Informe Parcial

Nicolas Parra Moya

Johan Ramirez Camargo

Andres Ducuara

Pensamiento Algoritmico

Universidad Sergio Arboleda

Resumen:

El presente informe busca dar resolución a un ejercicio de programación. Creando un análisis, una implementación en dos lenguajes: Python y C + +. Además de la documentación adecuada en un informe.

Palabras clave: Juego, Código, Variables, Condicionales.

Introducción:

Para el primer parcial de Pensamiento Algorítmico se le dio a cada grupo un problema el cual se tenía que resolver. El caso que se presentó fue de un juego llamado Doom.

Desarrollo:

I) Descripción del problema.

“En el clásico juego Doom, los demonios son enemigos que persiguen al jugador con diferentes velocidades dependiendo de su tipo. Tu tarea es calcular cuánto tiempo tardará un demonio en alcanzar al jugador, dada la distancia entre ellos y la velocidad del demonio.”

Se pide hallar el tiempo que tardan en recorrer de ciertos enemigos. Por lo cual se ha de requerir indicar que tipo de enemigos son, y especificarlos para que no haya confusión. Además de asignarles a cada uno su velocidad.

Por lo tanto los requisitos funcionales han de ser la capacidad del programa de permitir solo una entrada específica, definida como cada tipo de enemigo correspondiente (Baron of Hell, Imp, o Zombieman) además de saber identificar en cada uno su variable de velocidad. Aparte, ser capaz de recibir la entrada de la distancia, asignada por el usuario respectivamente. Para que así, al tener los dos datos de entrada, ejecute el programa y se logre obtener el tiempo en que se tardaría. Cuál serían las fechas de salida.

Si al contrario no es el caso de que los valores de entrada fueran alguna de las variables ya definidas como lo son los enemigos previamente mencionados, que el programa muestre un mensaje de que la variable ingresada (El tipo de demonio) no es válida.

II) Solución del problema.

Se quiso diseñar un código sencillo, siguiendo el temario visto durante las sesiones de clase. Por ende, hasta el momento del desarrollo de este informe se ha avanzado en el tema de estructuras de selección.

La metodología de solución fue de manera manual. A fuerza bruta. Por lo tanto no se utilizó ningún algoritmo para la realización del código . En un primer momento se buscó desglosar el problema. Esto identificando que se tenía, que se quería hacer y cuál era el resultado esperado:

- Lo que se tenía o sabía era acerca de sobre qué variables se iba a trabajar (Tipo\_demonio: Zombieman, Baron of Hell y Imp).

- Después el planteamiento del problema era el cálculo del tiempo. Para eso, ya se sabía que esta era la razón de cambio entre la distancia y la velocidad. Por lo que se tuvo algo con lo que iniciar.

- Por último se vio que el resultado esperado era el tiempo. Pero, también al mismo tiempo se tuvo la cuestión de qué pasaría si el valor resultante no fuese exacto. Por lo tanto se buscó definir la variable tiempo como un decimal.

Debido a esto se planteó un código que ingrese un primer valor de entrada, buscando definir una variable (tipo\_demonio). Además de tener una segunda entrada que pueda recibir el valor de distancia desde la cual se va a medir el tiempo.

Se inicia la velocidad en 0. Esto para tener un parámetro si es que la variable ingresada no es válida.

Se usan una serie de condicionales **if** y **elif** para verificar que el tipo de demonio sea el correcto, y asignar la variable de velocidad respectiva de cada uno.

Además, se agrega un condicional para comprobar que el tipo de enemigo sea válido. Es decir, si el demonio no es alguno de los ya mencionados la velocidad será cero y se mostrará un mensaje de: “Tipo de demonio no válido”. Si por el contrario se verifica que la condición se cumple. se calculará el tiempo usando la fórmula.

**Flowchart**

