APCS Final Project grading rubric. /50 Name(s)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Complexity: 0 2 4 6 8 10

Proposal 0 2 4 6 8 10

Testing 0 1 2 3 4 5

Documentation 0 1 2 3 4 5

Code Execution 0 2 4 6 8 10

Participation 0 1 2 3 4 5

Self Evaluation 0 1 2 3 4 5

**Complexity**

✔ Multiple Classes

✔ Inheritance

✔ Data Structures

✔ Polymorphism

Sorting

Sequencing

Selection

✔ Abstraction

✔ Iteration

Recursion

The code demonstrates the complexity of the Java programming language and includes the concepts of object-oriented programming taught in this course. In addition to the obvious use of multiple classes (REQUIRED), these might include the use of data structures, inheritance, polymorphism, sorting, sequencing, selection, abstraction, iteration and recursion. Note that your project MUST include some of these ideas. It will most likely NOT include all of them. The more concepts you use and the scale to which they are used together will determine the overall complexity.

**Proposal**

✔ Purpose of the Project

✔ Explanation of Implementation

✔ Project Roles

The proposal details the purpose of your project (what will it do? who is it for? what makes it unique if it is based on an existing game/program?). The proposal should also include a thorough explanation of how the program will be implemented. This may be done in written form or by using UML diagrams. Finally, this document should provide a *preliminary* breakdown of which team member will be responsible for each major component of your design (no update required if changes to assignment occur during the project).

**Testing**

…

All methods in classes (not simple methods like getters/setters) are tested. Unit tests or runners are acceptable. Tests should cover a majority of situations including valid and invalid conditions.

**Documentation**

✔ OK so far!

Key elements such as description, parameters (inputs) and return values are specified at the beginning of each class describing its’ purpose. Each method that is not self-explanatory has documentation.

**Execution**

✔ OK so far!

The program works (does not crash) and can be used for the purpose stated in the proposal.

**Participation**

✔ OK so far!

When you are in the classroom, you are utilizing the time to work on your project. You actively participate with your group members on the project. Yes, this is subjective – so don't let me catch you not working….

**Self Evaluation**

…

You are capable of describing your role in your group’s project and have a good understanding of the concepts used in the program.