**C Sc 335 Analysis and Design Artifacts for the Final Project, Fall 2015**

**1. Team Name:**  A.R.E.S.

**2. Team Members**: Paul Hein Ryan Wong

Mark Hadley Mingcheng Dai

**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 Colonist | The colonist hierarchy stores all information and methods necessary for referencing, moving, and determining the status of a colonist object |
| 2 Tile | The tile hierarchy houses all the necessary information for determining the status of a tile and houses methods necessary for updating that status as well as returning tile information |
| 3 Building | The building hierarchy includes all information about the specific building and houses methods which are necessary to perform actions on that building such as to be built, to collect resources, or to gain a special resource, or to remove |
| 4 AresGUI | The AresGUI class loads the Map class along with any necessary controls that will be visible at all times during the game, this houses all event listeners which will be used to facilitate gameplay |
| 5 Map | The map class builds and displays the game map on the game panel so the player has an active view of what is currently happening in game. |
| 6 Selector | The selector hierarchy allows for specific selectors to be implemented that allow the user to make certain decisions in separate JOptionPanes/JPanels. This hierarchy is used when any button from AresGUI is pressed in order to deal with user interaction. |
| 7 Manager | The manager hirearchy is the model code behind user interaction that runs all of the logic for any user input for the decisions that need to be made in-game about either movement, tasks, buildings, etc. |

*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.