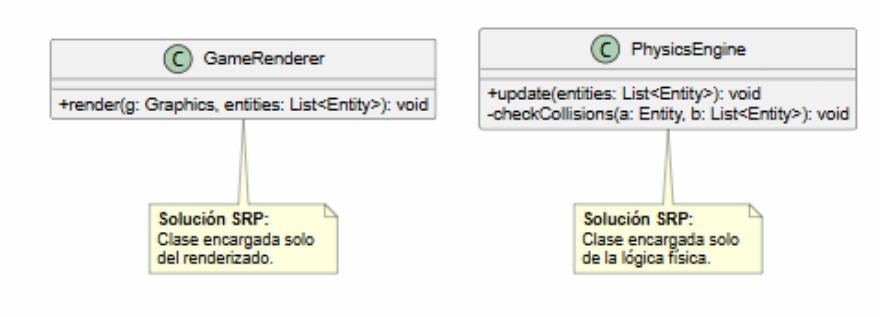
Proposed code

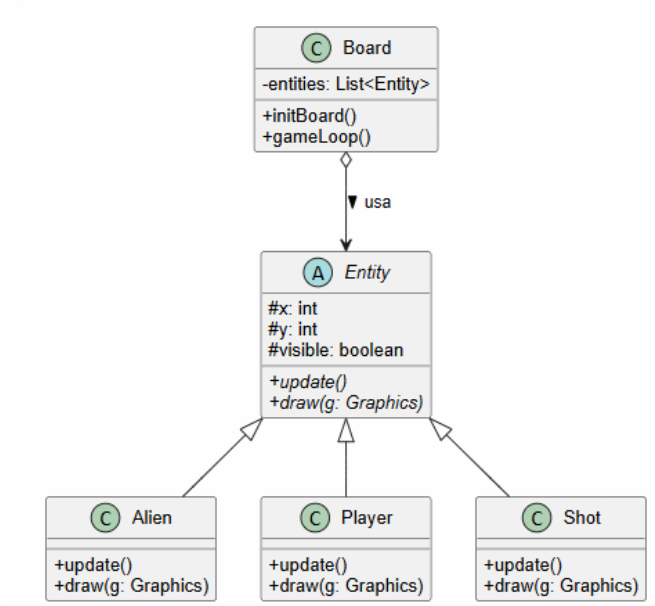
1. Single Responsability Principle (SRP)

```java
// Clase encargada solo del renderizado
public class GameRenderer {
    public void render(Graphics g, List<Entity> entities) {
        for (Entity e : entities) {
            e.draw(g);
        }
    }
}

// Clase encargada solo de la lógica física
public class PhysicsEngine {
    public void update(List<Entity> entities) {
        for (Entity e : entities) {
            e.update();
            checkCollisions(e, entities);
        }
    }
    private void checkCollisions(Entity a, List<Entity> b) { /* Lógica */ }
}
```



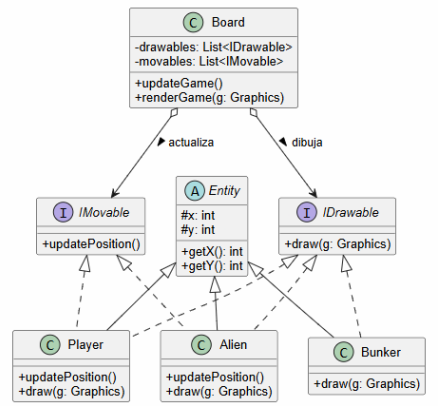
2. Open/Closed Principle (OCP)

```java
public abstract class Entity {
    protected int x, y;
    protected boolean visible;

    public abstract void update();
    public abstract void draw(Graphics g);

    // Getters y Setters comunes
}
```



3. Interface Segregation Principle (ISP)

```
public interface IDrawable {
    void draw(Graphics g);
}

public interface IMovable {
    void move();
}
```



4. Dependency Inversion Principle (DIP)

```java
public class GameBoard {
    private final IRenderer renderer;
    private final List<Entity> entities;

    public GameBoard(IRenderer renderer, EntityFactory factory) {
        this.renderer = renderer;
        this.entities = factory.createInitialLevel();
    }

    public void gameLoop(Graphics g) {
        renderer.render(g, entities); // No depende de Swing directamente
    }
}

public interface IRenderer {
    void render(Graphics g, List<Entity> entities);
}
```