**Portada**

* **Título del Proyecto**: Battle Ship Guerra de Mundos
* **Autor:** Donald Miranda
* **Fecha:** 21/08/2024
* **Institución:** Universidad Técnica Nacional
* **Curso:** Principios de Programación

**Índice**

1. Introducción
2. Objetivo General
3. Metodología del Trabajo
4. Situaciones Adversas
5. Conclusión
6. Bibliografía

**Introducción**

Guerra de mundos es mi proyecto hecho en Python que se basa en una batalla de vehículos de guerra en mar abierto donde se utilizara un tablero 8x8 donde el usuario indicara que posición desea atacar y así empezar una batalla entre el usuario defensor y el usuario invasor.

El proyecto fue elegido por el motivo de que es un desafío complementar lógicas que hagan funcionar este tipo de código, en este caso este juego, es divertido y como dije antes todo un desafío hacerlo funcionar sin errores, por lo tanto, se me hizo mas agradable este proyecto que la opción 1.

**Objetivo General**

El objetivo general de este proyecto es desarrollar un juego de Battle Ship utilizando el lenguaje de programación Python, con una dinámica fácil, generado a partir de un tablero de juego donde el jugador puede indicar las coordenadas para atacar, también la implementación de reglas y decisiones donde el algoritmo analiza constantemente condiciones para una buena ejecución del juego y por ultimo neutralizar, corregir o manejar cualquier error generado por el usuario para una excelente experiencia a la hora de jugar.

**Metodología del Trabajo**

1. **Diseño del Tablero**: Crear un tablero de juego utilizando listas en Python.
2. **Posicionamiento de Barcos**: Implementar una función para colocar los barcos en el tablero de manera aleatoria.
3. **Interfaz de Usuario**: Desarrollar una interfaz de texto donde el jugador pueda ingresar a un menú agradable y pueda elegir entre algunas opciones como iniciar el juego, por ejemplo.
4. **Lógica del Juego**: Programar la lógica para verificar si un ataque es un acierto o un fallo y actualizar el tablero en consecuencia, también implementación de condiciones que serán constantemente analizadas por el algoritmo para comprobar si hay ventajas entre un usuario u otro o si simplemente x usuario ya perdió o ganó.
5. **Pruebas y Depuración**: Realizar pruebas exhaustivas para asegurar que el juego funcione correctamente y depurar cualquier error encontrado.

**Situaciones Adversas**

Durante el desarrollo del proyecto, se presentaron varias situaciones adversas, tales como:

* **Errores en la colocación de barcos**: Inicialmente, los barcos no se generaban aleatoriamente.
* **Problemas con la actualización del tablero**: Hubo dificultades para manejar las verificaciones de estadísticas y la actualización constante del tablero.
* **Depuración de errores**: Identificar y corregir errores en la lógica del juego fue un desafío constante.

**Conclusión**

El desarrollo de este juego de Battle Ship en Python permitió aplicar y reforzar conocimientos en programación, manejo de listas, y desarrollo de algoritmos. A pesar de las dificultades encontradas, el proyecto fue exitoso y el juego funciona según lo esperado.

**Bibliografía**

* Documentación oficial de Python
* Tutoriales y recursos en línea sobre desarrollo de decisiones.

En este apartado destaco mi constante insistencia por crear el juego con la mínima ayuda posible y al final el resultado fue todo un éxito y orgullo ya que requerí una ayuda de paginas o tutoriales extremadamente mínima, por eso no señalo ni indico links de ello.