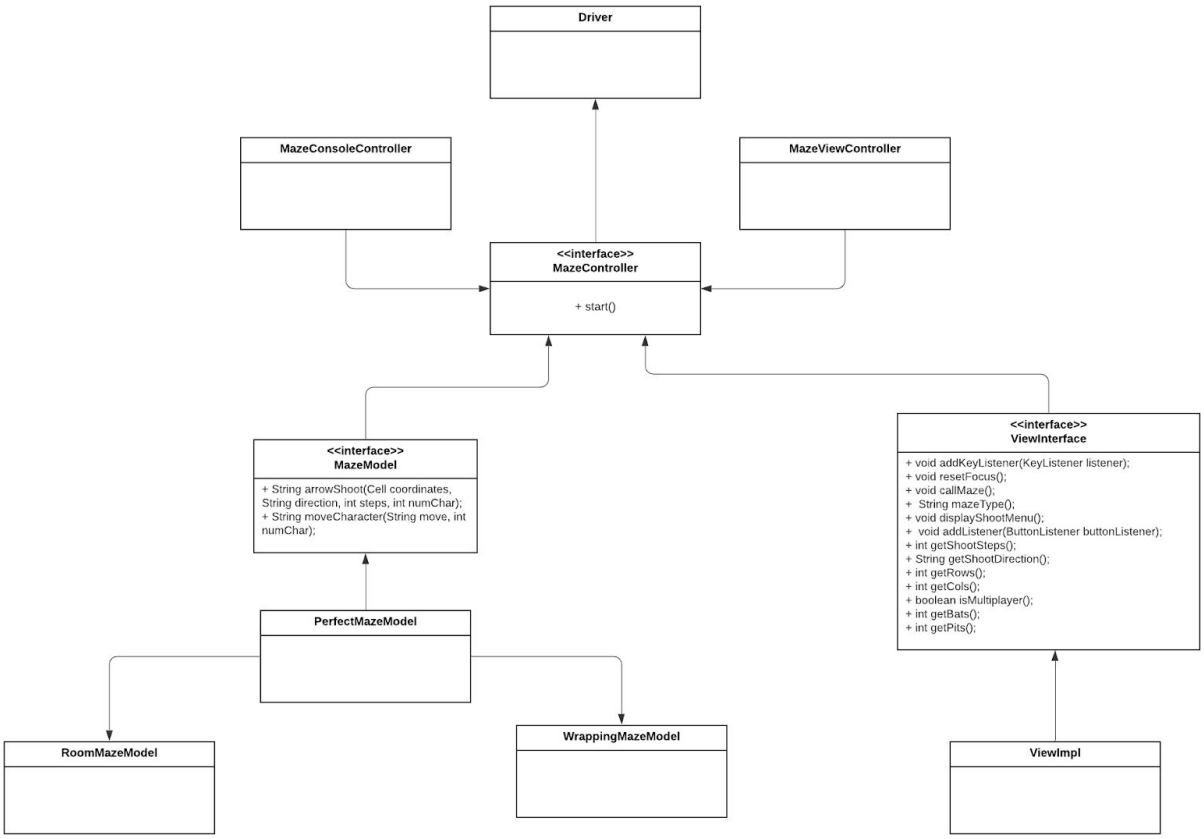
UML Diagram



Description

The JAR file can be run with the --text flag for the text version of the game, and --gui flag for the GUI version of the game. The text version is the same, aside from the multiplayer functionality. For the View, I have a ViewInterface interface that contains all the methods which can be accessed by the controller. At first, the program starts with the driver, and it passes in an empty model and a view into the controller. From there, the control goes to the view, and it takes in the details of the maze from the user. The control comes back into the controller and the controller creates the maze model with the specifications provided by the user. The maze along with the direction buttons along with the starting point of the character is displayed. After configuring the keyboard and button listeners, every time the user presses a button, the old frame closes and a new frame which has the new location of the character opens up and displays all the cells which the character has visited. In the model the only the Character data type was changed. Now, the model holds an array of Characters instead of a single Character object. So the character which is being referred to in a certain context can be obtained via an index.

Testing

1) Testing what happens when the user enters a room with a bat.
2) Testing whether the game gets over if the user enters a room with a bottomless pit.
3) Testing whether the game gets over when the user runs out of arrows.
4) Testing if the location of the user changes when he encounters a room with super bats.
5) Testing if the user can smell the stench if he is one room away from the wumpus.
6) Testing if the user can feel a draft if he is one room away from a pit.
7) Testing if the starting location of the user is within the maze or not.
8) Testing whether the game gets over when the user kills the wumpus.
9) Testing the user enters another room when he steps out of one.
10) Testing the percentage of bats and pits do not exceed the threshold value.

Challenges

1) Figuring out how to get the specifics of the maze input by the user in the view, back into the controller, in order to create the model.
2) Figuring out how to attach keyboard listeners in the view.
3) Figuring out how to set the focus back into the view in order for the keyboard and button click events to work.
4) Figuring out a testing plan for this assignment.