Requirements Specification

Software Engineering Group 6

12/3/2012: Requirements Specification, v1.0

March 2012 - Second Deliverable

Contents: **Page no:**

Introduction

This document contains the description of requirements as outlined by the customer. This information is structured as a set of scenarios and use cases to demonstrate how the user will interact with the program and the anticipated behaviour of the system. These are split into primary scenarios, which are inevitable for the gameplay, and the secondary scenarios which are optional actions. CRC cards were used to illustrate the collaborations and responsibilities of the proposed classes and class diagram was created for better understanding of how these will link together.

The players will have an option of playing the game in single mode, where a single user will be able to play against other player, or the tournament mode where multiple number of players will engage in a tournament, where a number of matches will be played until one brain is victorious. The players will have an opportunity to design their own worlds and ant brains, but the option of randomly generating worlds or using already made brains will be available. The program will have to check if the brains and the world supplied by the player are syntactically correct before the game can be started. Also the worlds generated by the system must be well structured and obey the rules of the game. The game will provide the examples of a well-structured components, to give the player understanding how they should look like. Also, the game will use graphical interface to visualise the world and the behaviour of the ants which will give the game dynamic and attractive look.

This documentation will serve the design team as a guide to build the high-level design specification of the project.

Customer Requirements

1. Program must check if the brain supplied by the player is syntactically well formed.

2. Program must check if the world supplied by the player is syntactically well formed.

3. Program must visualise and display given world.

4. Program must be able to generate random and well-formed ant world.

- Program must be able to randomly generate a world that will follow the rules specified in requirement 2.

5. Program must allow two players to play the game.

- Program allows two players to upload brains that they have created.
- Players must be able to upload previously created world or choose randomly generated world.
- Program must display information which species of ants is controlled by which player.
- Program will assess the final scores and determine the winner of the match.

6. Program must allow arbitrary number of players to compete in a tournament.

7. The software engineering team will supply their version of the brain following the rules specified in point 1.

8. The program will output appropriate sound effects to improve user experience.

Use Cases

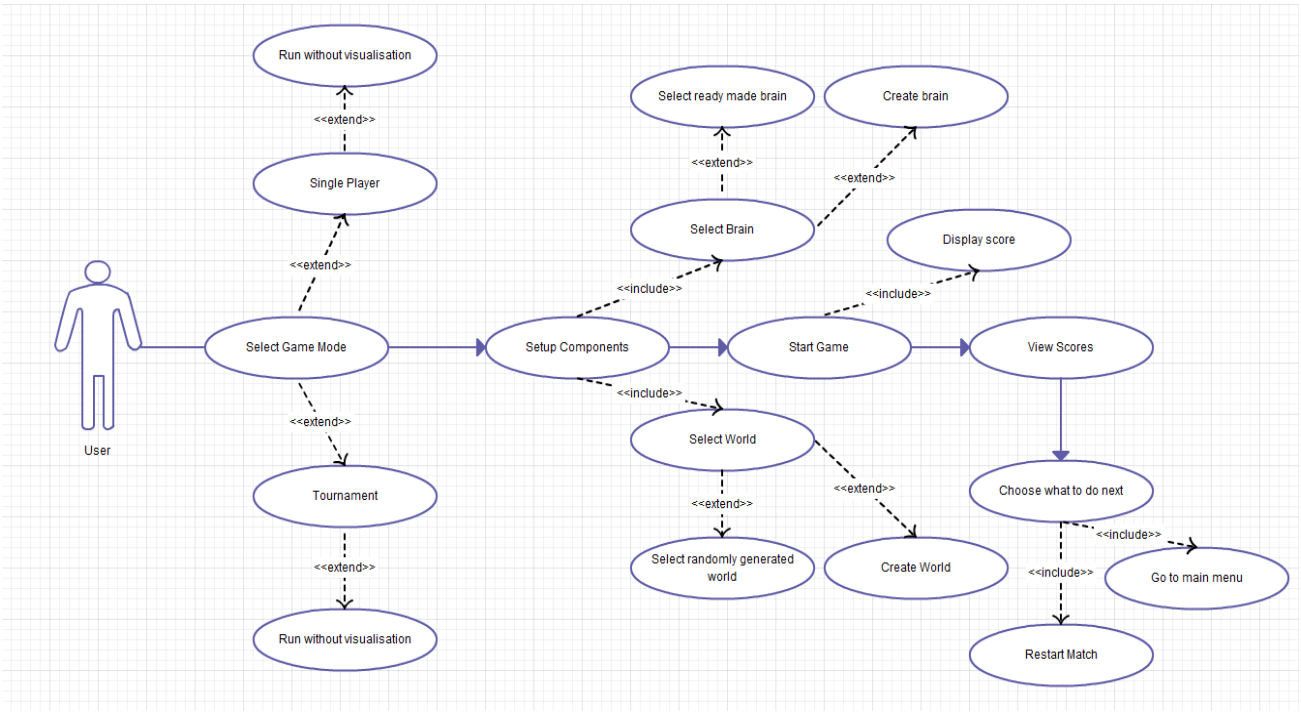
1. User interaction

1.1 Use case hierarchy

Note: each line represents a use case. Lines beginning with the bar symbol (|) which appear at the same indentation level are alternative paths. Numbered lines must be completed in the order specified, while lines beginning with a lower-case alpha character which appear at the same indentation level can be completed in any order.

Decide which game mode to run
- | Run Single Match
  - **1** Setup Game Components
    - **a** Choose Brain { 2 }
      - | Select ready-made brain.
      - | Upload/create custom brain
    - **b** Choose World
      - | Generate random world
      - | Upload/create custom world
  - **2** Begin Match
    - | Display current score
  - **3** Choose what to do next
- | Run Tournament
  - **1** Choose Brain { n | n >= 2 }
    - | Select ready-made brain.
    - | Upload/create custom brain
  - **2** Begin Tournament
  - **3** Choose what to do next

1.2 Use Case Diagram



1.3 User Interaction Use Case Descriptions

Use Case: **Decide which game mode to run**
Actor: User
Precondition: Application Launched
Description:

Upon launching the game, the user is presented with the main menu which has two options: Single Match and Tournament Mode. One of these options must be chosen to continue.

Post Condition: The user is at a setup screen for one of the modes

---

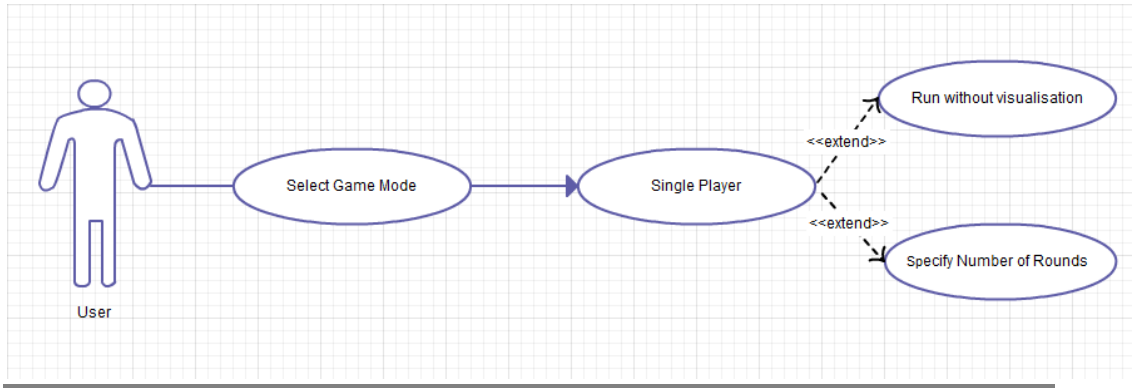
Use Case: **Run Single Match**
Actor: User
Precondition: Single Match option chosen from main menu
Description:

The Single Match setup screen requires that the user choose two brains and a world to be used (see the **Setup Game Components** use case).

There are two optional actions the user can take: deciding to run the game without visualisation, and specifying the number of rounds for which the game should run (the default is 300,000).

Once the user has chosen their desired game components, they can choose to begin the simulation (see the **Begin Match** use case).

Upon successful completion of a match, the user is presented with the result of the match and statistics gathered during its execution. They can then choose to take some further action (see the **Choose what to do next** use case).

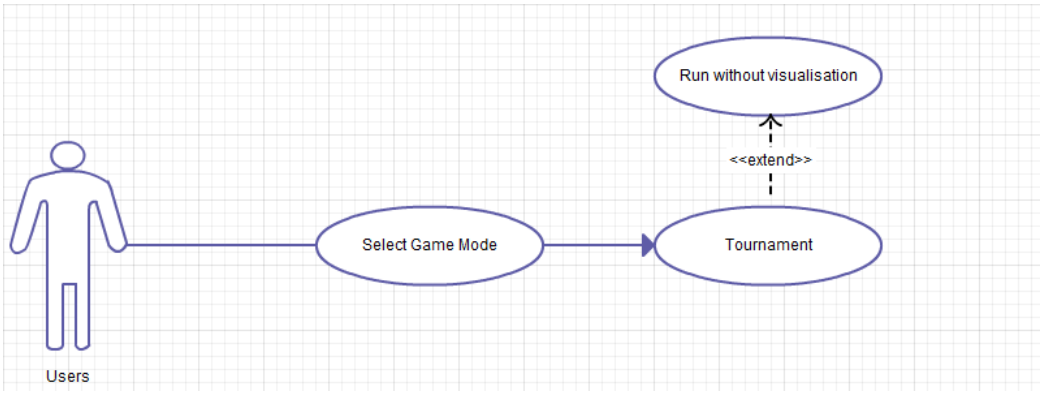


Use Case: **Select Tournament**
Actor: User
Precondition: Tournament option chosen from menu
Description:

In the tournament mode, it is possible for multiple users to upload the brains that they have created and put them in a match against each other until one is victorious. Just like in a single match mode, the users will be able to run the simulation without visualisation, but they won't be able to specify the number of rounds for each match (all matches will run for 300 000 rounds).

The game will randomly generate a group of worlds that will be used for the tournament. For example if the group will be set to 5, the game will generate 5 random worlds and for each match different worlds will be used. The users won't have an option of adding their own created worlds into the group.

After each match, a leader board screen will be displayed, specifying how many points each player has so far. The player that has won the match will be awarded 2 points and the player that has lost won't acquire any points. If the match was a draw, both players will be awarded 1 point each. At the end of the tournament, the winner will be the one with the most amount of points. If two top players at the end of the game have the same amount of points, a deciding match will be played between those two and the player who wins the deciding match will win the whole tournament. After the tournament is completed the users can take further action (see the **Choose what to do next** use case).



Use case: **Run without visualisation**

Actor: User

Precondition: User has selected one of two possible game modes

Description:

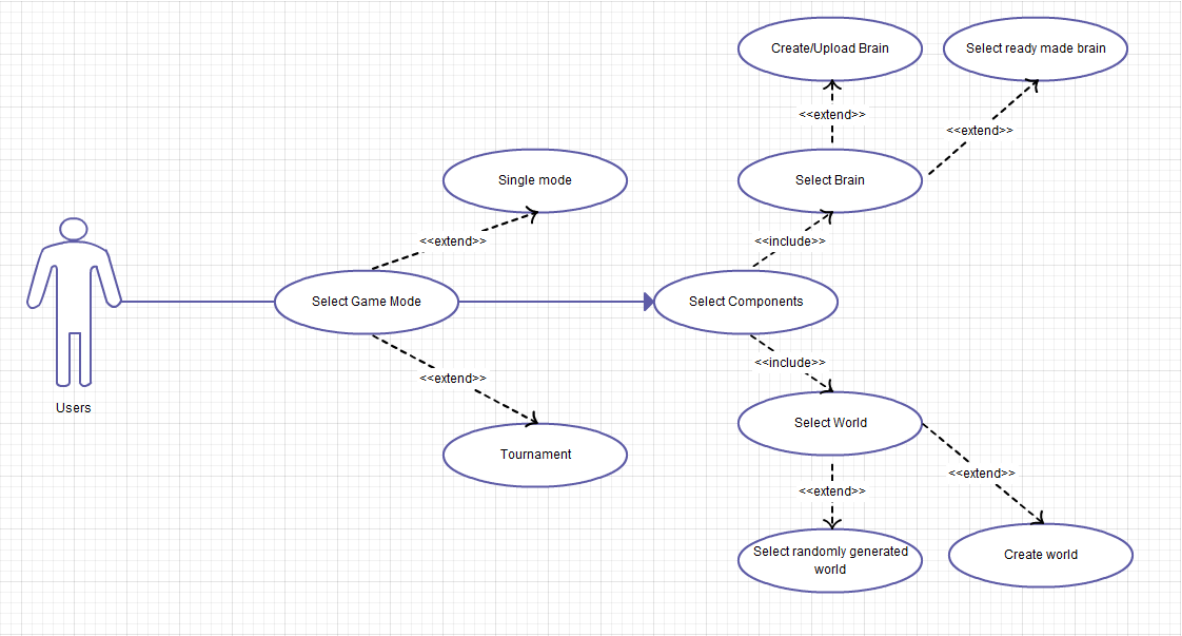
After selecting a game mode, the user will be able to choose an option of running the simulation without visualisation. If that option is selected, no graphical visualisation of the game will be displayed, drastically decreasing the time of the process. The users will only see the final score of the match.

Use Case: **Setup Game Components**
Actor: User
Precondition: Single Match or Tournament option chosen from main menu
Description:

The user must choose two ant brains and one world to use for the match (see the **Choose Brain** and **Choose World** use cases). Choices are not final and can be modified at any point before the match begins.

In the tournament mode, the users will have to choose a minimum of 2 ant brains, however the maximum depends on the number of players that take part in the tournament.

Post Condition: Two valid (i.e. syntax-checked) brains, and one valid world are available to be used in the match.



Use Case: **Choose Brains**
Actor: User
Precondition: The user is at one of the setup screens
Description:

There are two main methods by which the user can select an ant brain to use. The game will have a collection of built-in brains which the user can pick from, or the user can upload/create their own brain (see the **Select ready-made brain** and **Upload/create custom brain** use cases). In addition, there is the ability to combine these two methods by first selecting a ready-made brain and then customising it by modifying its source code.

Post Condition: A valid ant brain is available to be used in the match/tournament.

Use Case: **Select ready-made brain**
Actor: User
Precondition: The user is at one of the setup screens
Description:

By clicking a button or some other method, the user is presented with a list of at least five ready-made brains. The brains will be named according to their intelligence and/or primary design feature (e.g. killing the other team). The user selects one of these brains by clicking on its name or some other intuitive method.

Post Condition: A named brain appears as selected on the match/tournament setup screen.

Use Case: **Upload/Create custom brain**
Actor: User
Precondition: The user is at one of the setup screens
Description:

If user chooses to create/upload custom brain, he/she will be asked to provide a source code for the brain that will be used. The user can write values in the text file and then copy those values into the game. Then the source code will be checked for any syntax errors and if is correct, it will be uploaded into the game. Once all the brains have been uploaded the user can move on to uploading worlds (see the **Choose World** use case).

Use Case: **Choose Worlds**

Actor: User

Precondition: The user uploaded brains in the setup menu

Description:

The user has the option to create his own world or randomly generate one. Just like in the case of choosing brains, these two methods can be combined by creating random brains and then changing the source code.

In case of the tournament mode, the users will not be able to upload their own world, instead a group of random worlds will be created which later will be used in the game.

Use Case: **Upload/Create custom world**
Actor: User
Precondition: The user is at one of the setup screens
Description:

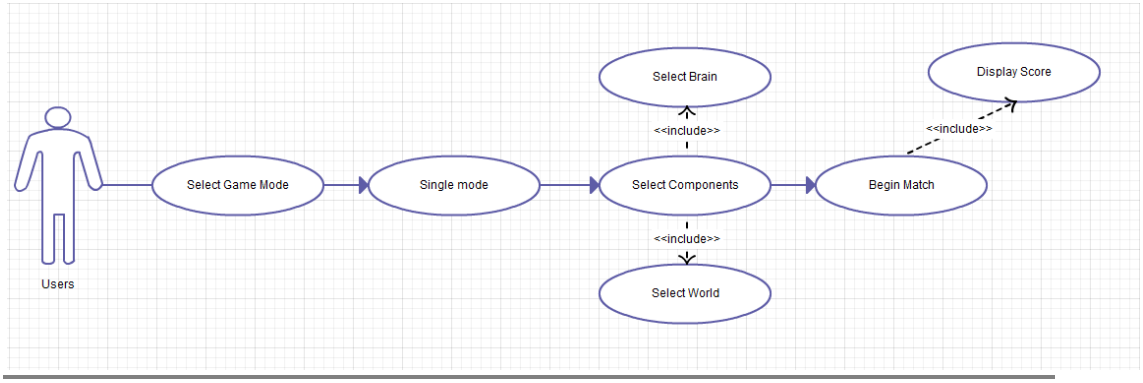
Just like in there case of the uploading custom brain, the user will be asked to provide the source code for the world. The user will be able to specify how many different components the world will contain (food, rocks etc.) and also the location of those elements.

Use Case: **Select randomly generated world**
Actor: User
Precondition: The user is at one of the setup screens
Description:

If the world is randomly generated, the user will be able to specify the number of components that the world will contain but not their location. In tournament mode, the whole world will be randomly generated (users will have no control over the number or the location of any components).

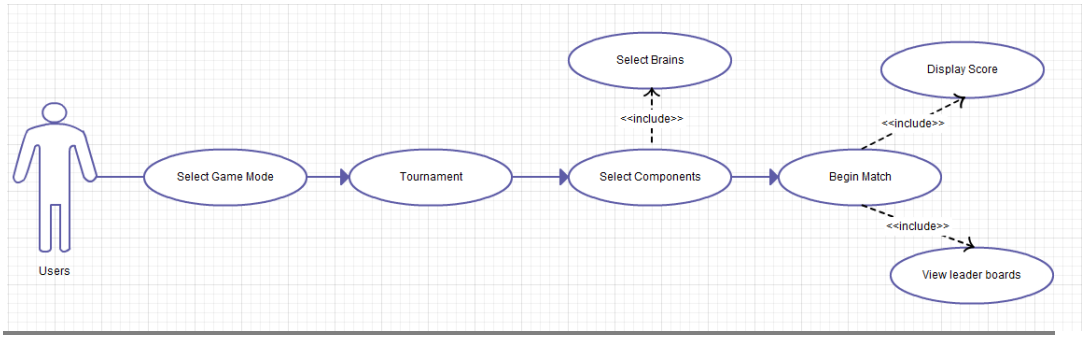
Use Case: **Begin Match**
Actor: User
Precondition: The used has selected all necessary components and clicked begin match
Description:

Once all the components were selected, the user is presented with graphical visualisation of the world, and from there he can observe the process of the game play. Also the user will be able to view the current scores. After the match is completed, the user will be presented with the final score screen. If the "run without visualisation" option was selected at the beginning of the game, the user will only see the final score screen.



Use Case: **Begin Tournament**
Actor: User
Precondition: The user has selected all necessary components and clicked begin match
Description:

At the beginning of the tournament, two first teams will be selected to face each other. The graphical visualisation of the world will be presented and the users will be able to observe the process of the game. The users will be also able to view the current score. At the end of the match the users will be presented with the final outcome of the match and points will be awarded to the winner. After that, the next two players will be selected and the process will continue until all players have played against each other. At the end the leader board will be displayed specifying which player has obtained the most points. From there the users will be able to take further action.



Use Case: **Choose what to do next**
Actor: User
Precondition: The user has seen the final outcome of the match/tournament

Description:

After the match/tournament is completed, the user will have an option to return to the main menu, restart the game using different brains but the same world will be used, using same brains on a different world or choose all new different components.

CRC Cards

AntWorld	
Responsibilities	Collaborators
• Contains ant objects • Contains food • Contains ant colonies • Keeps track of number of rounds	• Ant • AntHill • WorldCell • RandomGenerator

Ant	
Responsibilities	Collaborators
• Contain chemical marker • Contain brain • Collect food • Die • Avoid obstacles • Attack other species	• Brian Parser • Ant Brain

AntBrain	
Responsibilities	• Collaborators
• Holds instructions for ant behaviour	• Ant • Brain parser

WorldCells	
Responsibilities	Collaborators
Contain food Contain rock	• Ant world

Game	
Responsibilities	Collaborators
• Game start • Game statics • Allow multiple player to play	• Ant world • Tournament

Tournaments	
Responsibilities	Collaborators
• Allows arbitrary player to upload ant brain	• Ant brain

AntHill	
Responsibilities	Collaborators
• Sets number of ants of given species • Keeps count of food particles in the hills	• Ant world

BrainParser	
Responsibilities	Collaborators
• Check if supplied brain is correct/ has correct value	• Ant

GUI	
Responsibilities	Collaborators
• Visualise given ant world	• Ant world

RandomWorldGenerator	
Responsibilities	Collaborators
• Generate random ant world	• Ant world

WorldParser	
Responsibilities	Collaborators
• Checking syntax of the world • Checking that worlds conform to contest rules	• Ant World

Analysis Class Diagram: