

Manual Del Programador

Intersemestral programación orientada a objetos



1 de julio de 2015

universidad autonoma de san luis potosi

[Dirección de la compañía]

# DATOS DEL PROYECTO

* Nombre: Monster-Master.
* No. De Proyecto: 236.

# OBJETIVO DEL PROYECTO

Matar a los monstruos por medio de poderes y luchas.

# DESCRIPCION DEL PROYECTO

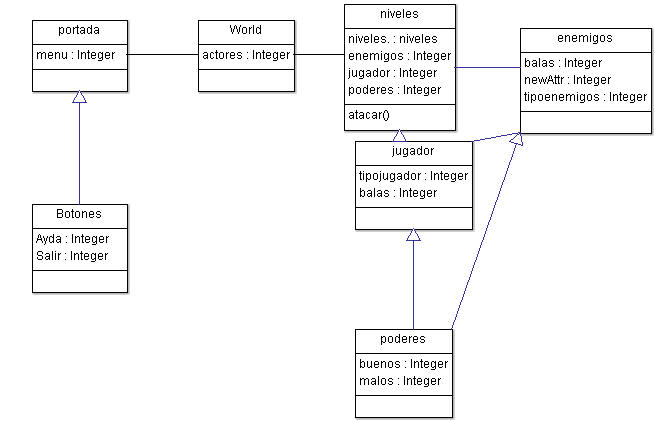
El Juego consiste en duelos de monstruos entre 2 jugadores, cada uno podrá tener un máximo de 3 monstruos aleatorios por juego, los monstruos los podrás escoger y sólo con el monstruo elegido podrá jugar, se cuenta con un total de 5 paquetes con las cuales podrás aumentar o disminuir el poder de tu monstruo, el monstruo podrá moverse hacia el enemigo para atacar o defenderse, aleatoriamente aparecerá paquetes que al tocarlos se agregará poder por un determinado tiempo, pero si los enemigos también lo tocan les pasará lo mismo.

Después de que mates a un enemigo aparecerá otro con más poderes y así hasta llegar a los 5 monstruos.

Aparte de luchar con el enemigo tendrá que esquivar trampas que caigan del cielo, que parezcan del suelo o sean aventados por el mismo escenario.

El juego termina cuando la barra de vida está llena en cualquiera de los 2 casos.

# DIAGRAMA DE CLASES UML



# CARACTERISTICAS Y COMPORTAMIENTO DE CADA CLASE

|  |  |
| --- | --- |
| Nombre De La Clase: | Niveles |
| Características: | Añade todos los objetos al mundo dependiendo del nivel en que estemos jugando |
| Nombre De La Clase: | Menú |
| Características: | Son los botones que te llevan dependiendo del que aplanes. |
| Nombre De La Clase: | Ayuda |
| Características: | Da las instrucciones para el juego. |
| Nombre De La Clase: | Portada |
| Características: | Crédito de los autores. |
| Nombre De La Clase: | Balas |
| Características: | Añade al mundo una escalera por la cual subirá el personaje para cambiar de plataforma. |
| Nombre De La Clase: | vidaJugador |
| Características: | Te da la cantidad de vida o tiempo del jugador |
| Nombre De La Clase: | counter |
| Características: | Cuenta la vida de los enemigos |
| Nombre De La Clase: | Enemigos |
| Características: | Carga las imágenes de los enemigos en pantalla y los dirige para que al encontrarse con el jugador resten vida a este al dispararle. |
| Nombre De La Clase: | Botones |
| Características: | Apoyan al menú, cada botón tendrá su propio comando. |
| Nombre De La Clase: | Poderes |
| 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: | Jugadores |
| Características: | Son los que tu como usuario podrás mover y tendrás que salvar. |
| Nombre De La Clase: | Textos |
| Características: | Es cada uno de las presentaciones. |

# HERENCIA Y POLIMORFISMO

La herencia se usa principalmente en las clases jugador, poderes 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 poderes hereda su método poderes a sus subclases que aunque claro cambian un poco en la manera de actuar.

# CRONOGRAMA DE ACTIVIDADES

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

# 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:

**Portada:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

public class Portada extends World

{

public int band;

/\*\*

\* Constructor for objects of class Portada.

\*

\*/

public Portada()

{

// Create a new world with 600x400 cells with a cell size of 1x1 pixels.

super(800, 600, 1);

GreenfootSound sound = new GreenfootSound("WeWerentAngels.mp3");

sound.play();

addObject(new Presentacion(sound), 400, 300);

//Imprimir la portada de nombres y esperar al usuario

}

public void detenM(GreenfootSound m)

{

m.stop();

}

}

**Clase padre - Niveles:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

import java.util.\*;

public class Niveles extends World

{

/\*\*

\* Constructor for objects of class Niveles.

\*

\*/

public Niveles()

{

// Create a new world with 600x400 cells with a cell size of 1x1 pixels.

super(800, 600, 1);

}

public void eliminaActores()

{

List<Botones> a = getObjects(Actor.class);

removeObjects(a);

}

}

Sub clase nivel1

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class TheBackGround here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class hNivel1 extends Niveles

{

private int mundo = 0;

private int ex;

Enemigos enems[]={null,null,null,null,null};

/\*\*

\* Constructor para el primer Nivel.

\*

\*/

public hNivel1(int op)

{

mundo = op;

pinta\_mundo();

}

private void pinta\_mundo()

{

if(mundo == 1)

{

addObject(new jugLight(), 400, 550);

for(int i = 0; i < 5 ; i++)

{

ex = Greenfoot.getRandomNumber(750);

if(ex==0)

{

ex+=20;

}

enems[i] = new enemLight();

addObject(enems[i],ex, 550);

}

}

if(mundo == 2)

{

addObject(new dJugLight(), 400, 550);

for(int i = 0; i < 5 ; i++)

{

ex = Greenfoot.getRandomNumber(750);

if(ex==0)

{

ex+=20;

}

enems[i] = new hEnemLight();

addObject(enems[i],ex, 550);

}

}

}

}

**Sub clase nivel2**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Nivel2 here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class hNivel2 extends Niveles

{

private int mundo = 0;

private int ex;

Enemigos enems[]={null,null,null,null,null};

/\*\*

\* Constructor for objects of class Nivel2.

\*

\*/

public hNivel2(int op)

{

// Create a new world with 600x400 cells with a cell size of 1x1 pixels.

mundo = op;

pinta\_mundo();

}

private void pinta\_mundo()

{

int ex;

if(mundo == 1)

{

addObject(new jugador(), 400, 550);

enems[0] = new enemigo();

enems[1] = new enemLight();

enems[2] = new enemLight();

enems[3] = new enemLight();

enems[4] = new enemigo();

for(int i = 0; i < 5 ; i++)

{

ex = Greenfoot.getRandomNumber(750);

if(ex==0)

{

ex+=20;

}

addObject(enems[i],ex, 550);

}

}

if(mundo == 2)

{

addObject(new dJug(), 400, 550);

enems[0] = new hEnem();

enems[1] = new hEnemLight();

enems[2] = new hEnemLight();

enems[3] = new hEnemLight();

enems[4] = new hEnem();

for(int i = 0; i < 5 ; i++)

{

ex = Greenfoot.getRandomNumber(750);

if(ex==0)

{

ex+=20;

}

addObject(enems[i],ex, 550);

}

}

}

}

**Sub clase nivel3**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Nivel3 here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class hNivel3 extends Niveles

{

private int mundo = 0;

private int ex;

Enemigos enems[]={null,null,null,null,null};

/\*\*

\* Constructor for objects of class Nivel3.

\*

\*/

public hNivel3(int op)

{

// Create a new world with 600x400 cells with a cell size of 1x1 pixels.

mundo = op;

pinta\_mundo();

}

private void pinta\_mundo()

{

if(mundo == 1)

{

addObject(new jugador(), 400, 550);

enems[0] = new enemPesado();

enems[1] = new enemPesado();

enems[2] = new enemPesado();

enems[3] = new enemPesado();

enems[4] = new enemPesado();

for(int i = 0; i < 5 ; i++)

{

ex = Greenfoot.getRandomNumber(750);

if(ex==0)

{

ex+=20;

}

addObject(enems[i],ex, 550);

}

}

if(mundo == 2)

{

addObject(new dJugPe(), 400, 550);

enems[0] = new hEnem();

enems[1] = new hEnem();

enems[2] = new hEnem();

enems[3] = new hEnem();

enems[4] = new hEnem();

for(int i = 0; i < 5 ; i++)

{

ex = Greenfoot.getRandomNumber(750);

if(ex==0)

{

ex+=20;

}

addObject(enems[i],ex, 550);

}

}

}

}

**Clase Ayuda:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Ayuda here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Ayuda extends World

{

/\*\*

\* Constructor for objects of class Ayuda.

\*

\*/

public Ayuda()

{

// Create a new world with 600x400 cells with a cell size of 1x1 pixels.

super(800, 600, 1);

addObject(new Texto\_Ayuda(), 400, 300);

}

}

**Clase Select Carácter:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

import java.util.\*;

/\*\*

\* Write a description of class Select\_character here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Select\_character extends World

{

private int raza = 0;

/\*\*

\* Constructor for objects of class Select\_character.

\*

\*/

public Select\_character(int c)

{

// Create a new world with 600x400 cells with a cell size of 1x1 pixels.

super(800, 600, 1);

raza = c;

select();

}

public void select()

{

int op=0;

op = raza;

addObject(new Pesado(op), 150, 350);

addObject(new Medio(op), 400, 350);

addObject(new Ligero(op), 650, 350);

}

public void eliminaActores()

{

List<Botones> a = getObjects(Botones.class);

removeObjects(a);

}

}

**Clase Menu:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

import java.util.\*;

/\*\*

\* Write a description of class Menu here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Menu extends World

{

/\*\*

\* Constructor for objects of class Menu.

\*

\*/

public Menu()

{

// Create a new world with 600x400 cells with a cell size of 1x1 pixels.

super(800, 600, 1);

Greenfoot.playSound("SC2\_uiBNetToast.mp3");

addObject(new Boton\_Jugar(), 400, 190);

addObject(new Boton\_Ayuda(), 400, 290);

addObject(new Boton\_Salir(), 400, 490);

addObject(new Titulo(), 400, 90);

//Greenfoot.playSound("WeWerentAngels.mp3");

}

public void eliminaActores()

{

List<Botones> a = getObjects(Botones.class);

removeObjects(a);

}

}

**Clase select class**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

import java.util.\*;

/\*\*

\* Write a description of class Select\_class here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Select\_class extends World

{

/\*\*

\* Constructor for objects of class Select\_class.

\*

\*/

public Select\_class()

{

// Create a new world with 600x400 cells with a cell size of 1x1 pixels.

super(800, 600, 1);

addObject(new Humanoides(), 200, 350);

addObject(new Demonios(), 600, 350);

addObject(new Texto\_select1(), 400, 100);

}

public void eliminaActores()

{

List<Botones> a = getObjects(Botones.class);

removeObjects(a);

}

}

**Clase GameOverText**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class gameOverText here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class gameOverText extends World

{

private int puntos = 0;

/\*\*

\* Constructor for objects of class gameOverText.

\*

\*/

public gameOverText()

{

// Create a new world with 600x400 cells with a cell size of 1x1 pixels.

super(800, 600, 1);

showText("Tu puntuación fue de: " + puntos, 400, 300);

showText("Para salir apriete el siguiente boton",400,400);

addObject(new botonSalirMenu(), 400, 490);

}

}

**Clase game over**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class gameOver here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class gameOver extends World

{

/\*\*

\* Constructor for objects of class gameOver.

\*

\*/

public gameOver()

{

// Create a new world with 600x400 cells with a cell size of 1x1 pixels.

super(800, 600, 1);

showText("Usted ha perdido, su record no será guardado", 400, 100);

showText("Para salir apriete el siguiente boton",400,400);

addObject(new botonSalirMenu(), 400, 490);

}

}

**Clase text- select1**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Texto\_select1 here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Texto\_select1 extends Actor

{

/\*\*

\* Act - do whatever the Texto\_select1 wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

// Add your action code here.

}

}

**Clase text- select12**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Texto\_select2 here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Texto\_select2 extends Actor

{

/\*\*

\* Act - do whatever the Texto\_select2 wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

// Add your action code here.

}

}

**Clase Balas:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Balas here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Balas extends Actor

{

/\*\*

\* Act - do whatever the Balas wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

//muevete();// Add your action code here.

}

protected void muevete(int dir)

{

int band\_dir = dir;

int band = Greenfoot.getRandomNumber(1);

int x = getX(), y = getY();

checaColision();

if(band\_dir == 1)

{

switch(band)

{

case 0:

setLocation(x+20,y);

break;

}

}

if(band\_dir == 2)

{

switch(band)

{

case 0:

setLocation(x-20,y);

break;

}

}

}

private void checaColision()

{

Actor a = this.getOneIntersectingObject(bala\_enem.class);

if(a != null )

{

//getWorld().removeObject(this);

this.getWorld().removeObject(a);

}

if(isAtEdge())

{

getWorld().removeObject(this);

}

}

}

**Clase Bala\_jug**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class bala\_jug here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class bala\_jug extends Balas

{

private int band\_dir = 0;

private int tipo\_bala = 0;

public bala\_jug(int dir,int p)

{

band\_dir = dir;

tipo\_bala=p;

}

/\*\*

\* Act - do whatever the bala\_jug wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

if(tipo\_bala == 1)

{

setImage("Balajug1.png");

}

if(tipo\_bala == 2)

{

setImage("Balajug2.png");

}

muevete(band\_dir);// Add your action code here.

//checaColision();

}

}

**Clase Bala\_enemigo**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class bala\_enem here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class bala\_enem extends Balas

{

private int band\_dir = 0;

public bala\_enem(int d)

{

if(d!=0)

{

band\_dir = d;

}

else

{

band\_dir = 2;

}

}

/\*\*

\* Act - do whatever the bala\_jug wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

muevete(band\_dir);// Add your action code here.

//checaColision();

}

}

**Clase counter:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

import java.awt.Color;

/\*\*

\* A Counter class that allows you to display a numerical value on screen.

\*

\* The Counter is an actor, so you will need to create it, and then add it to

\* the world in Greenfoot. If you keep a reference to the Counter then you

\* can adjust its value. Here's an example of a world class that

\* displays a counter with the number of act cycles that have occurred:

\*

\* <pre>

\* class CountingWorld

\* {

\* private Counter actCounter;

\*

\* public CountingWorld()

\* {

\* super(600, 400, 1);

\* actCounter = new Counter("Act Cycles: ");

\* addObject(actCounter, 100, 100);

\* }

\*

\* public void act()

\* {

\* actCounter.setValue(actCounter.getValue() + 1);

\* }

\* }

\* </pre>

\*

\* @author Neil Brown and Michael Kölling

\* @version 1.0

\*/

public class Counter extends Actor

{

private static final Color transparent = new Color(0,0,0,0);

private GreenfootImage background;

private int value;

private int target;

private String prefix;

public Counter()

{

this(new String());

}

/\*\*

\* Create a new counter, initialised to 0.

\*/

public Counter(String prefix)

{

background = getImage(); // get image from class

value = 0;

target = 0;

this.prefix = prefix;

updateImage();

}

/\*\*

\* Animate the display to count up (or down) to the current target value.

\*/

public void act()

{

if (value < target) {

value++;

updateImage();

}

else if (value > target) {

value--;

updateImage();

}

}

/\*\*

\* Add a new score to the current counter value. This will animate

\* the counter over consecutive frames until it reaches the new value.

\*/

public void add(int score)

{

target += score;

}

/\*\*

\* Return the current counter value.

\*/

public int getValue()

{

return target;

}

/\*\*

\* Set a new counter value. This will not animate the counter.

\*/

public void setValue(int newValue)

{

target = newValue;

value = newValue;

updateImage();

}

/\*\*

\* Sets a text prefix that should be displayed before

\* the counter value (e.g. "Score: ").

\*/

public void setPrefix(String prefix)

{

this.prefix = prefix;

updateImage();

}

/\*\*

\* Update the image on screen to show the current value.

\*/

private void updateImage()

{

GreenfootImage image = new GreenfootImage(background);

GreenfootImage text = new GreenfootImage(prefix + value, 22, Color.BLACK, transparent);

if (text.getWidth() > image.getWidth() - 20)

{

image.scale(text.getWidth() + 20, image.getHeight());

}

image.drawImage(text, (image.getWidth()-text.getWidth())/2,

(image.getHeight()-text.getHeight())/2);

setImage(image);

}

}

**Clase VidaJugador:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class vidaJugador here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class vidaJugador extends Actor

{

private int golpe = 0;

/\*\*

\* Act - do whatever the vidaJugador wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public vidaJugador(int hit)

{

golpe = hit;

//System.out.println("Cont Hit: " + hit);

}

public void act()

{

// Add your action code here.

bajavida(golpe);

}

private void bajavida(int golpe)

{

if(golpe == 0)

{

setImage("VidaJugEnCero.png");

//System.out.println("Primero ");

}

if(golpe == 1)

{

setImage("VidaJug6.png");

//System.out.println("Segundo ");

}

if(golpe == 2)

{

setImage("VidaJug5.png");

//System.out.println("Segundo ");

}

if(golpe == 3)

{

setImage("VidaJug4.png");

//System.out.println("Tercer ");

}

if(golpe == 4)

{

setImage("VidaJug3.png");

//System.out.println("Cuarto ");

}

if(golpe == 5)

{

setImage("VidaJug2.png");

//System.out.println("Quinto ");

}

if(golpe == 6)

{

setImage("VidaJug1.png");

//System.out.println("Sexto ");

}

if(golpe == 7)

{

setImage("VidaJug.png");

//System.out.println("Septimo ");

}

//System.out.println("contador: " + golpe);

}

}

**Clase Enemigos:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Enemigos here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Enemigos extends Actor

{

private int mundo = 0;

private int ex;

private int rand=Greenfoot.getRandomNumber(400);

private int dir = 1;

private int cont = 0;//milisegundos

private int cont2 = 0;//Segundos

private int cont3 = 0;//Segundos

private int cont4 = 0;//Segundos

private int contGolpe = 0;

private int vida = 5;

private int golpe = 0;

private int puntos2 = 0;

private int jump = -20;

private int puntos = 0;

private int verticalSpeed = 5;

Poderes p = new Poderes();

private int poderb = 0;

private int poderm = 0;

/\*\*

\* Act - do whatever the enemigo\_1 wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

// Add your action code here.

mov\_rand();

}

public void mov\_rand()

{

String cad;

int accion=Greenfoot.getRandomNumber(50);

checkFall();

cont ++;

cad = "Contador 1: " + cont + "Contador 2: " + cont2;

getWorld().showText(cad, 500, 50);

if(cont == 100)

{

cont2++;

if(cont2 == 3)

{

pintaPoderes();

cont=0;

cont2=0;

}

if(poderm != 0)

{

cont3++;

if(cont3 == 5)

{

poderm = 0;

cont3 = 0;

}

}

if(poderb != 0)

{

cont4++;

if(cont4 == 10)

{

poderm = 0;

cont4 = 0;

}

}

}

if(poderm == 0)

{

if(accion==1)

{

muevete();

}

if(accion==2)

{

muevete();

}

if(cont==100)

{

dispara();

cont = 0;

}

if(accion==4)

{

if(onGround())

{

verticalSpeed = jump;

fall();

}

}

}

if(isTouching(bala\_jug.class))

{

if(poderb == 0)

{

contGolpe+=2;

}

if(poderb == 1)

{

contGolpe++;

}

if(contGolpe == 20)

{

golpe++;

if(golpe == 7)

{

//getWorld().removeObject();

vida--;

if(vida == 0)

{

Greenfoot.setWorld(new gameOverText());

}

}

contGolpe = 0;

}

removeTouching(bala\_jug.class);

puntos = 0;

}

}

private void muevete()

{

int band\_dir = Greenfoot.getRandomNumber(40);

int mov = 0;

switch(band\_dir)

{

case 1:

//setImage("space\_pirates2.png");

setLocation(getX()+20,getY());

mov = 1;

break;

case 2:

//setImage("space\_pirates1.png");

setLocation(getX()-20,getY());

mov = 2;

break;

}

dir = mov;

}

public void pintaPoderes()

{

int poder = Greenfoot.getRandomNumber(6);

int x = Greenfoot.getRandomNumber(781);

int y = 200;

if(x == 0)

{

x = 20;

}

if(poder == 0)

{

getWorld().addObject(new MinimizarPoder(),x,y);

}

if(poder == 1)

{

getWorld().addObject(new Invulnerable(),x,y);

}

if(poder == 2)

{

getWorld().addObject(new DetenerUsuario(),x,y);

}

if(poder == 3)

{

getWorld().addObject(new DetenerEnemigo(),x,y);

}

if(poder == 4)

{

getWorld().addObject(new DañoUsuario(),x,y);

}

if(poder == 5)

{

getWorld().addObject(new AgrandarVida(),x,y);

}

}

private void dispara()

{

int band = 0;

if (dir!=0)

{

band = dir;

}

getWorld().addObject(new bala\_enem(band),getX(),getY());

}

private void fall()

{

setLocation(getX(),getY() + verticalSpeed);

verticalSpeed = verticalSpeed + 2;

}

private boolean onGround()

{

if(getY()>=550)

{

return true;

}

else

{

return false;

}

}

private void checkFall()

{

if(onGround())

verticalSpeed = 0;

else

fall();

}

public void detenEnem()

{

if(poderm == 0)

{

poderm = 1;

}

}

public void minP()

{

if(poderb == 0)

{

poderb = 1;

}

}

}

**Clase enemigo1**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class enemigo\_1 here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class enemigo extends Enemigos

{

/\*\*

\* Act - do whatever the enemigo\_1 wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

mov\_rand();

if(getX()<400)

{

setImage("space\_pirates2.png");

}

}

}

**Clase enemPesado**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class enemPesado here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class enemPesado extends Enemigos

{

/\*\*

\* Act - do whatever the enemigo\_1 wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

mov\_rand();

if(getX()<400)

{

setImage("HK-1\_der.png");

}

}

}

**Clase hEnemLight**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class hEnemLight here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class hEnemLight extends Enemigos

{

/\*\*

\* Act - do whatever the enemigo\_1 wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

mov\_rand();

if(getX()<400)

{

setImage("MC-1\_der.png");

}

}

}

**Clase hEnem:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class hEnem here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class hEnem extends Enemigos

{

/\*\*

\* Act - do whatever the enemigo\_1 wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

mov\_rand();

if(getX()<400)

{

setImage("Samus\_Aran2.png");

}

}

}

**Clase enemLight**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Presentacion here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Presentacion extends Actor

{

private GreenfootSound s;

/\*\*

\* Act - do whatever the Presentacion wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public Presentacion(GreenfootSound m)

{

s = m;

}

public void act()

{

if(Greenfoot.mouseClicked(this))

{

Portada p = (Portada) getWorld();

Greenfoot.setWorld(new Menu());

p.detenM(s);

p.removeObject(this);

}

}

}

**Clase Botones**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Botones here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Botones extends Actor

{

/\*\*

\* Act - do whatever the Botones wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

// Add your action code here.

}

}

**Clase Humanoides:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Humanos here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Humanoides extends Botones

{

private int op = 0;

/\*\*

\* Act - do whatever the Humanos wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

Select\_class m = (Select\_class) getWorld();

if(Greenfoot.mouseClicked(this))

{

m.eliminaActores();

op = 1;

Greenfoot.setWorld(new Select\_character(op));

Greenfoot.playSound("Protoss\_Probe.wav");

}

}

}

**Clase Ligero:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Ligero here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Ligero extends Botones

{

private int op = 0;

private int nivel = 0;

/\*\*

\* Act - do whatever the Pesado wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public Ligero(int ra)

{

op = ra;

}

public void act()

{

// Add your action code here.

if(op == 1)

{

setImage("ligeroH.png");

nivel = 1;

}

else if(op == 2)

{

setImage("ligeroH\_d.png");

nivel = 2;

}

entraL();

}

private void entraL()

{

int niv = 0;

niv = nivel;

if(Greenfoot.mouseClicked(this))

{

if(nivel == 1)

{

Greenfoot.setWorld(new hNivel1(op));

}

if(nivel == 2)

{

Greenfoot.setWorld(new hNivel1(op));

}

Greenfoot.playSound("Protoss\_Probe.wav");

}

}

}

**Clase Boton-jugar:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Boton\_Jugar here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Boton\_Jugar extends Botones

{

/\*\*

\* Act - do whatever the Boton\_Jugar wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

Menu m = (Menu) getWorld();

if(Greenfoot.mouseClicked(this))

{

m.eliminaActores();

Greenfoot.setWorld(new Select\_class());

Greenfoot.playSound("SC2\_uiBNetToast.mp3");

}

}

}

**Clase Boton-Salir:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Boton\_Salir here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Boton\_Salir extends Botones

{

/\*\*

\* Act - do whatever the Boton\_Salir wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

salida();// Add your action code here.

}

private void salida()

{

Menu m = (Menu) getWorld();

if(Greenfoot.mouseClicked(this))

{

m.eliminaActores();

Greenfoot.playSound("SC2\_uiBNetToast.mp3");

Greenfoot.setWorld(new Portada());

Greenfoot.stop();

}

}

}

**Clase Demonios:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Demonios here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Demonios extends Botones

{

private int op = 0;

/\*\*

\* Act - do whatever the Demonios wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

Select\_class m = (Select\_class) getWorld();

if(Greenfoot.mouseClicked(this))

{

m.eliminaActores();

op = 2;

Greenfoot.setWorld(new Select\_character(op));

Greenfoot.playSound("Protoss\_Probe.wav");

}

}

}

**Clase Pesado:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Pesado here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Pesado extends Botones

{

private int op = 0;

private int nivel = 0;

/\*\*

\* Act - do whatever the Pesado wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public Pesado(int ra)

{

op = ra;

}

public void act()

{

// Add your action code here.

if(op == 1)

{

setImage("pesadoH.png");

nivel = 5;

}

else if(op == 2)

{

setImage("pesadoH\_d.png");

nivel = 6;

}

entraP();

}

private void entraP()

{

int niv = 0;

niv = nivel;

Select\_character m = (Select\_character) getWorld();

if(Greenfoot.mouseClicked(this))

{

if(nivel == 5)

{

Greenfoot.setWorld(new hNivel3(op));

}

if(nivel == 6)

{

Greenfoot.setWorld(new hNivel3(op));

}

Greenfoot.playSound("Protoss\_Probe.wav");

}

}

}

**Clase Medio:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Medio here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Medio extends Botones

{

private int op = 0;

private int nivel = 0;

/\*\*

\* Act - do whatever the Pesado wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public Medio(int ra)

{

op = ra;

}

public void act()

{

// Add your action code here.

if(op == 1)

{

setImage("medioH.png");

nivel = 3;

}

else if(op == 2)

{

setImage("medioH\_d.png");

nivel = 4;

}

entraM();

}

private void entraM()

{

int niv = 0;

niv = nivel;

if(Greenfoot.mouseClicked(this))

{

if(nivel == 3)

{

Greenfoot.setWorld(new hNivel2(op));

}

if(nivel == 4)

{

Greenfoot.setWorld(new hNivel2(op));

}

Greenfoot.playSound("Protoss\_Probe.wav");

}

}

}

**Clase Boton SalirMenu:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class botonSalirMenu here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class botonSalirMenu extends Botones

{

/\*\*

\* Act - do whatever the botonSalirMenu wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

salida();// Add your action code here.

}

private void salida()

{

if(Greenfoot.mouseClicked(this))

{

Greenfoot.playSound("SC2\_uiBNetToast.mp3");

Greenfoot.setWorld(new Menu());

}

}

}

**Clase Ayuda:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Boton\_Ayuda here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Boton\_Ayuda extends Botones

{

/\*\*

\* Act - do whatever the Boton\_Ayuda wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

Menu m = (Menu) getWorld();

if(Greenfoot.mouseClicked(this))

{

m.eliminaActores();

Greenfoot.setWorld(new Ayuda());

}

}

}

**Clase Texto ayuda:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Texto\_Ayuda here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Texto\_Ayuda extends Actor

{

/\*\*

\* Act - do whatever the Texto\_Ayuda wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

Ayuda m = (Ayuda) getWorld();

if(Greenfoot.mouseClicked(this))

{

Greenfoot.setWorld(new Menu());

m.removeObject(this);

}

}

}

**Clase vida enemigo:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class vidaEnemigo here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class vidaEnemigo extends Actor

{

int golpe = 0;

/\*\*

\* Act - do whatever the vidaJugador wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public vidaEnemigo(int hit)

{

golpe = hit;

}

public void act()

{

// Add your action code here.

bajavidaEn();

}

private void bajavidaEn()

{

int h = 0;

h = golpe;

if(h == 0)

{

setImage("VidaEnemEnCero.png");

}

if(h == 1)

{

setImage("VidaEnem6.png");

}

if(h == 2)

{

setImage("VidaEnem5.png");

}

if(h == 3)

{

setImage("VidaEnem4.png");

}

if(h == 4)

{

setImage("VidaEnem3.png");

}

if(h == 5)

{

setImage("VidaEnem2.png");

}

if(h == 6)

{

setImage("VidaEnem1.png");

}

if(h == 7)

{

setImage("VidaEnem.png");

}

}

}

**Clase poderes:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Poderes here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Poderes extends Actor

{

/\*\*

\* Act - do whatever the Poderes wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

// Add your action code here.

baja();

}

public void baja()

{

setLocation(getX(),getY()+20);

if(isAtEdge())

{

getWorld().removeObject(this);

}

}

}

**Clase minimizar poderes:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class MinimizarPoder here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class MinimizarPoder extends Poderes

{

private int contaP = 0;

/\*\*

\* Act - do whatever the MinimizarPoder wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

// Add your action code here.

contaP++;

if(contaP == 50)

{

baja();

contaP=0;

}

}

}

**Clase Agradar vida:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class AgrandarVida here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class AgrandarVida extends Poderes

{

private int contaP = 0;

/\*\*

\* Act - do whatever the AgrandarVida wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

contaP++;

if(contaP == 50)

{

baja();

contaP=0;

}

}

}

**Clase invunerable:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class Invulnerable here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class Invulnerable extends Poderes

{

private int contaP = 0;

/\*\*

\* Act - do whatever the Invulnerable wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

contaP++;

if(contaP == 50)

{

baja();

contaP=0;

}

}

}

**Clase detener usuario:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class DetenerUsuario here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class DetenerUsuario extends Poderes

{

private int contaP = 0;

/\*\*

\* Act - do whatever the DetenerUsuario wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

contaP++;

if(contaP == 50)

{

baja();

contaP=0;

}

}

}

**Clase detener enemigo:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class DetenerEnemigo here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class DetenerEnemigo extends Poderes

{

private int contaP = 0;

/\*\*

\* Act - do whatever the DetenerEnemigo wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

contaP++;

if(contaP == 50)

{

baja();

contaP=0;

}

}

}

**Clase daño usuario:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class DañoUsuario here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class DañoUsuario extends Poderes

{

private int contaP = 0;

/\*\*

\* Act - do whatever the DañoUsuario wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

contaP++;

if(contaP == 50)

{

baja();

contaP=0;

}

}

}

**Clase jugadores:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Esta es la clase principal de los jugadores, incluye los movimientos y validaciones sobre otros

\* objetos que Interactuan en los mundos

\* @author (Edgar Daniel Rivera Rangel / Nora María Pedraza Cisneros)

\* @version (a version number or a date)

\*/

public class Jugadores extends Actor

{

private int salto = 10;

private int poder = 2;

private int disp = 0;

private int puntos = 0;

private int mundo = 0;

private int jump = -20;

private int verticalSpeed = 5;

private int mili = 0;

private int cont1 = 0;

private int cont2 = 0;

private int cont3 = 0;

private int mp;

private int inv;

private int su;

private int du;

private int de;

private int av;

//Modificar if's mas abajo

/\*\*

\* Act - do whatever the jugador wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

mover();

}

/\*\*

\*

\*/

protected void mover()

{

String cad;

int i;

int x = getX();

int y = getY();

int band\_dir=1;

int band\_mov = 1;

int band = Greenfoot.getRandomNumber(4000);

Actor bala = getOneIntersectingObject(bala\_enem.class);

Enemigos e = new Enemigos();

getWorld().addObject(new vidaJugador(disp), 200 , 100 );

checkFall();

mili++;

if(mili == 50)

{

if(inv != 0)

{

cont1++;

if(cont1 == 5)

{

inv=0;

cont1=0;

}

}

if(su != 0)

{

cont2++;

if(cont2 == 5)

{

su=0;

cont2=0;

}

}

if(mp != 0)

{

cont3++;

if(cont3 == 5)

{

mp=0;

cont3=0;

poder = 2;

}

}

mili = 0;

}

if(su == 0)

{

if (Greenfoot.isKeyDown("Right"))

{

setLocation(getX()+1,getY());

band\_dir = 1;

}

if (Greenfoot.isKeyDown("Left"))

{

setLocation(getX()-1,getY());

band\_dir = 2;

}

if (Greenfoot.isKeyDown("Space"))

{

if(onGround())

{

verticalSpeed = jump;

fall();

}

}

if(Greenfoot.isKeyDown("F"))

{

getWorld().addObject(new bala\_jug(band\_dir,poder),x,y);

}

}

if(Greenfoot.isKeyDown("escape"))

{

Greenfoot.setWorld(new Menu());

}

if(isTouching(MinimizarPoder.class))

{

e.minP();

poder = 1;

removeTouching(MinimizarPoder.class);

}

if(isTouching(Invulnerable.class))

{

inv = 1;

removeTouching(Invulnerable.class);

}

if(isTouching(DetenerUsuario.class))

{

su= 1;

removeTouching(DetenerUsuario.class);

}

if(isTouching(DetenerEnemigo.class))

{

e.detenEnem();

removeTouching(DetenerEnemigo.class);

}

if(isTouching(DañoUsuario.class))

{

disp++;

removeTouching(DañoUsuario.class);

}

if(isTouching(AgrandarVida.class))

{

disp=disp-1;

removeTouching(AgrandarVida.class);

}

if(isTouching(bala\_enem.class))

{

if(bala != null)

{

if(inv == 0)

{

disp++;

}

if(disp == 7)

{

Greenfoot.setWorld(new gameOver());

}

}

removeTouching(bala\_enem.class);

}

}

private void fall()

{

setLocation(getX(),getY() + verticalSpeed);

verticalSpeed = verticalSpeed + 2;

}

private boolean onGround()

{

if(getY()>=550)

{

return true;

}

else

{

return false;

}

}

private void checkFall()

{

if(onGround())

verticalSpeed = 0;

else

fall();

}

}

**Clase jugador:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

import java.lang.\*;

/\*\*

\* Write a description of class jugador here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class jugador extends Jugadores

{

/\*\*

\* Act - do whatever the jugador wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

mover();

}

}

**Clase djuglight**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class dJugLight here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class dJugLight extends Jugadores

{

/\*\*

\* Act - do whatever the jugador wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

mover();

}

}

**Clase djug:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class dJug here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class dJug extends Jugadores

{

/\*\*

\* Act - do whatever the jugador wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

mover();

}

}

**Clase djugPe:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class dJugPe here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class dJugPe extends Jugadores

{

/\*\*

\* Act - do whatever the jugador wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

mover();

}

}

**Clase jugLight:**

import greenfoot.\*; // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/\*\*

\* Write a description of class jugLight here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class jugLight extends Jugadores

{

/\*\*

\* Act - do whatever the jugador wants to do. This method is called whenever

\* the 'Act' or 'Run' button gets pressed in the environment.

\*/

public void act()

{

mover();

}

}

**Clase SmootMover:**

import greenfoot.\*; // (World, Actor, GreenfootImage, and Greenfoot)

/\*\*

\* A variation of an actor that maintains a precise location (using doubles for the co-ordinates

\* instead of ints). This allows small precise movements (e.g. movements of 1 pixel or less)

\* that do not lose precision.

\*

\* @author Poul Henriksen

\* @author Michael Kolling

\* @author Neil Brown

\*

\* @version 3.0

\*/

public abstract class SmoothMover extends Actor

{

private double exactX;

private double exactY;

/\*\*

\* Move forward by the specified distance.

\* (Overrides the method in Actor).

\*/

@Override

public void move(int distance)

{

move((double)distance);

}

/\*\*

\* Move forward by the specified exact distance.

\*/

public void move(double distance)

{

double radians = Math.toRadians(getRotation());

double dx = Math.cos(radians) \* distance;

double dy = Math.sin(radians) \* distance;

setLocation(exactX + dx, exactY + dy);

}

/\*\*

\* Set the location using exact coordinates.

\*/

public void setLocation(double x, double y)

{

exactX = x;

exactY = y;

super.setLocation((int) (x + 0.5), (int) (y + 0.5));

}

/\*\*

\* Set the location using integer coordinates.

\* (Overrides the method in Actor.)

\*/

@Override

public void setLocation(int x, int y)

{

exactX = x;

exactY = y;

super.setLocation(x, y);

}

/\*\*

\* Return the exact x-coordinate (as a double).

\*/

public double getExactX()

{

return exactX;

}

/\*\*

\* Return the exact y-coordinate (as a double).

\*/

public double getExactY()

{

return exactY;

}

}