**C Sc 335 Analysis and Design Artifacts for the Final Project, Fall 2014**

*Due: 11:59pm Sunday 9-Nov to the common repository on GitHub*

**1. Team Name:**  Hodor

**2. Team Members**: Dana Schoppers Joshua Koike

Jason Tom Trevor Fasulo

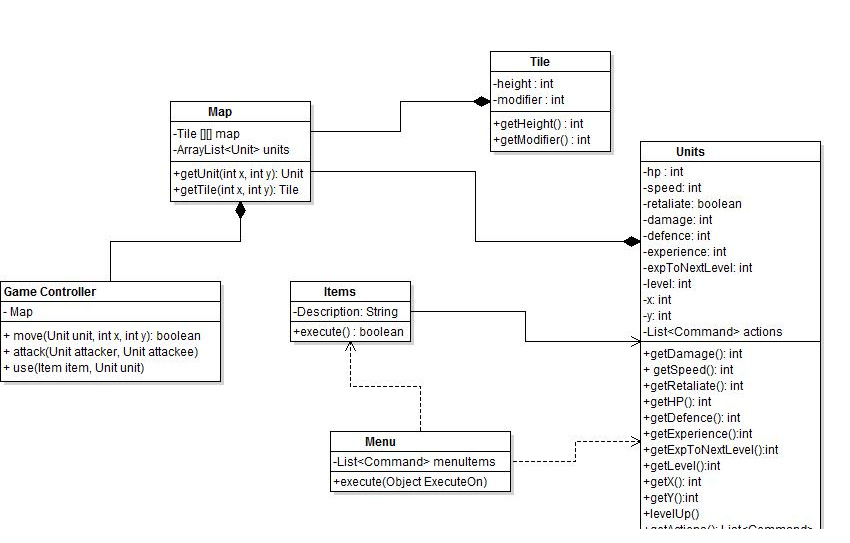
**3. Candidate Objects or Class Hierarchies**

List the seven most important objects, or the name of a hierarchy, and the main responsibility of each

|  |  |
| --- | --- |
| **Candidate Object** | **Single Responsibility in 1 or 2 sentences** |
| 1. Map | Keeps track of the terrain and where the units are. A Map is comprised of tiles and units. |
| 1. Tile | Responsible for knowing the type of terrain (ex. Modifying movement speed) |
| 1. Units | The playable characters of the game. Keeps track of each unit’s stats, items, and location. |
| 1. Menus | Game menu: Instantiates the map and tiles to start a new game.  Context menu: Responsible for handling the many different actions an object can perform. |
| 1. Items | Responsible for changing a unit’s stats. |
| 1. Game Controller | Responsible for controlling the movement around the map as well as controlling the actions of the units. |
| 1. Game View | Displays the map, tiles, and units for the user to see. |

*These Class and Sequence Diagrams may be written by hand and scanned or drawn with a UML editor such as Violet* [*http://sourceforge.net/projects/violet/files/violetumleditor/*](http://sourceforge.net/projects/violet/files/violetumleditor/) *and / or the sequence diagram editor or* [*https://www.websequencediagrams.com/#*](https://www.websequencediagrams.com/)

4. Class Diagram: Your team UML Class Diagram must show at least all of your candidate objects from above. Show any relationships between them the classes such as inheritance or interface implementation. Draw general associations such as dependency or aggregation. Label some to help explain things. Add any multiplicity adornments that seem appropriate. Use notes to explain things if you feel it will help. Each UML class must show the class name. For full credit, each class must have an average of at least one attribute per class. There must be an average of at least 2.0 methods per class, which may be implicit (no need to repeat methods) if the class implements a Java interface with methods shown there.



**5. Sequence Diagram:** Your team UML Sequence Diagram should show the most important scenario you can think of. Your sequence diagram should show most of your objects from above and how they communicate with each other.

http://www.websequencediagrams.com/cgi-bin/cdraw?lz=dGl0bGUgTW92ZW1lbnQgU2VxdWVuY2UKCk1hcC0-VW5pdDogc2VsZWN0KCkKVW5pdC0-Q29tbWFuZCBsaXN0OiBnZXRBY3Rpb25zKCkKAA8MACAJADkITW92ZQAaCi0-R2FtZUNvbnRyb2xsZXI6IGV4ZWN0dXRlKCkKAA0OLT5NYXA6IHVwZGEAGQUK&s=napkin