

MANUAL TÉCNICO

ESCAPE DEL LABERINTO

Proceso EscapeLaberinto():

Este proceso representa el método main dentro del código en java, en la cual se utiliza para inicializar los demás procesos, este proceso inicia llamando al Proceso menuPrincipal().

Proceso menuPrincipal():

En este proceso se muestra dentro de un ciclo while las cinco opciones que tiene el juego como: jugar, crear mapas, reportes, visualizar mapas y salir. La elección del menú se reciben en una variable llamada opcion, que sera introducida en una condicional de if else. Donde si la opcion es igual 1 entonces comienza el juego donde se llama a la función jugar(), si la opcion es 2 entonces el usuario entrará a un menú donde podrá crear un mapa a gusto personal y también podrá colocarle un nombre llamando al proceso tamañoMapa(), seguidamente de diseñoMapa() y mostrarMapa(), si la opcion es igual a 3 se mostraran las estadísticas generales del juego llamando al proceso reporteGeneral(), si la opcion es igual a 4 el usuario podrá visualizar el mapa que esta creado por defecto y los mapas creados por él, estos mapas se mostraran a través del llamado del proceso mostrarMapa(), si la opcion es 5 entonces se le mostrara al usuario un mensaje “Saliendo del Juego” y terminara la ejecución del programa.

Función jugar()

Se encarga de ejecutar la función principal del juego que permite al usuario elegir un mapa para jugar, desplazarse por medio de comandos y al finalizar el juego se mostrarán as estadísticas finales, esto lo logra hacer haciendo uso de la llamada al proceso juego(), también se le mostrará al usuario que presionando el número 1 puede regresar al menú principal haciendo uso de la llamada al proceso menuPrincipal(), siempre y cuando no haya comenzado a jugar o haya terminado el juego.

Proceso partida()

Proceso donde el jugador tiene un turno al igual que el bot, al comenzar el juego se le da el primer turno al jugador y se le muestra los comandos que puede ejecutar, después de ejecutarlo se le da el turno al bot y este ejecuta un comando el cual no es visible para el usuario.

Proceso bot()

El bot es un jugador del laberinto la cual es controlado por el algoritmo del juego, su misión principal es atrapar al jugador, el bot a igual que el usuario puede ejecutar comandos pero no son visibles en pantalla, el jugador puede estar a una distancia máxima de 5x5 de la posición del jugador, si el bot ejecuta el comando mirar entonces en el siguiente turno avanzará una posición hacia donde está el jugador.

Proceso turnos()

Este proceso regula los turnos del usuario y del bot, haciendo que uno pueda ejecutar alguna acción después del otro, teniendo un orden y cumpliendo con las reglas del juego.

Proceso mapaPrincipal()

Este proceso se encarga de crear el mapa inicial que será utilizado en caso de que el usuario no desee crear algún otro mapa, este tiene una dimensión de 30x30 con varias salidas y caminos que el usuario puede utilizar.

Proceso mapasGuardados()

Este proceso se encarga de guardar el mapa inicial que viene por defecto en el juego como también los mapas creados por el usuario.

Proceso tamañoMapa()

Proceso que se encarga de la creación del tamaño de los mapas, a través de variables como; filas, columnas y mapas[espacio][][] el usuario puede definir el tamaño del mapa que desea. Para definir el tamaño se utiliza el número de filas y columnas que ingresa el usuario y haciendo uso de un ciclo for anidados que se encargan de crear las filas y columnas del arreglo que está contenido dentro del arreglo mapas[espacio], se utiliza un arreglo tridimensional para que el arreglo que almacena los otros arreglos sea la posición en donde se guardara el mapa para poder tener un orden y poder visualizarlo en el proceso mapasguardados(), cuando el ciclo for termina de crear el mapa y la posición del mapa entonces se tendrá un arreglo como sigue: mapas[i] = mapas[filas][columnas] para posteriormente “dibujar” los espacios correspondientes para visualizar el laberinto como tal.

Proceso diseñoMapa()

Este proceso cuenta con variables que definirán visualmente que habrá en el laberinto, es decir, donde habrá una pared, una casilla vacía, una casilla con oro, una salida, la posición del usuario y la posición del bot. En este se hace uso de otros ciclos for anidados que permiten “dibujar” o sobrescribir los valores de los arreglos para así poder crear la posición de las paredes y salidas, al finalizar le pide al usuario colocarle un nombre al mapa para posteriormente guardarlo.

Proceso mostrarMapa()

Se encarga de mostrar visualmente el mapa que el usuario creo y también de poder ser utilizar dentro de la función jugar().

Proceso comandos()

En este proceso primero se le muestra al usuario los comandos que puede utilizar para las diferentes acciones dentro del juego, como ver el oro requerido para poder hacer uso de una de las salidas, oro que posee actualmente, movimientos que puede realizar en 4 direcciones diferentes, levantar el oro en una de las casillas, mirar que hace que se le muestre al usuario una matriz de 5x5 donde podrá visualizar la posición del jugador, del bot y del oro, salir hace que cuando el usuario llegue a una de las salidas pueda ejecutar el comando pueda salir del mapa y ganar siempre y cuando tenga el oro suficiente para hacer uso de esa salida, si el usuario ingresa un comando inexistente se volverá a repetir el turno, pero si ejecuta un comando inexistente 3 veces pierde el juego.

Proceso oroRequerido()

Muestra el oro que se requiere para cada salida que este en el mapa del laberinto, ya sea el mapa inicial o el mapa creado por el usuario.

Proceso movimiento()

Este proceso muestra la forma en la que el usuario puede ingresar el comando respectivo para moverse ya sea arriba, abajo, izquierda o derecha y al igual modo que el proceso comandos() si ingresa 3 veces un comando inexistente pierde el juego.

Proceso levantar()

Se encarga de que cuando el usuario ejecute el comando levantar si éste se encuentra en una casilla donde hay oro pueda recogerlo, de lo contrario se le mostrará un texto donde diga que no ha levantado ninguna cantidad de oro.

Proceso oro()

Es el encargado de almacenar el oro que se vaya recolectando dentro dentro de una variable llamada oroTotal, este dato servirá para que el usuario pueda ver el oro que ha logrado conseguir y también para mostrar el oro recaudado en las estadísticas finales y generales.

Proceso reportesFinales()

En este proceso se mostrará los resultados al terminar una partida ya sea que se haya completado o que el usuario haya perdido la partida, el proceso mostrara datos como: oro recolectado, movimientos realizados, movimientos realizados por el bot, veces que el bot intento atrapar al usuario y las veces que el usuario estuvo a la vista del usuario.

Proceso reporteGeneral()

Este proceso se encarga de mostrar los promedios de las partidas jugadas por el usuario, donde se mostrará la cantidad de veces que el usuario fue atrapado por el bot, cantidad de partidas ganadas, oro promedio ganado por partida, el promedio de movimientos, el mapa más jugado, mapa menos jugado, mapa donde se perdió más, mapa donde más se gano y el total de mapas creados.