**C Sc 335 Analysis and Design Artifacts for Final Project**

**1. Team Name:**  \_\_\_\_\_\_\_\_404 Error\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2. Team Members**: \_ \_\_\_Maxwell Faridian\_\_\_\_\_\_\_\_\_\_ Katherine Walters\_\_\_\_\_\_\_\_\_\_

\_\_\_Jonathon Davis\_\_\_\_\_\_\_\_\_\_\_\_ \_Ethan Ward\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3. Candidate Objects or Class Hierarchies**

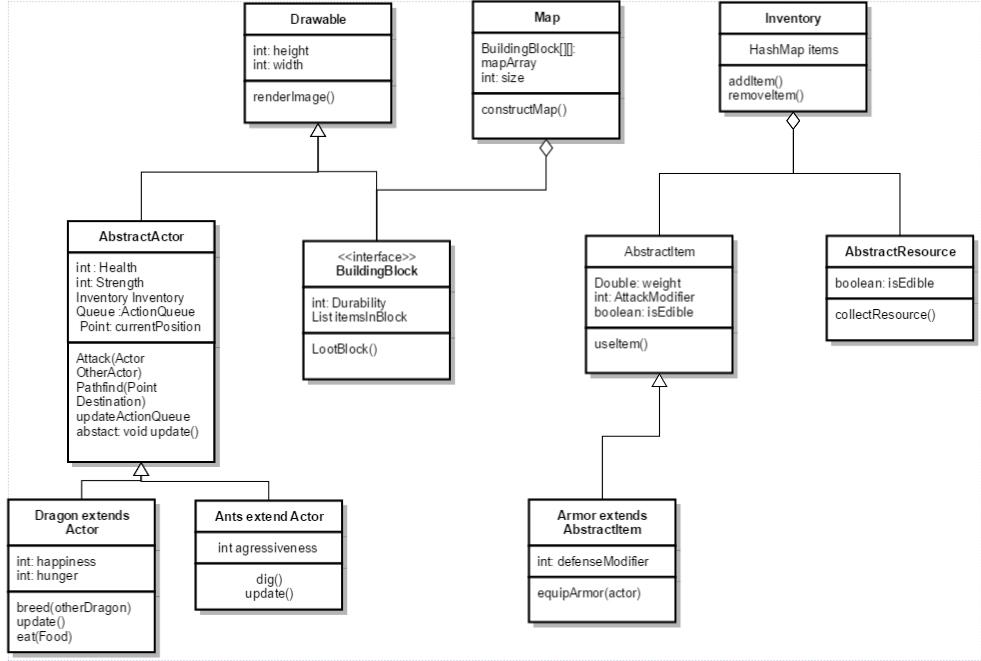
List the most important objects, or the name of an inheritance hierarchy, and the main responsibility.

|  |  |
| --- | --- |
| **Candidate Object** | **Single Responsibility in 1 or 2 sentences** |
| AbstractActor | Define basic functionality of npcs  Attack Actor, Pathfinding, ActionQueue, Health, Strength, Inventory  Armor slot |
| Inventory | Collection stores AbstractItems |
| BuildingBlock | Interface, player built buildings, obstacles like hard rock, ant tunnels  Durability, Loot/salvage, |
| AbstractItem | Defines basic functionality of an item: weight, attack modifier, defense modifier, edible, durability |
| AbstractResource | Interface for resources, wood, metal, stone, food |
| Map | “2d array” of BuildingBlocks  Userdefined small, medium, large |
| Dragon | Extends Actors, player controlled characters, breed, do jobs, fight, sleep, eat, needs |
| Ants | Dig, Kill, More attacks per population |
| Armor | Equipable in armor slot |
| Drawable | Able to be rendered |

**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.

*This class diagram may be written by hand and scanned or drawn with any*

*UML editor or drawing program*



**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.

*This Sequence Diagram may be written by hand and scanned or drawn with any program or sequence diagram editor such as* [https://www.websequencediagrams.com/#](https://www.websequencediagrams.com/)

