

2012

STAR WARS: SERPIENTES Y ESCALERAS

Manual Técnico

Este documento describe la implementación necesaria para desarrollar el
StarWars: Serpientes y Escaleras.



Índice

1. Descripción Del Software.....	3
Especificaciones	
Restricciones	
Limitantes	
Funcionalidad Basica	
 2. Requerimientos Mínimos de Software y Hardware.....	 5
Hardware	
Software	
 3. Código Fuente de algunos métodos.....	 6
4. Contacto.....	21

1. Descripción del Software

El juego de mesa conocido como **escaleras y serpientes** es un juego de azar jugado sobre un tablero con casillas ilustrado sobre el que los participantes compiten por llegar en primer lugar a la meta. Debe de llegarse justo a la última casilla, por ejemplo si un jugador está a tres casillas de la meta y el dado sale cuatro debe avanzar tres y retroceder una, hasta caer justamente en la casilla última.

«**Star Wars – Serpientes y Escaleras**»® es un software que ha sido basada en la Programación Orientada a Objetos (POO) utilizando el lenguaje de programación Java™ con la cual se construyo una aplicación visual y cómoda para el usuario.

Especificaciones:

- ❖ Desarrollo multimedia de Star Wars.
- ❖ Reproducir Sonidos o Música de Star Wars.
- ❖ Deben jugar de dos o hasta 4 jugadores por partida.
- ❖ El nombre de los jugadores puede editarse.

Para su implementación fue necesario el manejo de excepciones, creación (constructores) de objetos, manejo de varios paquetes como awt, swing, Jlayer(para reproducir sonido .mp3), el manejo de Graphic y Graphic 2D para el movimiento de las fichas .



Funcionalidad Básica

- ❖ Del menú principal uno puede escoger el número de jugadores por partida.
- ❖ El usuario podrá editar los nombres que vienen por defecto.
- ❖ Al Ser Ejecutado podrá oírse una serie de canciones de Star Wars.
- ❖ En la ventana del tablero los usuarios podrán presionar el dado que indicara cuanto avanzara su correspondiente ficha.

Restricciones

«**Star Wars – Serpientes y Escaleras**»® solo se puede jugar con dos jugadores a cuatro, no menos de dos, ni más de 4.

Requerimientos Mínimos De Software y Hardware

Para que se pueda desarrollar «**Star Wars – Serpientes y Escaleras**»® se necesitan mínimo los siguientes componentes en su equipo:

Hardware:

- ❖ Procesador AMD® Athlon™ 2.7GHz o superior.
- ❖ 3 GB de espacio disponible en disco duro. (para instalar
- ❖ **JDeveloper**®).
- ❖ Unidad de CD-ROM (Documentar).
- ❖ Monitor con resolución SVGA o superior.
- ❖ Teclado.
- ❖ Mouse.
- ❖ Puestos USB (Para respaldar información).

Software:

- ❖ Microsoft® Windows® XP/Vista/7
- ❖ Visualizador de documentos PDF(p.e. Acrobat Reader™)
- ❖ «**Star Wars – Serpientes y Escaleras**»® código fuente (CD-ROM)
- ❖ **JDeveloper**® para desarrollar (Tener Incluido el **JDK**).
- ❖ Las librerías JLayer, Graphic2d y ciertas versiones de Synthetica para look and feel.



3. Codigo Fuente

Metodo Para el Sonido del dado o las casillas

Se utilizo la libreria JLayer para el manejo de archivos de audio en este metodo lo que se hace es recibir el nombre o la direccion del archivo a reproducir y utilizar las instrucciones necesarias para abrir y reproducir.

```
void setAudio(String N){
    AudioFileFormat aff;
    AudioInputStream ais;
    File sf;
    sf = new File(N);
    try{
        aff = AudioSystem.getAudioFileFormat(sf);
        ais = AudioSystem.getAudioInputStream(sf);
        AudioFormat af = aff.getFormat();
        DataLine.Info info=new DataLine.Info(Clip.class,ais.getFormat(),
            ((int) ais.getFrameLength()*af.getFrameSize()));
        Clip audio = (Clip)AudioSystem.getLine(info);
        audio.open(ais);
        audio.loop(0);
    }catch(UnsupportedAudioFileException ee){}
    catch(IOException ea){}
    catch(LineUnavailableException LUE){};
}
```

Metodo Paint con el que se dibujaron las fichas y el numero de las casillas

Este metodo es para dibujar en pantalla las fichas del numero de jugadores que elije el usuario.

Las variables necesarias son:

int Jugadores //q guarda el numero total de jugadores

int tam//para el tamaño de la ficha

int[] x,y//vectores para saber en q posicion se van a dibujar los jugadores.

```
public void paint(Graphics g) {  
    //tamaño del circulo  
    int tam=12;  
    //pintar el jpanel  
    super.paint(g);  
    //saber cuantos circulos vamos a dibujar  
    switch (Jugadores){  
        case 2:  
            //jugador 1  
            g.setColor(Color.blue);  
            g.fillOval(x[0],y[0], tam, tam);  
            //juador 2  
            g.setColor(Color.red);  
            g.fillOval(x[1],y[1], tam, tam);  
            break;  
        case 3:  
            //jugador 1  
            g.setColor(Color.blue);  
            g.fillOval(x[0],y[0], tam, tam);  
            //juador 2  
            g.setColor(Color.red);  
            g.fillOval(x[1],y[1], tam, tam);  
            //juador 3  
            g.setColor(Color.green);  
            g.fillOval(x[2],y[2], tam, tam);  
            break;  
        case 4:  
            //jugador 1  
            g.setColor(Color.blue);  
            g.fillOval(x[0],y[0], tam, tam);  
            //juador 2  
            g.setColor(Color.red);  
            g.fillOval(x[1],y[1], tam, tam);  
            //juador 3  
            g.setColor(Color.green);  
            g.fillOval(x[2],y[2], tam, tam);  
            //jugador 4  
            g.setColor(Color.yellow);  
            g.fillOval(x[3],y[3], tam, tam);  
            break;  
    }  
  
    //Pintar A Las Casillas Los Numeros al Tablero  
    //Casilla 1  
    g.setColor(Color.WHITE);
```

"The_Spartans" Never Die, They're Just Missing In Action


```
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("1",80 ,75);
//Casilla 2
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("2",150 ,75);
//Casilla 3
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("3",220 ,75);
//Casilla 4
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("4",290 ,75);
//Casilla 5
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("5",360 ,75);
//Casilla 6
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("6",430 ,75);
//Casilla 7
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("7",500 ,75);
//Casilla 8
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("8",570 ,75);
//Casilla 9
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("9",640 ,75);
//Casilla 10
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("10",145 ,140);
//Casilla 11
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("11",215 ,140);
//Casilla 12
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("12",285 ,140);
//Casilla 13
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("13",355 ,140);
//Casilla 14
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("14",425 ,140);
//Casilla 15
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("15",495 ,140);
//Casilla 16
```

"The_Spartans" Never Die, They're Just Missing In Action



```
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("16",565,140);
//Casilla 17
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("17",635,140);
//Casilla 18
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("18",145,205);
//Casilla 19
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("19",215,205);
//Casilla 20
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("20",285,205);
//Casilla 21
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("21",355,205);
//Casilla 22
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("22",425,205);
//Casilla 23
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("23",495,205);
//Casilla 24
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("24",565,205);
//Casilla 25
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("25",635,205);
//Casilla 26
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("26",145,270);
//Casilla 27
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("27",215,270);
//Casilla 28
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("28",285,270);
//Casilla 29
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("29",355,270);
//Casilla 30
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("30",425,270);
```

"The_Spartans" Never Die, They're Just Missing In Action



```
//Casilla 31
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("31",495,270);
//Casilla 32
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("32",565,270);
//Casilla 33
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("33",635,270);
//Casilla 34
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("34",145,335);
//Casilla 35
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("35",215,335);
//Casilla 36
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("36",285,335);
//Casilla 37
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("37",355,335);
//Casilla 38
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("38",425,335);
//Casilla 39
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("39",495,335);
//Casilla 40
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("40",565,335);
//Casilla 41
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("41",635,335);
//Casilla 42
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("42",145,400);
//Casilla 43
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("43",285,400);
//Casilla 44
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("44",355,400);
//Casilla 45
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
```

"The_Spartans" Never Die, They're Just Missing In Action



```
g.drawString("45",425,400);
//Casilla 46
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("46",495,400);
//Casilla 47
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("47",565,400);
//Casilla 48
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("48",635,400);
//Casilla 49
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("49",145,465);
//Casilla 50
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("50",215,465);
//Casilla 51
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("51",285,465);
//Casilla 52
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("52",355,465);
//Casilla 53
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("53",425,465);
//Casilla 54
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("54",495,465);
//Casilla 55
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("55",565,465);
//Casilla 56
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("56",635,465);
//Casilla 57
g.setColor(Color.WHITE);
g.setFont(new Font("Tahoma",Font.BOLD,10));
g.drawString("57",705,465);
}
```

Metodos para el Dado

Para este metodo no basamos en los metodos random de 0 a 5 y de ahí sumar uno para sacar numero entre 1 y 6, en funcion de este boton las fichas se moveran

```
int Dado(){
    //se invoca al metodo random para asignar un numero aleatorio al dado
    int n=(int)(Math.random()*6);
    n++;
    //se muestra el numero en una imagen para el boton
    BDado.setIcon(new ImageIcon("Dado"+n+".png"));
    return n;
}

private void BDado_actionPerformed(ActionEvent f) {
    //se reproduce un sonido
    CAudio A=new CAudio();
    A.setAudio("SDado.wav");
    //esto es para que al oprimir el boton del dado se cambie una imagen de fondo
    if(E==1){
        E=2;
    }else{
        E=1;
    }
    Inicio1.setIcon(new ImageIcon("Estrellas"+E+".png"));
    //invocamos a metodo dado para saber que numero toco y luego iniciamos la animacion
    dado=this.Dado();
    this.Animacion();
    //deshabilitamos boton en lo q esta la animacion
    BDado.setEnabled(false);
}
```

Metodos Para el movimiento de las piezas

El metodo getTurno os regresa el turno actual en el que va el juego cuando se llega al turno maximo se reinicia el conteo para evitar que se cicle.

```
void getTurno(){
    //ciclo de jugadores
    if(turno==Jugadores){
        turno=1;
    }else{
        turno++;
    }
}
```

El metodo animacion llama a los respectivs metodos para que se lleve a cabo el movimiento de la pieza actual. Primero llamamos a getTurno para saber el turno actual luego instanciamos el objeto para el retraso de las piasas y por ultimo empezamos la animacion con timer.restart()

```
void Animacion(){
    this.getTurno();
    timer = new Timer(5, this);
    timer.restart();
}
```

Timer.restart() activa el siguiente metodo el cual llama a una seria de funciones que simularan el movimiento.

Con Moviminetos() movemos la pieza tantas casillas como la variable dado tenga asignada.

Con Detener() pausamos el movimiento para que le llegue el turno a un siguiete jugador

Repaint() actualiza el lugar donde se dibujaron las fichas de los jugadores.

```
public void actionPerformed(ActionEvent e){
    //mover la pieza
    this.Movimientos();
    this.Detener();
    //redibujar ficha
    repaint();
}
```

Para movimientos que es donde se mueve la pieza se utilizaron 3 condiciones muy importantes con las que sabremos hacia donde se moverá la pieza

La primera condicion `if(c[turno-1]==1 || c[turno-1]==3 || c[turno-1]==5)` es para que se mueva a la derecha si la pieza esta en filas impares 1,3,5, mientras no llegue al final de esa fila(coordenada 610)

La seungda condicion es para moverse hacia la izquierda mientras no llegue al final de esa fila(coordenada 120), esto si esta en la fila 2,4,6.

La tercera condicion (else) es para saber si esta en la ultima fila por lo que el jugador esta a punto de ganar, se debe saber con las banderas `b[turno]` hacia donde se esta moviendo la pieza para que al estar en la ultima casillas y si cae 6 regresar 5 casillas y al siguiete turno volver a vanzar a la derecha.

```
void Movimientos(){
    //columnas nones avanzan a la derecha
    if(c[turno-1]==1 || c[turno-1]==3 || c[turno-1]==5){
        //mientras no llegue a la ultima casilla de la fila se mueve en x
```

"The_Spartans" Never Die, They're Just Missing In Action

```

if(x[turno-1]!=610){
    x[turno-1]=x[turno-1]+5;
    //contador de movimientos
    auxtimerX++;
}else{
    //en la ultima casilla de la fila debe de bajar
    this.bajar(turno);
}
}else{
    if(c[turno-1]==2 || c[turno-1]==4 || c[turno-1]==6){
        //columnas pares avanzan a la izquierda
        if(x[turno-1]!=120){
            x[turno-1]=x[turno-1]-5;
            //contador de movimientos
            auxtimerX++;
        }else{
            this.bajar(turno);
        }
    }else{
        //ultima fila
        if(c[turno-1]==7){
            //si es diferente de 680 sigue avanzando a la derecha
            if(x[turno-1]!=680 && b[turno-1]==false){
                x[turno-1]=x[turno-1]+5;
                //contador de movimientos
                auxtimerX++;
                //bandera para avanzar
                b[turno-1]=false;
            }else{
                if(x[turno-1]==680 && b[turno-1]==false){
                    x[turno-1]=x[turno-1]-5;
                    //contador de movimientos
                    auxtimerX++;
                    //bandera para regreso
                    b[turno-1]=true;
                }else{
                    if(x[turno-1]!=680 && b[turno-1]==true){
                        x[turno-1]=x[turno-1]-5;
                        //contador de movimientos
                        auxtimerX++;
                        //bandera de regreso
                        b[turno-1]=true;
                    }
                }
            }
        }
    }
}
}
}
}
}
}
}

```

Al bajar no moveremos en las coordenadas y[turno] 5 pixeles durante 13 veces por lo que el total de pixeles seran 65 que es donde empieza la siguiente fila, la variable f[turno] se debe modificar para que en el

siguiete movimiento entre a la condicion de derecha o izquierda según sea el caso.

Para los movimientos hacia la dercha o izquierda se utiliza la misma mecanica de 5 pixeles durante 14 para que sumen un total de 70 pixeles de dezplazamiento.

```
void bajar(int t){
    //moverse en y
    y[turno-1]=y[turno-1]+5;
    //contador de movimientos
    auxtimerY++;
    //condicion de fin de movimiento (se compara con 13 para equilibrar los pixeles horizontales a
    verticales
    if(auxtimerY==13){
        //reiniciar contador de movimientos
        auxtimerY=0;
        dado--;
        //contador de fila
        c[turno-1]=c[turno-1]+1;
    }
}
```

Para detener la animacion es necesaria una variable que cuente los movimientos totales y asi poder salir del ciclo en este caso tenemos a auxtimerX que cuenta cuantas veces se ha movido, al momento de cortar el movimiento se compara auxtimeX con (14*dado) que es el numero de pixeles totales que se va a mover, cuando la condicion se cumpla reiniciamos contador para el siguiente jugador, ponemos la bandera b[turno] en false para que en el siguiete turno del mismo jugador se mueva a la derecha en la ulñtima fila.

Timer.stop() para deternet la animacion

En su caso si ya llego a la casilla final mostrar un mensaje de victoria
This.PreNombre() es para que en el tablero se ilumine el nombre del siguiente jugador

This.SW_SE() es para saber si callo en alguna casillas de sable laser.

```
void Detener(){
    //condicion para cortar movimiento
    if(auxtimerX==(14*dado)){
        //reinicir contador
        auxtimerX=0;
        //regresar bandera de final de fila
        b[turno-1]=false;
        //terminar animacion
        timer.stop();
        //habilitar dado
        BDado.setEnabled(true);
        //condicion de victoria
        if(x[turno-1]==680){
```

"The_Spartans" Never Die, They're Just Missing In Action




```

        JOptionPane.showMessageDialog( this,Nombre[turno-1]+ " ha ganado!!!","Felicidades!!!",
        JOptionPane.INFORMATION_MESSAGE);
        t.dispose();
        FMPrincipal nuevo=new FMPrincipal();
        nuevo.setVisible(true);
    }
    // System.out.println(x[turno-1]);
    // System.out.println(y[turno-1]);
    this.coordenada(x[turno-1], y[turno-1]); //Metodo Para El Sonido De Casillas

    this.PreNombre(turno);
    //medoto de serpiente y escaleras
    this.SW_SE();
}
}

```

Metodos Para las caidas o subidas

Con SW_SE() sabermos si callo en una escalera o en una serpiente

```

void SW_SE(){
    this.Escalera();
    this.Serpiente();
}

```

Escalera() compara el lugar donde esta la ficha del jugador ejemplo si esta en el pixel $x=610$ y en la fila $f=2$ quiere decir que estamos en el sable 2 por lo tanto debemos de movernos la casilla (540,y) en la fila=4, donde y es el numero de pixeles que se movio en x multiplicado por 2, y restandole 10 para que quede en el pixel correcto de tal jugador(jugador azul seria $((610-540)*2)-10=(70)*2-10=140-10=130$ (540,130);

Esto se podria mejorar ya que la implementacion fue diseñada para simular movimiento en diagonal pero no se retrasa el tiempo y solo se ve cuando estaba en la casilla taal y llega a la otra casilla.

Para las serpiente se utilizo la misma mecanica que en escaleras.

```

void Escalera(){
    if(x[turno-1]==120 && f[turno-1]==1){//escalera 1
        this.SonidoEscalera();
        while(x[turno-1]!=190){
            x[turno-1]=x[turno-1]+1;
            y[turno-1]=y[turno-1]+1;
            f[turno-1]=2;
            repaint();
        }
        y[turno-1]=y[turno-1]-5;
        repaint();
    }else{
        if(x[turno-1]==610 && f[turno-1]==2){//escalera 2
            this.SonidoEscalera();
            while(x[turno-1]!=540){

```

"The_Spartans" Never Die, They're Just Missing In Action


```

    }
}
void Serpiente(){
    if(x[turno-1]==400 && f[turno-1]==2){//serpiente 1
        this.SonidoSerpiente();
        while(x[turno-1]!=470){
            x[turno-1]=x[turno-1]+1;
            y[turno-1]=y[turno-1]-1;
            f[turno-1]=1;
            repaint();
        }
        y[turno-1]=y[turno-1]+5;
        repaint();
    }else{
        if(x[turno-1]==400 && f[turno-1]==3){//serpiente 2
            this.SonidoSerpiente();
            while(x[turno-1]!=260){
                x[turno-1]=x[turno-1]-1;
                y[turno-1]=y[turno-1]-1;
                f[turno-1]=1;
                repaint();
            }
            y[turno-1]=y[turno-1]+10;
            repaint();
        }else{
            if(x[turno-1]==400 && f[turno-1]==4){//serpiente 3
                this.SonidoSerpiente();
                while(x[turno-1]!=470){
                    x[turno-1]=x[turno-1]+1;
                    y[turno-1]=y[turno-1]-2;
                    f[turno-1]=2;
                    repaint();
                }
                y[turno-1]=y[turno-1]+10;
                repaint();
            }else{
                if(x[turno-1]==540 && f[turno-1]==5){//serpiente 4
                    this.SonidoSerpiente();
                    while(x[turno-1]!=470){
                        x[turno-1]=x[turno-1]-1;
                        y[turno-1]=y[turno-1]-1;
                        f[turno-1]=4;
                        repaint();
                    }
                    y[turno-1]=y[turno-1]+5;
                    repaint();
                }else{
                    if(x[turno-1]==120 && f[turno-1]==7){//serpiente 5
                        this.SonidoSerpiente();
                        while(x[turno-1]!=190){
                            x[turno-1]=x[turno-1]+1;
                            y[turno-1]=y[turno-1]-4;
                            f[turno-1]=5;

```

```
        repaint();
    }
    y[turno-1]=y[turno-1]+20;
    repaint();
}else{
    if(x[turno-1]==400 && f[turno-1]==7){//serpiente 6
        this.SonidoSerpiente();
        while(x[turno-1]!=260){
            x[turno-1]=x[turno-1]-1;
            y[turno-1]=y[turno-1]-1;
            f[turno-1]=5;
            repaint();
        }
        y[turno-1]=y[turno-1]+10;
        repaint();
    }
}
}
}
}
}
}
```



4. Contacto:

The_Spartans_Company™

adair_1403@hotmail.com

luislyly@hotmail.com

Facultad de Ingeniería, Cerro de Coatepec s/n.

Toluca, Edo. de México. C.P. 50130

(722)1839419

