Recommendations for changes to the game engine

Problems

1. The use of private/protected variables throughout the engine: The current engine does not let us have access to important variables which makes it difficult to implement certain functionalities. This meant excessive code was to be made instead of just having a simple get method for the variable.
    - ActorLocations stores the location of all the actors in the map, and there was no way to access this variable. Instead we had to code an inefficient method that checks every location on the map for an actor and adds it to a list which is then returned. This class is called ListActors if you wish to see the code

Design Changes

1. To simply create a getter method for important variables to allow other classes access to the variable. For the List of all actors problem simply putting a getActorLocations method would solve the problem of writing up the ListActors Class.

Advantages and disadvantages

1. Advantages would be the reduction of redundant code as in our case, if there was access to the actorLocations, we would not have to have written ListActors. A disadvantage would be that writing a get method for every variable would lead to an excess of code with unnecessary getter methods. However this can be solved by using them just for important variables such as actorLocations in GameMap

What was Done Right

Overall the engine was designed very well as each class had a specific purpose and called the relevant classes, methods to complete that specific purpose. Each class also was unique and didn't have any repeated code in any classes. This meant that abstraction was done well as classes that were similar and branched from the same superclass were done in a way that no code was repeated. An example would be the Action class as there were many actions that were required but are done in a way that each action was unique.