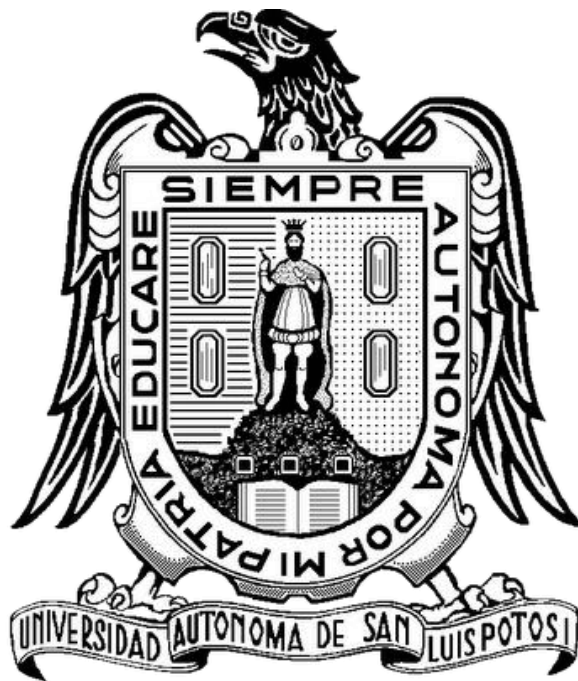


26 DE JUNIO DE 2015



MANUAL DEL PROGRAMADOR

INTER SEMESTRAL PROGRAMACIÓN ORIENTADA A OBJETOS

ROGELIO DANIEL GONZALEZ NIETO
UNIVERSIDAD AUTONOMA DE SAN LUIS POTOSI

Índice

DATOS DEL PROYECTO	2
OBJETIVO DEL PROYECTO	2
DESCRIPCION DEL PROYECTO	2
DESCRIPCION DE CADA NIVEL	2
DIAGRAMA DE CLASES UML	4
CARACTERISTICAS Y COMPORTAMIENTO DE CADA CLASE.....	6

DATOS DEL PROYECTO

- Nombre: Chavo's Kong.
- No. De Proyecto: 237.

OBJETIVO DEL PROYECTO

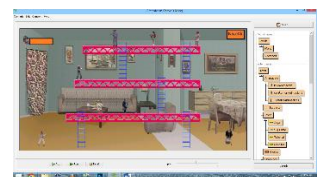
Conseguir que el jugador esquive los obstáculos y llegue hasta la última plataforma para rescatar a la Popis.

DESCRIPCION DEL PROYECTO

El juego consiste de plataformas como las mostradas en la imagen, el jugador tendrá que ir subiendo a través de ellas para poder rescatar a la princesa de los enemigos que la tienen secuestrada. Existirán obstáculos tales como manzanas, plátanos, triciclos que irán cayendo y tendrán que ser esquivadas, también caerán tortas de jamón que si el jugador las agarra ganara vida. También aparecerán algunos personajes extra sobre las plataformas tales como Don Ramón o el Sr. Barriga los cuales golpearan al chavo y le harán perder bastante vida. Al lograr rescatar a la princesa el jugador ganara el nivel y pasara al siguiente.

DESCRIPCION DE CADA NIVEL

- Nivel 1: El nivel 1 se ambientara en el patio principal de la vecindad del chavo, el jugador tiene que esquivar los obstáculos arrojados por la chilindrina, para que el jugador gane el nivel es necesario llegar hasta la Popis.
- Nivel 2: El nivel 2 se ambientara en el departamento de Don Ramón, es este nivel además de esquivar los obstáculos abra que esquivar también los golpes de don Ramón, Quico y Jaimito el Cartero.
- Nivel 3: El nivel 3 se ambientara en el departamento de Quico y de Doña Florinda, como en el nivel pasado abra que esquivar los obstáculos y también los golpes del señor Barriga, Don Ramón y Jaimito el Cartero.



- Nivel 4: El nivel 4 se ambientara en la salón de clases de la escuela, como en los niveles pasados el objetivo es llegar hasta la Popis esquivando a los enemigos
- Nivel 5: El nivel 5 se ambientara en el 2do patio de la vecindad, acabando el nivel el juego terminara, el objetivo sigue siendo el mismo.

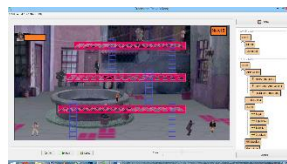
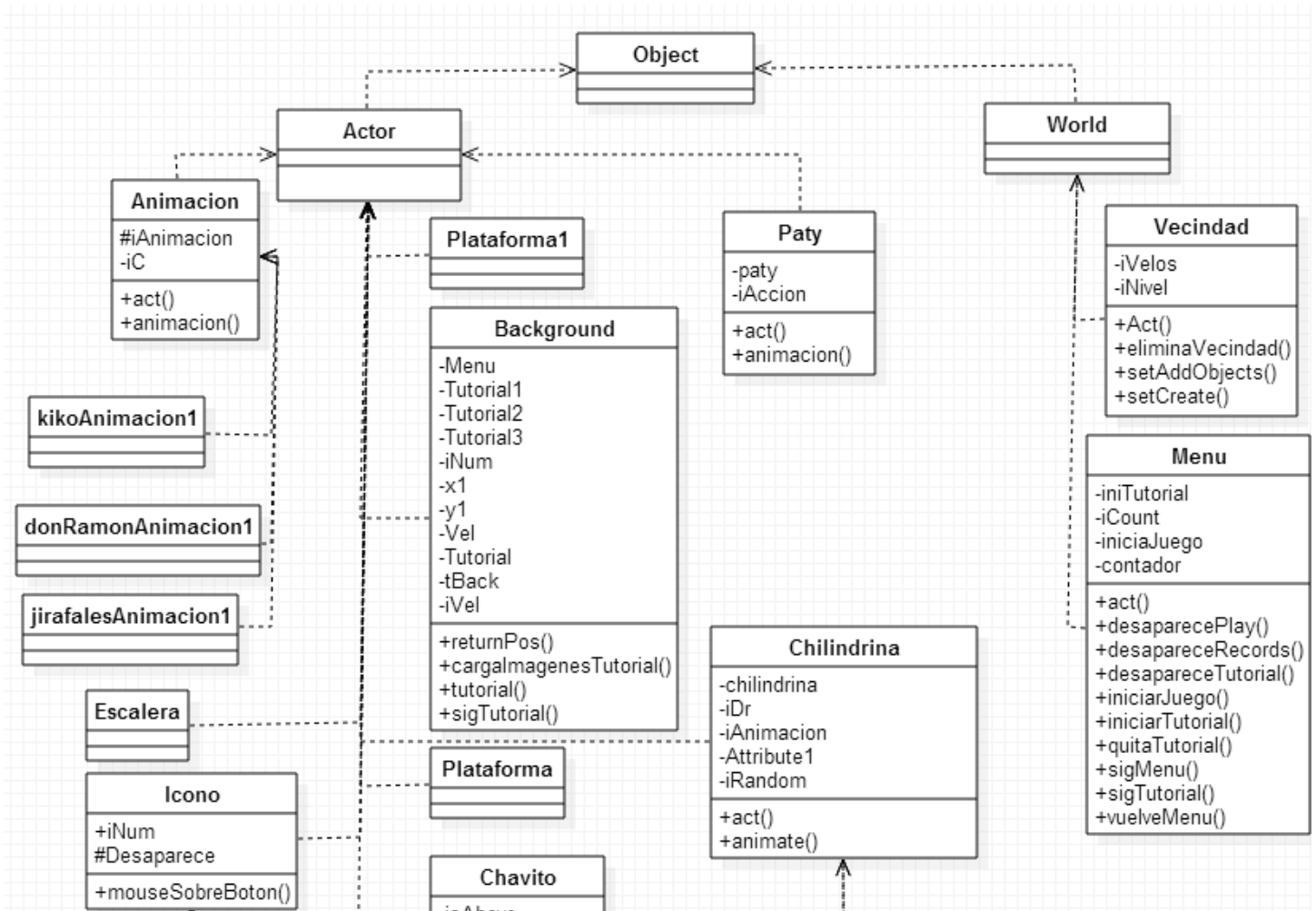
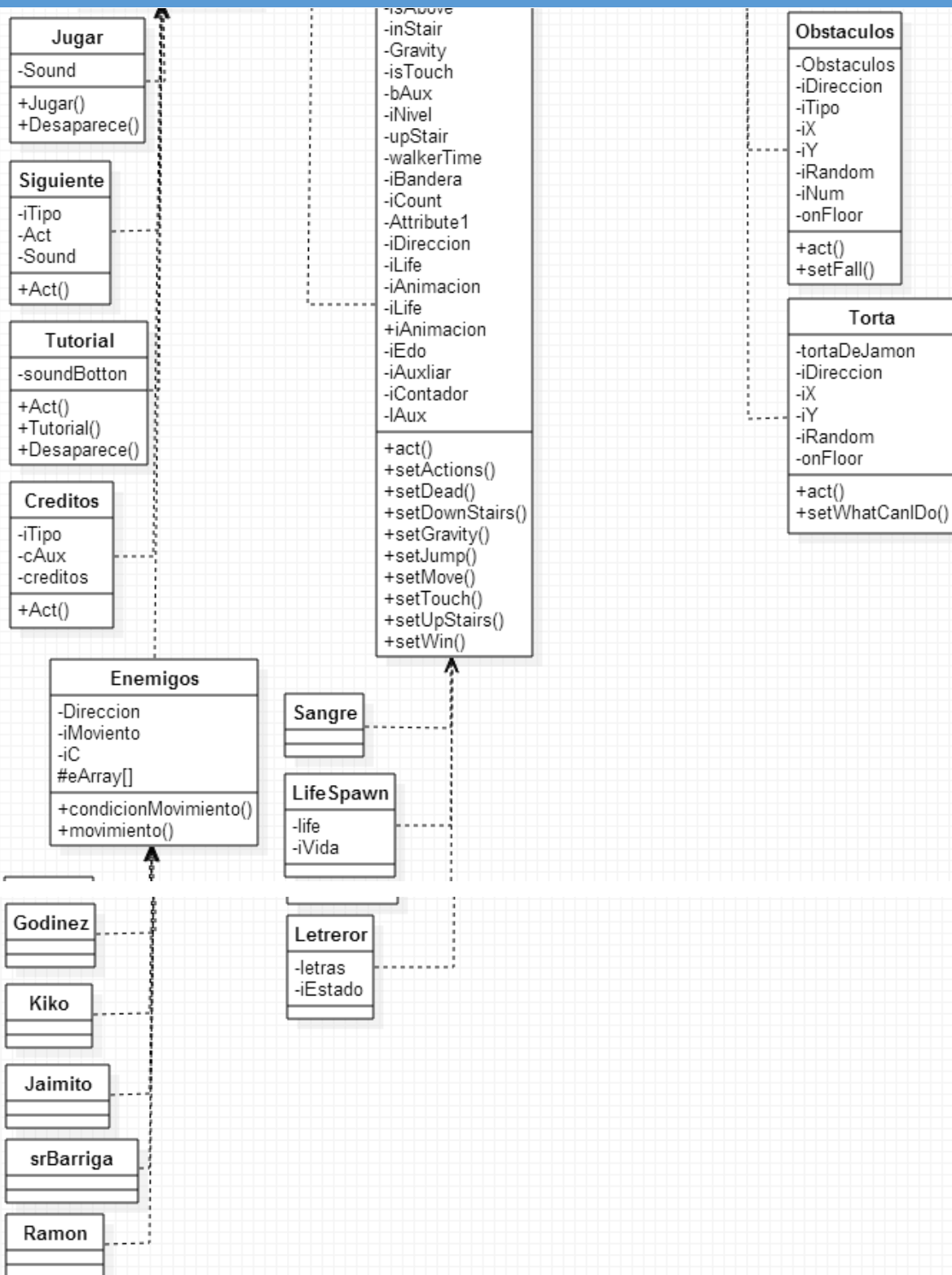


DIAGRAMA DE CLASES UML





CARACTERISTICAS Y COMPORTAMIENTO DE CADA CLASE

Nombre De La Clase:	Vecindad
Características:	Añade todos los objetos al mundo dependiendo del nivel en que estemos jugando
Nombre De La Clase:	Menú
Características:	Añade todo lo relacionado con el menú, al darle clic en cada botón nos llevara a su respectiva acción.
Nombre De La Clase:	Animación
Características:	Brinda el efecto de animación a los personajes extras de cada nivel.
Nombre De La Clase:	Icono
Características:	Cada que se hace clic en un botón este lo lleva a su respectivo menú
Nombre De La Clase:	Escalera
Características:	Añade al mundo una escalera por la cual subirá el personaje para cambiar de plataforma.
Nombre De La Clase:	Plataforma
Características:	Añade una plataforma al mundo en la cual se moverá el personaje.
Nombre De La Clase:	Background
Características:	Añade las imágenes de los tutoriales y de los créditos, y las muestra en pantalla.
Nombre De La Clase:	Enemigos
Características:	Carga las imágenes de los enemigos en pantalla y los anima para que al encontrarse con el jugador resten vida a este. También hereda dicho comportamiento a sus subclases para que cada una cargue imágenes diferentes
Nombre De La Clase:	Chilindrina
Características:	Envía obstáculos al personaje para que este los tenga que esquivar, de caso contrario le restaran vida, también envía tortas de jamón para que le sumen vida al personaje.
Nombre De La Clase:	Chavito
Características:	Es la clase principal de todo el juego, esta se encarga de controlar al personaje en su totalidad, darle y quitarle vida, entre otras cosas.
Nombre De La Clase:	Obstáculos
Características:	Cada que tocan al jugador disminuye su vida
Nombre De La Clase:	Torta de Jamón
Características:	Cada que tocan al jugador aumenta su vida.

HERENCIA Y POLIMORFISMO

La herencia se usa principalmente en las clases animación y enemigo, debido a que cada sub clase de dichas clases tiene el mismo comportamiento pero varían en la imagen que muestra en pantalla, la clase Enemigo hereda su método animación a sus subclases.

La clase animaciones hereda su método animación a sus subclases, aunque ambos métodos se llamen igual no son el mismo ya que se encargan de otorgar un comportamiento diferente a los objetos de dichas clases.

CRONOGRAMA DE ACTIVIDADES

El proyecto se encuentra alojado dentro del servidor de greenfoot, para acceder a él se necesita visitar el siguiente link:

<http://www.greenfoot.org/scenarios/14514>

El siguiente enlace es al video de YouTube con una muestra del funcionamiento del proyecto:

<https://www.youtube.com/watch?v=IOQhjcYsCZO>

El manual de usuario se anexa en la misma carpeta en la que se encuentra este documento.

CODIGO DE EL PROYECTO

El proyecto fue programado en lenguaje java con el editor Greenfoot y el compilador jdk, a continuación anexo el código de dicho proyecto:

- Clase Menú:


```
public class Menu extends World
{
    private Tutorial tAyuda;
    private Jugar jJuego;
    private Siguiente sFlecha;
    private Background bEscenario;
    private Background bAyuda;
    private Credits cQuien;
    private int iIniTutorial;
    private int iCount;
    private boolean iniciaJuego;
    private int contador;
    public Menu()
```



```

{
    //Crea el mundo del menu
    super(1020, 544, 1);
    iCount = 0;
    iIniTutorial = 0;
    contador=0;
    iniciaJuego = false;
    setMenu();
}

```

```

public void act(){
    if(!iniciaJuego)
    {
        if(iIniTutorial == 1)
        {
            if(iCount!=115)
            {
                iCount++;
            }
            if(iCount == 110)
            {
                bAyuda = new Background(0,0,0,2);
                addObject(bAyuda,(getWidth()/2),(getHeight()/2));
            }
        }
    }
    else
    {
        contador++;
        if(contador==120){
            Greenfoot.setWorld(new Vecindad());
        }
    }
}

```

```

/**
 * Añade los botones del menu
 */
public void setMenu(){//Crea El Menu
    iIniTutorial = 0;
    bEscenario = new Background(0,0,0,0);
    addObject(bEscenario,(getWidth()/2),(getHeight()/2));
    jJuego = new Jugar();
    addObject(jJuego,230,450);
}

```

```

    tAyuda = new Tutorial();
    addObject(tAyuda,480,450);
    cQuien = new Creditos(1);
    addObject(cQuien,730,450);
}

public void desapareceTutorial(){
    jJuego.desaparece();
}

public void desapareceRecords(){
    jJuego.desaparece();
    tAyuda.desaparece();
}

public void desaparecePlay(){
    tAyuda.desaparece();
}

public void sigTutorial(){
    bAyuda.sigTutorial();
    ilniTutorial = 0;
}

public void iniciaTutorial(){
    ilniTutorial = 1;
}

public void quitaTutorial(){
    removeObject(bAyuda);
    removeObject(sFlecha);
    removeObject(bEscenario);
    ilniTutorial = 0;
    iCount = 0;
    setMenu();
}

public void vuelveMenu(){
    setMenu();
}

public void iniciarJuego()
{
    contador = 0;
    iniciaJuego = true;
}
}

```

- Clase Vecindad:

```

public class Vecindad extends World
{
    private GreenfootSound soundBoton = new GreenfootSound("musica-01.wav");
    private List listAux = new ArrayList();
    private List listAux2 = new ArrayList();
    private List<Enemigos> listAux3 = new ArrayList<Enemigos>();
    private List<Animacion> listAux4 = new ArrayList<Animacion>();
    private Chavito chavo;
    private Chilindrina chilin;
    private Plataforma1 plataforma;
    private Plataforma2 plataforma1;
    private Plataformita p1;
    private Plataforma pla1;
    private Plataforma pla2;
    private Plataforma pla3;
    private Escalera stair;
    private Escalera stair2;
    private Escalera stair3;
    private Escalera stair4;
    private Escalera stair5;
    private Escalera stair6;
    private Escalera stair7;
    private Escalera stair8;
    private Paty popis;
    private Vecindad1 esc1;
    private Vecindad2 esc2;
    private Vecindad3 esc3;
    private Vecindad4 esc4;
    private Vecindad5 esc5;
    private int iNivel;
    private LifeSpawn barra;
    private donRamonAnimacion1 ramon;
    private jirafalesAnimacion1 jirafa;
    private kikoAnimacion1 federico;
    private srBarriga barriga;
    private Godinez godi;
    private Jaimito cartero;
    private Kiko cachetes;
    private Ramon ronDamon;
    private int iVelos;
    /**
     * Crea todos los objetos;
     */
    public Vecindad()
    {
        super(1020, 544, 1);
        barra = new LifeSpawn(0);
        iNivel = 1;
        iVelos = iNivel;
        chavo = new Chavito();
    }
}

```

```

chilin = new Chilindrina();
plataforma = new Plataforma1();
plataforma1 = new Plataforma2();
pla1 = new Plataforma();
pla2 = new Plataforma();
pla3 = new Plataforma();
stair = new Escalera();
stair2 = new Escalera();
stair3 = new Escalera();
stair4 = new Escalera();
stair5 = new Escalera();
stair6 = new Escalera();
stair7 = new Escalera();
popis = new Paty();
esc1 = new Vecindad1();
esc2 = new Vecindad2();
esc3 = new Vecindad3();
esc4 = new Vecindad4();
esc5 = new Vecindad5();
ramon = new donRamonAnimacion1();
jirafa = new jirafalesAnimacion1();
federico = new kikoAnimacion1();
barriga = new srBarriga();
godí = new Godínez(iVelos*2);
cartero = new Jaimito(iVelos*2);
cachetes = new Kiko(iVelos*3);
ronDamon = new Ramon(iVelos*3);
}

/**
 * Llama al metodo que crea todo el mundo.
 */
public void act()
{
    setAddObjects(iNivel);
}

public void setAddObjects(int iAcomodar)
{
    /**
     * Dependiendo del nivel
     * se añadiran en cierto orden las ecaleras
     * y tambien las plataformas.
     */
    soundBoton.play();
    iNivel = iAcomodar;
    if(iNivel == 1){
        addObject(esc1,(getWidth()/2),(getHeight()/2));
        addObject(barra,80,50);
        addObject(pla1,511,245);
    }
}

```

```

    addObject(pla2,473,395);
    addObject(pla3,549,95);
    addObject(plataforma,511,282);
    addObject(stair,280,450);
    addObject(stair2,750,450);
    addObject(stair3,640,298);
    addObject(stair4,465,145);
    addObject(stair5,750,145);
    addObject(chavo,100,485);
    addObject(chilin,600,40);
    addObject(popis,800,40);
    addObject(ramon,(getWidth()-80),485);
    addObject(barriga,300,190);
    addObject(godi,400,340);
    addObject(cartero,500,40);
}
if(iNivel == 2){
    addObject(esc2,(getWidth()/2),(getHeight()/2));
    addObject(barra,80,50);
    addObject(pla1,511,245);
    addObject(pla2,473,395);
    addObject(pla3,549,95);
    addObject(plataforma,511,282);
    addObject(stair,300,450);
    addObject(stair2,750,450);
    addObject(stair3,500,298);
    addObject(stair4,600,145);
    addObject(stair5,400,145);
    addObject(stair7,700,145);
    addObject(chavo,100,485);
    addObject(chilin,600,40);
    addObject(popis,800,50);
    addObject(jirafa,(getWidth()-80),485);
    addObject(cachetes,300,190);
    addObject(ronDamon,400,340);
    addObject(cartero,500,40);
}
if(iNivel == 3){
    addObject(esc3,(getWidth()/2),(getHeight()/2));
    addObject(barra,80,50);
    addObject(pla1,511,245);
    addObject(pla2,473,395);
    addObject(pla3,549,95);
    addObject(plataforma,511,282);
    addObject(stair,280,450);
    addObject(stair2,750,450);
    addObject(stair3,640,298);
    addObject(stair4,465,145);
    addObject(stair5,750,145);
    addObject(chavo,100,485);

```

```

    addObject(chilin,600,40);
    addObject(popis,800,50);
    addObject(federico,(getWidth()-80),485);
    addObject(ronDamon,300,190);
    addObject(barriga,400,340);
    addObject(cartero,500,40);
}
if(iNivel == 4){
    addObject(esc4,(getWidth()/2),(getHeight()/2));
    addObject(barra,80,50);
    addObject(pla1,511,245);
    addObject(pla2,473,395);
    addObject(pla3,549,95);
    addObject(plataforma,511,282);
    addObject(stair,300,450);
    addObject(stair2,750,450);
    addObject(stair3,500,298);
    addObject(stair4,600,145);
    addObject(stair5,400,145);
    addObject(stair7,700,145);
    addObject(chavo,100,485);
    addObject(chilin,600,40);
    addObject(popis,800,50);
    addObject(ramon,(getWidth()-80),485);
    addObject(jirafa,(getWidth()-180),485);
    addObject(barriga,300,190);
    addObject(godi,400,340);
    addObject(cartero,500,40);
}
if(iNivel == 5){
    addObject(esc5,(getWidth()/2),(getHeight()/2));
    addObject(barra,80,50);
    addObject(pla1,511,245);
    addObject(pla2,473,395);
    addObject(pla3,549,95);
    addObject(plataforma,511,282);
    addObject(stair,280,450);
    addObject(stair2,750,450);
    addObject(stair3,640,298);
    addObject(stair4,465,145);
    addObject(stair5,750,145);
    addObject(chavo,100,485);
    addObject(chilin,600,40);
    addObject(popis,800,50);
    addObject(federico,(getWidth()-80),485);
    addObject(jirafa,(getWidth()-180),485);
    addObject(barriga,300,190);
    addObject(godi,400,340);
    addObject(cartero,500,40);
}

```

```

}

/**
 * Remueve todos los objetos al cambiar de nivel.
 */
public void eliminaVecindad(int Acomodar)
{
    removeObject(chilin);
    removeObject(barra);
    removeObject(chavo);
    removeObject(popis);
    removeObject(pla1);
    removeObject(pla2);
    removeObject(pla3);
    removeObject(plataforma);
    removeObject(plataforma1);
    removeObject(stair);
    removeObject(stair2);
    removeObject(stair3);
    removeObject(stair4);
    removeObject(stair5);
    removeObject(stair6);
    removeObject(stair7);
    removeObject(p1);
    listAux = getObjects(Obstaculos.class);
    listAux2 = getObjects(Torta.class);
    listAux3 = getObjects(Enemigos.class);
    listAux4 = getObjects(Animacion.class);
    removeObject(listAux);
    removeObject(listAux2);
    removeObject(listAux3);
    removeObject(listAux4);
    iNivel = iNivel + 1;
    setCreate(iNivel);
}

/**
 * Despues de eliminar el mundo crea uno nuevo con nuevas condiciones.
 */
public void setCreate(int Acomodar)
{
    //iNivel = iNivel + 1;
    if(Acomodar == 1){
        esc1 = new Vecindad1();
    }
    if(Acomodar == 2){
        esc2 = new Vecindad2();
    }
    if(Acomodar == 3){
        esc3 = new Vecindad3();
    }
}

```

```

    }
    if(Acomodar == 4){
        esc4 = new Vecindad4();
    }
    if(Acomodar == 5){
        esc5 = new Vecindad5();
    }
    barra = new LifeSpawn(0);
    chavo = new Chavito();
    chilin = new Chilindrina();
    plataforma = new Plataforma1();
    plataforma1 = new Plataforma2();
    pla1 = new Plataforma();
    pla2 = new Plataforma();
    pla3 = new Plataforma();
    stair = new Escalera();
    stair2 = new Escalera();
    stair3 = new Escalera();
    stair4 = new Escalera();
    stair5 = new Escalera();
    stair6 = new Escalera();
    stair7 = new Escalera();
    popis = new Paty();
    setAddObjects(iNivel);
}
}
}

```

- Clase Animación:

```

public class Animacion extends Actor
{
    protected GreenfootImage[] eArray = new GreenfootImage[3];
    public int iAnimacion;
    private int iC;
    /**
     * Inicializa en null el array, ya que despues las subclases lo inicializaran de manera correcta
     */
    public Animacion()
    {
        iAnimacion = 0;
        for(iC=0; iC<3; iC++)
        {
            eArray[iC] = null;
        }
    }

    /**
     * Animacion1 es la animacion a 3 imagenes
     */
    public void Animacion1()

```



```

{
    if(iAnimacion == 0){
        setImage(eArray[0]);
    }
    if(iAnimacion == 30){
        setImage(eArray[1]);
    }
    if(iAnimacion == 60){
        setImage(eArray[2]);
    }
    if(iAnimacion == 80){
        iAnimacion = -1;
    }
    iAnimacion++;
}

/**
 * Animacion2 es la animacion a 2 imagenes
 */
public void Animacion2()
{
    if(iAnimacion == 0){
        setImage(eArray[0]);
    }
    if(iAnimacion == 30){
        setImage(eArray[1]);
    }
    if(iAnimacion == 60){
        iAnimacion = -1;
    }
    iAnimacion++;
}
}

```

▪ Clase kikoAnimacion1:

```

public class kikoAnimacion1 extends Animacion
{
    /**
     * Hereda de su superclase la variable eArray y la inicializa con las imagenes
     * correspondientes a la animacion
     */
    public kikoAnimacion1(){
        eArray[0] = new GreenfootImage("kikoanimacion1.png");
        eArray[1] = new GreenfootImage("kikoanimacion2.png");
        eArray[2] = new GreenfootImage("kikoanimacion3.png");
    }

    /**

```

```

    * Hereda el metodo Animacion1 de su superclase animacion
    */
    public void act(){
        Animacion1();
    }
}

```

- Clase donRamonAnimacion1:

```

public class donRamonAnimacion1 extends Animacion
{
    /**
     * Hereda de su superclase la variable eErray y la inicialisa con las imagenes
     * correspondientes a la animacion
     */
    public donRamonAnimacion1(){
        eArray[0] = new GreenfootImage("dranimacion1.png");
        eArray[1] = new GreenfootImage("dranimacion2.png");
    }

    /**
     * Hereda Animaicon2 de su superclase Animacion
     */
    public void act(){
        Animacion2();
    }
}

```

- Clase jirfalesAnimacion1:

```

public class jirafalesAnimacion1 extends Animacion
{
    /**
     * Hereda de su superclase la variable eErray y la inicialisa con las imagenes
     * correspondientes a la animacion
     */
    public jirafalesAnimacion1(){
        eArray[0] = new GreenfootImage("jirafalesanimacion1.png");
        eArray[1] = new GreenfootImage("jirafalesanimacion2.png");
        eArray[2] = new GreenfootImage("jirafalesanimacion3.png");
    }
}

```

```

/**
 * Hereda el metodo Animacion1 de su superclase animacion
 */
public void act(){
    Animacion1();
}
}

```

- Clase Icono:

```

public class Icono extends Actor
{
    private int iNum;
    protected int desaparece;
    public void act()
    {
        iNum = 0;
    }

    public void mouseSobreBoton(){
        if(Greenfoot.mouseMoved(this)){
            this.getImage().scale(160,60);
        }
        else{
            this.getImage().scale(150,50);
        }
    }
}

```

- Clase Jugar:

```

public class Jugar extends Icono
{
    private GreenfootSound soundBoton = new GreenfootSound("Sound1.wav");
    public void act()
    {
        if(Greenfoot.mouseClicked(this))
        {
            //this.sonidoBoton();
            soundBoton.play();
            World aux=getWorld();
            ((Menu)aux).iniciarJuego();
        }
    }
}

```

```

    }
    mouseSobreBoton();
}

public Jugar()
{
    this.getImage().scale(150,50);
    desaparece=0;
}

public void desaparece()
{
    desaparece=1;
}
}

```

- Clase Siguiente:

```

public class Siguiente extends Icono
{
    private int iTipo;
    private Actor act;
    private GreenfootSound soundBoton = new GreenfootSound("Sound1.wav");
    public Siguiente(int tipo, Actor actor){
        iTipo = tipo;
        act = actor;
    }

    public void act()
    {
        if(iTipo == 1){
            this.getImage().scale(40,40);
            if(Greenfoot.mouseClicked(this))
            {
                soundBoton.play();
                this.getImage().scale(60,60);
                World aux=getWorld();
                ((Menu)aux).sigTutorial();
            }
        }
        if(iTipo == 2){
            if(Greenfoot.mouseClicked(this)){
                soundBoton.play();
                World aux=getWorld();
                ((Menu)aux).vuelveMenu();
                getWorld().removeObject(act);
                getWorld().removeObject(this);
            }
        }
    }
}

```

```
}
```

- Clase Tutorial:

```
public class Tutorial extends Icono
{
    private GreenfootSound soundBoton = new GreenfootSound("Sound1.wav");
    public void act()
    {
        if(Greenfoot.mouseClicked(this))
        {
            World aux = getWorld();
            ((Menu)aux).iniciaTutorial();
            soundBoton.play();
        }
        mouseSobreBoton();
    }

    public Tutorial()
    {
        this.getImage().scale(150,50);
        desaparece = 0;
    }

    public void desaparece(){
        desaparece = 1;
    }
}
```

- Clase Creditos:

```
public class Creditos extends Icono
{
    private int iTipo;
    private Creditos cAux;
    private GreenfootImage creditos = new GreenfootImage("Creditos1.jpg");
```

```

private GreenfootSound soundBoton = new GreenfootSound("Sound1.wav");
public Credits(int tip){
    desaparece = 0;
    iTipo = tip;

    if(tip == 1){
        this.getImage().scale(150,50);
    }

    if(tip == 2){
        setImage(creditos);
    }
}

```

```

public void act()
{
    if(Greenfoot.mouseClicked(this))
    {
        soundBoton.play();
        this.getImage().scale(150+10,50+10);
        cAux = new Credits(2);
        getWorld().addObject(cAux,510,(544/2));
        //getWorld().addObject(cAux,320,(getWorld().getHeight()/2));
        //(getWidth()/2),(getHeight()/2));
        getWorld().addObject(new Siguiente(2,cAux),871,300);
    }
    else{
        if(iTipo == 1){
            this.getImage().scale(150,50);
        }
    }
}
}

```

- Clase Enemigos:

```

public class Enemigos extends Actor
{
    private int iDireccion;
    private int iMovimiento;
    private int iC;
    protected GreenfootImage[] eArray = new GreenfootImage[10];
    public Enemigos()
    {
        iDireccion = 1;
        iMovimiento = 0;
        for(iC=0; iC<10; iC++)
        {
            eArray[iC] = null;
        }
    }
}

```

```

    }
}

public void condicionMovimiento()
{
    if(isTouching(Plataforma.class)){
        movimiento();
    }
    else{
        if(!isTouching(Plataforma.class)&&(iDireccion == 1)){
            iDireccion = 2;
            move(-2);
            setImage(eArray[1]);
            iMovimiento = 1;
        }
        if(!isTouching(Plataforma.class)&&(iDireccion == 2)){
            iDireccion = 1;
            move(2);
            setImage(eArray[0]);
            iMovimiento = -1;
        }
    }
}

}

public void movimiento()
{
    /**
     * iMovimiento se encarga de la animacion del personaje.
     */
    if(iDireccion == 1){
        move(2);
        if(iMovimiento == 1){
            setImage(eArray[0]);
        }
        if(iMovimiento == 4){
            setImage(eArray[2]);
        }
        if(iMovimiento == 7){
            setImage(eArray[4]);
        }
    }
    if((!isTouching(Chavito.class))&&(iMovimiento == 10)){
        iMovimiento = 0;
        setImage(eArray[0]);
    }
    if(isTouching(Chavito.class)){
        if(iMovimiento == 13){
            setImage(eArray[6]);
            iMovimiento++;
        }
    }
}

```

```

        if(iMovimiento == 16){
            setImage(eArray[8]);
            iMovimiento = 0;
        }
    }
    iMovimiento++;
}
if(iDireccion == 2){
    move(-2);
    if(iMovimiento == -1){
        setImage(eArray[1]);
        iMovimiento--;
    }

    if(iMovimiento == -4){
        setImage(eArray[3]);
        iMovimiento--;
    }
    if(iMovimiento == -7){
        setImage(eArray[5]);
        iMovimiento--;
    }
    if((!isTouching(Chavito.class))&&(iMovimiento == -10)){
        iMovimiento = 0;
        setImage(eArray[1]);
    }
    if(isTouching(Chavito.class)){
        if(iMovimiento == -13){
            setImage(eArray[7]);
            iMovimiento--;
        }
        if(iMovimiento == -16){
            setImage(eArray[9]);
            iMovimiento = 0;
        }
    }
    iMovimiento--;
}
}
}
}

```

- Clase Godinez:

```

public class Godinez extends Enemigos
{
    public Godinez(int iVel)
    {
        eArray[0] = new GreenfootImage("godinez1.png");
        eArray[1] = new GreenfootImage("godinez1izq.png");
    }
}

```



```

    eArray[2] = new GreenfootImage("godinez2.png");
    eArray[3] = new GreenfootImage("godinez2izq.png");
    eArray[4] = new GreenfootImage("godinez3.png");
    eArray[5] = new GreenfootImage("godinez3izq.png");
    eArray[6] = new GreenfootImage("godinez4.png");
    eArray[7] = new GreenfootImage("godinez4izq.png");
    eArray[8] = new GreenfootImage("godinez5.png");
    eArray[9] = new GreenfootImage("godinez5izq.png");
}

public void act()
{
    condicionMovimiento();
}
}

```

- Clase Kiko:

```

public class Kiko extends Enemigos
{
    public Kiko(int iVel)
    {
        eArray[0] = new GreenfootImage("kiko1.png");
        eArray[1] = new GreenfootImage("kiko1izq.png");
        eArray[2] = new GreenfootImage("kiko2.png");
        eArray[3] = new GreenfootImage("kiko2izq.png");
        eArray[4] = new GreenfootImage("kiko3.png");
        eArray[5] = new GreenfootImage("kiko3izq.png");
        eArray[6] = new GreenfootImage("kiko4.png");
        eArray[7] = new GreenfootImage("kiko4izq.png");
        eArray[8] = new GreenfootImage("kiko5.png");
        eArray[9] = new GreenfootImage("kiko5izq.png");
    }

    public void act()
    {
        condicionMovimiento();
    }
}

```

- Clase Jaimito:

```

public class Jaimito extends Enemigos
{
    public Jaimito(int iVel)
    {
        eArray[0] = new GreenfootImage("jaimito1.png");
        eArray[1] = new GreenfootImage("jaimito1izq.png");
        eArray[2] = new GreenfootImage("jaimito2.png");
        eArray[3] = new GreenfootImage("jaimito2izq.png");
    }
}

```

```

        eArray[4] = new GreenfootImage("jaimito3.png");
        eArray[5] = new GreenfootImage("jaimito3izq.png");
        eArray[6] = new GreenfootImage("jaimito4.png");
        eArray[7] = new GreenfootImage("jaimito4izq.png");
        eArray[8] = new GreenfootImage("jaimito5.png");
        eArray[9] = new GreenfootImage("jaimito5izq.png");
    }

    public void act()
    {
        condicionMovimiento();
    }
}

```

- Clase srBarriga:

```

public class srBarriga extends Enemigos
{
    public srBarriga()
    {
        eArray[0] = new GreenfootImage("srbarriga1.png");
        eArray[1] = new GreenfootImage("srbarriga1izq.png");
        eArray[2] = new GreenfootImage("srbarriga2.png");
        eArray[3] = new GreenfootImage("srbarriga2izq.png");
        eArray[4] = new GreenfootImage("srbarriga3.png");
        eArray[5] = new GreenfootImage("srbarriga3izq.png");
        eArray[6] = new GreenfootImage("srbarriga4.png");
        eArray[7] = new GreenfootImage("srbarriga4izq.png");
        eArray[8] = new GreenfootImage("srbarriga5.png");
        eArray[9] = new GreenfootImage("srbarriga5izq.png");
    }
    public void act()
    {
        condicionMovimiento();
    }
}

```

- Clase Ramon:

```

public class Ramon extends Enemigos
{
    public Ramon(int iVel)
    {
        eArray[0] = new GreenfootImage("ramon1.png");
        eArray[1] = new GreenfootImage("ramon1izq.png");
        eArray[2] = new GreenfootImage("ramon2.png");
        eArray[3] = new GreenfootImage("ramon2izq.png");
        eArray[4] = new GreenfootImage("ramon3.png");
    }
}

```

```

        eArray[5] = new GreenfootImage("ramon3izq.png");
        eArray[6] = new GreenfootImage("ramon4.png");
        eArray[7] = new GreenfootImage("ramon4izq.png");
        eArray[8] = new GreenfootImage("ramon5.png");
        eArray[9] = new GreenfootImage("ramon5izq.png");
    }

    public void act()
    {
        condicionMovimiento();
    }
}

```

- Clase Background:

```

public class Background extends Actor
{
    private GreenfootImage menu = new GreenfootImage("Fondo.jpg");
    private GreenfootImage tutorial1;
    private GreenfootImage tutorial2;
    private GreenfootImage tutorial3;
    private int iNum;
    private int x1;
    private int y1;
    private int vel;
    private int tutorial;
    private int tBack;
    private int iVel;
    private int iNivel = 1;
    public Background(int iPos, int x, int y, int iFondo){
        iNivel = 1;
        tBack = iFondo;
        if(tBack == 0){
            setImage(menu);
        }
        if(tBack == 1){
            iNum = iPos;
            x1 = x;
            y1 = y;
            iVel = 1;
        }
        if(tBack == 2){
            cargalmagenesTutorial();
        }
    }

    public void act()

```

```

{
    if(tBack == 0){
        setLocation(this.getX()-iVel,this.getY());
        if(getX() == -400){
            this.returnPos();
        }
    }
    if(tBack == 2){
        Tutorial();
    }
}

public void returnPos(){
    setLocation(950,200);
}

public void cargalmagenesTutorial(){
    tutorial1 = new GreenfootImage("Tutorial1.png");
    tutorial2 = new GreenfootImage("Tutorial2.jpg");
    tutorial3 = new GreenfootImage("Tutorial3.jpg");
}

public void sigTutorial(){
    if(tutorial != 400){
        tutorial++;
    }
    if(tutorial == 400){
        World aux = getWorld();
        ((Menu)aux).quitaTutorial();
    }
}

public void Tutorial(){
    if(tutorial == 0){
        setImage(tutorial1);
    }
    if(tutorial == 200){
        setImage(tutorial2);
    }
    if(tutorial == 400){
        setImage(tutorial3);
    }
    sigTutorial();
}
}

```

- Clase Chavito:

```
public class Chavito extends Actor
```

```

{
    private Thread t;
    private GreenfootImage[] chavo = new GreenfootImage[14]; //El arreglo de tipo GreenfootImage
    guarda todas las imagenes que pertenecen al movimiento del actor
    private boolean isAbove;
    private boolean inStair;
    private boolean gravity;
    private boolean isTouch;
    private boolean bAux;
    private int iNivelAux;
    private int iNivel;
    private int upStair;
    private int walkTimer;
    private int iBandera;
    private int iBandera2;
    private int iCount;
    private int iDireccion;
    private int iLife;
    private int iAnimacion;
    private int iEdo;
    private int iAuxiliar;
    private int iContador;
    private LifeSpawn lAux;
    //private GamePad xbox360;
    /**
     * public Chavito() se encarga de inicialisar todas las variables que actuaran con el personaje
    principal del juego
     */
    public Chavito()
    {
        t = new Thread();
        chavo[0] = new GreenfootImage("chavo1.png"); //camina1
        chavo[1] = new GreenfootImage("chavo1_izq.png"); //camina1 izq
        chavo[2] = new GreenfootImage("chavo2.png"); //camina2
        chavo[3] = new GreenfootImage("chavo2_izq.png"); //camina2 izq
        chavo[4] = new GreenfootImage("chavo3.png"); //agachado
        chavo[5] = new GreenfootImage("chavo3_izq.png"); //agachado izq
        chavo[6] = new GreenfootImage("chavo4.png"); //subir escalera
        chavo[7] = new GreenfootImage("chavo5.png"); //brinco
        chavo[8] = new GreenfootImage("chavo5_izq.png"); //brinco izq
        chavo[9] = new GreenfootImage("chavo6.png"); //medio muerto
        chavo[10] = new GreenfootImage("chavo6_izq.png"); //medio muerto izq
        chavo[11] = new GreenfootImage("chavo7.png"); //muerto
        chavo[12] = new GreenfootImage("chavo7_izq.png"); //muerto izq
        chavo[13] = new GreenfootImage("chavo4_izq.png"); //subir
        iLife = 0;
        upStair = -1;
        walkTimer = 0;
        iBandera = 0;
        iBandera2 = 0;
    }
}

```

```

iCount = 0;
iDireccion = 0;
iAnimacion = 0;
iEdo = 0;
iNivel = 1;
iAuxiliar = 0;
isAbove = false;
gravity = false;
inStair = false;
isTouch = false;
bAux = false;
}

/**
 * Esta clase se encarga de todo lo que puede hacer el chavo siempre y cuando este tenga aun
vida
 */
public void act()
{
    if(iLife != 5){
        setActions(); //Ejecuta todas las acciones que puede realizar el actor
    }
    if((iLife == 5)||iLife>5){
        setDead(); //Corresponde a la animacion de la muerte del actor =(
    }
}

/**
 * Se encarga de checar que tecla esta presionada para asi determinar que hacer
 */
public void setActions()
{
    if(!isTouching(Paty.class)){
        setJump(); //metodo que ejecuta el salto
        setCrouch(); //metodo que ejecuta para agacharse
        setMove(); //metodo que ejecuta el movimiento
        setUpStairs(); //metodo que ejecuta subir escaleras
        setDownStairs(); //metodo que ejecuta el bajar escaleras
        setGravity(); //metodo que ejecuta la gravedad cuando cae de una plataforma
        setTouch(); //Si es tocado por un obstaculo pierde vida y si es tocado por una torta gana
vida
    }
    if(isTouching(Paty.class)){
        setWin(); //Si el chavo toca a la popis el jugador pasara al sig nivel
    }
}

/**
 * Se encarga del salto del personaje
 */

```

```

public void setJump()
{
    if (Greenfoot.isKeyDown("space")){
        iBandera = 1;
    }
    if (iBandera == 1 || iBandera2 == 1){
        if(iBandera==1){
            if(iDireccion == 0){
                setImage(chavo[7]);
            }
            if(iDireccion == 1){
                setImage(chavo[8]);
            }
            setLocation(getX(),getY()-1);
            iCount++;
            if(iCount == 65 || isTouching(Plataformita.class)==true){
                iBandera=0;
                iBandera2=1;
            }
        }
    }

    if(iBandera2==1){
        if(iDireccion == 0){
            setImage(chavo[7]);
        }
        if(iDireccion == 1){
            setImage(chavo[8]);
        }
        setLocation(getX(),getY()+1);
        iCount--;
        if(iCount == 0 || isTouching(Plataformita.class)){
            iBandera2=0;
            if(iDireccion == 0){
                setImage(chavo[0]);
            }
            if(iDireccion == 1){
                setImage(chavo[1]);
            }
        }
    }
}

/**
 * Animacion del chavo agachandose.
 */
public void setCrouch()
{
    if (Greenfoot.isKeyDown("down") && !isTouching(Escalera.class)){
        if(iDireccion == 0){

```

```

        setImage(chavo[4]);
    }
    if(iDireccion == 1){
        setImage(chavo[5]);
    }
}

/**
 * Todo el movimiento del actor
 */
public void setMove()
{
    if(getX()<970 && inStair == false){
        if (Greenfoot.isKeyDown("right")){
            iDireccion = 0;
            move(walkTimer);
            walkTimer ++;
            if(gravity == false){
                if(walkTimer == 0){
                    setImage(chavo[0]);
                }
                if(walkTimer == 3){
                    setImage(chavo[2]);
                }
                if(walkTimer == 6){
                    walkTimer = 0;
                    setImage(chavo[0]);
                }
            }
        }
    }

    if(getX())>50 && inStair == false){
        if (Greenfoot.isKeyDown("left")){
            iDireccion = 1;
            move(walkTimer);
            walkTimer --;
            if(walkTimer == 0){
                setImage(chavo[1]);
            }
            if(walkTimer == -3){
                setImage(chavo[3]);
            }
            if(walkTimer == -6)
            {
                walkTimer = 0;
                setImage(chavo[1]);
            }
        }
    }
}

```



```

    }
}

/**
 * Animacion del chavo subiendo escaleras.
 */
public void setUpStairs()
{
    if( getY() < 400){
        isAbove = true;
    }
    if(      isTouching(Escalera.class)      &&      (isTouching(Plataforma1.class)      ||
isTouching(Plataforma2.class))) {
        if(Greenfoot.isKeyDown("up")){
            upStair++;
            setImage(chavo[6]);
            setLocation(getX(),getY()-1);
            if(upStair == 0){
                setImage(chavo[6]);
            }
            if(upStair == 3 ){
                setImage(chavo[13]);
            }
            if(upStair == 6){
                upStair = -1;
            }
        }
    }
}

/**
 * Se encarga de que el personaje puede bajar las escaleras siempre que este arriva de una
plataforma
 */
public void setDownStairs()
{
    if(isTouching(Escalera.class) && Greenfoot.isKeyDown("down") && getY()<475 && isAbove
== true){
        upStair++;
        setImage(chavo[6]);
        setLocation(getX(),getY()+1);
        if(iDireccion == 0){
            if(upStair == 0){
                setImage(chavo[0]);
            }
            if(upStair == 10){
                setImage(chavo[13]);
            }
        }
        if(upStair == 20){
            upStair = 0;
        }
    }
}

```

```

        setImage(chavo[0]);
    }
}
if(iDireccion == 1){
    if(upStair == 0){
        setImage(chavo[1]);
    }
    if(upStair == 10){
        setImage(chavo[13]);
    }
    if(upStair == 20){
        upStair = 0;
        setImage(chavo[1]);
    }
}
if(getY() == 475 ){//no esta tocando plataforma
    isAbove = false;
    inStair = false;
}
}
}

/**
 * Cuando el personaje se encuentra fuera de las plataformas automaticamente cae hasta el piso
 */
public void setGravity()
{
    if((isAbove == true) && !isTouching(Plataforma1.class)){
        while(getY() < 475){
            setLocation(getX(),getY()+1);
            if(iDireccion == 0){
                setImage(chavo[0]);
            }
            if(iDireccion == 1){
                setImage(chavo[1]);
            }
            if(getY() == 475){
                if(iDireccion == 0){
                    setImage(chavo[4]);
                }
                if(iDireccion == 1){
                    setImage(chavo[5]);
                }
                isAbove = false;
            }
        }
    }
}
}

/**

```

```

* Cuando el jugador toca a la popis se cambia de nivel
*/
public void setWin()
{
    if(isTouching(Paty.class)){
        if(iNivel != 5){
            Letreros lAuxiliar1 = new Letreros( 1 );
            Letreros lAuxiliar2 = new Letreros( 2 );
            for(iContador=0;iContador<4500;iContador++){
                if(iContador == 1){
                    getWorld().addObject(lAuxiliar1, 500, 100);
                    getWorld().addObject(lAuxiliar2, 500, 400);
                }
                if(iContador == 4498){
                    getWorld().removeObject(lAuxiliar1);
                    getWorld().removeObject(lAuxiliar2);
                }
            }
            World wAux1 = getWorld();
            iLife = 0;
            ((Vecindad)wAux1).eliminaVecindad(iNivel);
        }
        if(iNivel == 5){
            Letreros lAuxiliar1 = new Letreros( 1 );
            getWorld().addObject(lAuxiliar1, 500, 100);
        }
    }
}

/**
 * Si es tocado por un enemigo pierde vida.
 * Si es tocado por una torta aumenta su vida.
 */
public void setTouch()
{
    if(isTouching(Torta.class)){
        if(iLife!=0){
            iLife--;
            LifeSpawn lAux = new LifeSpawn(iLife);
            getWorld().addObject(lAux,80,50);
            removeTouching(Torta.class);
        }
        if(iLife==0){
            iLife = -3;
            removeTouching(Torta.class);
        }
    }
    if((isTouching(Obstaculos.class) == true)){
        iLife++;
        LifeSpawn lAux = new LifeSpawn(iLife);
    }
}

```

```

        removeTouching(Obstaculos.class);
        getWorld().addObject(lAux,80,50);
    }
    if((isTouching(Enemigos.class) == true) && (isTouch == false)){
        iLife = iLife + 3;
        LifeSpawn lAux = new LifeSpawn(iLife);
        getWorld().addObject(lAux,80,50);
        isTouch = true;
        Sangre blood = new Sangre();
        int iX = getX();
        int iY= getY();
        getWorld().addObject(blood,iX,iY);
        for(int i=0;i<300;i++);
        getWorld().removeObject(blood);
    }
    if((isTouching(Enemigos.class) == false )){
        isTouch = false;
    }
}

/**
 * Cuando el chavo muere muestra una animacion en pantalla
 */
public void setDead()
{
    if(iAnimacion == 0){
        if(iDireccion == 0){
            setImage(chavo[9]);
        }
        if(iDireccion == 1){
            setImage(chavo[11]);
        }
    }
    if(iAnimacion == 10){
        if(iDireccion == 0){
            setImage(chavo[10]);
        }
        if(iDireccion == 1){
            setImage(chavo[12]);
        }
    }
    if(iAnimacion == 20){
        Letreros lAuxiliar = new Letreros(0);
        getWorld().addObject(lAuxiliar, 500, 300);
    }
    iAnimacion++;
}
}

```

- Clase Sangre:

```

public class Sangre extends Chavito
{
    public GreenfootImage sangre;
    public Sangre()
    {
        sangre = new GreenfootImage("sangre.png");
        setImage(sangre);
    }
    public void act()
    {
        // Add your action code here.
    }
}

```

- Clase LifeSpawn:

```

public class LifeSpawn extends Chavito
{
    private GreenfootImage[] life = new GreenfootImage[6];
    private int iVida;
    public LifeSpawn(int iTi)
    {
        if((iTi>=0) && (iTi<=5)){
            life[0] = new GreenfootImage("LifeSpawn1.png");
            life[1] = new GreenfootImage("LifeSpawn2.png");
            life[2] = new GreenfootImage("LifeSpawn3.png");
            life[3] = new GreenfootImage("LifeSpawn4.png");
            life[4] = new GreenfootImage("LifeSpawn5.png");
            life[5] = new GreenfootImage("LifeSpawn6.png");
            setImage( life[iTi] );
        }
    }

    public void act()
    {

    }
}

```

- Clase Letreros:

```

public class Letreros extends Chavito
{
    private GreenfootImage[] letras = new GreenfootImage[3]; //El arreglo de tipo GreenfootImage
    guarda todas las imagenes que pertenecen a los letreros de peder, ganar y sig nivel
    private int iEstado;
    /**
     * Act - do whatever the Letreros wants to do. This method is called whenever
     * the 'Act' or 'Run' button gets pressed in the environment.

```

```

*/
public Letreros(int iEdos)
{
    letras[0] = new GreenfootImage("perdiste.png");
    letras[1] = new GreenfootImage("ganaste.png");
    letras[2] = new GreenfootImage("siguiente.png");
    iEstado = iEdos;
    setImage( letras[iEdos] );
}
public void act()
{

}
}

```

- Clase Paty:

```

public class Paty extends Actor
{
    private GreenfootImage[] paty = new GreenfootImage[3];
    private int iAccion;
    public Paty()
    {
        paty[0] = new GreenfootImage("paty1.png");
        paty[1] = new GreenfootImage("paty2.png");
        paty[2] = new GreenfootImage("paty3.png");
        iAccion = 0;
    }

    public void act()
    {
        animacion();
    }

    public void animacion()
    {
        if(iAccion == 0){
            setImage(paty[0]);
        }
        if(iAccion == 20){
            setImage(paty[1]);
        }
        if(iAccion == 40){
            setImage(paty[2]);
        }
        if(iAccion == 60){
            iAccion = 0;
        }
        iAccion++;
    }
}

```

```
}

```

- Clase Chilindrina:

```
public class Chilindrina extends Actor
{
    private GreenfootImage[] chilindrina = new GreenfootImage[10]; //El arreglo de tipo
    GreenfootImage guarda todas las imagenes que pertenecen al movimiento de el enemigo
    private int iDr;
    private int iAnimacion;
    private int iAux;
    private Random iRandom;
    /**
     * Este constructor carga las imagenes de la animacion de la chilindrina
     */
    public Chilindrina()
    {
        chilindrina[0] = new GreenfootImage("chilin1.png");
        chilindrina[1] = new GreenfootImage("chilin1_izq.png");
        chilindrina[2] = new GreenfootImage("chilin2.png");
        chilindrina[3] = new GreenfootImage("chilin2_izq.png");
        chilindrina[4] = new GreenfootImage("chilin3.png");
        chilindrina[5] = new GreenfootImage("chilin3_izq.png");
        chilindrina[6] = new GreenfootImage("chilin4.png");
        chilindrina[7] = new GreenfootImage("chilin4_izq.png");
        chilindrina[8] = new GreenfootImage("chilin5.png");
        chilindrina[9] = new GreenfootImage("chilin5_izq.png");
        iDr = 1;
        iAnimacion = 0;
        iRandom = new Random();
        iAux = 0;
    }

    /**
     * Llama al metodo animate();
     */
    public void act()

```

```

{
    animate();
}

/**
 * Este metodo se encarga de todo lo que puede hacer la chilindrina
 */
public void animate()
{
    //Si direccion es igual a 1, se mostrara las imagenes del lado derecho, si es izquierda se
    ostraran del lado derecho
    if(iDr == 1){
        if(iAnimacion == 10){
            setLma[1]);
        }
        if(iAnimacion == 25){
            setLma[3]);
        }
        if(iAnimacion == 40){
            setLma[5]);
        }
        if(iAnimacion == 55){
            setLma[7]);
        }
        if(iAnimacion == 70){
            setLma[9]);
            iAux = iRandom.nextInt(6);
            if(iAux != 5){
                getWorld().addObject(new Obstaculos(iRandom.nextInt(4)),600,50);
            }
            if(iAux == 5){
                getWorld().addObject(new Torta(),600,50);
            }
        }
        if(iAnimacion == 85){
            iAnimacion = 0;
        }
        iAnimacion++;
    }
    if(iDr == 2){
        if(iAnimacion == 10){
            setLma[0]);
        }
        if(iAnimacion == 25){
            setLma[2]);
        }
        if(iAnimacion == 40){
            setLma[4]);
        }
        if(iAnimacion == 55){

```



```

        setImage(chilindrina[6]);
    }
    if(iAnimacion == 70){
        setImage(chilindrina[8]);
        iAux = iRandom.nextInt(5);
        if(iAux != 5){
            getWorld().addObject(new Obstaculos(iRandom.nextInt(4)),600,50);
        }
        if(iAux == 5){
            getWorld().addObject(new Torta(),600,50);
        }
    }
    if(iAnimacion == 85){
        iAnimacion = 0;
    }
    iAnimacion++;
}
}
}

```

- Clase Obstaculos:

```

public class Obstaculos extends Chilindrina
{
    private GreenfootImage[] obstaculos = new GreenfootImage[4];
    private int iDireccion;
    private int iTipo;
    private int iX;
    private int iY;
    private Random iRandom;
    private int iNum;
    private Boolean onFloor;
    public Obstaculos(int iTi)
    {
        obstaculos[0] = new GreenfootImage("manzana.png");
        obstaculos[1] = new GreenfootImage("triciclo.png");
        obstaculos[2] = new GreenfootImage("pelota.png");
        obstaculos[3] = new GreenfootImage("platano.png");
        iDireccion = -1;
        iTipo = iTi;
        setImage( obstaculos[iTi] );
        iRandom = new Random();
        iNum = iRandom.nextInt(2);
        onFloor = false;
    }

    public void act()
    {
        setFall();
    }
}

```

```

public void setFall()
{
    /**
     * Mientras onFloor sea falso significa que el obstaculo se esta moviendo en las plataformas
     * Si onFloor es verdadero el obstaculo se esta moviendo en el piso
     */
    if(onFloor == false){
        if(isTouching(Escalera.class) || (!isTouching(Plataforma.class))){
            setLocation(getX(),getY()+2);
        }
        if(!isTouching(Escalera.class) || (isTouching(Plataforma.class))){
            if(iDireccion == 1){
                move(2);
            }
            if(iDireccion == -1){
                move(-2);
            }
        }
        if((isTouching(Escalera.class)) && (!isTouching(Plataforma.class))){
            iDireccion = ((iDireccion) * (-1));
        }
        if( (getY()==500)){
            onFloor = true;
        }
    }
    if(onFloor == true){
        if(iDireccion == 1){
            move(2);
        }
        if(iDireccion == -1){
            move(-2);
        }
        if( (getX()<=20) || (getX())>1000) ){
            getWorld().removeObject(this);
        }
    }
}
}

```

- Clase Torta:

```

public class Torta extends Chilindrina
{
    private GreenfootImage tortaDeJamon;
    private int iDireccion;
    private int iX;
    private int iY;
    private Random iRandom;
    private int iNum;

```

```

private Boolean onFloor;
public Torta(){
    tortaDeJamon = new GreenfootImage("torta.png");
    iDireccion = -1;
    setImage(tortaDeJamon);
    iRandom = new Random();
    iNum = iRandom.nextInt(2);
    onFloor = false;
}

public void act() {
    setWhatCanItDo();
}

public void setWhatCanItDo(){

    if(onFloor == false){
        if(isTouching(Escalera.class) || (!isTouching(Plataforma.class))){
            setLocation(getX(),getY()+2);
        }
        if(!isTouching(Escalera.class) || (isTouching(Plataforma.class))){
            if(iDireccion == 1){
                move(2);
            }
            if(iDireccion == -1){
                move(-2);
            }
        }
        if((isTouching(Escalera.class)) && (!isTouching(Plataforma.class))){
            iDireccion = ((iDireccion) * (-1));
        }
        if( (getY()==500)){
            onFloor = true;
        }
    }
    if(onFloor == true){
        if(iDireccion == 1){
            move(2);
        }
        if(iDireccion == -1){
            move(-2);
        }
        if( (getX()<=20) || (getX())>1000) ){
            getWorld().removeObject(this);
        }
    }
}
}

```

}