**Manual tecnico**

Sistema operativo: Windows 10

IDE: Apache Netbeans 12.6

Lenguaje: Java

JDK: 15

**public class Proyecto {**

//cubo

public static String cubo[][][];

//cubo

static int vecesAtrapado=0;

static int partidasGanadas=0;

static int oroTotal=0;

static int partidasJugadas=0;

static int movimientosTotal=0;

static int mapaMasJugado=0;

static int mapasCreados=0;

static int selectorMapa=0;

static int oroNecesario1=0;

static int filas;

static int columnas;

*Se definen las variables globales que usara el programa:*

1. *Cubo para guardar diferentes matrices/mapas*
2. *Variables de tipo entero para llevar un conteo de las acciones realizadas que servirán para el apartado de “reportes”*

**public static void main(String[] args) {**

int opcion=0;

Scanner scanner = new Scanner(System.in);

Scanner scanner1 = new Scanner(System.in);

while(opcion!=5){

System.out.println("ESCAPE DEL LABERINTO");

System.out.println("Menu Principal");

System.out.println("1. Jugar");

System.out.println("2. Crear mapa");

System.out.println("3. Reportes");

System.out.println("4. Visualizar mapas");

System.out.println("5. Salir del juego");

opcion=scanner.nextInt();

if(opcion==1){

//logica para jugar

partidasJugadas+=1;

int opcionMapa=0;

//Funcion para jugar

System.out.println("Seleccione un mapa: ");

System.out.println("1. Mapa predeterminado");

System.out.println("2. Mapas Creados");

opcionMapa=scanner1.nextInt();

if(opcionMapa==1){

LogicaJuego1(mapaPrederminado());

}else if(opcionMapa==2){

System.out.println("Seleccione cual mapa creado(1,2,3...etc)");

selectorMapa=scanner.nextInt();

LogicaJuegoMapasCreados();

}

}else if (opcion==2){

Crearmapa();

}else if(opcion==3){

Reportes();

}else if(opcion==4){

System.out.println("Que mapa quier ver");

System.out.println("1. Mapa predeterminado");

System.out.println("2. Mapas Creados");

int opcionMapaVer=scanner1.nextInt();

if(opcionMapaVer==1){

MostrarMapa(mapaPrederminado());

}if(opcionMapaVer==2){

System.out.println("Seleccione cual mapa creado quiere ver(1,2,3...etc)");

selectorMapa=scanner.nextInt();

mostrarCubo();

}

}

}

System.out.println("Hasta luego :)");

}

1. *El menú del juego*
2. *Se declara una variable para saber la opción del manu*
3. *Se tiene un if para saber que opción del menú eligió*
4. *Dependiendo del if se realiza cierta función o procedimiento*
5. *En la opción 1 (jugar) se crea una variable para saber en que mapa quiere jugar para luego ejecutar el procedimiento para jugar*
6. *En la opción 2 (crear mapa) entra al proceso creado para hacer mapas personalizados*
7. *En la opción 3 (Reportes) entra al proceso creado para los reportes*
8. *En la opción 4 (Ver mapa) entra al proceso creado para ver los mapas*
9. *En la opción 5 (Salir) se termina el proceso del código*

***public static String[][] mapaPrederminado() {***

*String mapaPre[][]={{"#","#","#","#","#","#","#","#","S","#","#","#","#","#","#","#","#","#","#","#","#","#","#","#","#","#","#","#","#","#"},*

*{"S","O","O","O","O","O","O","#","O","O","O","O","O","O","O","O","O","#","O","O","O","O","O","O","O","O","#","O","O","S"},*

*{"#","#","#","#","#","#","O","O","#","#","#","O","#","#","#","O","#","O","O","#","#","#","#","#","#","O","O","#","O","#"},*

*{"#","O","O","G","O","#","O","#","G","G","#","O","#","G","#","O","#","G","#","O","#","O","O","O","#","#","O","O","O","#"},*

*{"#","O","#","#","O","#","O","O","O","O","#","G","#","O","#","O","#","#","O","O","#","O","#","O","O","#","O","#","O","#"},*

*{"#","G","#","O","O","#","O","#","#","#","#","#","#","O","#","O","O","O","#","O","#","O","#","#","#","#","O","#","O","#"},*

*{"#","O","#","#","O","O","O","O","O","O","#","O","#","O","O","O","O","O","O","O","O","O","#","O","O","O","O","#","#","#"},*

*{"#","#","O","O","O","#","O","#","O","#","O","O","#","#","O","#","#","O","#","#","#","#","#","O","#","O","#","#","G","#"},*

*{"#","O","O","#","O","#","O","O","O","#","O","#","O","#","O","O","#","O","#","O","O","O","O","O","#","O","O","#","G","#"},*

*{"#","O","#","#","O","#","#","#","O","#","O","#","O","#","#","O","#","O","#","#","#","O","#","#","#","#","G","#","O","#"},*

*{"#","O","O","O","O","#","O","#","O","#","O","#","O","#","O","O","#","O","O","G","#","O","O","#","O","#","O","#","O","#"},*

*{"#","#","#","G","#","O","O","O","O","#","O","#","O","#","#","O","#","#","#","#","#","O","#","o","#","#","O","#","O","#"},*

*{"#","O","O","#","O","O","#","#","#","O","O","#","O","#","O","O","O","O","O","O","O","O","O","O","O","#","O","#","O","#"},*

*{"#","#","O","#","#","O","#","O","O","#","O","#","O","O","O","#","O","#","#","O","#","O","O","#","O","#","O","#","O","#"},*

*{"#","O","O","#","O","O","#","O","#","#","O","#","G","#","O","#","#","O","O","#","O","O","#","O","O","#","O","#","O","#"},*

*{"#","#","O","#","#","O","O","O","G","#","O","#","#","#","O","#","O","O","O","O","O","#","O","O","#","#","O","#","O","#"},*

*{"#","O","O","#","O","O","#","O","#","#","O","O","#","O","O","#","O","#","#","#","#","#","O","#","O","#","G","#","O","#"},*

*{"#","#","O","O","O","#","#","#","O","#","O","#","O","#","O","O","O","#","G","O","O","O","O","O","O","O","#","#","O","#"},*

*{"#","G","#","O","#","O","O","O","O","#","O","O","#","#","#","#","#","#","#","#","#","#","#","#","#","O","O","O","O","#"},*

*{"#","G","#","O","O","O","#","#","O","O","O","O","O","G","O","O","O","G","O","O","O","G","O","O","#","#","O","#","O","#"},*

*{"#","O","#","#","#","#","#","O","O","#","#","#","#","#","#","#","#","#","#","#","#","#","#","O","O","#","O","#","#","#"},*

*{"#","O","#","O","O","O","#","O","#","O","O","O","O","O","O","O","O","O","O","O","O","O","O","#","O","#","O","O","O","S"},*

*{"#","O","O","O","#","O","#","O","O","#","O","#","#","#","#","#","#","O","#","#","#","O","O","O","O","#","O","#","#","#"},*

*{"#","#","#","#","#","O","#","O","#","O","O","#","G","O","O","O","#","O","O","O","#","O","#","#","#","#","O","#","O","#"},*

*{"#","O","#","O","O","O","#","O","G","#","O","#","#","#","#","O","#","#","#","O","#","O","O","O","O","#","O","O","O","#"},*

*{"#","O","#","#","#","O","#","#","#","#","O","#","O","O","O","O","#","G","#","O","#","#","#","O","#","G","#","#","O","#"},*

*{"#","O","#","O","O","O","O","O","O","O","O","O","O","#","O","O","#","O","#","O","O","O","O","O","#","O","#","O","O","#"},*

*{"#","O","#","#","#","#","#","#","#","#","#","O","#","O","#","#","#","O","#","O","#","#","#","#","#","O","O","O","#","#"},*

*{"#","O","O","O","O","O","G","#","O","O","O","O","#","O","O","O","O","O","O","O","O","O","O","O","O","O","#","O","G","#"},*

*{"#","#","#","#","#","#","#","#","#","#","#","#","#","S","#","#","#","#","#","#","#","#","#","#","#","#","#","#","#","#"}};*

*return mapaPre;*

*}*

1. *Se define una variable de tipo String para crear una matriz en donde se almacenará el mapa predeterminado, se escribe uno a uno todo lo que lleva la matriz*
2. *Se retorna el mapa Predeterminado*

**public static void LogicaJuego1(String mapaPre[][]){**

int posicionFilaJugador=0;

int posicionColumnaJugador=0;

int posicionFilaBot=0;

int posicionColumnaBot=0;

int opcionJugar=0;

int Oronecesario=13;

//ubicando al jugador

do{

posicionFilaJugador=(int)(Math.random()\*30);

posicionColumnaJugador=(int)(Math.random()\*30);

}while((mapaPre[posicionFilaJugador][posicionColumnaJugador])!="O");

mapaPre[posicionFilaJugador][posicionColumnaJugador]="J";

//ubicando al bot

do{

posicionFilaBot = (int)(Math.random()\*30);

posicionColumnaBot=(int)(Math.random()\*30);

}while((mapaPre[posicionFilaBot][posicionColumnaBot])!="O");

mapaPre[posicionFilaBot][posicionColumnaBot]="B";

Scanner scanner2 = new Scanner(System.in);//scanner para numero

System.out.println("\nBienvenido al juego\n");

System.out.println("Las reglas son las siguentes: \n");

System.out.println("Debe reunir la cantidad de oro que se le solicita");

System.out.println("Debe llegar a una de las salidas con esa cantidad de oro que se le solicita");

System.out.println("Debe evitar ser atrapado por el enemigo");

System.out.println("Si lo atrapa el enemigo pierde pero si llega a la salida con la cantidad de oro necesaria gana y el juego se acaba\n");

System.out.println("\nPara moverse use el comando MOVER y luego use los comandos N (norte),S (sur), E (este) o O (oeste) para decidir en que ubicacoin se movera");

System.out.println("Para ver el oro necesario para salir esriba el siguiente comando ORO\_REQUERIDO");

System.out.println("Para ver el oro que tiene escriba el comando ORO");

System.out.println("Para levantar el oro de una casilla escriba el comando LEVANTAR");

System.out.println("Para usar las casillas de salida escriba el comando SALIR\n");

//Comenzar el juego

System.out.println("Presione cualquier NUMERO del 0 al 9 para comenzar: \n");

opcionJugar=scanner2.nextInt();

if(opcionJugar>=0 && opcionJugar<=9){

LogicaComandos(mapaPre, posicionFilaJugador, posicionColumnaJugador, posicionFilaBot, posicionColumnaBot, Oronecesario);

}

}

1. *Proceso creado solo para el mapa predeterminado*
2. *Se definen 2 variables de tipo entero para la posición en donde estará el jugador*
3. *Se definen 2 variables de tipo entero para la posición en donde estará el bot*
4. *Se usa un bucle do while para asignarle un valor aleatorio a la posición de la fila y la columna hasta que la posición sea un espacio libre en la matriz (Se aplica tanto al bot como al jugador)*
5. *Se crea un pequeño menú explicando como funciona, las reglas y los comandos del juego*
6. *Luego se le pide un numero para continuar y entra a la lógica de comandos*

**public static void LogicaComandos(String mapaPre [][], int posicionFilaJugador, int posicionColumnaJugador, int posicionFilaBot, int posicionColumnaBot,int Oronecesario){**

Scanner mover = new Scanner(System.in);//scanner para saber la direccion en la que se va a mover

Scanner scanner3 = new Scanner(System.in);//scanner para saber que comando va a realizar

int ororecolectado=0;

boolean salida= false;

String comando;

String movimiento;

int movimientosEchos=0;

boolean atrapado=false;

int cantidadMovimientosBot=0;

int errores=0;

while(((ororecolectado < Oronecesario )&&(!salida))||(!atrapado)||(errores<3)){

System.out.println("Escriba el comando que quiere realizar (recuerde escribirlo en mayuscula y correctamente): ");

comando=scanner3.nextLine();

if("MOVER".equals(comando)){

movimientosTotal+=1;

System.out.println("A que direccion se quiere mover(N,S,E o O): ");

movimiento = mover.nextLine();

if("S".equals(movimiento)){

if( "O".equals(mapaPre[posicionFilaJugador+1][posicionColumnaJugador])){

mapaPre[posicionFilaJugador][posicionColumnaJugador]="O";

posicionFilaJugador+=1;

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="J";

movimientosEchos+=1;

}else if("G".equals(mapaPre[posicionFilaJugador+1][posicionColumnaJugador])){

mapaPre[posicionFilaJugador][posicionColumnaJugador]="O";

posicionFilaJugador+=1;

mapaPre[posicionFilaJugador][posicionColumnaJugador]="JG";

movimientosEchos+=1;

}else if("S".equals(mapaPre[posicionFilaJugador+1][posicionColumnaJugador])){

mapaPre[posicionFilaJugador][posicionColumnaJugador]="O";

posicionFilaJugador+=1;

mapaPre[posicionFilaJugador][posicionColumnaJugador]="JS";

movimientosEchos+=1;

} else if("#".equals(mapaPre[posicionFilaJugador+1][posicionColumnaJugador])){

System.out.println("Esta accion no se pudo realizar por que quiere moverse a una pared");

System.out.println("Pierde un turno");

}

} else if("N".equals(movimiento)){

if( "O".equals(mapaPre[posicionFilaJugador-1][posicionColumnaJugador])){

mapaPre[posicionFilaJugador][posicionColumnaJugador]="O";

posicionFilaJugador-=1;

mapaPre[posicionFilaJugador][posicionColumnaJugador]="J";

movimientosEchos+=1;

}else if("G".equals(mapaPre[posicionFilaJugador-1][posicionColumnaJugador])){

mapaPre[posicionFilaJugador][posicionColumnaJugador]="O";

posicionFilaJugador-=1;

mapaPre[posicionFilaJugador][posicionColumnaJugador]="JG";

movimientosEchos+=1;

}else if("S".equals(mapaPre[posicionFilaJugador-1][posicionColumnaJugador])){

mapaPre[posicionFilaJugador][posicionColumnaJugador]="O";

posicionFilaJugador-=1;

mapaPre[posicionFilaJugador][posicionColumnaJugador]="JS";

movimientosEchos+=1;

}else if("#".equals(mapaPre[posicionFilaJugador-1][posicionColumnaJugador])){

System.out.println("Esta accion no se pudo realizar por que quiere moverse a una pared");

System.out.println("Pierde un turno");

}

}else if("E".equals(movimiento)){

if( "O".equals(mapaPre[posicionFilaJugador][posicionColumnaJugador+1])){

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="O";

posicionColumnaJugador+=1;

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="J";

movimientosEchos+=1;

}else if("G".equals(mapaPre[posicionFilaJugador][posicionColumnaJugador+1])){

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="O";

posicionColumnaJugador+=1;

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="JG";

movimientosEchos+=1;

}else if("S".equals(mapaPre[posicionFilaJugador][posicionColumnaJugador+1])){

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="O";

posicionColumnaJugador+=1;

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="JS";

movimientosEchos+=1;

}else if("#".equals(mapaPre[posicionFilaJugador][posicionColumnaJugador+1])){

System.out.println("Esta accion no se pudo realizar por que quiere moverse a una pared");

System.out.println("Pierde un turno");

}

}else if("O".equals(movimiento)){

if("O".equals(mapaPre[posicionFilaJugador][posicionColumnaJugador-1])){

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="O";

posicionColumnaJugador-=1;

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="J";

movimientosEchos+=1;

}else if("G".equals(mapaPre[posicionFilaJugador][posicionColumnaJugador-1])){

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="O";

posicionColumnaJugador-=1;

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="JG";

movimientosEchos+=1;

}else if("S".equals(mapaPre[posicionFilaJugador][posicionColumnaJugador-1])){

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="O";

posicionColumnaJugador-=1;

(mapaPre[posicionFilaJugador][posicionColumnaJugador])="JS";

movimientosEchos+=1;

}else if("#".equals(mapaPre[posicionFilaJugador][posicionColumnaJugador-1])){

System.out.println("Esta accion no se pudo realizar por que quiere moverse a una pared");

System.out.println("Pierde un turno");

}

}else{

System.out.println("Error de comando no existente, 3 errores y perdera ");

errores=errores+1;

}

}else if("ORO\_REQUERIDO".equals(comando)){

System.out.println("El oro requerido para salir es de: "+Oronecesario);

}else if("ORO".equals(comando)){

System.out.println("El oro recoletcado es de: "+ororecolectado);

}else if("LEVANTAR".equals(comando)){

if((mapaPre[posicionFilaJugador][posicionColumnaJugador])=="JG"){

mapaPre[posicionFilaJugador][posicionColumnaJugador]="J";

ororecolectado+=1;

oroTotal+=1;

}else{

System.out.println("No se pudo realizar esta accion porque no hay oro en esta casilla");

System.out.println("Pierde un turno");

}

}else if("MIRAR".equals(comando)){

Mirar(mapaPre,posicionFilaJugador,posicionColumnaJugador);

}else if("SALIR".equals(comando)){

if("S".equals(mapaPre[posicionFilaJugador][posicionColumnaJugador])&&(ororecolectado>=Oronecesario)){

System.out.println("FELICIDADES :3");

System.out.println("Ha logrado salir del laberinto");

salida=true;

//reportes

System.out.println("Cantidad de oro recolectado: "+ororecolectado);

System.out.println("Cantidad de movimientos echos: "+movimientosEchos);

partidasGanadas+=1;

}

}else{

System.out.println("Error de comando no existente, 3 errores y perdera ");

errores=errores+1;

}

//Movimiento del bot

int movimientoBot=0;

movimientoBot=(int)(Math.random()\*5);

if(movimientoBot==1){

if( "O".equals(mapaPre[posicionFilaBot+1][posicionColumnaBot])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="B";

cantidadMovimientosBot+=1;

}else if("G".equals(mapaPre[posicionFilaBot+1][posicionColumnaBot])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="BG";

cantidadMovimientosBot+=1;

}else if("S".equals(mapaPre[posicionFilaBot+1][posicionColumnaBot])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="BS";

cantidadMovimientosBot+=1;

} else if("#".equals(mapaPre[posicionFilaBot+1][posicionColumnaBot])){

}else if("J".equals(mapaPre[posicionFilaBot+1][posicionColumnaBot])){

System.out.println("Te han atrapado");

atrapado=true;

}

}else if(movimientoBot==2){

if( "O".equals(mapaPre[posicionFilaBot-1][posicionColumnaBot])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="B";

cantidadMovimientosBot+=1;

}else if("G".equals(mapaPre[posicionFilaBot-1][posicionColumnaBot])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="BG";

cantidadMovimientosBot+=1;

}else if("S".equals(mapaPre[posicionFilaBot-1][posicionColumnaBot])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="BS";

cantidadMovimientosBot+=1;

} else if("#".equals(mapaPre[posicionFilaBot-1][posicionColumnaBot])){

}else if("J".equals(mapaPre[posicionFilaBot-1][posicionColumnaBot])){

System.out.println("Te han atrapado");

atrapado=true;

}

}else if(movimientoBot==3){

if( "O".equals(mapaPre[posicionFilaBot][posicionColumnaBot+1])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="B";

cantidadMovimientosBot+=1;

}else if("G".equals(mapaPre[posicionFilaBot][posicionColumnaBot+1])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="BG";

cantidadMovimientosBot+=1;

}else if("S".equals(mapaPre[posicionFilaBot][posicionColumnaBot+1])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="BS";

cantidadMovimientosBot+=1;

} else if("#".equals(mapaPre[posicionFilaBot][posicionColumnaBot+1])){

}else if("J".equals(mapaPre[posicionFilaBot][posicionColumnaBot+1])){

System.out.println("Te han atrapado");

atrapado=true;

}

}else if(movimientoBot==4){

if( "O".equals(mapaPre[posicionFilaBot][posicionColumnaBot-1])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="B";

cantidadMovimientosBot+=1;

}else if("G".equals(mapaPre[posicionFilaBot][posicionColumnaBot-1])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="BG";

cantidadMovimientosBot+=1;

}else if("S".equals(mapaPre[posicionFilaBot][posicionColumnaBot-1])){

mapaPre[posicionFilaBot][posicionColumnaBot]="O";

posicionFilaBot+=1;

mapaPre[posicionFilaBot][posicionColumnaBot]="BS";

cantidadMovimientosBot+=1;

} else if("#".equals(mapaPre[posicionFilaBot][posicionColumnaBot-1])){

}else if("J".equals(mapaPre[posicionFilaBot][posicionColumnaBot-1])){

System.out.println("Te han atrapado");

atrapado=true;

}

}else if(movimientoBot==5){

Mirar(mapaPre,posicionFilaBot,posicionColumnaBot);

}

}

if(atrapado=true){

System.out.println("Cantidad de oro recolectado: "+ororecolectado);

System.out.println("Cantidad de movimientos echos: "+movimientosEchos);

System.out.println("Cantidad de movimientos echos por el bot"+cantidadMovimientosBot);

vecesAtrapado+=1;

}

if(errores>3){

System.out.println("Cantidad de oro recolectado: "+ororecolectado);

System.out.println("Cantidad de movimientos echos: "+movimientosEchos);

}

}

1. *Proceso creado solo para el mapa predeterminado*
2. *Se definen las variables que nos servirán para saber cuando termina el juego mediante un bucle while*
3. *Se definen las variables también para llevar un conteo del oro recolectado y las acciones*
4. *Se definen 2 variables de tipo boolean para saber si el jugador salió o si el bot lo atrapo*
5. *Realizamos un while que se seguirá repitiendo mientras el oro recolectado sea menor al oro necesario y la variable salir sea falsa o si la variable atrapada sea falsa o si el número de errores siga siendo menor que 3*
6. *Dentro del while lo primero que se le pregunta es que comando quiere realizar*
7. *Si elige el comando mover se le preguntara la dirección y dependiendo de la dirección se tendrán varios if para saber a que casilla se quiere mover  
   Si a la casilla que se quiere mover es un espacio libre, la posición actual de jugado en la matriz se cambiara por un espacio libre, se le sumara o restara a las fila o columna (Dependiendo a donde se quiera mover) de la posición del jugador y esta nueva posición en la matriz se cambiara por el jugador*
8. *Si a donde se quiere mover es una pared se le mostrara un mensaje explicándole por que no se puede realizar la acción*
9. *Si elige el comando de oro\_requerido se le mostrara la variable con la cantidad de oro que necesita*
10. *Si elige el comando de oro se le mostrara la variable en donde esta la cantidad de oro que tiene*
11. *Si elige el comando levantar se verificará primero con un if si la casilla en donde esta el jugador hay oro, si es así se le quitara el oro de la casilla y se le sumara 1 a la variable donde esta la cantidad de oro que tiene. Si no hay una casilla se le mostrara un mensaje diciendo por que no se puede realizar esta acicon*
12. *Si elige el comando mirar se le llevara al proceso para mostrar una cuadricula del mapa 5x5 desde su posición*
13. *Si elige el comando salir se verificara con un if si esta en una casilla donde hay una salida y si tiene la cantidad de oro necesario para salir, si es así, se le dejara salir y se le mostrara el reporte de final de partida, tendrá una partida mas ganada y se le llevara al menú principal*
14. *Luego se le concede un valor aleatorio hasta 5 al movimiento del bot*
15. *Si el movimiento es 1 ira al norte, si es 2 ira al sur, si es 3 ira al oeste, si es 4 ira al este y si es 5 ejecutara el proceso de mirar*
16. *Si el bucle termina por otra razón habrá 2 if al final para saber si perdió por que lo atrapo el bot o por que cometió 3 errores*

**public static void Mirar(String mapa[][],int posicionf, int posicionc){**

int posicionMinFila;

int posicionMaxFila;

int posicionMinColumna;

int posicionMaxColumna;

posicionMinFila=posicionf-2;

posicionMaxFila=posicionf+2;

posicionMinColumna=posicionc-2;

posicionMaxColumna=posicionc+2;

//Por si el jugador esta pegado a una pared

if(posicionMinFila<0){

posicionMinFila=0;

}

if(posicionMaxFila>29){

posicionMaxFila=29;

}

if(posicionMinColumna<0){

posicionMinColumna=0;

}

if(posicionMaxColumna>29){

posicionMaxColumna=29;

}

for(int i=posicionMinFila;i<=posicionMaxFila;i++){

for(int j=posicionMinColumna;j<=posicionMaxColumna;j++){

System.out.print(mapa[i][j]+" ");

}

System.out.println("");

}

}

1. *Se crean 4 variables para la posición mínima y máxima que se quiere mostrar*
2. *Luego se verifica por medio de if que el jugador este pegado a una pared, si lo esta, se cambia el valor de las posiciones mínimas o máximas*
3. *Luego se crear un for que va desde la posición mínima de la columna y de la fila mostrando la porción del mapa 5x5*

**public static void Crearmapa(){**

Scanner scanner = new Scanner(System.in);

int limiteDeMapas=10;

int cantidadSalidas=0;

mapasCreados=1;

cubo= new String [limiteDeMapas][][];

System.out.println("\nBienvenido al creado de mapas\n");

System.out.println("Definamos tamaño al mapa\n");

System.out.println("Cantidad de filas: ");

filas=scanner.nextInt();

System.out.println("Cantidad de Columnas: ");

columnas =scanner.nextInt();

cubo[mapasCreados]=new String [filas][columnas];

System.out.println("\nComenzemos con las salidas");

System.out.println("Cuantas salidas tendra el mapa: ");

cantidadSalidas=scanner.nextInt();

//ubicando las salidas

for(int i=0;i<cantidadSalidas;i++){

System.out.println("En donde ira la salida "+(i+1));

System.out.println("Fila de la salida:");

int filaSalida=scanner.nextInt();

System.out.println("Columna de la salida: ");

int columnaSalida=scanner.nextInt();

cubo[mapasCreados][filaSalida][columnaSalida]="S";

}

System.out.println("\nUbiquemos ahora el oro en el mapa");

System.out.println("Cual es la cantidad de oro que tendra su mapa: ");

int cantidadDeOro=scanner.nextInt();

Oronecesario();

//ubicando el oro en el mapa

for(int i=0;i<cantidadDeOro;i++){

System.out.println("En donde ira el oro "+(i+1));

System.out.println("Fila del oro:");

int filaOro=scanner.nextInt();

System.out.println("Columna del oro: ");

int columnaOro=scanner.nextInt();

cubo[mapasCreados][filaOro][columnaOro]="G";

}

System.out.println("\nAhora ubiquemos todas las paredes y espacios libres");

int espaciosRestantes = (filas\*columnas)-(cantidadSalidas)-(cantidadDeOro)-1;

//ubicando espacios y paredes

for(int i=0;i<=espaciosRestantes;i++){

System.out.println("Que desea poner:");

System.out.println("1. Pared");

System.out.println("2. Espacio");

int espacioOpared=scanner.nextInt();

if(espacioOpared==1){

System.out.println("Fila: ");

int filaPared=scanner.nextInt();

System.out.println("Columna");

int columnaPared=scanner.nextInt();

cubo[mapasCreados][filaPared][columnaPared]="#";

}else if(espacioOpared==2){

System.out.println("Fila: ");

int filaEspacio=scanner.nextInt();

System.out.println("Columna");

int columnaEspaciod=scanner.nextInt();

cubo[mapasCreados][filaEspacio][columnaEspaciod]="0";

}

}

System.out.println("\nFelicidades su mapa ha sido creado\n");

mapasCreados+=1;

}

1. *Si elige la opción 2 en el menú lo llevara a la opción de crear mapa*
2. *Con el cubo que definimos como variable global se irán guardando los mapas necesarios*
3. *Se lleva un contador de mapas que es una variable global para que no se reinicie*
4. *Se le pregunta primero cuantas salidas tendrá y en que fila y columna se ponda*
5. *Se le pregunta luego cuanto oro tendrá, el oro necesario para salir y la ubicación*
6. *Luego solo se ponen espacios o paredes en la cantidad que le resta del mapa*

**public static void Reportes(){**

System.out.println("\nCantidad de veces que ha sido atrapado: "+vecesAtrapado);

System.out.println("\nCantidad de partidas ganadas: "+partidasGanadas);

float promedioOro=(oroTotal/partidasJugadas);

System.out.println("\nPromedio del oro recoletcado por partida: "+promedioOro);

float promedioMovimientos=(movimientosTotal/partidasJugadas);

System.out.println("\nPromedio de movimientos echos por partida"+promedioMovimientos

}

**public static void MostrarMapa(String mapa1[][]){**

for(int i=0;i<30;i++){

for(int j=0;j<30;j++){

System.out.print(mapa1[i][j]+" ");

}

System.out.println("");

}

}

*Mostrar el mapa predeterminado*

**public static void mostrarCubo(){**

for(int i=0;i<filas;i++){

for(int j=0;j<columnas;j++){

System.out.print(cubo[selectorMapa][i][j]+" ");

}

System.out.println("");

}

}

*Mostrar los mapas creados*

**public static void Oronecesario(){**

Scanner scanner = new Scanner(System.in);

System.out.println("Digite la cantidad de oro necesario para salir: ");

oroNecesario1 =scanner.nextInt();

}

*Preguntar el oro necesario para los mapas creados*