

THE ROCKET GAME

GUÍA DEL TALLER

Configurando el “Mundo”

1. Inicie Greenfoot
2. Cree un nuevo escenario en Java:
 - Scenario -> New Java Scenario
 - Ingrese el nombre y el directorio
3. En el nuevo escenario, haga doble click en MyWorld
4. En el editor de código, busque el método MyWorld() y modifique el tamaño del canvas a 800 x 600:

```
super(800, 600, 1);
```

5. Presione el botón “Compile” y cierre la ventana de código
6. En la ventana principal vaya a Tools y seleccione “Recompile Scenario”
7. Luego Tools -> Save the World
8. Haga right-click en MyWorld y seleccione “Set Background” en el menu flotante
9. En la lista de imagenes, expanda “Backgrounds” y seleccione el cielo estrellado
10. Recompile y Guarde el escenario

Agregando una nueva clase: nave espacial

11. En el navegador de la derecha, haga right-click en “Actor” y seleccione “New Subclass en el menu flotante
12. Digite “Rocket” como el nombre de la nueva clase.
13. En las categorías de imágenes, expanda “Transport” y seleccione la imagen del cohete (es la última)
14. En el navegador, haga right-click en Rocket y seleccione new Rocket()
15. Mueva el mouse y haga click para colocar en nuevo cohete en el medio y a la izquierda de la pantalla (mundo).
16. Ahora, abra el editor de código de Rocket.
17. En el método act, agregue la siguiente línea de código:

```
move(1);
```

18. Presione el botón Compile
19. Salve y luego recompila el mundo.
20. Presione el botón “Run”

21. Mueva ahora el cohete a la derecha de la pantalla
22. Modifique el método act, para que sea move(-1);
23. Compile el código, salve y luego recompile el mundo.
24. Ejecute su programa.
25. Presione "Reset" para reiniciar su mundo.
26. Coloque el cohete de vuelta en la izquierda de su pantalla, pero en la esquina inferior.
27. Haga click derecho en el cohete en la pantalla, y en el menú flotante seleccione "inherited from Actor" -> void turn(int)
28. Ingrese -45
29. Modifique el código para que de nuevo el cohete se mueva para adelante.
30. Cree 3 instancias adicionales de Rocket, colóquelas en la pantalla, gírelas en direcciones diferentes y ejecute su programa.

Moviendo el cohete con las teclas

31. Abra el código de la clase Rocket y agregue los siguientes métodos y variables:

```
private static int moveStep = 5;

public void moveUp()
{
    setLocation(getX(), getY() - moveStep);
}

public void moveDown()
{
    setLocation(getX(), getY() + moveStep);
}

public void handleMovement()
{
    if (Greenfoot.isKeyDown("up"))
        moveUp();
    else if (Greenfoot.isKeyDown("down"))
        moveDown();
}
```

32. Modifique el método act() para que sea solamente un llamado a handleMovement()
33. Borre el cohete que tiene actualmente en el mundo y cree uno nuevo. Coloquelo en la mitad de la pantalla, totalmente a la izquierda.
34. Salve, recompile y ejecute su programa, y use las teclas up and down para mover el cohete en la pantalla.

¡Abriendo fuego! Agregando una nueva clase “Bullet”

35. En el navegador, cree una nueva clase “Bullet”, y asocie la imagen de la esfera roja que se encuentra bajo la categoría “other”
36. En el código de Bullet, modifique el act para que la bala se mueva 10 pixeles hacia adelante:

```
move(10);
```

37. Cree una nueva instancia de Bullet y colóquela justo delante del cohete en el mundo.
38. Salve, Recompile y ejecute su programa.
39. Hay que hacer desaparecer la bala una vez que llegue al final de la pantalla. Para ello modifique el código en el act() de Bullet y para que sea:

```
move(10);  
if (isAtEdge())  
    getWorld().removeObject(this);
```

40. Salve, Recompile y ejecute su programa.
41. Ahora, hay que hacer que cada vez que se presione la tecla F (Fuego!) una nueva bala se cree. Para ello, hay que modificar el código de Rocket
42. Agregue una nueva variable estática:

```
private static int shipSize = 50;
```

43. Agregue el método fire():

```
public void fire()  
{  
    getWorld().addObject(new Bullet(), getX()+shipSize,getY());  
}
```

44. Agregue el siguiente código al final del método handleMovement():

```
public void handleMovement() {  
    if (Greenfoot.isKeyDown("up"))  
        moveUp();  
    else if (Greenfoot.isKeyDown("down"))  
        moveDown();  
  
    String pressKey = Greenfoot.getKey();  
    if (pressKey != null) {  
        if (pressKey.equals("f"))  
            fire();  
    }  
}
```

45. Remueva del mundo la bala que había creado antes.
46. Salve, Recompile y ejecute su programa.

Agregando los “aliens”

47. Cree una nueva clase “Alien” y asigne como imagen la silueta verde bajo la categoría de people.
48. Agregue una característica (speed) y dos métodos “constructores” a la clase Alien:

```
private int speed;

public Alien()
{
    speed = 1;
}

public Alien(int newSpeed)
{
    speed = newSpeed;
}

public void act()
{
    move(-1 * speed);
}
```

49. Cree dos instancias de Alien, una con speed en 1 y otra con speed en 10. Coloquelas a la derecha de la pantalla.
50. Salve, recompile y ejecute su programa.
51. Ahora, agregue un método más a la clase Alien, que determinará si el juego acaba (un alien llega hasta la base):

```
public boolean gameFinished()
{
    return (isAtEdge());
}
```

52. Modifique el metodo act de la clase Alien, para que verifique si el juego terminó y coloque el mensaje correspondiente en la pantalla:

```
public void act()
{
    move(-1 * speed);

    if (gameFinished())
    {
        getWorld().showText("GAME OVER", getWorld().getWidth()/2,
                                getWorld().getHeight()/2);
        Greenfoot.stop();
    }
}
```

53. Salve, recompile y ejecute su programa.

Destruyendo los aliens

Ahora hace falta agregar la lógica que haga “morir” a un alien cuando una bala lo alcance. Para ello, hay que modificar el código de Bullet.

54. En el código de Bullet, modifique el método `act()`, como sigue:

```
public void act()
{
    move(10);

    if (isTouching(Enemy.class)) {
        removeTouching(Enemy.class);
        getWorld().removeObject(this);
    }
    else
        if (isAtEdge()) {
            getWorld().removeObject(this);
        }
}
```

55. Salve, recompile y ejecute su programa.

Finalizando el juego: colocando todo junto

Es hora de poner el juego a funcionar, ya que tenemos todos los elementos. Para ello, hay que mantener un numero creciente de aliens en la pantalla, y aumentar su velocidad cada vez mas para que el juego se haga interesante. Todo esto lo haremos en la clase MyWorld, como sigue.

56. Abra el editor de código de la clase MyWorld y agregue las siguientes variables:

```
private static int minNumberOfEnemies = 3;
public int numberOfEnemies;
public int enemySpeed; // higher numbers are slower
public int times;
public int score;
```

57. Modifique el constructor de MyWold, para que inicialice el estado:

```
public MyWorld()
{
    super(800, 600, 1);

    numberOfEnemies = 0;
    enemySpeed = 1;
    score = 0;
    times = 0;

    prepare();
}
```

58. Agregue 4 nuevos métodos, incluido el método act:

```
public void reduceEnemies()
{
    numberOfEnemies--;
    score += 100;
}

public void checkEnemies()
{
    if (numberOfEnemies <= minNumberOfEnemies) {
        int toAdd = Greenfoot.getRandomNumber(minNumberOfEnemies+1);
        for (int i = 0; i < toAdd; i++) {
            addObject(new Alien(enemySpeed), getWidth()-40,
                           Greenfoot.getRandomNumber(getHeight()));
        }

        numberOfEnemies += toAdd;

        times++;

        if ((times % 10) == 0)
            enemySpeed++;
    }
}

public void printScore()
{
    showText("Score: " + score, 100, getHeight() - 20);
}
```

```
public void act()
{
    checkEnemies();
    printScore();
}
```

59. Salve, recompile y ejecute su programa.

60. Ahora, hay que modificar la clase Bullet, para que elimine los aliens que van desapareciendo, de modo que se creen mas. Para ello en el código de act() de Bullet, para agregar un llamado a reduceEnemies:

```
if (isTouching(Alien.class)) {
    ((MyWorld) getWorld()).reduceEnemies();
}
```

61. Salve, recompile y juegue!!

Únase a Oracle Academy

Visite academy.oracle.com para tener más detalles del programa y registrar gratuitamente a su institución.

Como programa educativo global y filantrópico de Oracle, Oracle Academy promueve la educación en computación en todo el mundo, para aumentar el conocimiento, la innovación, el desarrollo de habilidades y la diversidad en los campos de la tecnología. Oracle Academy comprende y valora a los educadores como socios que están capacitados para facilitar el aprendizaje innovador de los estudiantes dentro y fuera del aula. Nos relacionamos con miles de instituciones educativas y educadores en más de 120 países, ayudando a millones de estudiantes a prepararse para la universidad y la carrera.

Oracle Academy continúa el compromiso de Oracle con la educación, que se extiende por 25 años y es parte del legado de la empresa. Oracle Academy ofrece a los educadores y a sus estudiantes recursos gratuitos para enseñar y aprender informática. Esto incluye currículo, recursos de aprendizaje, tecnologías de nube de Oracle, software, soporte y recursos de certificación profesional.

Con Oracle Academy, los estudiantes reciben experiencia práctica con las últimas tecnologías, lo que los ayuda a prepararse para la universidad y la carrera en la era de la computación en la nube, la inteligencia artificial, el aprendizaje automático, la ciencia de datos, la Internet de las cosas y más.

Oracle Academy trabaja con socios públicos y privados para proporcionar las herramientas y la capacitación que los educadores necesitan para involucrar, inspirar y preparar a los estudiantes para que se conviertan en innovadores y líderes del futuro.