https://realpython.com/python-gui-tkinter>, 71.

[2] Sweigart, A. (2012). Making Games with Python & Pygame.

[3] <https://www.pygame.org/>

[4] <https://docs.python.org/es/3/library/index.html>

[5] HARRIS, Charles R., et al. Programación de matrices con NumPy. Nature , 2020, vol. 585, núm. 7825, págs. 357-362.