

| Space Invaders | Lesson 05 | Reorganización código |
|--|-----------|-----------------------|
| clases de ejemplo : <code>com.gamedev.dax04.SpaceInvaders</code> <code>com.gamedev.dax04.Entity</code> <code>com.gamedev.dax04.Player</code> | | |

En este ejemplo creamos la astronave del jugador y implementamos el bucle del juego.

Metemos mano al código para limpiar las cosas.

Creamos una clase abstracta Entity, que representa la super-class de todos los objetos del juego (alienígenas, el jugador, los disparos, las bombas, etc.).

```
public abstract class Entity
```

Las entidades del juego:

- saben donde están

```
protected int x;
protected int y;
```

- saben su dirección de movimiento (vx: -1 = izquierda, 0 = parado, +1 = derecha; vy: -1 = arriba, 0 = parado, +1 = abajo)

```
protected int vx;
protected int vy;
```

- saben moverse

```
public abstract void move()
```

- saben dibujarse

```
public abstract void draw(Graphics g)
```

Creamos la clase Player que extiende Entity

```
public class Player extends Entity
```

y implementamos los metodos de la super-clase move() y draw()

```
public void move() {
    x = x + vx;
    if(x <= 0) x = 0;
    if(x > 560) x = 560;
}
```

```
public void draw(Graphics g) {
    g.setColor(Color.WHITE);
    g.fillRect(x, y, 40, 10);           // base
    g.fillRect(x+18, y-7, 4, 7);       // cañon
    g.dispose();
}
```

Modificamos la clase original implementando el bucle del juego, que es basicamente un loop infinito en donde:

- se mueven las entidades
- se controla si se han producido colisiones entre entidades
- se dibujan en la pantalla las entidades
- se hace una pausa de unos milisegundos

```
public void run(){
    Graphics g = getGraphics();
    // init
    init();
    while(true){
        // move objects
        move();
        // check collision
        // paint
        paintComponent(g);
        // sleep
        try{
            Thread.sleep(2);
        }catch(Exception ex){}
    }
}

public void init(){
    player = new Player();
}

public void move(){
    player.move();
}

public void paintComponent(Graphics g){
    setBackground(Color.BLACK);
    player.draw(g);
    repaint();
}
```