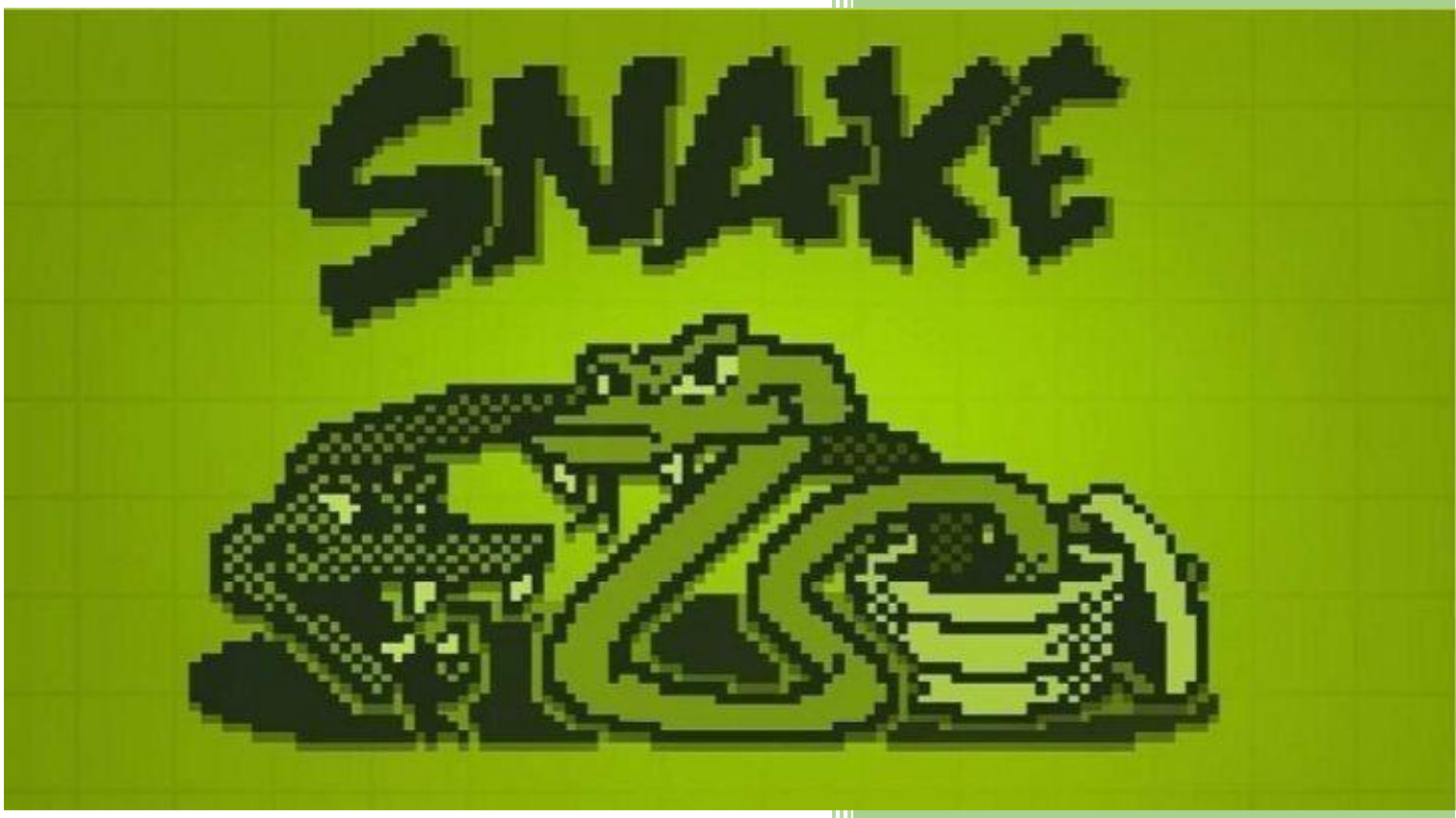
MANUAL TECNICO



Adrián Ismael Orlando García Balan

Phyenyx Computer Inc. ®

202102775 UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

INDICE

OBJETVOS

Para el usuario

Poder desarrollar su mentalidad de programado y poder ver como con líneas de código se le puede dar vida a un juego tan divertido.

Poder darle un momento de diversión al usuario de manera que pueda relajarse y desesterarse jugando esta pieza de arte.

Objetivo Principal

Que el usuario pueda comprender y entender lo que se está realizando dentro del juego de manera que el mismo pueda entender el entorno en el que se desempeña, pueda ver las funciones y las nuevas prácticas que incluye este desafío.

REQUISITOS

- Equipo Pentium II o Superior
- Mínimo 2GB en RAM
- Sistema Operativo Windows 7 o Superior
- Resolución Grafica 800*600
- Tener instalado java en su versión más reciente

HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

Java

Herramientas utilizadas para el desarrollo JAVA El lenguaje de programación de Java es una herramienta de desarrollo orientada a objetos, fue diseñado para que no dependieran en muchas implementaciones, el cual permite a los desarrolladores ejecutar en cualquier dispositivo sin necesidad de recompilar el código, el cual se considera multiplataforma.

NetBeans

Netbeans es un IDE (Integrated Development Environment) o entorno de desarrollo integrado, que es gratuito y de código abierto. Si quieres saber qué es Netbeans, en primera instancia, se debe destacar que sirve para el desarrollo de aplicaciones web, corporativas, de escritorio y móviles que utilizan plataformas como Java y HTML5, entre otras.

Github

Github es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, y que fue comprada por Microsoft en junio del 2018. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargarte la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.

Instalación De Herramientas Necesarias

Requisitos generales pre-instalación para el sistema de escritorio Para ejecutar el programa de escritorio se necesita de Java 8.0 instalado con las siguientes características para la ejecución del programa de escritorio. Soporte en procesador Intel 1.4.0 GHz entre otros. Memoria RAM 2 GB. Espacio en disco: 124 MB El programa se descarga del siguiente enlace en la página oficial de Java https://www.java.com/es/download/.

Ilustración 1 JAVA *Fuente. Propia*

DESCRIPCION DEL PROGRAMA

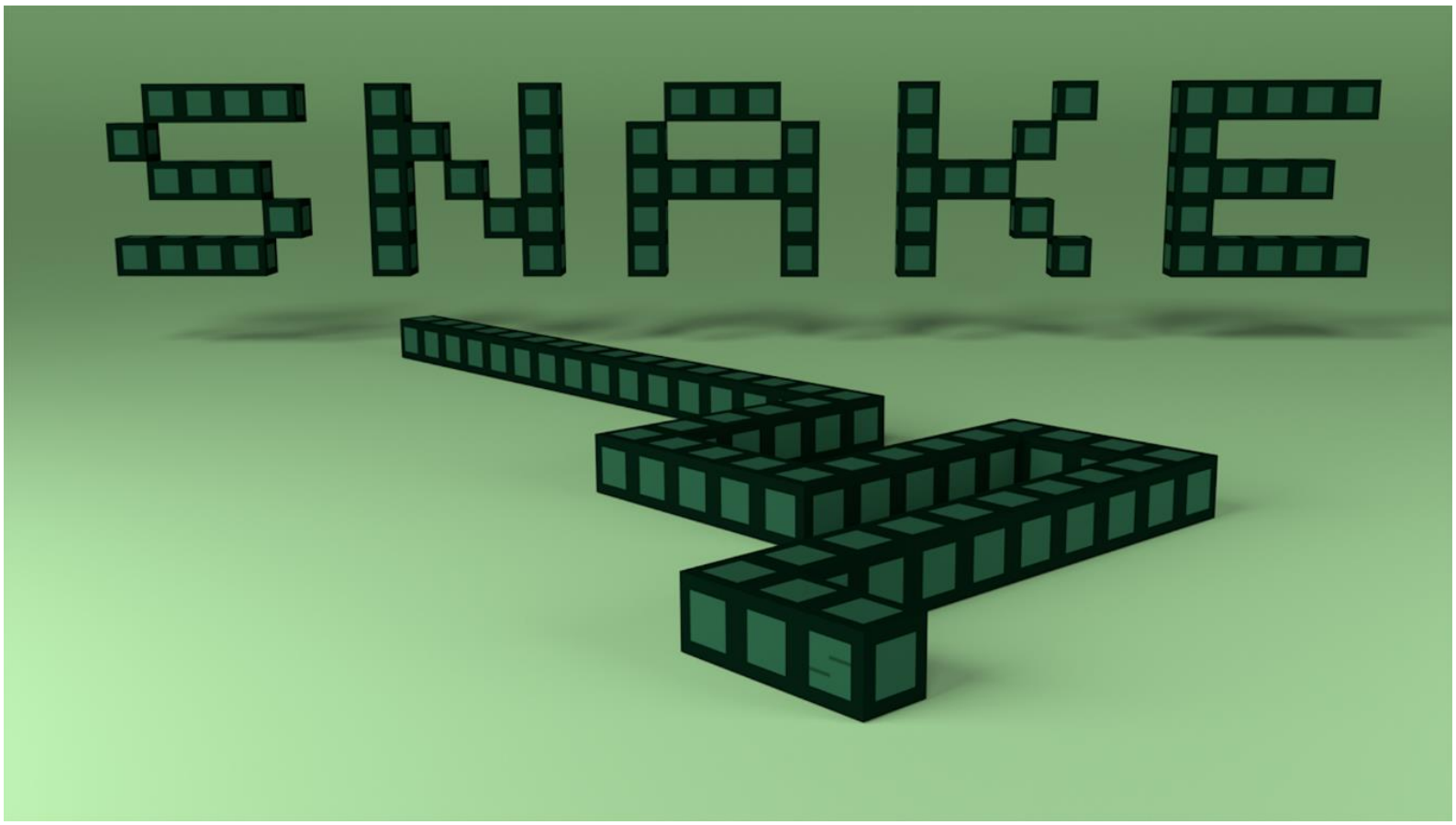
En el juego, el jugador o usuario controla una larga y delgada criatura, semejante a una serpiente, que vaga alrededor de un plano delimitado, recogiendo alimentos (o algún otro elemento), tratando de evitar golpearse contra su propia cola o las "paredes" que rodean el área de juego. Cada vez que la serpiente se come un pedazo de comida, la cola crece más, provocando que aumente la dificultad del juego. El usuario controla la dirección de la cabeza de la serpiente (arriba, abajo, izquierda o derecha) y el cuerpo de la serpiente la sigue. Además, el jugador no puede detener el movimiento de la serpiente, mientras que el juego está en marcha.

Dificultades

- Fácil
- Medio
- Difícil

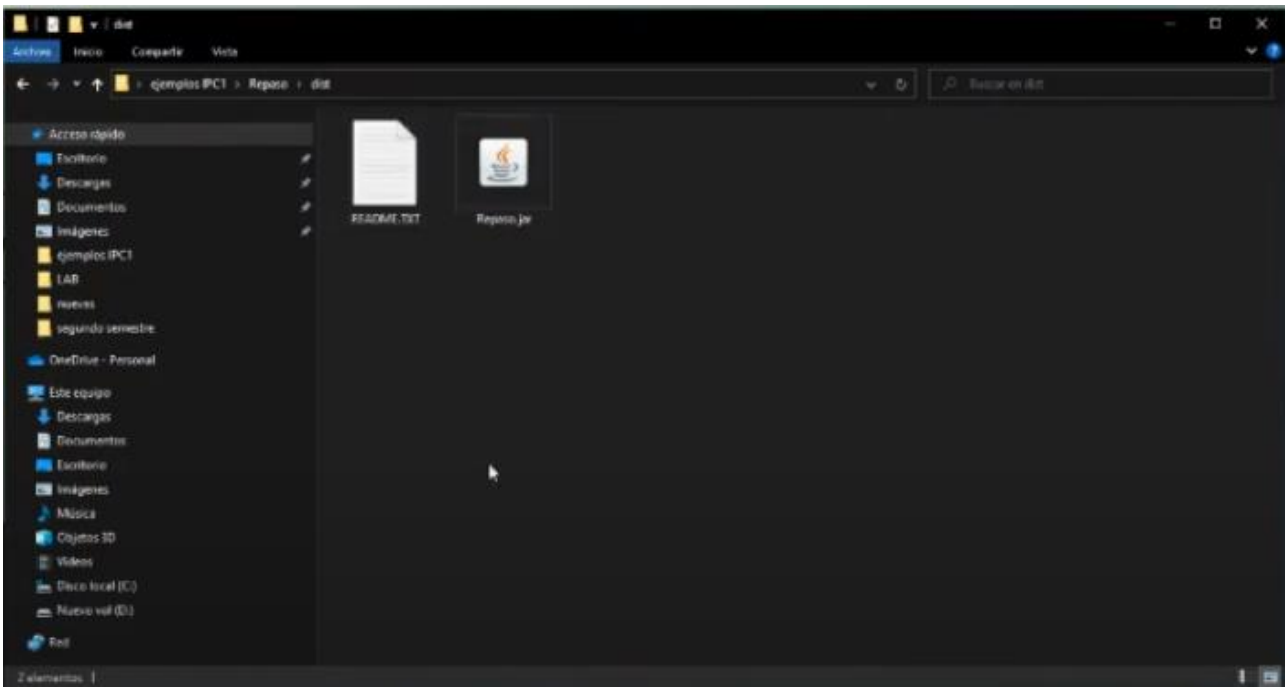
Funciones

Comerse la manzana hasta llegar a la cantidad deseada de 30 puntos.



INSTALACION DEL JUEGO SNAKE

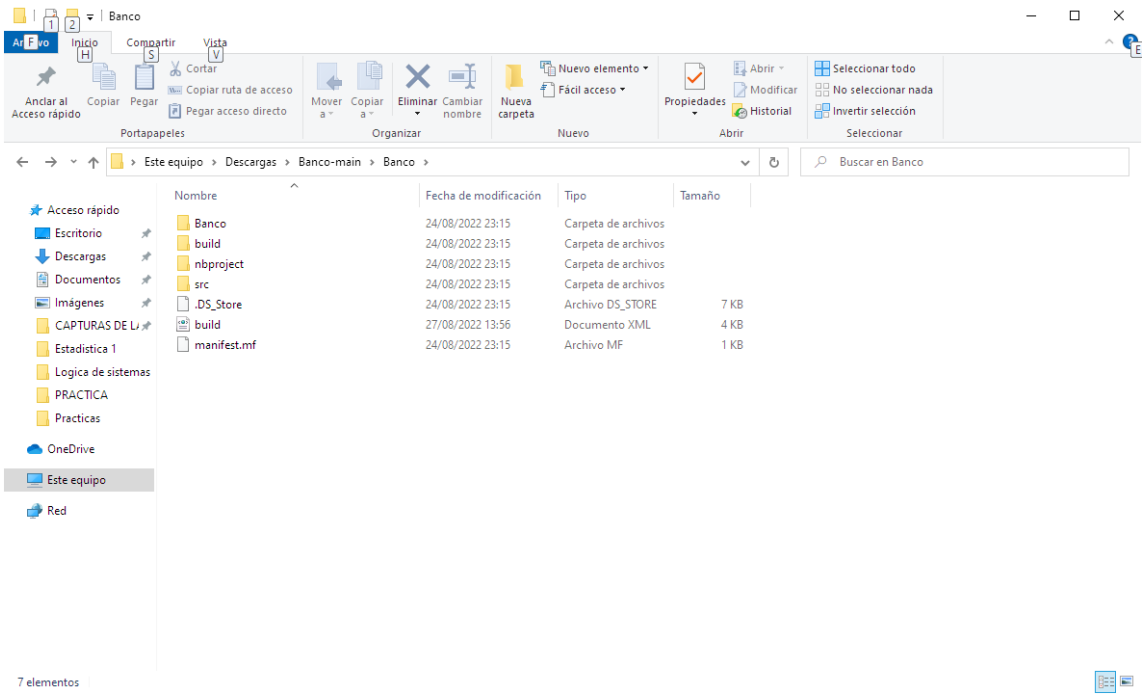
Descargar el ejecutable .jar el cual contienen el programa. Esto de la siguiente manera.



Fuente: Clase de Laboratorio de Introducción a la Programación y Computación 1

Dirigirse a la carpeta donde se encuentra el ejecutable .jar del programa “Snake 1.1”

Una vez ahí haremos lo siguiente:



Una vez en el sitio donde se encuentra nuestro ejecutable .jar ya posicionados en el destino de la carpeta, lo ejecutaremos como cualquier programa. Le daremos doble clic e inmediatamente ingresara al programa, mostrándonos su interfaz gráfica.

CODIGO IMPORTANTE DEL INICIO

Del inicio no hay mucho que hablar solo que hacemos el llamado para lo que será nuestro jFrame principal en el cual se contendrá lo que es nuestro juego de Snake el cual lo llamaremos para poder hacerlo visible.

```
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        Snake1 in = new Snake1();
        in.setVisible(true);
        this.setVisible(false);


    }
```

CODIGO IMPORTANTE DEL FRAME PRINCIPAL

Una vez dentro ejecutaremos cada una de las instrucciones necesarias para que el juego se ejecute con total normalidad y así la experiencia del usuario sea más placentera ya agradable.

CODIGO IMPORTANTE DE FORMACION DE SNAKE

```
public class Snake1 extends javax.swing.JFrame implements KeyListener, Runnable{

    int snake = -1;
    boolean valido=true, activado=false, co=false;
    int direccion, puntos=0, contador;
    Point ultimo;
    int x,y;

    //Para el timer con hilo
     String minutos;
     String horas;
     String segundos;
     String ampm;
     Calendar calendario;
     //Hilo
     Thread h1;


     //Para los movimiento
     int cantidad_ar;
     int cantidad_ab;
     int cantidad_iz;
     int cantidad_de;

     String Facil;
     String Medio;
     String Dificil;
```

Dentro de la clase del Jframe Declararemos nuestras variables las cuales nos ayudaran a darle vida al juego.

CODIGO IMPORTANTE DE CONSTRUCTOR

```
public Snake1() {
    initComponents();
    this.setLocationRelativeTo(null);
    setIconImage(getIconImage());
    h1 = new Thread(this);
    h1.start();
    setSize(530,650);
    do {
        x=(int)Math.floor(Math.random()*23);
        y=(int)Math.floor(Math.random()*23);
        valido=false;



    } while (valido);
    jPanel1.setBounds(0, 0, 530, 650);
    JButton cabeza = new JButton();
```

```
cabeza.setBounds(x*20, y*20,20, 20);
cabeza.setContentAreaFilled(false);
cabeza.setOpaque(true);
cabeza.setBackground(Color.YELLOW);
cabeza.addKeyListener(this);

jPanell.add(cabeza);

JButton comida = new JButton();
comida.setBounds(x*20, y*20,20, 20);
comida.setContentAreaFilled(false);
comida.setOpaque(true);
comida.setBackground(Color.RED);
comida.addKeyListener(this);
jPanell.add(comida);
jPanell.addKeyListener(this);
```

Dentro de nuestro constructor lo que haremos será declarar las formas, partes esenciales, como lo será el tamaño de nuestro Jframe, la cabeza de la serpiente, su comida, y así mismo el tamaño de la misma, de qué color se ira mostrando.

CODIGO IMPORTANTE MOVIMIENTO DE LA SERPIENTE

Para el movimiento de la serpiente se decidio lo siguiente ye es que la serpiente se pudiera mover en lo que es el espacio de nuestro “Jpanel” el cual va a cumpli como nuestra cancha de juegos.

Todo esto dentro de un SWITCH el cual se hará cargo de cumplir cada uno de los casos

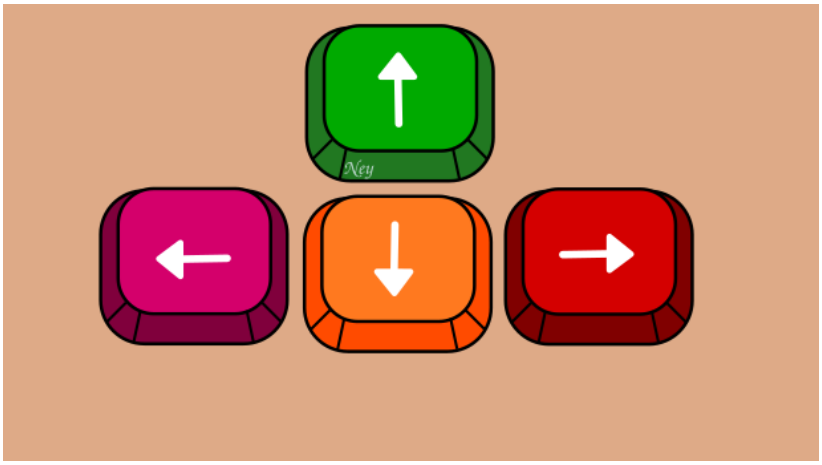
```
switch(direccion){
    case 1:
    cabeza.setLocation(cabeza.getX()+20, cabeza.getY());
    if (cabeza.getX()>=comida.getX()&cabeza.getX()<=comida.getX()+20) {
        if (cabeza.getY()>=comida.getY()&cabeza.getY()<=comida.getY()+20) {
            PartesSerp.add(cabeza.getLocation());
            serp.add(new JButton());
            snake++;
            serp.get(snake).setContentAreaFilled(false);
            serp.get(snake).setOpaque(true);
            serp.get(snake).setBounds(new Rectangle(PartesSerp.get(snake),new Dimension(2
            serp.get(snake).setBackground(Color.ORANGE);
            jPanell.add(serp.get(snake));


            do {
                for (int e = 0; e<=serp.size()-1; e++){
                    x=(int)Math.floor(Math.random()*23);
                    y=(int)Math.floor(Math.random()*23);

                    valido=(new Point(x*20, y*20).equals(serp.get(e).getLocation()));
                    if(!valido)break;
                }
}while (valido);
            comida.setLocation(x*20, y*20);
            puntos++;
            co=true;

   }

    break;
```

```
        case 2:
        cabeza.setLocation(cabeza.getX()-20, cabeza.getY());
        if (cabeza.getX()>=comida.getX()&cabeza.getX()<=comida.getX()+20) {
            if (cabeza.getY()>=comida.getY()&cabeza.getY()<=comida.getY()+20) {
                PartesSerp.add(cabeza.getLocation());
                serp.add(new JButton());
                snake++;
                serp.get(snake).setContentAreaFilled(false);
                serp.get(snake).setOpaque(true);
                serp.get(snake).setBounds(new Rectangle(PartesSerp.get(snake),new Dimension(20,20)));
                serp.get(snake).setBackground(Color.ORANGE);
                jPanell.add(serp.get(snake));


                do {
                    for (int e = 0; e<=serp.size()-1; e++){
                        x=(int)Math.floor(Math.random()*23);
                        y=(int)Math.floor(Math.random()*23);

                        valido=(new Point(x*20, y*20).equals(serp.get(e).getLocation()));
                        if(!valido)break;
                    }
    }while (valido);
                comida.setLocation(x*20, y*20);
                puntos++;
                co=true;
       }
}
        break;
```

```
    case 3:
    cabeza.setLocation(cabeza.getX(), cabeza.getY()+20);
    if (cabeza.getX()>=comida.getX()&cabeza.getX()<=comida.getX()+20) {
        if (cabeza.getY()>=comida.getY()&cabeza.getY()<=comida.getY()+20) {
            PartesSerp.add(cabeza.getLocation());
            serp.add(new JButton());
            snake++;
            serp.get(snake).setContentAreaFilled(false);
            serp.get(snake).setOpaque(true);
            serp.get(snake).setBounds(new Rectangle(PartesSerp.get(snake),new Dimension(20,20)));
            serp.get(snake).setBackground(Color.ORANGE);
            jPanell.add(serp.get(snake));


            do {
                for (int e = 0; e<=serp.size()-1; e++){
                    x=(int)Math.floor(Math.random()*23);
                    y=(int)Math.floor(Math.random()*23);

                    valido=(new Point(x*20, y*20).equals(serp.get(e).getLocation()));
                    if(!valido)break;
                }
}while (valido);
            comida.setLocation(x*20, y*20);
            puntos++;
            co=true;
    }

    break;
```

```
    case 4:
    cabeza.setLocation(cabeza.getX(), cabeza.getY()-20);
    if (cabeza.getX()>=comida.getX()&cabeza.getX()<=comida.getX()+20) {
        if (cabeza.getY()>=comida.getY()&cabeza.getY()<=comida.getY()+20) {
            PartesSerp.add(cabeza.getLocation());
            serp.add(new JButton());
            snake++;
            serp.get(snake).setContentAreaFilled(false);
            serp.get(snake).setOpaque(true);
            serp.get(snake).setBounds(new Rectangle(PartesSerp.get(snake),new Dimension(20,20)));
            serp.get(snake).setBackground(Color.ORANGE);
            jPanell.add(serp.get(snake));


            do {
                for (int e = 0; e<=serp.size()-1; e++){
                    x=(int)Math.floor(Math.random()*23);
                    y=(int)Math.floor(Math.random()*23);

                    valido=(new Point(x*20, y*20).equals(serp.get(e).getLocation()));
                    if(!valido)break;
                }
}while (valido);
            comida.setLocation(x*20, y*20);
            puntos++;
            co=true;
    }
}
    break;
   }
```

De esta manera se cumplirá con lo que son las instrucciones de juego en la cual nuestra serpiente sabrá como moverse y de la manera en la que el usuario interactúe con las teclas para jugar esta misma vaya haciendo sus movimientos.



CONDICIONES PARA ACABAR EL JUEGO

Para que el juego se termine existirán varias condiciones las cuales resumirles de la siguiente manera.

1. Que la serpiente choque con las 4 paredes de lo que sería el campo de juego
2. Que la serpiente choque con su propia cola
3. Que el jugador llegue a la cantidad máxima de puntos y gane

Tomando en cuenta esto se crearon los siguientes Ifs anidados o hechos por un bucle (for) el cual nos va a permitir que recorra la serie de instrucciones que se le indiquen al momento de que llegue a ese proceso.

```
for (int i = 0; i <=PartesSerp.size()-1 ; i++) {

    if(cabeza.getLocation().equals(PartesSerp.get(i))){



        contador++;
        System.out.println("contador");

        if (co) {
            if (contador==2) {
                JOptionPane.showMessageDialog(null, "Perdiste, tus puntos son" +puntos);
                GenerarHTML();
                System.exit(0);
            }
        }else{
            JOptionPane.showMessageDialog(null, "Perdiste, Tus puntos Son: "+puntos);
            GenerarHTML();
            System.exit(0);
        }
    }
}
```

Otro es:

```
for (int i = 0; i <=serp.size()-1; i++) {

    serp.get(i).setLocation(PartesSerp.get(i));

}
if (cabeza.getLocation().getX()>=500) {
    JOptionPane.showMessageDialog(null, "Perdiste, Tus puntos Son: "+puntos);
    GenerarHTML();

    System.exit(0);

}
if (cabeza.getLocation().getY()>=500) {
    JOptionPane.showMessageDialog(null, "Perdiste, Tus puntos Son: "+puntos);
    GenerarHTML();
    System.exit(0);
}
if (cabeza.getLocation().getX()<0) {
    JOptionPane.showMessageDialog(null, "Perdiste, Tus puntos Son: "+puntos);
    GenerarHTML();
    System.exit(0);
}
if (cabeza.getLocation().getY()<0) {
    JOptionPane.showMessageDialog(null, "Perdiste, Tus puntos Son: "+puntos);
    GenerarHTML();
    System.exit(0);
}
if (puntos==30) {
        JOptionPane.showMessageDialog(null, "Ganaste, tus puntos son: " +puntos);
        GenerarHTML();
        System.exit(0);

}

}
```

Estos son los que permitirán que nuestra serpiente se desplace y que al momento de que choque con las paredes o choque con su misma cola o que incluso el usuario gane, pues pueda llevarse su merecido mensaje que indicara que gano.

Como usted puede ver a medida que la serpiente vaya comiendo, empezara a crecer hasta ir tomando el tamaño de 30 espacios.

Como podemos ver este es otro momento en el cual la serpiente ya lleva más manzanas comidas dentro del juego y vamos notando su crecimiento.

INSTRUCCIONES EN LOS BOTONES

```
    private void facilMousePressed(java.awt.event.MouseEvent evt) {
        Facil="Facil";
        Medio="_____";
        Dificil="_____";
    }

    private void medioMousePressed(java.awt.event.MouseEvent evt) {
        Medio="Media";
        Facil="_____";
        Dificil="_____";
    }

    private void medioActionPerformed(java.awt.event.ActionEvent evt) {
        facil.setEnabled(false);
        medio.setEnabled(false);
        dificil.setEnabled(false);
        difi.setText("Dificultad: " +Medio);
    }

    private void dificilActionPerformed(java.awt.event.ActionEvent evt) {
        facil.setEnabled(false);
        medio.setEnabled(false);
        dificil.setEnabled(false);
        difi.setText("Dificultad: " +Dificil);
    }

    private void dificilMousePressed(java.awt.event.MouseEvent evt) {
        Dificil="Dificil";
        Facil="_____";
        Medio="_____";
    }
```

Estas serán también algunas de las instrucciones que dará lo que son los botones los cuales permitirán que el usuario tenga un paseo más grato por lo que es el juego.

MOVIMIENTOS DE LA SERPIENTE

Se solicitó un contador de movimiento por lo que se explicara lo que sucedió

CONTADOR DE MOVIMIENTOS

Se generó el método “KeyPressed” el cual lo que hará es permitirnos crear una serie de instrucciones los cuales se imprimirán por consola y no solo nos dirán que tecla está presionando el usuario, sino que también nos sumara la cantidad de veces que el usuario presione las mismas

```
@Override
public void keyPressed(KeyEvent e) {
    System.out.println("Presiona");
    if(e.getKeyCode()==KeyEvent.VK_UP){
        System.out.println("Arriba: " +cantidad_ar);
        //Contador de movimientos
        cantidad_ar++;

        if(direccion!=3)
            direccion=4;
            //ARRIBA
        }
        if (e.getKeyCode()==KeyEvent.VK_DOWN) {
            System.out.println("Abajo: " +cantidad_ab);
            //Contador de movimientos
            cantidad_ab++;

            if (direccion!=4)
                direccion=3;
                //ABAJO
            }
            if (e.getKeyCode()==KeyEvent.VK_LEFT) {
                System.out.println("Izquierda: " +cantidad_iz);
                //Contador de movimientos
                cantidad_iz++;

                if (direccion!=1)
                    direccion=2;
                    //IZQUIERDA
                }
                if (e.getKeyCode()==KeyEvent.VK_RIGHT) {
                    System.out.println("Derecha: " +cantidad_de);
                    //Contador de movimientos
                    cantidad_de++;

                    if (direccion!=2)
                        direccion=1;
                        //DERECHA
```

El cual se vera asi, este es el que nos permitira ver la cantidad de teclazos que da nuestro usuario al momento de presionar la tecla que corresponde al movimiento de nuestra serpiente.

GENERADOR DE HTML

Se solicitó que el usuario pudiera ver las estadísticas de juego por medio de un HTML, el cual se generara al momento de que el usuario pierda o gane su partida, no importando esto; El usuario podrá ver el HTML generado en los archivos del juego con el cual podrá ver las estadísticas de juego y corroborar todo lo que hizo dentro de su partida.

```
//Generador

public  void GenerarHTML() {
    FileWriter fichero=null;
    PrintWriter pw = null;

    //Try catch

    try {
        fichero = new FileWriter("202102775.html");
        pw = new PrintWriter(fichero);

        pw.println("<HTML>");
        pw.println("<body>");
        pw.println("<font face=nunito,arial,verdana>");
        //Titulo del HTML
        pw.println("<title>Reporte de Jugador</title>");
        pw.println("<center>");

        //Encabezado con mi informacion
        pw.println("<h1>Adrian Ismael Orlando Garcia Balan - 202102775</h1>");
        pw.println("<p>");
        pw.println("<p>");

        //Dificultad
        pw.println("<h2>Dificultad prueba</h2>");
        pw.println("<p>");
        pw.print("Jugo el nivel: " +Facil);
        pw.println("<p>");
        pw.print("Jugo el nivel: " +Medio);
        pw.println("<p>");
        pw.print("Jugo el nivel: " +Dificil);
        pw.println("<p>");
        pw.println("<p>");

        //Intervalo
        pw.println("<h2>Intervalo</h2>");
        pw.println("<p>");
        pw.println("<p>");
```

```
        //Intervalo
        pw.println("<h2>Intervalo</h2>");
        pw.println("<p>");
        pw.println("<p>");

        //Tamaño de la serpiente
        pw.println("<h2>Tamaño de Serpiente en puntos</h2>");
        pw.print("Su puntaje es: " +puntos);
        pw.println("<p>");
        pw.println("<p>");

        //Historial de movimientos
        pw.println("<h2>Historial de Movimientos</h2>");
        pw.println("Sus Movimientos Hacia Arriba Fueron: " +cantidad_ar);
        pw.println("<p>");
        pw.println("Sus Movimientos Hacia abajo Fueron: " +cantidad_ab);
        pw.println("<p>");
        pw.println("Sus Movimientos Hacia La Izquierda Fueron: " +cantidad_iz);
        pw.println("<p>");
        pw.println("Sus Movimientos Hacia La Derecha Fueron: " +cantidad_de);
        pw.println("<p>");
        pw.println("<p>");

        //Tiempo transcurrido
        pw.println("<h2>Tiempo Transcurrido</h2>");
        pw.println("Usted jugo en la hora: " +horas+":"+minutos+":"+segundos);
        pw.println("</font>");
        pw.println("</body>");
        pw.println("</HTML>");



    } catch (Exception i) {

        i.printStackTrace();
    }finally{
```

```
            //Tiempo transcurrido
            pw.println("<h2>Tiempo Transcurrido</h2>");
            pw.println("Usted jugo en la hora: " +horas+":"+minutos+":"+segundos);
            pw.println("</font>");
            pw.println("</body>");
            pw.println("</HTML>");



        } catch (Exception i) {

            i.printStackTrace();
        }finally{

            try {
                if(null != fichero){
                    fichero.close();
                }
            } catch (Exception i) {
                i.printStackTrace();
            }
        }
}
```

Adrian Ismael Orlando Garcia Balan - 202102775

Dificultad prueba

Jugo el nivel: _____

Jugo el nivel: _____

Jugo el nivel: Dificil

Intervalo

Tamaño de Serpiente en puntos

Su puntaje es: 30

Historial de Movimientos

Sus Movimientos Hacia Arriba Fueron: 28

Sus Movimientos Hacia abajo Fueron: 21

Sus Movimientos Hacia La Izquierda Fueron: 26

Sus Movimientos Hacia La Derecha Fueron: 23

Tiempo Transcurrido

Usted jugo en la hora: 04:53:42

TIEMPO EN EL QUE JUEGA

Se solicitó que el usuario pudiera ver la hora de juego en la cual estaba dentro de su partida y del juego por lo cual se creó un temporizador el cual tomara la hora exacta del dispositivo en el que estaba trabajando y así poder tener constancia de eso también

THREAD – HILOS

Se solicitó que algunas funciones y/o método fuera funcionando por medio de lo que es un hilo por que se decidió crear el hilo para lo que es el temporizador del juego el cual por medio de un Thread se decidió que llevara la cantidad de segundos que lleva

```
ti.setText(horas+":"+minutos+":"+segundos);
try {
    Thread.sleep(1000);
} catch (InterruptedException e) {
}
```

```
    @Override
    public void run() {
        Thread ct = Thread.currentThread();

        while(ct==hl){
            calcula();

            ti.setText(horas+":"+minutos+":"+segundos);
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
            }
        }
    }
//Calculadora de horas
    private void calcula() {
        Calendar calendario = new GregorianCalendar();
        Date fechaHoraActual = new Date();

        calendario.setTime(fechaHoraActual);
         ampm = calendario.get(Calendar.AM_PM) == Calendar.AM ? "AM" : "PM";
         if (ampm.equals("PM")) {
            int h = calendario.get(Calendar.HOUR_OF_DAY) - 12;
            horas = h > 9 ? "" + h : "0" + h;
            if(h==00){
                    horas="12";
             }else{
                    horas=h>9?""+h:"0"+h;
             }
        } else {
            horas = calendario.get(Calendar.HOUR_OF_DAY) > 9 ? "" + calendario.get(Calendar.HOUR_OF_DAY) : "0" + calendario.get(Calendar.HOUR_OF_DAY);
        }
        minutos = calendario.get(Calendar.MINUTE) > 9 ? "" + calendario.get(Calendar.MINUTE) : "0" + calendario.get(Calendar.MINUTE);
        segundos = calendario.get(Calendar.SECOND) > 9 ? "" + calendario.get(Calendar.SECOND) : "0" + calendario.get(Calendar.SECOND);
    }
```

De esta manera culminamos con el manual del juego SNAKE por parte de la compañía desarrolladora Phenyx Computer.