

Informal Specification and Test Plan

I will begin by writing a class called "GUI" and this class will be responsible for displaying the GUI and dealing with the interactions from the users. The methods it will contain are:

- A method called "gui()" which will be responsible for creating a new JFrame and the JPanel for the GUI.
 - o The JPanel will be split into 3 sections
 - One section to display the movement buttons
 - One section to display the map and text information
 - One section to display any other buttons e.g. 'quit' or 'pickup'
 - o A method called "buttons" which will be responsible for creating all the buttons and linking them to their respective action
 - For example if the "move up" button was pressed the moveCommand() from GameLogic will be called and the player will be moved north
 - To create my buttons I will use JButtons and use an action listener to link the button with its respective action

This class is what I will use to try and gain the mandatory 15 points.

As I am not using a server/client in my coursework 3 I have been limited when it comes to picking my 30 extra marks but I have chosen to add a graphic user pane. This will not have its own class it will be a method added to the "GUI" class. As I have not done a graphic pane before I am unsure of how to do this but my plan is to loop through the map and to add an image depending on what character it is. For example if the current character is '#' put the image of a wall in its place.

Next I will add more interactions for the player. This will not have its own class but will be added to GameLogic. I plan to add doors that are locked and the only way to get access through them is to pickup a key to unlock the door. The methods I will add to GameLogic are:

- A method called "openDoor()" which will return "Success!" or "Fail!" depending on if the user has picked up a key prior to trying to open the door
 - o If the user doesn't have a key they will not be able to walk through the door
- I will extend my pre existing pickup() method so that it takes into account keys and not just coins

Finally I will create a "god's eye view" pane. I plan for this to be an extra method in the GUI class. It will be very similar to what I am planning to create for the graphic pane. I might even use the same method as I use for the graphic pane as it should be the same code except instead of printing the look reply it shall print the whole map. There will be a button in the main gui which can be pressed and the gods eye view map will be printed in a new separate gui.

My test plan will involve the testing of each element once it is implemented. Each time a change is made I will run my program and check that it should do what is expected.

Action	Test	Expected Result
Create a basic GUI	Run the program	A basic GUI will be displayed on the screen
Add buttons to the GUI	Run the program and click on the newly added button	The button should do its corresponding action e.g. if the button is "move up" the player should move north one space each time the button is clicked
Add a graphic pane	Run the program	When the map is printed to the screen it should display images instead of characters like '#', '.' and 'P'
Add extra interactions	Run the program and carry out the new interaction	The new addition should do its corresponding action e.g. if a key is added to the map the player should be able to pick it up and unlock a door with it
Create a God's eye view pane	Run the program and click on the 'God's eye view' button	The full map of the game should be displayed to the user

Intended Features	Points	Works? (self evaluation)	Works? & marks (checker evaluation)	Special instructions
Basic GUI	15	Yes		None
Graphic Pane	10	Yes		None
God's eye view	10	Yes		None
Extra interactions	10	Yes		A sword must be picked up to slay the monster and a key must be picked up before trying to unlock a door
Total points	45			

Sketch

This is a sketch of the initial idea for my GUI. As explain above the main JPanel is split into three sections, the map will be printed in the center of the GUI with the buttons to play the game surrounding it.

