Samuel Nathaniel Guasco Cedeño

Lógica de Programación

02/03/2025

Evaluación en Contacto con el Docente

Índice

1. Introducción

Este informe detalla el desarrollo del juego "Piedra, Papel o Tijeras" como parte del proyecto final de la materia. Durante ocho semanas de trabajo, aplicamos los conocimientos adquiridos en programación para crear un juego funcional con múltiples características y mejoras progresivas.

El propósito principal de este proyecto fue fortalecer nuestras habilidades en lógica de programación, estructuración de código y resolución de errores**.** Además, nos permitió desarrollar una mentalidad analítica para optimizar procesos y mejorar la experiencia del usuario mediante validaciones adecuadas y una interfaz de texto clara.

A lo largo del desarrollo, aplicamos distintos conceptos como el uso de funciones, validaciones de entrada, estructuras de control y temporizadores**,** lo que nos permitió crear un programa robusto y modular. Gracias a la retroalimentación del foro y del profesor, fuimos refinando el código hasta alcanzar una versión optimizada y sin errores críticos.

2. Desarrollo del Proyecto

El proyecto se dividió en fases de desarrollo que se alinearon con el cronograma de la materia. A continuación, describimos el proceso llevado a cabo durante las ocho semanas:

Semana 1-2: Aprendizaje teórico y desarrollo de diagramas de flujo y UML. Estudiamos la lógica del juego antes de implementarla en código.

Semana 3-4: Creación de la estructura base del código en Python, incluyendo funciones principales y un menú interactivo.

Semana 5-6: Implementación de validaciones para evitar errores como “IndexError” optimización de entradas y corrección de bucles.

Semana 7-8: Incorporación de un temporizador, mejoras en el formato de la interfaz de texto y ajustes en base a la retroalimentación del foro.

2.1 Herramientas utilizadas:

Lenguaje de programación: Python.

Entorno de desarrollo: Visual Studio Code.

Control de versiones: GitHub, para gestionar cambios y compartir el código.

Librerías:

- Random: Para la generación de elecciones aleatorias.
- Getpass: Para ocultar la entrada en modo multijugador.
- Threading: Para implementar el temporizador en paralelo.
- Time: Para controlar pausas y definir tiempos límite.
- Sys: Para gestionar entradas del usuario después de que se agote el tiempo.

3. Características del Juego

El juego fue desarrollado en Python, utilizando una arquitectura modular basada en funciones para garantizar claridad y eficiencia en el código.

3.1 Modos de juego implementados:

Jugador vs Computadora: Se enfrenta a un oponente virtual que elige opciones de manera aleatoria.

Modo Multijugador: Dos jugadores ingresan sus opciones de forma secreta para mayor emoción.

3.2 Validaciones Implementadas:

Se restringieron entradas inválidas “IndexError”.

Se añadió un temporizador que determina el resultado si un jugador no responde a tiempo.

Se mejoró la experiencia del usuario con mensajes claros y amigables.

3.3 Estadísticas del juego:

Se almacena el historial de partidas ganadas, perdidas y empates.

Se ofrece un registro detallado de cada ronda para referencia del usuario.

4. Resultados y Logros

Durante el desarrollo del proyecto, logramos transformar una idea simple en un juego completamente funcional con múltiples características y validaciones que mejoraron la experiencia del usuario. Uno de los principales logros fue la creación de una estructura modular bien organizado, lo que nos permitió agregar funcionalidades de manera progresiva sin comprometer la estabilidad del código. Además, implementamos un sistema de validaciones robusto que evita errores comunes, como selecciones fuera de rango “IndexError” y entradas inválidas, garantizando así una interacción fluida y sin fallos para los jugadores.

Otro logro importante fue la incorporación del **temporizador**, lo que añadió una mecánica extra de desafío al juego. Esto permitió que los jugadores tuvieran un tiempo límite para tomar decisiones, evitando que una partida se quede detenida indefinidamente. También logramos optimizar la gestión de las estadísticas de juego, permitiendo un seguimiento detallado de las victorias, derrotas y empates en cada sesión. Finalmente, la retroalimentación recibida en el foro nos ayudó a perfeccionar la lógica del programa, hacer correcciones en la experiencia del usuario y asegurar que el código fuese lo más eficiente posible.

Conclusiones y Reflexiones

El desarrollo del juego "Piedra, Papel o Tijeras" nos permitió fortalecer nuestras habilidades en programación estructurada, depuración de errores y optimización de código. A través de este proyecto, aprendimos la importancia de una buena planificación antes de escribir código, ya que la creación de diagramas de flujo y UML al inicio facilitó la implementación y organización de las funciones. También comprendimos la relevancia de validar adecuadamente las entradas del usuario, lo que mejora significativamente la usabilidad del software y previene fallos inesperados.

Más allá de la parte técnica, este proyecto nos dejó importantes aprendizajes sobre trabajo, iteración en el desarrollo de software y la importancia del feedback. La implementación de mejoras basadas en la retroalimentación del foro y del profesor fue clave para perfeccionar nuestro código, y nos enseñó que la optimización y depuración son procesos fundamentales en cualquier desarrollo de software. En el futuro, podríamos expandir este proyecto con una interfaz gráfica y nuevas funcionalidades, aplicando lo aprendido para seguir mejorando nuestras habilidades en programación.

Referencias

Beazley, D. M., & Jones, B. K. (2013). *Python Cookbook*. O'Reilly Media.

Lutz, M. (2013). *Learning Python*. O'Reilly Media.

Van Rossum, G., & Drake, F. L. (2009). *The Python Language Reference Manual*. Network Theory Ltd.

Pilgrim, M. (2004). *Dive into Python*. Apress.

McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.

Downey, A. B. (2012). *Think Python: How to Think Like a Computer Scientist*. O'Reilly Media.

Martelli, A. (2006). *Python in a Nutshell*. O'Reilly Media.

Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.

Python Software Foundation. (2024). *Python Documentation*. Recuperado de https://docs.python.org/3/

Stack Overflow. (2024). *Threads in Python: Best Practices*. Recuperado de https://stackoverflow.com