

U. Sergio Arboleda

Parcial n°1 Pensamiento Algorítmico Go2

-Luis Miguel López Acero

-Julián Ponguta

1.1. Análisis del problema

Descripción del problema:

- Realizar el aumento de una variable (el demonio) que persigue al jugador en determinados tiempos en m/s sabemos que para calcular el tiempo teniendo la distancia y la velocidad es:

Identificación de los requisitos funcionales y no funcionales:

Funcionales:

- Datos de entrada al programa
- Ingresar el nombre del demonio
- Ingresar la distancia que queremos saber evaluando el tiempo.

No funcionales:

- Las operaciones que hacen dentro del código a lo que ingresemos (con procesos del código)
-

Análisis de casos de uso principales:

Identificación de entradas, procesos y salidas esperadas

- entrada: nombre de los demonios.
- distancia del

1.2. Justificación de la solución:

Explicación detallada de la estrategia elegida para resolver el problema:

La estrategia a tomar fue leer detenidamente el documento, analizar la pregunta problema y realizar el código con operador lógico if-elif (creo que también podría haber usado el operador "while, do while" pero no me quedo muy claro en clase.

Usando el if-elif hacemos una cadena en donde el flujo hace varios pasos para comprobar una condición en este caso (que nos de el valor de m/s)

El programa funciona en visual studio code

(en el proceso de ejecución no vota ningún error)

Justificación de las estructuras de datos y algoritmos seleccionados:

- Condicionales if-elif
Si una de las variables da como resultado falso el programa nos dice "tipo de demonio no valido " esto hace una serie de comprobaciones y nos da un único

valor de verdadero tomando el valor y el nombre del demonio como verdadero y así se cumpla la condición

Comparación con posibles soluciones alternativas y razones de la elección final:

- lo intenté hacer con el (while, for, do while) pero realmente a mi compañero y a mí se nos hizo complejo implementarlo

3.1. Documentación Python

Comentarios explicativos para secciones complejas:

- Si los use en el código están claramente evidenciados (no se argumentar esto profe)

Instrucciones claras para ejecutar el programa (En los distintos sistemas operativos):

Windows:

1. **Guarda el código:** Guarda el código Python en un archivo llamado .py.
2. **Abre la terminal:** Abre el "Símbolo del sistema" (cmd).
3. **Navega a la carpeta:** Usa cd para ir a la carpeta donde guardaste .py.
4. **Ejecuta:** Ejecuta python .py.
5. **Ingresa datos:** Sigue las instrucciones en pantalla.(lo que pide el sistema)
6. **Presiona Enter:** Para salir.

Linux:

1. **Guarda el código:** Guarda el código Python en un archivo llamado .py.
2. **Abre la terminal:** Abre la terminal.
3. **Navega a la carpeta:** Usa cd para ir a la carpeta donde guardaste .py.
4. **Ejecuta:** Ejecuta python3 .py (o python .py si Python 3 es tu versión predeterminada).
5. **Ingresa datos:** Sigue las instrucciones en pantalla.
6. **Presiona Enter:** Para salir.

Ejemplos de uso:

3.2. Documentación C++

Comentarios explicativos para secciones complejas:

- Si los use en el código están claramente evidenciados (no se argumentar esto profe)

Instrucciones claras para compilar y ejecutar el programa (En los distintos sistemas operativos):

C++

Windows (con MinGW o WSL):

1. **Guarda el código:** Guarda el código C++ en un archivo llamado .cpp.
2. **Abre la terminal:**
 - Si usas MinGW, abre "MinGW Shell".
 - Si usas WSL, abre la terminal de tu distribución de Linux.
3. **Navega a la carpeta:** Usa el comando `cd` para ir a la carpeta donde guardaste .cpp.
4. **Compila:** Ejecuta `g++ .cpp -o .exe`.
5. **Ejecuta:** Ejecuta .exe.
6. **Ingresa datos:** Sigue las instrucciones en pantalla.(lo que pide el programa)
7. **Presiona Enter:** Para salir.

Linux:

1. **Guarda el código:** Guarda el código C++ en un archivo llamado .cpp.
2. **Abre la terminal:** Abre la terminal.
3. **Navega a la carpeta:** Usa `cd` para ir a la carpeta donde guardaste .cpp.
4. **Compila:** Ejecuta `g++ .cpp`
5. **Ejecuta:** Ejecuta `./nombre` como lo guardaste.
6. **Ingresa datos:** Sigue las instrucciones en pantalla.(lo que pide hacer el programa)
7. **Presiona Enter:** Para salir

Ejemplos de uso: