

REQ 3 Design Rationale

The diagram represents the third requirement/part of an object-oriented system for a text-based “rogue-like” game inspired by Elden Ring.

Previously

Individual classes of enemies (LoneWolf, GiantCrab, etc) is implemented by the Resettable interface, we used the classes individually instead of the Enemy abstract class because currently all the enemies are Resettable (can despawn/respawn when reset) but in future additions of enemy may be not Resettable.

FlaskOfCrimsonTears extends from the Item abstract class, because it has common attributes and methods with other item classes. So, it is logical to use abstraction to avoid repetitions (DRY).

ConsumeAction and RestAction classes extended the Action abstract class. Since all action classes have similar attributes and methods, it is logical to extend ConsumeAction and RestAction from Action abstract class to avoid code repetitions (DRY).

SiteOfLostGrace class extends from Ground abstract class, because it is a new environment class and all environment classes have similar attributes and methods, so to avoid repetition we use abstraction (DRY).

The ResetManager manages and depend on the Resettable interface rather than directly to the item or actor classes, this way any classes that implements the Resettable interface can be used with the ResetManager (DIP).

New Changes

Due to Pile of Bones not being an actor, we have decided to just implement the Enemy abstract class because all the enemies are Resettable, this is to avoid code repetition for the Reset method for each enemies.

FlaskOfCrimsonTears implements a Resettable while also extending from the Item abstract class, this change was applied so that in the future we can add items as Resettable as well instead of Resetting it all on the Player class.

Added ResetAction to initialize the reset, instead of letting all of the reset happening on the ResetManager, so that in the future new Behaviours and other classes can utilize the Reset.