Introduction to java programming
Assignment 2

Siyuan Zhao
ID: s1682851

October 2016

1 Briefly describe your class design, including the function and relationships between the important classes. You may do this using text and/or diagrams, but the result must be clear and concise.

I have the following class.

- Class **Location** is used to describe the location in the world. It has the following attributes. **String location** is the name of the location. **int totalViewDegree** is the total number of directions of views. **int currentViewDegree** is the current direction. **HashMap ⟨Integer, Location⟩ exits** stores the exits location and corresponding direction. It has the following methods:
  - **public Location(String locationName, int degree)**: It is the constructer to define the attributes.
  - **public void setExit(int degree, Location location)** sets the exits of next location.
  - **public void rotateRight** changes **currentView** to define the behaviour of rotate right.
  - **public void rotateLeft** defines the behaviour of rotate right.
  - **public Location moveForward()** defines the behaviour of move forward and it returns the next location. The **currentViewDegree** is changed in the meantime to set the new **currentView** when people come back to this location.
  - **public boolean isForwardable** checks whether people can move forward in current direction.
  - **public String getLocation** is a getter function to get the attribute **Location**.
  - **public String getCurrentLocationName** returns a string combining the location name and direction.
  - **public boolean equals(Object obj)** checks whether one location equals to another location in content. Only name is checked because if the name is equal, the location should be the same.

- Class **Room** is a subclass of class **Location** the only difference is the method of **moveForward**. It describes a special location when there is only four direction so the **currentViewDegree** is changed differently when people come back.

- Class **PortableItem** describes the portable item to be picked up and put down by user. It has attributes **String name** which is the name of the item and **Location currentLocation** which is the place that it is placed. When it is **null** type, it means the item has not been put in any locaition yet. This class has the following functions.

  - **public PortableItem(String name)** is the constructor.
  - **public String getName()** returns the name of the item.
  - **public Location getCurrentLocation()** returns the current location of the item.
  - **public void putDownItem(Location location)** put down the item and change the current location.
  - **public void pickUpItem()** pick up the item and make the current location become **null** type.
  - **public boolean isBeenPutDown()** to check if the item has been put down in any locations.

- Class **TheLabWorld** describes the the lab world in university of Edinburgh in forest hill. It has the following attributes **Location currentLocation** is the current location where the user is. **public PortableItem[] itemList** is the portable items list.

  - **TheLabWorld()** constructor to build the world.
  - **private void createWorld()** is the method to create the world. All locations and portable items needed is built here.
  - **public PortableItem[] getItemList()** gets the item list array.
  - **public void putDownItem(int i)** puts down an item specified by variable **int i**.
  - **public void pickUpItem(int i)** pick up one item specified by **int i**.
  - **public boolean canPickUp(int i)** check if one item can be picked up.
  - **public boolean canPutdown(int i)** check if one item can be put down.
  - **public void processCommand(String command)** Process the given command from "right", "left" and "forward".
  - **public Location getCurrentLocaiton()** gets current location.
  - **String getCurrentLocationName()** get current location's name.
  - **public String getCurrentPictureName()** returns the file url.
  - **public boolean isForwardable()**checks if people can forward in current location.

- Class **WorldController** is the controller for the world. It has attributes **TheLabWorld lab**, **ImageView imageView, mapView** for the image view on the left of the pane. **ImageView basketView, waterView, rabbitView** for image view on the left of the pane and **ImageView basket, water, rabbit** for the image view on the top left of the plane. **Button forward** is for the forward command, MenuItem pickUp0, pickUp1, putDown0, putDown1, pickUp2, putDown2 for the menu item and **Text showPosition** for the text indicating the current location for the user.

  - **public void Initialise()** initialise the lab world.
  - **public void updateItem()** updates the image view of item in the pane.
  - **public void pickUpBasket(ActionEvent event)** is the pick up action for the basket item.
  - **public void putDownBasket(ActionEvent event)**is the put down action for the basket item.
  - **public void pickUpWater(ActionEvent event)**Pick up action for the bottle of water item.
  - **public void putDownWater(ActionEvent event)**put down action for the bottle of water item.
  - **public void pickUpRabbit(ActionEvent event)**Pick up action for the rabbit item.
  - **public void putDownRabbit(ActionEvent event)** put down action for the rabbit item.
  - **public void updateMenu()** update the menu set some of menu item disable.
  - **private void processButton(String command)** process the button.
  - **public void pressButtonRight(ActionEvent event)** process the button to rotate right.
  - **public void pressButtonLeft(ActionEvent event)** process the button to rotate right.
  - **public void pressButtonForward(ActionEvent event)** process the button to move forward.

2 Briefly describe one or two significant choices that you had to make in the model design and explain why you chose your design over some plausible alternative.

At first I want to use string to defines the direction of view such as "left", "right", "forward" and "backward". However, what if I need the direction such as "right forward" to turn to the one of the doors in the small wall. Actually, Using string to describe angles is not smart. I finally use integer to describes directions. They are 1-based indices. and "1" indicates the starting directions which is forward and I actually don't care how many directions do I need which is convenient for the further developing.

3 Add any extra comments that you would like to make about your solution, or the assignment in general. In particular:

3.1 You should note any external resources from which you took code, and any significant col- laborations with others.

I did this work independently.

3.2 It would also be useful if you could indicate here if you would be prepared to allow us to publish your screenshots – many students produce interesting interfaces and images and it is useful to be able to show these to others – for example, the course website shows some previous implementations.

I would be prepared to publish your screenshots