|  |
| --- |
| Kingswood School |
| Survival of the fittest teaching aid |
| Comp4 project |

|  |
| --- |
| 01robinson |

Contents

[Section 1: Analysis 4](#_Toc381010694)

[The problem: identification and background 4](#_Toc381010695)

[Description of available tools 5](#_Toc381010696)

[Identification of prospective users 8](#_Toc381010697)

[Identification of user needs and acceptable limitations 8](#_Toc381010698)

[Data sources and destinations 9](#_Toc381010699)

[Data volumes 10](#_Toc381010700)

[Analysis Data Dictionary 10](#_Toc381010701)

[Structure and Data Flow Diagrams 10](#_Toc381010702)

[Entity-relationship (E-R) model 10](#_Toc381010703)

[Object analysis diagram 11](#_Toc381010704)

[Objectives for the proposed system 11](#_Toc381010705)

[Appraisal of potential solutions 12](#_Toc381010706)

[Justification of chosen solution 14](#_Toc381010707)

[Section 2: Design 15](#_Toc381010708)

[Overall system design 15](#_Toc381010709)

[Description of modular structure of system 18](#_Toc381010710)

[Definition of data requirements (Design Data Dictionary) 18](#_Toc381010711)

[Description of record structure 18](#_Toc381010712)

[File organisation and processing 18](#_Toc381010713)

[Validation required 18](#_Toc381010714)

[Identification of storage media 18](#_Toc381010715)

[Identification of suitable algorithms for data transformation, pseudo code of these algorithms 18](#_Toc381010716)

[Class definitions (diagrams) and details of object behaviours and methods 29](#_Toc381010717)

[User interface design (HCI) rational 29](#_Toc381010718)

[UI sample of planned data capture and entry designs 31](#_Toc381010719)

[UI sample of planned valid output designs 32](#_Toc381010720)

[Description of measures planned for security and integrity of data 33](#_Toc381010721)

[Description of measures planned for system security and integrity of data 33](#_Toc381010722)

[Overall test strategy 33](#_Toc381010723)

[Section 3: Technical Solution 34](#_Toc381010724)

[Section 4: System Testing 34](#_Toc381010725)

[Section 5: System Maintenance 34](#_Toc381010726)

[Section 6: User Manual 34](#_Toc381010727)

[Section 7: Evaluation 34](#_Toc381010728)

[Appendix A: Availability of Teaching Materials on Evolution 34](#_Toc381010729)

[Online research 34](#_Toc381010730)

[Interviews 35](#_Toc381010731)

[Appendix B: Evolution – the main concepts 36](#_Toc381010732)

[Sources 36](#_Toc381010733)

[Ants and their behaviour 36](#_Toc381010734)

**Further work required to Analysis section**

# Section 1: Analysis

## The problem: identification and background

### Problem identification

The problem addressed in this project is to develop a computer-based tool which will enable secondary school teachers of Biology at a school in Bath to illustrate how evolution occurs in a population. This problem was identified from interviews with Biology teachers which found that the subject of evolution is complex and difficult to teach to younger pupils (aged around 12-13).

These interviews found that pupils in this age range find it difficult to grasp the abstract ideas of evolution as this involves following an argument with several stages to it and a number of novel concepts such as ‘genes’, ‘mutation’, ‘selection’ and ‘inheritance’. The interviews also found that presentations and handouts failed to capture the dynamic nature of evolutionary change and ‘bring it to life’. Further, due to a shortage of suitable tools and materials it is difficult for teachers to set meaningful classroom exercises on evolution and to encourage pupils to do independent learning outside of the classroom.

As a result of these problems in teaching the topic teachers have found that some pupils can have common misconceptions about natural selection. For example, some pupils persist in believing in Lamarckism - the idea that an organism can pass on characteristics that it acquired during its lifetime to its offspring. It was felt, in the interviews, that a computer-based tools could illustrate the process of evolution, enable pupils to have more of a ‘hands-on experience’ and could be an excellent way for pupils to understand for themselves why ideas such as Lamarckism are wrong - rather than being taught simply that they are wrong.

Online research and face-to-face interviews with teachers (see Appendix A for details) has shown that there are many types of videos, handouts, presentations and other written materials currently available for school teachers on the subject of evolution. In addition, there are some computer-based tools which simulate how evolution occurs and which can, for example, allow different timescales and types of populations to be selected and different simulations to be run. These computer-based tools are evaluated and a new solution is developed in this project.

TODO

* Mention connection with teacher
* Mention how ants is chosen as the biological concept being simulated

### Background - Teaching context

The teacher who were initially interviewed teach Biology teachers and each have two classes a year of 15 – 20 mixed ability twelve to thirteen year old pupils. Evolution is taught as part of a Combined Sciences course in an introduction to the natural world and a 40 minute lesson is devoted to it. In these lessons the teachers cover the following:

* Some context – the ideas of Charles Darwin.
* A definition of evolution.
* An overview of how it occurs – introducing a number of ideas including the existence of differences within a population, members of a population being more/less suited to their environment, natural selection and the lengthy timescales over which evolution generally occurs.
* Examples are introduced – these can include more material on Darwin, on his travels and the Galapagos Islands, and well-known examples of evolution such as Darwin’s finches or the evolution of butterflies in contemporary Britain and their response to pollution.
* What evolution is not/common misconceptions – Lamarckism.
* Lessons are a mixture of explanation by the teacher, using video or written materials to illustrate the subject, taking questions and classroom discussion.

Computers with internet access are available during lessons with one computer for every pupil.

## Description of available tools

The Biology teacher who has been interviewed is not currently using any computer-based simulation tools as part of their lessons on evolution. However, some tools do exist and this section assesses the leading examples.

The tools assessed fall into two categories: a) tools focused specifically on evolution, and b) more broadly-based tools - for example considering the impact of disasters on animal populations. The table below summarises the tools which have been assessed.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Assessment** |
| **“Who Wants to Live a Million Years?”** | * Aimed at 12- 14 age group * Focused on evolution * Learning and simulation sections, together with a quiz, glossary and Darwin biography * Shows impact of a change in environment on a population and the effects of natural selection | * Learning and simulation sections * Poor treatment of mutations * Poor representation of gradual evolution processes * Not very engaging/fun * Minimal user interaction * Overall – too high level |
| **“Stop Disasters!”** | * Aimed at 16+ age group * Produced by the United Nations * Focused on disaster recovery * Shows the impact of disasters | * Engaging and sophisticated * Good level of user interaction * Good information and links |
| **“Cambrian EXPLOSION”** | * Aimed at 16+ age group * Created by Dimitri Bilenkin * Focused on the Cambrian explosion, a period of time when evolution accelerated creating a large diversity of new species in a short period of time * Shows the evolution of locomotion in a variety of creatures using a genetic algorithm | * Very good treatment of mutation and concept of evolution. * Good variety of settings which can be changed to focus simulation * Good use of statistics/data presenting them to the user as a easy to read graph * Simple and easy to use * Little documentation of complex settings * Great use of simulation output (visual representation of evolution) |

The tool “Who Wants to Live a Million Years?” (<http://science.discovery.com/games-and-interactives/charles-darwin-game.htm>) is an example of a free independent learning tool designed to help teach the concept of evolution. It is a flash-based web application aimed at teaching the basic concepts of evolution by demonstration. The content and graphics indicate that it is aimed at pupils from about 12 – 14 years.

The tool contains a learning section (‘Learn about Evolution’) and a simulation section (‘Play the Survival Game’). The learning section shows the user both text and pictures in a fun, animated way to help engage users. The simulation section proceeds as follows:

* It asks the user to choose a starting population and select variations within the population.
* It then demonstrates the impact of several changes in the environment (e.g. a warmer climate, a new predator or a natural disaster) on the numbers of the population - to show how only those best suited to the new environment will survive and reproduce while the others will not.
* If a population is wiped out the user is asked if they want to return to the beginning and select a different set of characteristics to test whether they will improve survivability.
* The simulation takes about 2 minutes to complete each time, with users expected to run several simulations.

A number of criticisms of this tool can be made:

* The simulation gives a misleading representation of natural selection as it presents mutations as a non-random event (as selected by the user) while in reality they are totally random and cannot be chosen.
* It is not clear what the different mutations are - different versions of an animal are shown in crude terms (e.g. one has longer fur, another has a longer body).
* The results of a change in environment are simply presented – there is no explanation why the different versions are increasing or decreasing in number
* The simulation is also not very engaging; there is minimal user interaction (which does not make it fun) with just a few clicks from start to finish.
* Overall, the tool is aimed at young children to provide some ‘fun’ but it provides only a very crude view of evolution.

Another example of a flash-based web application based in flash is “Stop Disasters!” ([http://www.stopdisastersgame.org](http://www.stopdisastersgame.org/)). The focus of “Stop Disasters!” is not evolution but preventing and responding to disasters – it is produced by the United Nations and the International Strategy for Disaster Reduction. It is aimed at older children than “Who Wants to Live a Million Years” and has a game section, a section on high scores from the game, information on the largest disasters in recent history and teacher resources about disasters such as tsunamis.

The simulation tool can be used to teach students about disaster prevention, as part of geography or other lessons. The objective of the game is to assess the risk of a disaster and to reduce the damage to the human population when one does occur. The simulation section proceeds as follows:

* The user selects a scenario e.g. ‘Coastal Village – tsunami’, and difficulty level (which is generated by the level of detail of the maps used).
* A disaster occurs and the user is given a mission e.g. ‘construct accommodation for 400 people with a budget of $35,000).
* The user then spends the budget to purchase various materials.
* A score is shown e.g. number of population successfully housed.
* The simulation takes about 10 minutes to run, depending on the choices made by the user.

“Stop Disasters!” is much more sophisticated and interesting than “Who Wants to Live a Million Years?” as it allows the user to interact with it more fully, it has better graphics and feels more like a game with choices – rather than a few simple steps over which the user has little influence. It does not deal with evolution but is a good example of a sophisticated interactive learning tool.

**Note**: Although “Stop Disasters!” is not specifically relevant to biology it is useful to evaluate as it shows how a simulation can be used as part of the teaching curriculum.

A web based application using JavaScript is “Cambrian EXPLOSION” (<http://www.cambrianexplosion.com/>). The application is focused on showing the evolution of locomotion in creatures. It uses a generic algorithm to generate creatures with different mutations and then runs a simulation of those species measuring how far they move in a specific amount of time. From this it calculates the generations fitness and draws graphs of how the fitness various with each generation thus showing the concept of evolution (small positive mutations each generation add up as that species has a greater change of reproducing thus transferring the mutation onto the next generation). This tool is quite sophisticated and complex although its graphics are not as good as “Stop Disasters!” it makes up for this in its simulation customizability.

In summary, the current tools which have been analysed have a number of advantages and disadvantages and none have been judged to meet all the relevant criteria. While “Who Wants to Live a Million Years?” is focused on evolution it has a number of drawbacks in the user experience and in the level of detail it shows about evolutionary processes. “Stop Disasters!” has a better user experience, being fun and informative and with significant user interaction, but is not focused on evolution. “Cambrian EXPLOSION” is a good simulation of evolution in species however it may be too advanced for the 12-13 year age group and also it focuses only on the evolution of locomotion and not the general evolution of animals.

My aim for the project is to create a tool which simulates ants and how they evolve over time due to the mutation of their characteristics. I will use ideas from the above solutions to similar problems to create this tool.

## Identification of prospective users

Two types of users of the system have been identified – teachers in the classroom and pupils.

Teacher users are teachers of Biology who conduct classes for 12-13 year olds at Kingswood School in which an introduction to evolution is required. Class sizes are from 10-15 pupils. The teacher users have an adequate knowledge of computing and application. Teachers will use the tool to demonstrate the basic concepts of evolution to the pupils, and will act as an administrator for the tool i.e. deciding when it is to be used and setting it for homework.

Pupil users are aged 12-13, are of mixed ability and have already had some background in science. They lack any detailed knowledge of evolution, although they are likely to have heard of it, and may find the idea challenging. The pupil users know how to follow simple instructions on a computer and use simple applications. Pupils will use the tool to see how the concepts of evolution taught to them in the classroom can be illustrated and ‘brought to life’.

## Identification of user needs and acceptable limitations

Analysis of user needs and acceptable limitations has been undertaken for both sets of users – teachers and pupils.

### User needs – teachers

Teachers require a tool that quickly and easily illustrates the concept of evolution in a manner that is understandable and interesting to children and which is easy to administer. Teachers’ more detailed requirements are:

1. As a teaching tool there **should not be a steep learning curve** i.e. the tool should not be over complex in its ideas, as this would defeat the point of simplifying the teaching of the concept.
2. As the tool is to be used as both a teaching tool and a tool to prompt independent learning/further study into the subject area by pupils the tool must be able to be **used within the classroom** (or computer room) as well as **by the pupils in their own time or for homework at home**.
3. The tool needs to be **able to be used by 12-13 year old pupils**, either individually or in small groups of 4-6, and must be **extremely simple to use** with minimum tutoring required – no more than 5 minutes.
4. **Setting up and running a simulation should take no more than 5 minutes** per simulation by pupils.
5. **Total time for using the tool** – from instruction to the analysis of results – **should be about 22 minutes.**
6. The tool must be **portrayed in a way to best engage this age group,** together with any linking content suitable for this age group. The tool must be fun and engaging.
7. All pupils in the classroom and outside the classroom should have **equal access to the tool** and there should be **no minimum system requirements** (e.g. owning a smart phone), so that everyone can use it.
8. When first introduced, **the tool must be demonstrated by a teacher through a projector** so that the pupils can get the most out of the tool and so that it is used correctly and all features used to their maximum effectively.

### User needs – pupils

1. The tool must be **suitable for 12-13 year olds** in terms of the complexity of content and the ease of use.
2. It must be **fun and engaging** – that is, pupils should enjoy each simulation and want to run several.
3. It must be capable of **communicating clear learnings**.
4. It must be capable of being **accessed by all pupils** both inside and outside the classroom.

### Software

1. The tool must **work on the major web browsers**:
   1. Internet explorer 9 and upwards (current school version).
   2. Firefox version 20 and upwards.
   3. Chrome version 22 and upwards.
2. The tool must be able to **run on Windows 7** which is the current school operating system.

### Hardware

1. The **tool must** **run smoothly** on the computers provided in the ICT rooms.
2. Only a mouse and keyboards can be used as input devices for the system as these are all that are provided by every work station.
3. The tool should be able to **scale up to resolutions larger then 800x600** so that it can be displayed through a school projector onto an interactive whiteboard.

### Simulation

1. The tool should **not be expected to be completely accurate** and approximations may be made so not to increase the complexity of the simulation beyond the scope of the teaching.
2. The simulation should not be expected to model realistic ant movement it should only be expected to **model a 2d representation of ant movement**.
3. The simulation should not be expected to model a realistic environment.
   1. However should **simulate basic regenerating food sources**.
   2. Should only **simulate a 2d terrain**.
4. The simulation should not simulate the complete life cycle of ants i.e. ants are born fully grown, it should only **model the ant’s adult life**.
5. The simulation should not model the inner workings of the nest only the **concept of the nest** is where the Queen lives and new ants are born and where food is stored.

## Data sources and destinations

The pupils will input values through buttons and sliders into the simulation via the user interface. The values selected will be used to change how the simulation runs and will therefore require heavy processing (running through the simulation). The simulation will output the result in an animation onto the screen.

## Data volumes

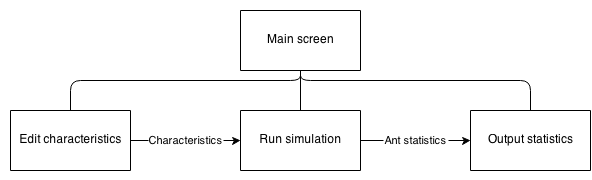
No data is saved as all data is processed in real time and outputted data is shown to the user immediately. So the size of the program is the only data volume, this will be less than 1mb.

## Analysis Data Dictionary

* **Characteristic** – A variable feature of an ant which changes it in some way e.g. how fast it can move.
* **Species** – A collection of unique characteristics define an new species in the simulation
* **Mutation** – A random event which alters a characteristic
* **Ant** **nest (or just nest)** – The home of the Queen ant and where it reproduces as well as a general food store
* **Queen** **ant** – The head of the species and the only ant capable of reproducing
* **Worker Ant** – An ant whose purpose is to collect food and supply it to the nest
* **Soldier Ant** – An ant whose purpose is to guard the nest and other ants from attack. And also to attack ants of a different species.
* **Pheromones** – Chemicals deposited by ants when they move which can be detected by antennae’s and used to alert other ants of danger or as a trial to food. Pheromones concentration slowly degrades over time due to evaporation.
* **Trial pheromone** – Deposited by worker ants and used as trials to and from food.
* **Fitness** – The ability to survive and reproduce
* **Food** – A source of nutrition which is in constant demand by ants. Without a constant supply of food an ant will die. Food will be needed by the nest to create new ants.

## Structure and Data Flow Diagrams

### Structure diagram



## Entity-relationship (E-R) model

This is not applicable to my project as I do not have any database manipulation.

## Object analysis diagram

## Objectives for the proposed system

### Ant simulation objectives

Make these SMART objectives

* Specific – Objectives should be specific to the problem being solved and should specify what should be achieved
* Measureable – You need to be able to measure whether you are meeting the objectives or not?
* Achievable – Are the objectives you set achievable and attainable?
* Realistic – Can you realistically achieve the objectives with the resources you have?
* Time – When do you want to achieve the set of objectives?

1. A tool which **illustrates the concept of evolution**. This concept is summarised in Appendix B. The main ideas of the concept of evolution should all be covered:
   1. **Organisms have characteristics** which determine their species.
   2. As a result of the mutation of characteristics, **variations capable of being inherited** exist within populations of organisms.
   3. Organisms **produce more offspring than can survive**.
   4. These **offspring**, with their different inheritances, **vary in their ability to survive and reproduce**.
   5. In conditions with competition between organisms for survival and reproduction those **organisms with traits that give them an advantage over their competitors pass these advantageous traits on**, while traits that do not confer an advantage are not passed on to the next generation. As a result we have the ‘survival of the fittest’ and a gradual change in populations – they change or may even die out.
2. The simulation should be able to generate **random food placement**.
3. The simulation should be able to simulate basic **growth of food**.
4. The tool should provide a **simulation involving movement** since evolution involves dynamic change and a static tool is not appropriate.
5. The simulation must simulate the three basic types of ants
   1. The **worker ant** – Locates and collects food
   2. The **soldier ant** – Guards and attacks ant species
   3. The **queen ant** – Creates a nest
6. The simulation must **model a basic nest** capable of:
   1. Producing new ants
   2. Being attacked and eventually destroyed.
7. The ants should have variable basic characteristics Including:
   1. **Speed** the ants move
   2. The **reproduction rate**
   3. The **amount of food ants are able to carry**
   4. The ants **eyesight** i.e. the distance the ant can see
   5. The ants **antenna size** i.e. the distance the ant can smell pheromones from
8. The simulation should be able to **introduce random mutations** (so pupils can see how the ants must suited to the environment will survive).
9. The simulation must be able to **show multiple different species at the same time**, so that their fitness can be compared by the pupils.
10. The simulation should model **energy intake from food** needed for ant to survive depending on the ants characteristics. For examples ants with more favourable characteristics such as moving faster should require much more food than ants which have less favourable characteristics.
11. The simulation must be able to model **ants fighting**.
12. The simulation must model **pheromone trails** including:
    1. Their **creation** when an ant is moving
    2. Their **evaporation** due to conditions
    3. **How ants respond** to the trials
       1. Following to find food
       2. Following back to nest

### Simulation interface features objectives

1. **Pause and play buttons** to pause and play simulation.
2. **Navigation buttons** to move around the simulation.
3. **Zooming** in and out of the simulation.
4. For each species (displayed when selecting nest) statistics measuring:
   1. The **amount of food stored in the nest** of a specific species
   2. **Number of ants** in species
5. **Editable characteristics** e.g. by sliders or buttons so that the user can change the simulation in real time.
6. **Selection of different species** so that their characteristics can be compared.

## Appraisal of potential solutions

### Smart phone app

A smart phone app written in Java (android) or Objective-C (IOS) would run the simulation on a phone.

This would fulfil the majority of the requirements:

* Due to smart phone portability the app could be used both inside and outside the classroom.
* Phones are internet-enabled and so the app can be downloaded from within the class room.
* The age group are familiar with the concept of apps and would be familiar with the interface.
* Smart phones are powerful enough to run a complex biological simulation and could maintain simulation accuracy.

However, there are some disadvantages from a smart phone solution. Not all pupils would have a smart phone and as a result not everyone in the class would be able to use the app. Furthermore, due to the fragmentation of smart phone operating systems the same app would need to be written for multiple different architectures (IOS, android …) - possibly making the project unviable. Lastly, if the exercise is done in class there would be no way to know if the pupils where using the tool or using their phones for another use.

### Web application

A web application written in HTML5 and JavaScript is another potential solution. Using the web has a number of advantages.

* The internet is available on school and home computers.
* Access to the tool would not be a problem as almost everyone has access to a computer (i.e. school computer lab) thus for filling the users requirements.
* Even young pupils are very familiar with web application interfaces and as such teachers would only be required to provide a short introduction and pupils could quickly learn how to use the application.
* There would be little issue with cross compatibility as both html and JavaScript are web standards and implemented in all modern browsers including mobile browsers.
* It would be very easy to setup as it is a essentially a webpage and therefore there is no need to download or install any applications just visit the webpage.

However the complexity of the project may increase due to the use of multiple languages (HTML, CSS, and JavaScript) and the added complication of configuring and maintaining a web server and domain. Furthermore a lot of time would have to be spent optimising the application to speed it up as JavaScript is much slower as it runs through a browser compared with natively written programs.

### Desktop application

A desktop application written in C/C++ or Java. This would have a number of advantages:

* The simulation could easily run due to C/C++’s speed and also access to a desktop.
* Like the web application in (HTML5, JavaScript) it could be used by all of the pupils at home or in the classroom for filling the objectives as C/C++ can be compiled to run on multiple different operating systems.

Although as the application would have to be cross compatible between the major operating systems (Windows, Mac and Linux) in order to make sure everyone could use it on their machine increasing the complexity of the program. Furthermore the increased setup time of the tool would make it less user friendly (i.e. user must download and install the program).

### Pen and paper simulation

A non-technical solution to the problem would be to create a set of rules and then use counters on a gridded board to simulate how the ant’s behaviour would change depending on their input values. Random mutations could be introduced by using dice.

* The simulation could be used all of the pupils.
* The simulation does not need computers and therefore the students do not have to change room to one equipped with computers, increasing the amount of time available.

However the simulation would be far too slow and the pupils may not be able to see the overall patterns emerging from it and therefore not learn the concepts of evolution. It would require more than one pupil to be fun and engaging and so cannot be done at home for homework. It would be tedious to record all of the statistics (a computer based solution could do this automatically). The scale of the simulation would be dramatically reduced to fit into the time allowed and would be less impressive as a result making the simulation less interesting to pupils.

## Justification of chosen solution

The chosen solution is a website based tool written in HTML5, CSS and JavaScript and using a school webserver backend to serve the website. This was chosen as the solution because it allows for all of the objectives to be achieved to their full with the least additional complexity.

* All simulation objectives can be accomplished with a well written optimised program in JavaScript.
* The simulation can be easily displayed in an aesthetically pleasing manner with CSS to engage the pupils.
* The tool requires very little preparation time as it only requires the browser which is already installed on school computers.
* The tool can be easily used at home by pupils as it is a website and can be navigated too from anywhere with internet and a web browser (which is assumed they have).
* The tool can be easily displayed through a projector and should be able to scale to any sized monitor (as it is a website).
* The tool will have a shallow learning curve as pupils already know how to navigate webpages.
* The tool will be able to run on existing hardware and software

# Section 2: Design

Diagrams needed:

* Overall system flow chart (website + simulation + scores + login)
* Ant behaviour diagrams (same as ant algorithm )
  + General behaviour
  + Solider behaviour
  + Worker behaviour
  + Queen behaviour
* User interface design
  + UI for webpages (login, main)
  + UI for simulation
* Class/object diagrams
  + OO diagram for ants (Queen?)
  + OO diagram for map/environment

## Overall system design

### The simulation

The simulation will simulate the principles of survival of the fittest by using ants. It will both simulate ant society structure as well as a basic environment for the ants to move in.

#### Evolution

Ants in the simulation will have a set of characteristics; these characteristics define the ant’s behaviour. For example one characteristic of an ant is its eye sight, if a particular ant has very good eye sight, it can spot food from further away, thus being more able to survive. The characteristics implemented will be:

* **Speed –** The speed the ant can move
* **Antenna size –** The size of the ants antenna (determines how far away the ant can smell pheromones)
* **Antenna angle –** The angle through which the ant can detect pheromones from its current direction
* **Eye sight –** The range at which the ant can see objects
* **Eye angle –** The angle through which the ant can see objects from its current direction
* **Jaw size –** The size of the jaw of the ant (determines how much food an ant can carry)
* **Jaw strength –** The strength of the ants jaw (determines how much damage it can do when biting other ants)
* **Sting size –** The size of the ants string (determines how far the ant can attack)
* **Pheromone concentration –** The concentration of pheromones secreted by the ant

A species is a collection of specific values of characteristics. There will be multiple species in the simulation. Ants will all belong to a species, and it is the species which determines the ant’s characteristics. When ants are born they will inherit the species which their nest has. However when a queen is born, there is a chance that the species the queen inherits is mutated. A mutation is the change of a single value of a characteristic in a species. If a queen inherits the mutated species and if it goes onto creating a nest then all ants born from this nest, will inherit the mutation.

A mutation can either be beneficial or destructive e.g. a better eyesight is beneficial as it allows ants to see further, while a decreased speed is destructive as it means that the ant will move slower. Ants which inherit mutations which are beneficial are more likely to be successful in surviving, thus they will have more ants. This means that the species will be passed onto more generations. This is the basic process of evolution and how it will be implemented in my simulation.

To decrease the advantage of some mutations there is a cost associated with characteristics. If a characteristic is more beneficial it will cost the nest a lot more *energy* (health in the simulation) to create an ant.

#### Ant behaviour

All ants must eat a certain amount of food otherwise they will lose health. Ants will have different amounts of health depending on their characteristics, if health is below 0 the ant dies.

There will be three types of ant:

* **Worker ant** – Responsible for finding, collecting and bringing food back to the nest.
* **Soldier ant** – Responsible for attacking other ant colonies destroying their nest or killing ants from the other nest to disrupt their collection of food leading to the collapse of the nest because of starvation. And also responsible for defending ants from attack from other species and also defending the nest.
* **Queen ant** – Creates a nest, and reproduces creating new ants.

**Trial pheromones** are continually laid down by worker ants; they will slowly diffuse into surrounding areas and will also slowly evaporate in the environment. When an ant finds food it will try to make its way back to the nest, as it does so it will lay down pheromones. These pheromones will then be followed by other ants either to the nest or to the food. Ants which have followed the trail and found food will lay down their own pheromones. Each time more pheromones will be laid down making the path more concentrated. Ants will be more likely to follow very concentrated routes when looking for food. Soldier ants will act as sentries and stand along busy trials defending their worker ants.

Nests will be simulated very minimally, after the initial simulation starts and the first queen starts reproducing it will create both soldier ants and worker ants but also new queen ants in a user defined ratio. When a new queen is created there is a chance of random mutation and the characteristics which define the species change (creating a new species). The queen will then fly away from the nest where it was born and create a new nest and start creating new ants, if the queens characteristics where mutated and it forms a new species it will become competition for other species.

#### Interface

At the start of the simulation the user will adjust the characteristics of the first colony or choose random characteristics. Once the simulation is running the user will be able to edit ant’s characteristics or leave the starting characteristics and watch how the simulation plays out. During the simulation the user will be able to track the progress of their species as well as monitor the progress of other species viewing statistics like:

* Number of ants
* Number of nests
* Total food in nest

The user will be able to zoom in and out of the simulation as well as pan around the entire simulation using mouse gestures e.g. dragging and scrolling. It will also be possible to control the simulations time. The user will be able to:

* **Run/Pause** – start and stop the simulation so that the user will be able to stop the simulation if are distracted.
* **Step** – Go a single frame into the future, this is will be useful if the user wants to view an event which happens quickly e.g. an ant being chased by a soldier ant.
* **Restart** – Restart the simulation if it doesn’t go to their liking.

During the simulation information about the species in the simulation added to the page to allow the user to review the information and adjust characteristics or judge how well the species is doing. The information presented is:

* **Colour** – This will be useful as it will show the user the colour of the species so that they can be easily spotted on the simulation.
* **Number of ants** – This is a measure of success of the species as it show how the species has grown form just a single ant.
* **Number of nests** – This is a measure of potential success as nests create ants and therefore the more nests the higher the number of ants which can be created in the future.
* **Amount of food** – This is another measure of success as well as impending danger. If the amount of food is high then there is the likely hood that a lot of new ants will be created however if this is low it shows that the species nests are close to starvation.

Finally on the page the user will be able to adjust characteristics and see them update live onto the simulation (i.e. without a restart). This is a useful for the user as it allows them to explore how each characteristic affects ants in the simulation. And so it will be useful when the user first uses the simulation to quickly allow them to understand all of the characteristics. As the user adjusts the characteristics they will get instant feedback in the form of text which shows the species cost as well as the cost of each ant in terms of food. The user will also be able to do the control the characteristics as a whole:

* **Update** – Push the modified characteristics to the simulation, this is needed rather than an instant update because it acts as a buffer. If a user accidently edits a characteristic it should not be pushed to the simulation.
* **Default values** – The update all characteristics to default values, needed if the user has changed characteristics and has not been able to create a viable ant. The default values will be designed to create a species which is initially successful this a good demonstration to prospective users.
* **Random –** This will set all characteristics to random values, useful to the user to show how even extreme mutations can be successful.

To aid ease of use their will be a number of shortcuts available to the user. This will allow the following tasks to be completed with just a single button press. This will also allow the entire tool to be used without a mouse and only a keyboard.

|  |  |  |
| --- | --- | --- |
| **Task** | **Button** | **Rationale** |
| Run/Pause | Space bar | This will be the most common action and so it should be easily accessible. |
| Step | “s” key | First letter of “step” and so intuitive to users. |
| Restart | “r” key | First letter of “restart” and so intuitive to users. |
| Zoom in/out | “+” and “-” key | These keys are commonly used for zooming in and out and therefore properly known by the user to perform this action. |
| Pan around | Arrow keys | Again the arrow keys are used for moving/panning around in other applications (e.g. games or online maps) and so intuitive to the users. |

**Note:** exclusion of **update** as this should not be accidently pressed and so it should only be pressed on purpose. Both default values and random have been excluded from shortcuts as these actions do not occur common enough to warrant a shortcut (It is still possible to access all of these actions from the keyboard using a combination of the TAB key and arrow keys).

## Description of modular structure of system

### Ant

The properties of a basic ant are the following:

* **Position** – The location of the ant
* **Direction** – The direction the ant is facing
* **Prioritized direction** – The direction the ant wants to move in
* **Size** – The size of the ant
* **Species** – The species the ant belongs to
* **Nest** – The nest where the ant was born
* **Health** – The current amount of health the ant has
* **Health Threshold** – The level of health bellow which the ant is hungry
* **Health rate** – The rate at which an ant will lose health
* **Goal** – The current goal the ant is trying to complete e.g. find food
* **Target** – The target the ant is heading to e.g. a piece of food

An ant can also perform the following basic actions:

* **Take food –** Takes a single piece of food
* **Get food –** Head towards a piece of food
* **Use food –** Decide how to use food i.e. eat or carry
* **Find food target –** Finds a locate a target piece of food within viewing distance
* **Scan –** Scan surroundings for items of interest which can be seen e.g. other ants and food
* **Smell –** Similar to scan however only detects pheromones
* **Secrete –** Secrete a pheromone
* **Wonder –** Walk around the map either randomly or following pheromone trials
* **Die –** Kills the ant

### Worker

A worker ant is a specific type of ant, its purpose is to find, collect and deposit food to the nest. Because of this the worker ant must be able to perform the following actions:

* **Determine if it can carry food –** Does the ant have enough room to carry extra food
* **Deposit food**

### Queen

A queen ant is another specific type of ant. The Queen ant is responsible for creating new nests and therefore needs to be able to:

* **Pick nest site**
* **Create a nest**

### Species

A species is a collection of specific values of characteristics. A characteristic is a property of an ant e.g. an ant’s eye sight (how far the ant can see). The following basic characteristics will be implemented:

* **Speed –** The speed the ant can move
* **Antenna size –** The size of the ants antenna (determines how far away the ant can smell pheromones)
* **Antenna angle –** The angle through which the ant can detect pheromones from its current direction
* **Eye sight –** The range at which the ant can see objects
* **Eye angle –** The angle through which the ant can see objects from its current direction
* **Jaw size –** The size of the jaw of the ant (determines how much food an ant can carry)
* **Jaw strength –** The strength of the ants jaw (determines how much damage it can do when biting other ants)
* **Sting size –** The size of the ants string (determines how far the ant can attack)
* **Pheromone concentration –** The concentration of pheromones secreted by the ant

There is a chance that a species will be mutated, this will result in the change in the value of a characteristic. This can happen whenever an ant is created. However the mutation will only ever be passed on by queen ants, as normal ants cannot create new ants.

### Nest

The nest implemented will be a very simple abstraction of actual ant nests. The nest must have the ability to recreate new ants. The nest must also be able to store/receive food from workers.

### Pheromone

A pheromone is a chemical signal deposited by ants to influence the direction of other ants i.e. lead other ants to food. A pheromone has the following properties:

* **Concentration –** The concentration of the pheromone (a pheromone concentration slowly reduces over time due to evaporation)
* **Position –** The position of the pheromone
* **Species –** The species of the ant which deposited the pheromone. This is important as in some species of ants ants will only follow pheromones laid down by an ant of their own species.

### Food

A piece of food has the following properties:

* **Position** – The location of the piece of food
* **Size** – The size of the piece of food
* **Density** – The density/concentration of the food

## Definition of data requirements (Design Data Dictionary)

## Description of record structure

This is not applicable for my project as there are no databases used.

## File organisation and processing

This is not applicable for my project as there is no data stored or retrieved from files.

## Validation required

Input validation will be done by sliders, this will reduce the amount of validation of input needed as it restricts the user to input values within a specific range and of a specific type. This is done to reduce the number of validation errors the user receives and thus streamlining their use of the application.

However the following characteristics will require extra validation and may produce validation errors or warnings:

* Queen steps min and Queen steps max – As these two characteristics act as a range of values it is important for min to the less then max. If this is not the case the user will be given an error and how to correct it.
* Soldier, Queen and Soldier Food cost– If an ants food cost is less or equal to the species cost the user should receive a warning as it means that the ant will be created with its health less than or equal to zero. This means that the ant will die instantly when it is created. This is likely not what the user wants to do and so a warning is issued (This is not an error however and the simulation can be run with ants with health less than zero).

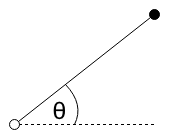
## Identification of storage media

This is not applicable for my project as there is no data stored.

## Identification of suitable algorithms for data transformation, pseudo code of these algorithms

### Angle between two points

Find the angle from the horizontal in radians between two coordinates.



coord

target

**DECLARE** integer dx

**DECLARE** integer dy

**SET** dx = targetX - coordX

**SET** dy = targetY - coordY

**RETURN** atan2(dy, dx) + Math.PI / 2;

### Boundary

The map will act like a torus i.e. if an ant goes off one side of the map, it will reappear on the other side of the map. This pseudo code will update a coordinate to its *wrapped* coordinate if needed.

**IF** coordX < boundsXMmin

**SET** coordX = boundsXMax - abs(coordX)

**ELSE IF** coordX >= boundsXMax

**SET** coordX = coordX - boundsXMax

**ENDIF**

**IF** coordY < boundsYMin

**SET** coordY = boundsYMax - abs(coordY)

**ELSE IF** coordY >= boundsYMax

**SET** coordY = coordY – boundsYMax

**ENDIF**

### Get a block of cells

It will be necessary to sample blocks of cells by ants to see what lies around them.

**DECLARE** block = []

**FOR** y = (coordY – sizeHeight) **to** (coordY + sizeHeight)

**FOR** x = (coordX – sizeWidth) to (coordX + sizeWidth)

block.push([coordX, coordY])

**ENDFOR**

**ENDFOR**

**RETURN** block

### C:\Dropbox\projects\Ant-Simulation\writeup\assests\Algorithms\getSector\Sector digram.pngGet a sector

Get sector will return the list of cells which lie within the sector at a particular coordinate facing a certain direction with an angle theta. This will be used by ants to sample the map in front of them to see any items of interests such as other ants, food and pheromones. Ants will have a certain viewing angle (the angle through which they can see).

**DECLARE** block = [];

**FOR** y = (coord.y – radius) **TO** (coord.y + radius)

**FOR** x = (coord.x – radius) **TO** (coord.x + radius)

**SET** searchCoord = {

x: x,

y: y

}

**SET** angle = angleTo(coord, searchCoord)

**SET** dist = distance(coord, searchCoord)

**SET** minSector = validateDirection(direction - angle / 2)

**SET** maxSector = validateDirection(direction + angle / 2)

**IF** angle >= minSector **AND** angle <= maxSector **AND** dist <= radius)

block.push(searchCoord

**ELSE** **IF** (direction <= angle / 2 **OR** direction >= Math.PI \* 2 - angle / 2) **AND** (angle <= maxSector **OR** angle >= minSector) **AND** dist <= radius

block.push(searchCoord)

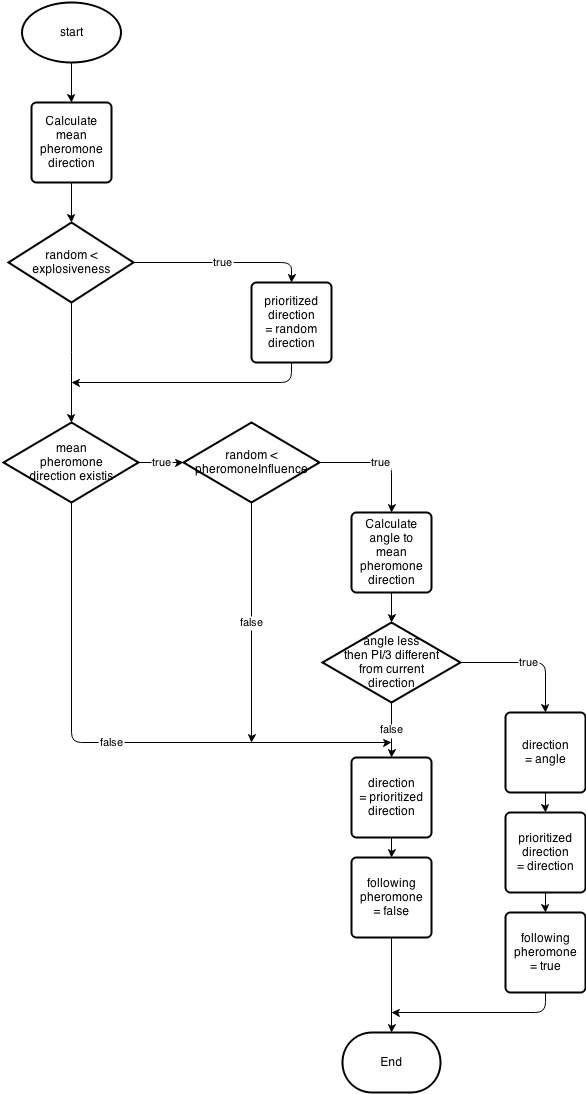
**ENDIF**

**ENDFOR**

**ENDFOR**

**RETURN** block

### Ant non goal driven movement



### Ant pheromone secretion

**FOR** pheromone in cell

**IF** pheromoneSpecies == thisSpecies

pheromoneConcentration += thisPheromoneConcentration

**IF** pheromoneConcentration > MAX\_PHEROMONE\_CONCENTRATION

pheromoneConcentration = MAX\_PHEROMONE\_CONCENTRATION

**ENDIF**

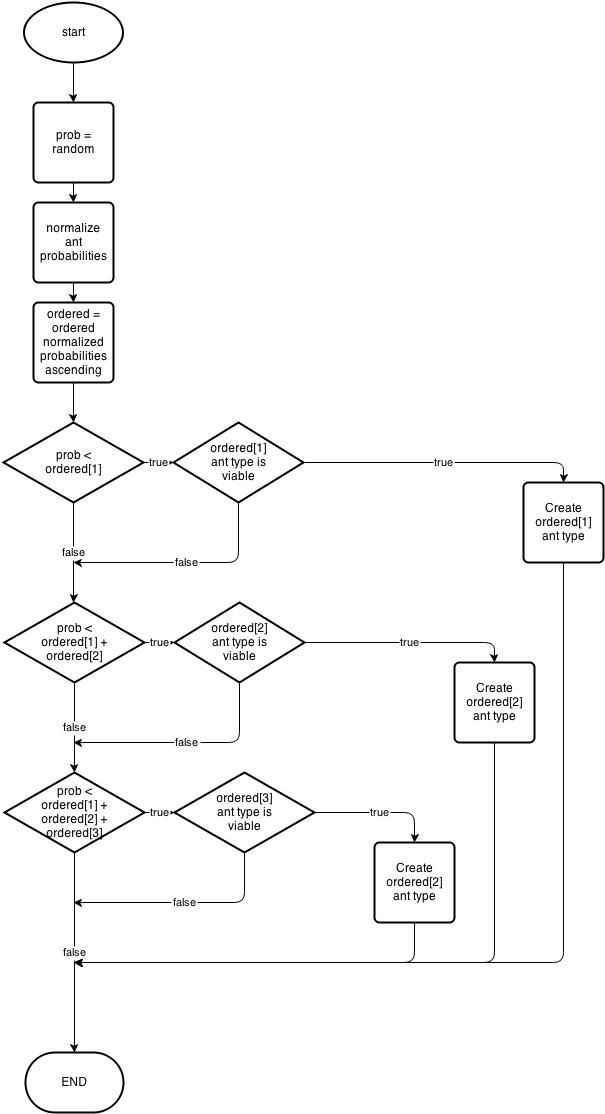
**ENDIF**

**ENDFOR**

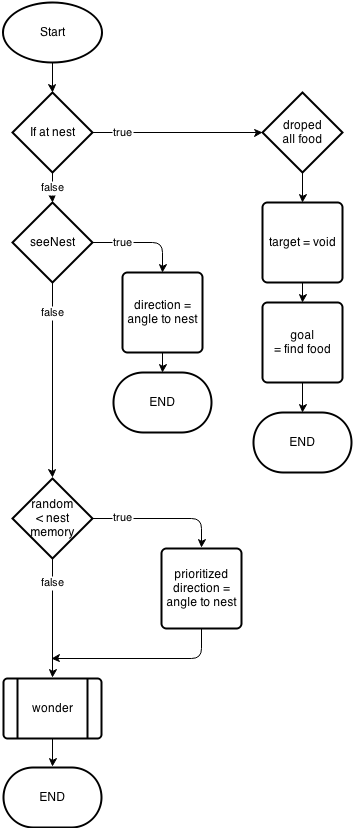
pheromone = new Pheromone(this.species.chars.pheromoneConcentration, thisCoord)

pheromone.addToMap()

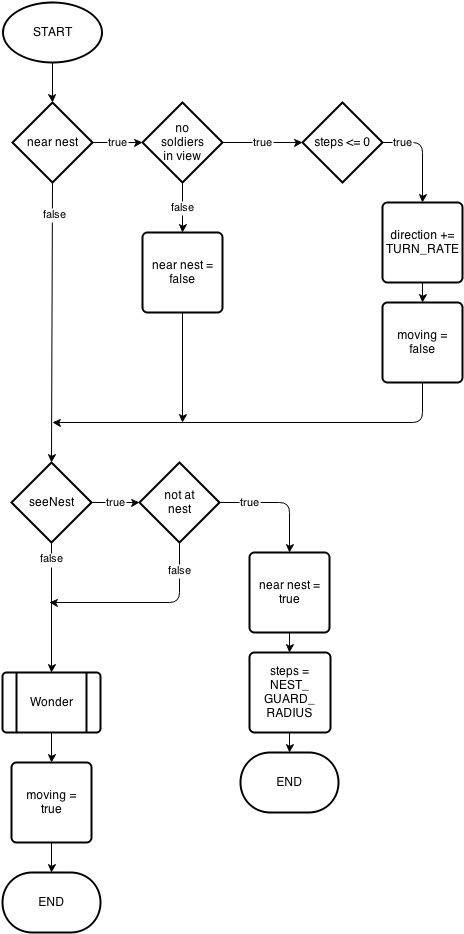
### Nest reproduction



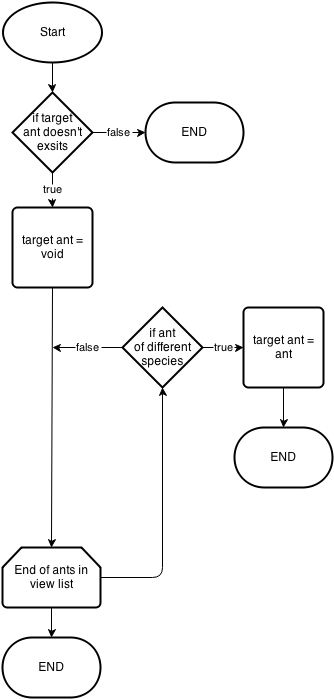
### Worker ant depositing food



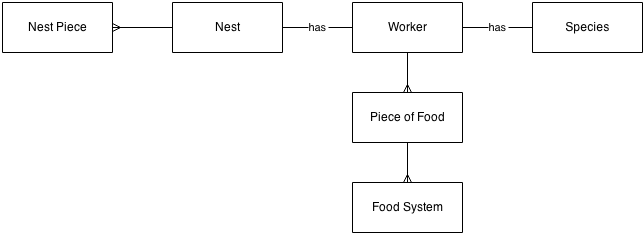
### Soldier nest guarding



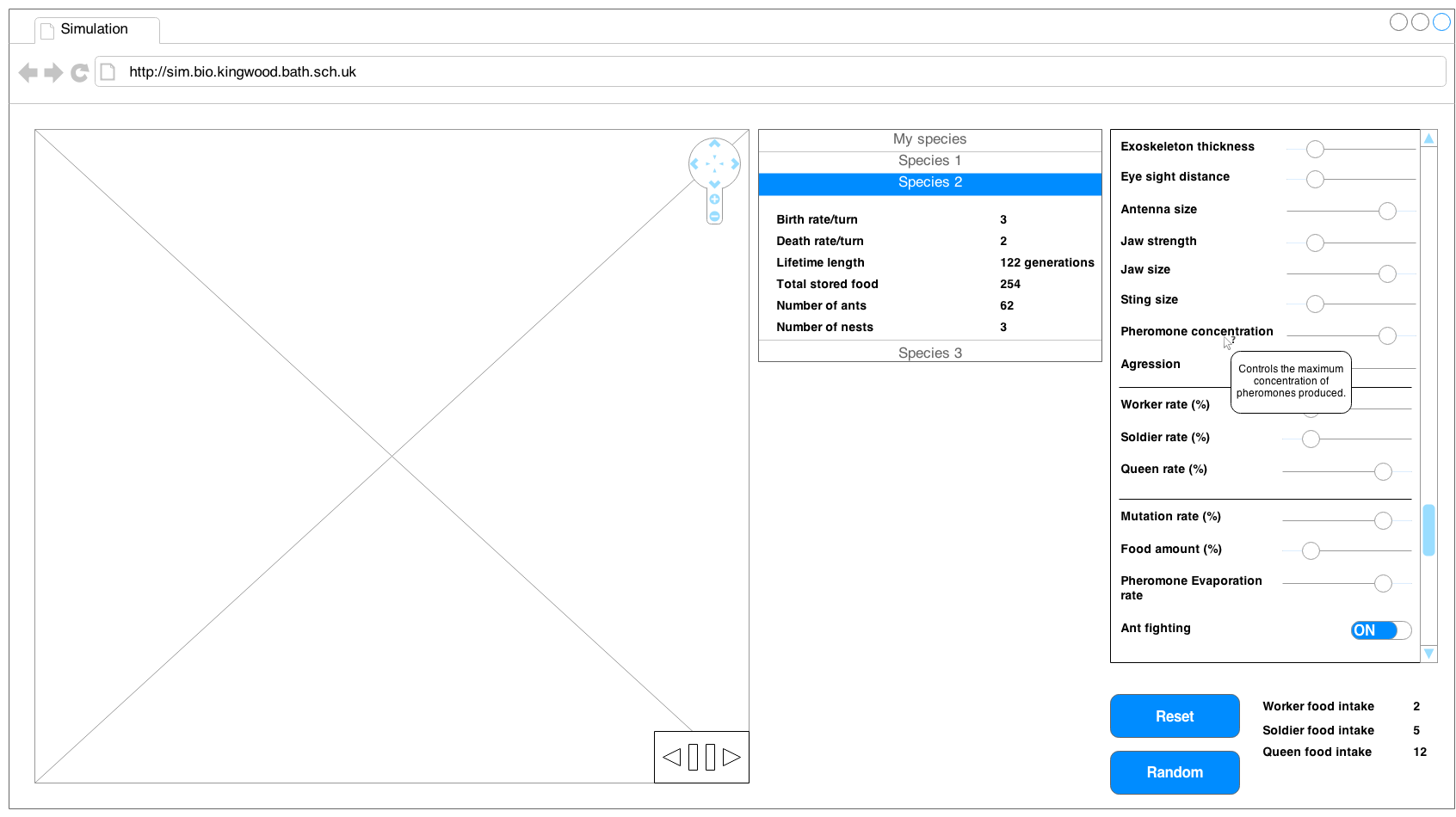
### Soldier pick target



## Class definitions (diagrams) and details of object behaviours and methods

\\fs001\01intake$\01ROBINSON\work\comp\comp4\Ant-Simulation-master\writeup\assests\Design\Entity relationship\ant\entity relationship - ant.png

## User interface design (HCI) rational



**6**

**5**

**4**

**3**

**2**

**1**

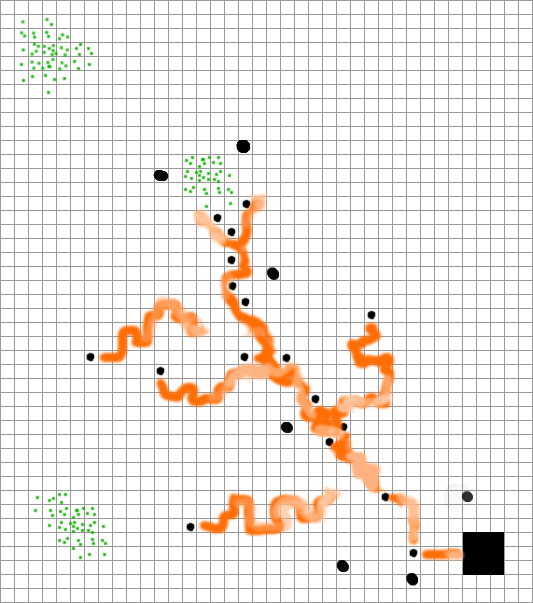
The tools interface is simple and is split up into three main sections:

* The simulation section on the left is where actions related to the presentation of the simulation are located. The main part of this is the main simulation however it also has position navigation buttons (item 1) which has both panning and zooming functions. And time navigation buttons (item 2) which has pausing and playing functions as well and single stepping forward buttons.
* The section in the middle of the screen is the statistics panel (item 3) which is where data outputted by the simulation is displayed to the pupil. The section has collapsing tiles for each species which display the real time statistics for that species.
* The section on the far right of the screen is the configuration panel (item 4); this is where the pupil can edit characteristics of the ants and general simulation settings. Relevant settings are grouped together within the panel to allow faster navigation. Tool tips (item 5) are used to show documentation of the settings and describe how they are used. At the bottom of the panel (item 6) is two buttons on for resetting the settings and the other for creating a random set of settings. Next to these buttons are the food intake indicators which show how much food each type of ant will require because of the chosen settings, these will update whenever a setting is changed so the user can immediately see the impact of the settings.

## UI sample of planned data capture and entry designs

Data within this tool is gathered from the editing of characteristics each characteristic has its own input field, the fields are chosen to limit the amount of user error in the input. Sliders and toggle switches are the only two data capture types. Sliders are used when a range of values within a limit must be chosen, they are chosen rather than spinners as they can be used with only the mouse, this is the same with a toggle button.

## UI sample of planned valid output designs



#### Colours

In the simulation each unique characteristic will have a specific colour, the strength of the characteristic gives the vibrancy of the colour, and the mix of these colours gives the species its colour. This is done so species with similar characteristics can be spotted and more easily compared.

Food is displayed as green a colour often associated with leaves, grass… which are the food the ants (most like leafcutters) eat in real life.

Pheromones in the simulation will be displayed in the colour of the ant species however a much lighter tint, the strength of the pheromone determines the opacity of the colour so if there is lots of pheromones with a higher concentration or newly laid pheromones then the colour will be more vivid showing that is more active and recent while older more stagnant trials will be very light, as to not attract attention from the pupil.

#### Shapes

* Worker ants – Displayed as circles, basic and easy to see however no to differentiate between different ants of the same species so no need for more complex shapes. Choose not to use ant like shapes as rotation of them would create too much on screen movement.
* Soldier ants – Displayed as larger circles, again no real need to have complex shape to differentiate individual ants by as this is not important, it is the main overall picture which counts.
* Nests – Displayed as large 3 by 3 squares, no need for complex shape will be easy to see overall picture with simple shapes.
* Food – Displayed as a collection of dots, the amount of dots in a square represents the amount of food in that square; this is done so the pupils can more easily estimate the relative food amounts in a certain area just by glancing at them.
* Pheromones – Displayed as long trials behind the ants much like they are in real life, this is done to show to the user the use of pheromones.

## Description of measures planned for security and integrity of data

There is no sensitive personal information recorded or entered into the system so the integrity of data is not applicable to my project.

## Description of measures planned for system security and integrity of data

## Overall test strategy

As part of the development strategy methods from test driven development will be used to create parts of the application. This means that for some parts of the project unit tests will be written which will be used to aid development. This will mean that a testing framework will be used during the development of the project to generate and automatically tests these tests.

White box test Ant, worker, queen, soldier and nest classes because methods of these classes will primarily contain logic and so it is important to test each non trivial logical path to insure that the procedures perform correctly. Utilities functions i.e. functions which do not belong to any class but provide useful actions which can be applicable to many situations, will be black box unit tested. This is because these functions will more mathematically oriented rather than logically e.g. the angle between two points.

As the tool will be used outside the classroom i.e. it is set for homework, it is important for the toll to work on multiple platforms. So it will be tested on multiple platforms (Operating systems and browsers) to make sure that it can be used outside the classroom. Due to the large numbers of possible operating systems and web browser combinations it is needed to limit these tests to only the most popular choices.

The simulation will be very intensive for large numbers of ants, so it will be needed to carry out a stress test to check that the simulation will be able to cope on the hardware available in the class room.

User interface functional testing will also be carried out to test that the user interface works as expected. This includes that buttons perform the correct actions and that inputs and outputs are displayed correctly.

Finally it will be necessary to test the simulation against real life biological models. As the simulation will model ants it should be able to be compared to real life ants and produce similar results/patterns. This is needed to show that the simulation truly models ants.

# Section 3: Technical Solution

# Section 4: System Testing

# Section 5: System Maintenance

# Section 6: User Manual

# Section 7: Evaluation

# Appendix A: Availability of Teaching Materials on Evolution

## Online research

The following sources illustrate the types of materials which are available for school teachers on the subject of evolution and related subjects (such as conservation).

|  |  |  |
| --- | --- | --- |
| **Website** | **Materials available** | **Assessment** |
| http://www.arkive.org/education/  (searched 2/10/13) | * Presentations, teacher notes and spreadsheets on numbers of Finch variations in the Galapagos Islands and on peppered moths * Aimed at 14-16 and 16-18 age groups | * Focus on conservation * Attractive materials * No online or computer simulation tools on evolution but some data which can be used in classrooms |
| More to be added |  |  |

## Interviews

Interview with Dr Sheffrin(Head of sciences and Biology teacher) have been conducted at Kingswood School, Bath on 18/10/13. The findings were as follows:

### How is the topic of evolution currently taught?

Taught in year 8, 11 and sixth form biology. Depending on the age group it is covered in various levels of complexity, it is taught over a series of 45 minute lessons of classes of roughly 15 and homework’s. The majority of the work is done using worksheets.

### What topics are covered?

* Mutation
* Natural selection
* Survival of the fittest
* Darwin vs. Lamarckism
* Adaptation
* Habitats and environments
* Characteristics

However for the sixth form topics on DNA and specific process are covered.

### Hardware and software available?

During classes an IT room is available with a computer for all of the pupils. There is little software specifically for use of biology teaching however there is lots of generic software such as web browsers available on all of the machines.

### How much time is available to use the tool?

Half of a lesson and a prep.

### Good example animals which can be modelled and will be interesting to observe?

Animals which have large populations are best to model as having more generations show a speeded up version of evolution which in animals with smaller populations takes a much longer amount of time and is therefore less interesting to simulate. Insects would probably be the best to simulate.

# Appendix B: Evolution – the main concepts

From Wikipedia – entry on ‘Evolution’

“Evolution by means of [natural selection](http://en.wikipedia.org/wiki/Natural_selection) is the process by which genetic mutations that enhance reproduction become and remain more common in successive generations of a population. It has often been called a "self-evident" mechanism because it necessarily follows from three simple facts:

* Heritable variation exists within populations of organisms.
* Organisms produce more progeny than can survive.
* These offspring vary in their ability to survive and reproduce.

These conditions produce competition between organisms for survival and reproduction. Consequently, organisms with traits that give them an advantage over their competitors pass these advantageous traits on, while traits that do not confer an advantage are not passed on to the next generation.

The central concept of natural selection is the [evolutionary fitness](http://en.wikipedia.org/wiki/Fitness_%28biology%29) of an organism.[[96]](http://en.wikipedia.org/wiki/Evolution#cite_note-Orr-96) Fitness is measured by an organism's ability to survive and reproduce, which determines the size of its genetic contribution to the next generation. However, fitness is not the same as the total number of offspring: instead fitness is indicated by the proportion of subsequent generations that carry an organism's genes.[[97]](http://en.wikipedia.org/wiki/Evolution#cite_note-Haldane-97) For example, if an organism could survive well and reproduce rapidly, but its offspring were all too small and weak to survive, this organism would make little genetic contribution to future generations and would thus have low fitness.

If an allele increases fitness more than the other alleles of that gene, then with each generation this allele will become more common within the population. These traits are said to be "selected *for*". Examples of traits that can increase fitness are enhanced survival and increased [fecundity](http://en.wikipedia.org/wiki/Fecundity). Conversely, the lower fitness caused by having a less beneficial or deleterious allele results in this allele becoming rarer — they are "selected *against*". Importantly, the fitness of an allele is not a fixed characteristic; if the environment changes, previously neutral or harmful traits may become beneficial and previously beneficial traits become harmful. However, even if the direction of selection does reverse in this way, traits that were lost in the past may not re-evolve in an identical form.”

Search undertaken 8/10/2013

# Sources

## Ants and their behaviour

<http://en.wikipedia.org/wiki/Ant>

<http://en.wikipedia.org/wiki/Eusociality>

<http://en.wikipedia.org/wiki/Pheromone>

<http://en.wikipedia.org/wiki/Ant_colony>

<http://en.wikipedia.org/wiki/Leafcutter_ant>