Lab 3 Team 4 – Design Rationale

Estus Flask – justification for design

Player has association with Estus Flask as it is in the player’s inventory at all times, it does not extend from the item class because it has no need to inherit functions such as drop item, pick up item, etc. Estus Flask also has an Action to heal the player, this is represented as the EstusFlaskAction class which inherits from the Action parent class. EstusFlaskAction does 1 thing before healing the player, that is to retrieve the Estus charges, if enough Estus charges are available the normal functionality of the Estus flask Is performed. \*This is illustrated as below:

Diagram

Description automatically generated

Opt fragment is used here because there is no alternative to this if statement.

\*Note that activation boxes correspond to EstusFlaskAction, EstusFlask and Player respectively.

Bonfire – justification for design

Application is dependent on Bonfire, as it is displayed in the game world. And because of this it is sensible for Bonfire to inherit from the Ground class. Since it provides an interaction to the player, a BonfireAction class which inherits from the Action is created. In this BonfireAction class heals the player to full health:

Diagram

Description automatically generated

It also uses the Reset Manager to remove undead using the cleanup() method and reset the positions of other enemies. Additionally Reset Manager also restores the charges of the estus flask

Diagram

Description automatically generated

\*Note the four activation boxes represent BonfireAction, ResetManager, EstusFlask and Skeletons in that order

Soft Reset / Dying – Justification for design

Diagram

Description automatically generated

PlayerDeathAction extends from the action class because when called it calls various other classes to retrieve the necessary information to perform the soft reset. TokenOfSouls inherits from the item class as it can be picked up by the player and displayed on the ground.

The PlayerDeathAction class is implemented to manage calls to other classes in a certain order. Upon death the player class executes the PlayerDeathAction, which makes use of the ResetManager to reset all classes that implement the resettable interface, and GameMap to determine the location at which to create the TokenOfSouls.

Graphical user interface, diagram

Description automatically generated

Creation of the token is shown above, the location of player upon death is first retrieved from GameMap, then if there is already an instance of the token it is destroyed. Then the token is created using the player location retrieved earlier, and the number of souls held by the player is transferred to it.