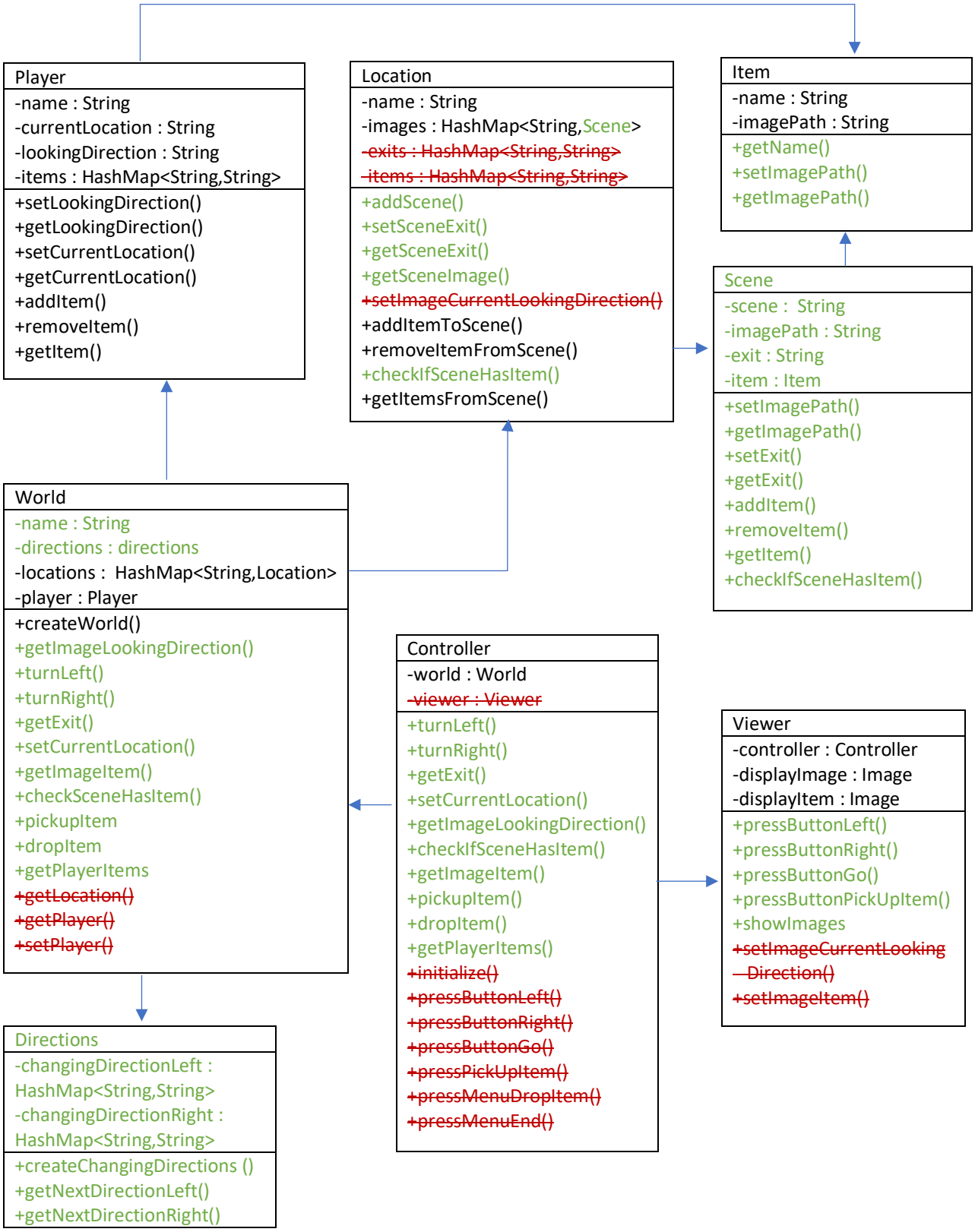
Class diagram



Design choices

I used the initial class design as a backbone for my application. However, after watching the lecture and starting to write the code I came up with some ideas how to improve the class design.
The new class design is shown above. Green text means the class or method is new and red text means the method was removed from the code.
First of all, I added the class Scene that contains the compass direction, the image file name and an item. I find that the code got easier to understand by adding this class. I further made the choice of restricting the amounts of items on a scene to one item only. That means if the scene already has an item then it is not possible for the user to drop another item on that scene.
Second of all, I added a second class Directions that is holding the information of the compass directions. Hence if the user wants to turn left and he is currently looking to the "north" then it returns the logical direction "west". I had to implement the logic somewhere and for me it makes sense to put it in a separate class since it makes the code easier to read and if I want to change the order of the compass direction, I could do that very easily.
Third of all I put everything that GUI related into the Viewer class, because the controller is supposed to be the intermediary between the model and the viewer. In my final design the ImageViews, Buttons, Menu and all action related to them got migrated to the Viewer class. By doing so I could not make the application start with the controller (which I intended to do) instead it is starting with the Viewer that is then creating an instance of the controller.

External resources

For adding items to the MenuBar I used code that I found on the website tutorials.jencov.com.
Link: http://tutorials.jenkov.com/javafx/menubar.html