University of Washington Outreach Program

This project replaces that given under assignment 8. Please see the deliverables listed in the default project description on the supplementary web page even if you have chosen your own project.

For this assignment, we will focus on defining and writing the Requirements Specification, the operating environment, and the Test Plan for the system.

Requirements Specification

From the default project description,

Requirements definition is the process of identifying and understanding what the needs of all interested parties are then documenting these needs as written definitions and descriptions. The focus is on *what* problem the system has to solve. The emphasis is on the world in which the system will operate not on the system itself.

The purpose of the *Requirements Specification* step is to capture and express a purely external view of the system. We refer to this view as the public interface of the system. We identify *what* needs to be done starting from the user's needs and requirements. Non-functional specifications also have to be added. We use these to explain constraints such as performance and timing constraints, dependability constraints, as well as cost, implementation and manufacturing constraints.

Much of this information can / must be captured through UML use cases – remember, graphics and text (including specifications and exceptions). These are part of your deliverables.

For this portion, the deliverables are as follows...

- 1. A high level description of the system. This should be no more than one paragraph. This is in the form of an abstract or executive summary.
- 2. A general and more detailed description of the system. This should include, for the case of default project, the rules and commands by which the game is played and identify how one wins or loses. For those who have chosen their own project, this should include a similar criteria for the user's interaction for your system.
- 3. The initial set of use case diagrams. Remember that these must include the graphics and textual components and that the textual components must include the description of the use case as well as any exceptions.
- 4. The initial functional decomposition of the system. Remember that this is a static structure that will eventually lead to the system architecture. It is not an activity diagram, sequence diagram, flow chart, or state diagram. It is capturing the major functions or jobs that comprise the system.

Environment Considerations

These are examining the environment in which your program is intended to be running.

- 1. What are the speed requirements?
- 2. What are the memory requirements?
- 3. What are the host requirements?
- 4. What are the display requirements?

Test Requirements

Test Plan. The Test Plan identifies *what* tests need to be carried out based upon the original requirements specification. It describes in general terms the following information:

- *What* is to be tested?
- The testing order within each type of test.
- Assumptions made.
- Algorithms that may be used.

Eventually, we will base the test specifications and test cases on this plan. It it important to remember that the Test Plan must be tied to the original specification.