**CSc 335 Analysis and Design Artifacts for Final Project**

*This must be in a private Github repo in a directory named documents*

*with your project manager added as a collaborator*

**1. Team Name:**  Team Weird Horses

**2. Team Members**: David Lamparter Kyle DeTar

Kyle Grady Brett Cohen

**3. Candidate Objects or Class Hierarchies**

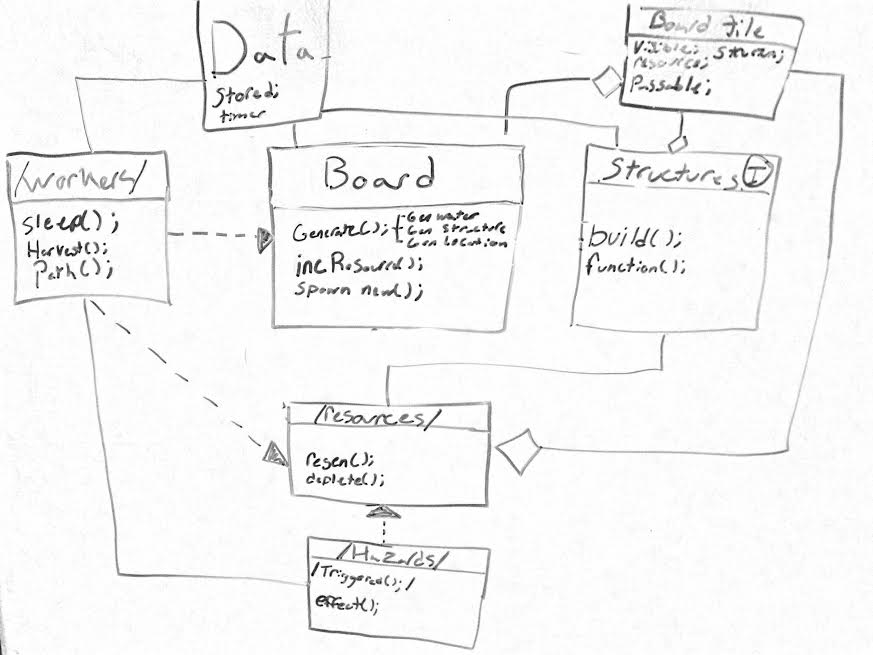
List the most important objects, or the name of an inheritance hierarchy, and the main responsibility.

|  |  |
| --- | --- |
| **Candidate Object** | **Single Responsibility in 1 or 2 sentences** |
| Workers | The worker will harvest resources to sustain the community. They might have preferences to what job they want to do, require sleep, and have a community moral. |
| Resources | Abstract class representing food, wood, and stone. We might use this class for community requirements of the resources, but that’s up for debate still. |
| Board | Randomly generates the map for the player to play on and contains all elements that exist in the map. Contains map tiles which hold each element of the game. |
| Structures | Interface class representing Town Center, Houses, Storages, and Churches (perhaps more to come as well) |
| Hazards | Seasons (require more resources during winter), poison bushes, discover (fall off a cliff, snake bite) |
| Map | Controls the GUI for the game based off the model (board). |
| Tools | Enum representing tools for workers to use |
| Tile | Represents one “space.” Utilized by Board and Game |
|  |  |
|  |  |

*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.

**This is our current UML model. We’ve added Sean to the google doc we have that has many more flushed out ideas that build from this model.**



**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.

**We determined the most important event to be when a resource is running low. Our diagram shows the sequence of events that will happen between our objects in this situation. Here’s a link to our sequence diagram below!**

**http://www.websequencediagrams.com/cgi-bin/cdraw?lz=dGl0bGUgR2F0aGVyIFJlc291cmNlIENvbW1hbmQgSXNzdWVkCgpQbGF5ZXItPk1hcDoADRBNYXAtPkJvYXJkOiBSZWxheXMgd2hhdCByAEUId2FzIGNsaWNrZWQKACMFLT5Xb3JrZXI6IEdpdmVzIHRhcmdldCBsb2NhdGlvbgoAGAYtPgCBBwg6IFRyYXZlbCB0byBhbmQgaGFydmVzAFgKCgCBMAgtPkhhegCBBAVJcyBpdCBkYW5nZXJvdXMKABIGAEcMWUVTAC0LAIEJCDAAgS4JcyBjb2xsZWN0ZWQAgQgJAIFfBwCBOAYgZGllcwCBSQgAghQFABEHcmVtb3ZlZCBmcm9tIGdhbWUAgh0GAII-BjogAIJGBiBubyBsb25nZXIgYWJsZSB0byB1c2UgdwCCFAUAgRwTTk8AgRsTMgCBAyxyZXR1cm5zIHRvIHN0b3JhZ2UAgSkUcG9zaXRpb24gdXBkYXQAg1cIAIE0CACBWQlhZHkgZm9yIGFubwCENQVvcmRlcg&s=napkin**