

Space Invaders	Lesson 07	Disparos
clases de ejemplo : <code>com.gamedev.dax06.SpaceInvaders</code> <code>com.gamedev.dax06.Entity</code> <code>com.gamedev.dax06.Player</code> <code>com.gamedev.dax06.TestAlien</code> <code>com.gamedev.dax06.Bullet</code>		

En este ejemplo añadimos los disparos del jugador (aun sin consecuencias).

- creamos la clase `Bullet`, siempre extendiendola de `Entity`
- usamos el flag boolean `inGame` para saber si el disparo está en juego o si ha salido de la pantalla.

```
private boolean inGame;
public boolean isInGame() {
    return inGame;
}
```

- Al constructor pasamos la coordenada x del player para saber donde dibujar la bala (el otro constructor, el sin parametros, es de coña, lo he metido para evitar de chequear en el bucle del juego si `bullet == null`).

```
public Bullet(int x) {
    this.x = x;
    ...
}
```

- Como regla, se puede disparar solo si el anterior disparo ha salido del `JFrame` (por esto se usa una sola instancia de `bullet`).
- En la clase principal del juego (`SpaceInvaders.java`):
- añadimos el disparo como campo de clase

```
Bullet bullet;
```

- el `bullet` se pinta y mueve en el bucle del juego como ya sabes:

```
bullet.move();
bullet.draw(g);
```

- Añadimos el `KeyListener` para la tecla 'barra espaciadora', que cuando se pulsa ejecuta el metodo `fire()`

```
if(keyCode== KeyEvent.VK_SPACE) {
    fire();
}
```

- El metodo `fire()` solo se ocupa de crear la nueva bala (si se puede) y la posiciona arriba del cañon del `Player`

```
public void fire() {
    if(!bullet.isInGame()) {
        bullet = new Bullet(player.getX() + 18);
        ...
    }
}
```

- He añadido un par de campos utiles para el debug a video (arriba izquierda) que son si el Player puede disparar y el numero de balas disparadas. Pijadas.

