

# IPPO Assignment 2 Draft Design

S2118625

November 13, 2020

## Classes And Methods:

### User

Responsibilities: store in which room is the user, which side of the room is the user looking at and which items the user is holding.

Methods: addItem (adds item to the items being hold), removeItem (removes item from the items being hold), roomLocation (returns the current room where the user is at), viewLocation (returns which side of the room is being looked at by the user e.g. North, South, Est, West).

Dependencies: Room and Item.

### View

Responsibilities: send commands from the physical user to the controller (implements GUI).

Methods: moveRight (tell the controller to move the user to the right), moveLeft (tell the controller to move the user to the left), enterRoom (tell the controller to change roomLocation of the user to the next room), pickItem (tell the controller to addItem in user and removeItem in room) , putItem (tell the controller to removeItem in user and addItem in room) .

Dependencies: Controller and Item.

### Controller

Responsibilities: receive the inputs from View and change the state of User and Room.

Methods: `newView` (create new View Class), `addRooms` (creates the rooms available), `createItems` (creates the available items), `addItem` (add items to rooms), `moveRight` (move the user to the right e.g. from North to West), `moveLeft` (move the user to the left e.g. from West to North), `enterRoom` (change `roomLocation` of the user to the next room if possible), `pickItem` (addItem in user and removeItem in room), `putItem` (removeItem in user and addItem in room).

Dependencies: View, User, Item, Room.

## **Item**

Responsibilities: retrieve image of item.

Methods: `getPicture` (returns from Service the picture of the item) .

Dependencies: Service.

## **Service**

Responsibilities: return images requested.

Methods: `getPicture` (returns the image requested) .

Dependencies:

## **Room**

Responsibilities: store the number of sides in the room, store on which sides there are available exits, which items does the room have and return pictures for the sides of the room.

Methods: `isExit` (return Boolean statement indicating if the side of the room contains an exit), `setExists` (indicate which sides have exits), `addItem` (adds item to the room), `removeItem` (removes item from the room), `getPicture` (requests from Service and then returns the image of one side of the room).

Dependencies: Item and Service.

## Design Choices

The use of a the class Service allows us to store all the images in one place. This in the future can make very easy the access of images from the web and the use of proxies. Alternative: make each respective class store their image.

Separating the View from the Controller makes very easy to test if the GUI is working as expected with tests or a Prototype for Controller. Alternative: The Controller receives the input directly.