Software Engineering

Software Requirements Specification

For

Ant-World

Prepared by: Eleanor Shakeshaft

Version Draft

08

**Fall**

Group 11

Requirements Specification

Version Draft

6th March 2012

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Robert Johnson Date

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mark Merriman Date

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Oliver McCarthy Date

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Andreas Nicolaou Date

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mark Purser Date

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Joanne Robert Date

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eleanor Shakeshaft Date

Approved by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ben Watt Date

**Revision Notice.**

|  |  |  |  |
| --- | --- | --- | --- |
| Document Revision History | | | |
| Revision | Date | Section | Description of Changes |
| Initial Review (Draft) | 2nd March 2012 |  |  |
| Initial Release (1.0) |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents.

1. Introduction. 5

1.1 Purpose. 5

1.2 Document Overview. 5

1.3 Intended Audience. 5

1.4 Project Scope. 5

*1.4.1* *Out of Scope* 5

1.5 References. 6

2. Overall Description. 6

2.1 Product Features. 6

2.2 User Classes and Characteristics. 6

2.3 Operating Environment. 6

2.4 Design and Implementation Constraints. 6

2.5 User Documentation. 6

3. System Requirements 7

3.1 Req 1 – User uploads ant brain. 7

3.2 Req 2 – Software checks ant-brain syntax. 7

3.3 Req 3 – Matches 2 users to play 7

3.4 Req 4 – Create ant-world. 7

3.5 Req 5 – Software checks ant-world conditions. 8

3.6 Req 6 – Run simulation. 8

3.7 Req 7 – Ant characteristics. 8

3.8 Req 8 – Log game statistics. 8

3.9 Req 9 – Calculate winner. 9

3.10 Req 10 – Send copy of simulation to user servers. 9

3.11 Req 11 – User to have control of viewing simulation. 9

3.12 Req 12 – Tournament. 9

3.13 Req 13 – Produce Highest Scores for website. 9

3.14 Req 14 – Upload Highest Scores to website. 10

3.15 Req 15 – TBD. 10

3.16 Req 16 – TBD. 10

4. External Interface Requirements. 10

4.1 User Interfaces. 10

4.2 Hardware Interfaces. 10

4.3 Software & Communications Interfaces. 11

5. Other Non functional Requirements. 11

5.1 Performance Requirements 11

5.2 Security Requirements 11

5.3 Software Quality Attributes 11

6. Analysis Models 11

6.1 Case Modelling 11

6.2 Sequence Diagram 11

6.3 Class Diagram 11

6.4 State Diagram 11

1. Introduction.
   1. Purpose.

This Software Requirements Specification (SRS) indentifies the requirements for the Ant-World Game (AWG) for the Software Engineering Module.

* 1. Document Overview.

Section 2 is an overview of the descriptions of requirements for the AWG, and the requirements are categorised and defined in section 3. Sections 4 and 5 contain general information to aid in the understanding of this specification.

* 1. Intended Audience.

This document is primarily intended for the design developers, but also as a reference for the project managers, programmers, QA testers, and documentation writers. Read over the overview sections and proceed through the sections that are most pertinent to your requirements.

* 1. Project Scope.

Our customer is a games designer who has an idea for a new computer game, and wants us to provide a solid implementation. The game is a competitive two-player strategy game where the players design and upload an ant-brain to pit against each other. The ants are placed into two ant colonies within a randomly-generated world simulation which also contains two anthills, some food sources, and several obstacles. The ant world will then simulate the behaviour of both kinds of ants; they must explore the world, find food and bring it back to their anthill. The winner is the anthill which has the most food at the end of 300,000 rounds. The highest scores and players will be put on a website along with statistics.

* + 1. *Out of Scope*

The software will not allow user profiles, it will not store ant-brain codes after the game(s) have completed.

* 1. References.

Customer Requirements: <http://www.sussex.ac.uk/Users/mfb21/se/project/require.html>

Quality Manual: <http://www.sussex.ac.uk/Users/mfb21/se/project/quality.html>

Configuration Management: <http://www.sussex.ac.uk/Users/mfb21/se/project/config.html>

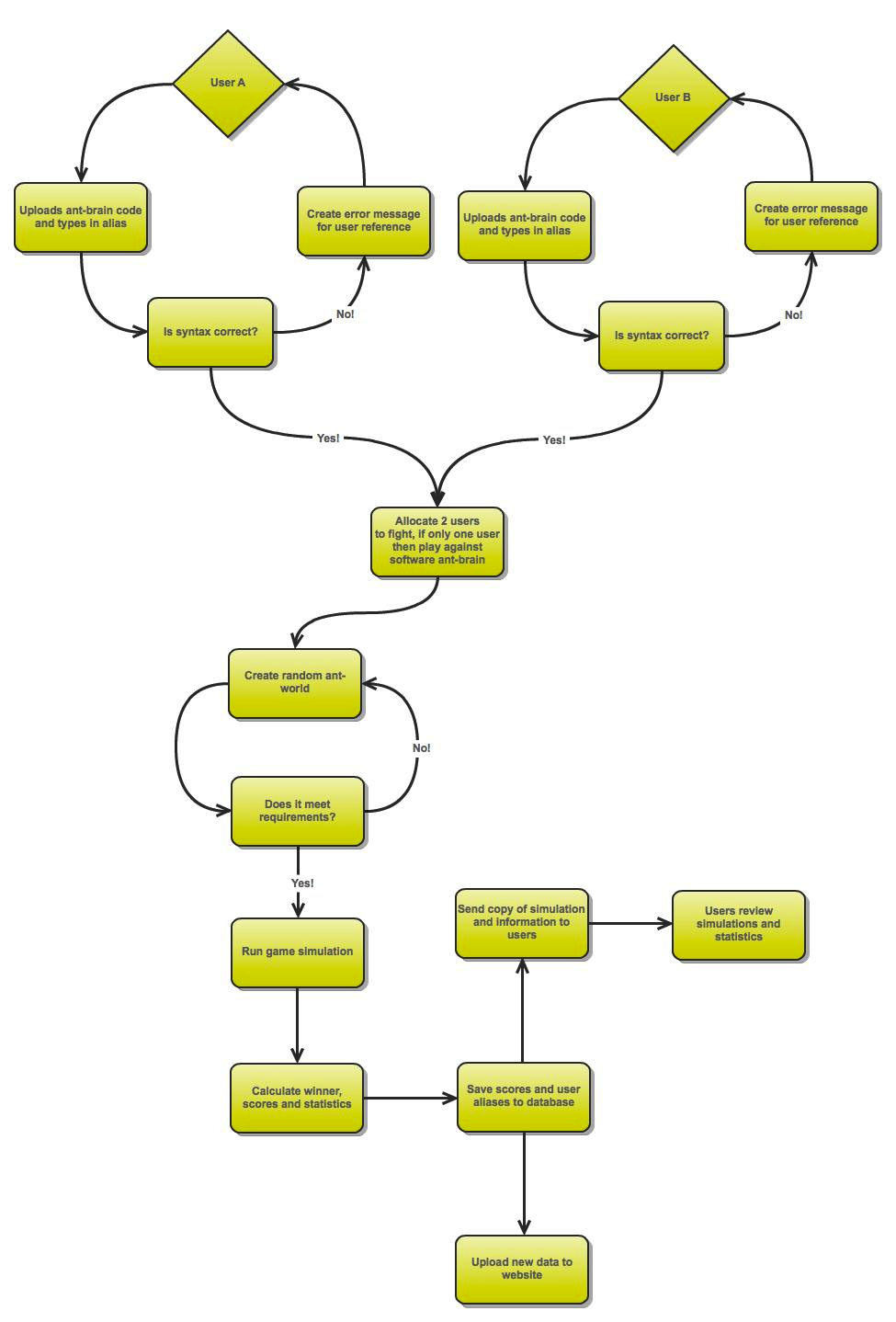
Project Website: <http://dl.dropbox.com/u/12957105/Website/home.html>

Project Plan: <http://dl.dropbox.com/u/12957105/Website/Project_Plan.pdf>

Google AI Challenge: <http://aichallenge.org/>

1. Overall Description.
   1. Product Features.

High-level flowchart of simple verses match:



High-level flowchart of tournament:

* 1. User Classes and Characteristics.

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the favored user classes from those who are less important to satisfy.>

* 1. Operating Environment.

The software will be designed to operate on PC’s only and be will be cross-platform. The highest scores will be displayed on a website.

Will user’s need to download software? Upload vis website? Client will need servcer, datatbase etc

* 1. Design and Implementation Constraints.

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

* 1. User Documentation.

The client will be given a document, along with the software, which will contain two sections:

Installation guide: This is written for the system administrator (who can be assumed to have reasonable computing knowledge, but knows nothing about gaming in general this software in particular) installing the program. It must be possible to install the software and get it to run from the installation guide only.

User Manual:  This documentation is to be read by users of the program (who cannot be assumed to have any particular computing knowledge), and shall describe the user interface and functionality of the system, in terms the user understands.

It will also detail the licensing agreement for the software.

The client will also be given a website which will hold the highest scores and statistics.

1. System Requirements
   1. Req 1 – User uploads ant brain.

***Description***

User opens software, fills in an alias for high-score and uploads their ant brain code.

***Stimulus/Response Sequences***

Step 1: User opens software / website?

Step 2: User starts new session & fills in alias.

Step 2: User loads ant-brain code document and submits.

***Functional Requirements***

Alias must be letters only and be less than 10 characters.

File uploaded must be in correct format = .txt file?

* 1. Req 2 – Software checks ant-brain syntax.

***Description***

Once ant-brain code is uploaded the software must check if it is syntactically correct.

***Stimulus/Response Sequences***

Step 1: User submits ant-brain (Req 1).

Step 2: Software checks the syntax of the code.

Step 3: Software responds to user whether syntax is correct or not.

***Functional Requirements***

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

* 1. Req 3 – Matches 2 users to play

***Description***

The software must allocate 2 corroborated users to play.

***Stimulus/Response Sequences***

***Functional Requirements***

* 1. Req 4 – Create ant-world.

***Description and Priority***

The software must generate a random ant-world for the game to take place.

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

The dimensions of the world are always 150 × 150 cells.

The cells on the perimeter are always rocky.

Every world contains exactly the same elements, of particular shapes: 2 anthills, 14 rocks, and 11 blobs of food. The anthills, in particular, are hexagons with sides of length 7. Also, a food blob is always a 5-by-5 rectangle, with each cell containing 5 food particles.

The positions and orientations of the elements are chosen randomly, subject to the constraint that there is always at least one empty cell between non-food elements. Also, no elements overlap. (The anthill elements are 6-ways-symmetric, so their orientation actually does not matter. All ants are initially facing in direction 0.)

* 1. Req 5 – Software checks ant-world conditions.

***Description and Priority***

Once ant-world is created the software must check if it is correct as per the specifications.

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

<Itemize

* 1. Req 6 – Run simulation.

***Description and Priority***

The game will commense and the entire simualtion will run through.

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

Win by having the most food in its anthill at the end of 300,000 rounds.

* 1. Req 7 – Ant characteristics.

***Description and Priority***

<Provide a short

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

Can communicate or leave trails by means of chemical markers in order to find their way back to the anthill and store their food.

Can sense (with limited capabilities), but not modify, the markers of the other species.

Different ant species can attack each other by surrounding them.

When the ant dies as a result of an attack becomes food.

* 1. Req 8 – Log game statistics.

***Description and Priority***

<Provide a short

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

<Itemize

* 1. Req 9 – Calculate winner.

***Description and Priority***

<Provide a short

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

Only the number of food particles currently in the anthill cells of each colour are counted, food being carried by an ant does not count, even if it is standing on its own anthill

* 1. Req 10 – Send copy of simulation to user servers.

***Description and Priority***

<Provide a short

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

<Itemize

* 1. Req 11 – User to have control of viewing simulation.

***Description and Priority***

<Provide a short

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

<Itemize

* 1. Req 12 – Tournament.

***Description and Priority***

<Provide a short

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

Be able to play tournaments, where an arbitrary number of players can upload ant-brains, who are all paired up to play against each other. They will fight twice on each contest world, each being red and black once. The winner gains 2 point, if it is a draw both players get 1 point. The overall tournament winner is the ant brain that wins the most individual games or has the biggest amount of food in its anthill.

* 1. Req 13 – Produce Highest Scores for website.

***Description and Priority***

<Provide a short

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

<Itemize

* 1. Req 14 – Upload Highest Scores to website.

***Description and Priority***

<Provide a short

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

<Itemize

* 1. Req 15 – TBD.

***Description and Priority***

<Provide a short

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

<Itemize

* 1. Req 16 – TBD.

***Description and Priority***

<Provide a short

***Stimulus/Response Sequences***

<List the

***Functional Requirements***

<Itemize

1. External Interface Requirements.
   1. User Interfaces.

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

* 1. Hardware Interfaces.

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

* 1. Software & Communications Interfaces.

The software must communicate with the database which will store the statistics and scores, the database will in turn communicate with the website so that the high scores can be updated. The software must also communicate with the user servers so that the simulations can be sent.

1. Other Non functional Requirements.
   1. Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

* 1. Security Requirements

Users will not have log-ins so no data is stored or kept about users. Only an alias will be input and stored for the high scores.

* 1. Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

1. Analysis Models
   1. Case Modelling
   2. Sequence Diagram
   3. Class Diagram
   4. State Diagram

Analysis Model.

This section describes the desired behaviour of the system.

The main abstractions required by the problem domain, and the classes and objects which will be used to represent these abstractions, should be identified and described here, together with their roles, responsibilities and collaborations.

The analysis model is given as a collection of scenarios illustrating each function point of the system or subsystem under consideration.

Scenarios may be described using CRC cards, object and class diagrams in UML notation, and/or finite state machines as appropriate. Scenarios should be divided into primary - those that pertain to the key behaviour of the system - and secondary - behaviour pertaining to exceptional circumstances.