Design Rationale for Enemies

The diagram represents an object-oriented system for the various enemies in the game.

Each enemy type is represented in its own class, which all extend the Enemy abstract class. This is done because they share common attributes and methods which all enemies use. The Enemy abstract class itself extends the Actor abstract class, as it also shares common attributes and methods with other Actor subclasses in the game.

Extending the Enemy abstract class achieves abstraction, and therefore adheres to the Open-closed principle. In other words, extending the Enemy class to add new enemy types into the game is easy and can be done without modifying existing classes.

It also follows the DRY principle of not repeating code, by reusing common attributes and methods through inheritance.