

Manual Técnico 2022

22 SEPTIEMBRE

Universidad San Carlos de Guatemala
Jose Andres Hinestroza García
202100316

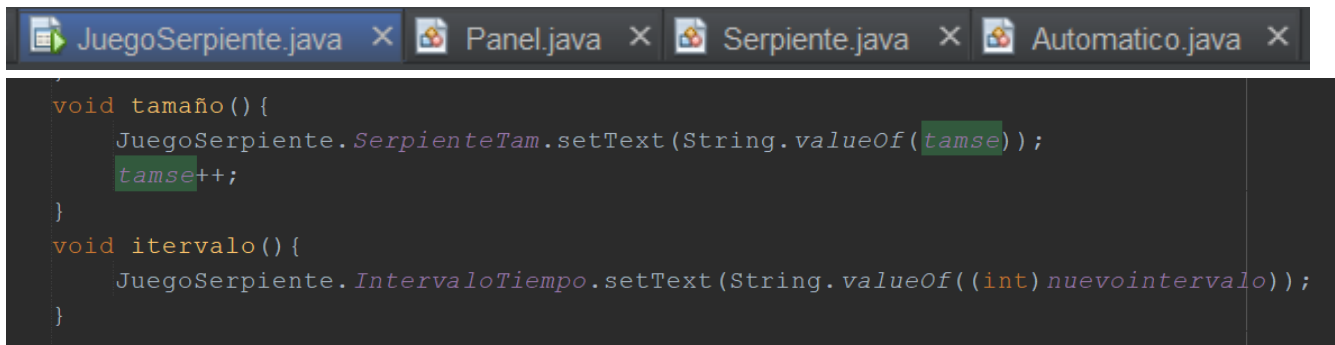


Nombre del
logotipo

Manual Técnico

La siguiente práctica se diseñó a base de lenguaje de Java en Apache NetBeans.

Se realizaron tres clases para poder distinguir entre cada menú en el cual el usuario podrá interactuar, identificándose de igual manera cada clase con el menú en el cual se va interactuar



```
void tamano() {  
    JuegoSerpiente.SerpienteTam.setText(String.valueOf(tamse));  
    tamse++;  
}  
  
void intervalo() {  
    JuegoSerpiente.IntervaloTiempo.setText(String.valueOf((int) nuevointervalo));  
}
```

En los siguientes métodos de la clase serpiente, nos servirá para poder modificar el label con el cual en el juego nos mostrará las características del juego



```
public void Dificultad() {  
    int dificultad = JuegoSerpiente.Nivel.getSelectedIndex();  
    JuegoSerpiente Jugando = new JuegoSerpiente();  
    System.out.println(dificultad);  
    if (dificultad == 0) {  
        nuevointervalo = (int) (MovimientoSerpiente.intervalo) * (1-0.03);  
        Jugando.IntervaloTiempo.setText(String.valueOf(nuevointervalo));  
        System.out.println("intervalo de "+nuevointervalo);  
        MovimientoSerpiente.intervalo = (int) nuevointervalo ;  
        intervalo();  
    }  
    if (dificultad == 1) {  
        nuevointervalo = (int) (MovimientoSerpiente.intervalo) * (1-0.06);  
        System.out.println("intervalo de "+nuevointervalo);  
        MovimientoSerpiente.intervalo = (int) nuevointervalo ;  
        intervalo();  
    }  
    if (dificultad == 2) {  
        nuevointervalo = (int) (MovimientoSerpiente.intervalo) * (1-0.09);  
        System.out.println("intervalo de "+nuevointervalo);  
        MovimientoSerpiente.intervalo = (int) nuevointervalo ;  
        intervalo();  
    }  
}
```

En el según el nivel con el cual nosotros queramos iniciar el juego, este empezará su partida, y su intervalo se irá modificando según lo estipulado.

```

public void jugar() {
    initComponents();
    SerpientePanel= new Serpiente(300,10);
    jPanel1.add(SerpientePanel);
    SerpientePanel.setBounds(10,10,800,800);
    SerpientePanel.setOpaque(false);
    Panel colorpanel = new Panel(300, 10);
    jPanel1.add(colorpanel);
    colorpanel.setBounds(10, 10, 800, 800);
}

```

En el nos dibujara en el panel la serpiente para que esta pueda iniciar, esta a su vez solo empezara su juego al momento de hacer su llamado, con el cual se hará mediante el boton de “empezar”

```

public void Comida() {
    boolean busqueda = false;
    int a = (int) (Math.random()*can); // busca numeros
    int b = (int) (Math.random()*can);
    for (int[] par : snake) {
        if (par[0] == a && par[1] == b) {
            busqueda = true;
            Comida();
            break;
        }
    }
    if (!busqueda) {
        this.comida[0] = a;
        this.comida[1] = b;
    }
}
}

```

En el metodo de la comida hace uso de la librería random puesto que con esta podemos ir variando las cantidad, coordenadas en la cual se nos presentara, siendo “can” la cantidad de cuadros los cuales nosotros hayamos colocado.

```
@Override
public void paint(Graphics pintar) {
    super.paint(pintar); // parta regraficar
    pintar.setColor(coloserpiente);
    for (int[] par : snake) { //serpiente
        pintar.fillRect(res / 2 + par[0] * tam, res / 2 + par[1] * tam, tam - 1, tam - 1);
    }
    pintar.setColor(colorcomida);
    pintar.fillRect(res / 2 + comida[0] * tam, res / 2 + comida[1] * tam, tam - 1, tam - 1);
}
```

En este metodo hace uso de un override puesto a que esta se estará sobrescribiendo cada vez que esta sea llamado.