Terrarium Simulator

Contents

[1. Analysis 2](#_Toc528571910)

[1.1 The problem: identification and background 2](#_Toc528571911)

[1.2 Description of the current system 2](#_Toc528571912)

[1.2.1 Problems with current system 2](#_Toc528571913)

[1.3 Objectives 2](#_Toc528571914)

[1.4 Data Modelling 2](#_Toc528571915)

[1.4.1 Analysis Entity Relationship Diagram // Object Diagrams // Storyboard [delete as appropriate] 2](#_Toc528571916)

[2 Design (12 marks) 2](#_Toc528571917)

[2.1 Hierarchy Chart 2](#_Toc528571918)

[2.2 Normalised Entity Relationship Diagram. 2](#_Toc528571919)

[2.2.1 Entity Description in standard notation 2](#_Toc528571920)

[2.2.2 Design Data Dictionary 2](#_Toc528571921)

[2.3. Form and algorithm Design 3](#_Toc528571922)

[2.3.1 Form1 3](#_Toc528571923)

[2.3.2 Form2 3](#_Toc528571924)

[2.3.1 Form3 3](#_Toc528571925)

[2.4 Report Design 4](#_Toc528571926)

[2.4.1 Report 1 4](#_Toc528571927)

[2.4.2 Report 2 4](#_Toc528571928)

[3 Testing (8 marks) 4](#_Toc528571929)

[3.1 Test Plan 4](#_Toc528571930)

[3.2 Test Results 4](#_Toc528571931)

[4 Evaluation (4 marks) 4](#_Toc528571932)

[4.1 Objectives comparison 4](#_Toc528571933)

[4.2 Improvements 4](#_Toc528571934)

[4.3 Analysis of 3rd party feedback 4](#_Toc528571935)

[5 Technical Solution (42 Marks) 4](#_Toc528571936)

# 1. Analysis

## 1.1 The problem: identification and background

I wanted a simple simulator for the interactions between various insects, soil and plants within an enclosed 2-dimensional space. Things like this are hard to find. The closest matches I’ve found are 3D, strategy games, or focus on interactions between the user and a small number of simulated creatures. This software would be for experimentation and observation, similar to a real-life ant farm but without risking the lives of real insects. It could also serve as a prototype for a more complex software involving more realistic insect behaviours, which could be used academically.

## 1.2 Description of the current system

Current versions of what I want are difficult to find examples of. They are either non-existent or obscure. Nearest matches are strategy games (such as Empires of the Undergrowth) and marine life simulators (such as Orb Farm), or Conway’s Game of Life. These are the three examples I will compare this software to for the purposes of analysis, as they each represent a different genre of software with similarities to the software I intend to create.

Empires of the Undergrowth is a real-time strategy game in which the user controls a colony of ants using pheromone markers. It features base-building, resource management and combat as its main gameplay, with various gamemodes such as a story-based campaign and a “battle arena” for pitting creatures against each other in simulated combat.



Figure I: Player-controlled ants in Empires of the Undergrowth

The game focuses on strategy over realism or observation, and it can only demonstrate insect behaviour with constant user interaction.

Orb Farm is a 2D ecosystem simulator inspired by falling sand\* games, that allows the user to fill an aquarium with various materials and organisms. Of the examples, it is the closest match to what I’m trying to create. It features various aquatic lifeforms and substrates, and the main focuses of gameplay are maintaining a balanced ecosystem and observing the actions and interactions within the aquarium. I hope to create something similar to Orb Farm, but with terrestrial life rather than marine, and I will be omitting Orb Farm’s complex environmental factors such as water O2 and CO2 levels. The creator of Orb Farm, Max Bittker, has also made another simulator called Sandspiel, which is not water-based, but it is seemingly quite experimental and favours pseudo-chemistry over life simulation; it more closely resembles the falling sand games which inspired it, and does not feature the appropriate elements to simulate insect behaviour in the way that I am looking for.



Figure : Orb Farm

Conway’s Game of Life is a cellular automaton simulation created by mathematician John Conway. It is a zero-player game and Turing complete. Like Orb Farm, Life is a 2D life simulator, but it is incredibly simple, with a small set of rules, but allowing for infinite possibilities. Life is capable of many things, but the average user would most likely be unable to simulate simple insect behaviour using it.

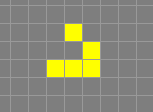


Figure : a simple automaton known as a glider in Conway's Game of Life

\*Falling sand games are a genre of sandbox game in which the user is given an empty space and a collection of elements which interact in various ways. Typically, gravity is a significant factor, and elements take the form of single-pixel particles, behaving like powder when placed in large quantities.

### 1.2.1 Problems with current system

Each of the systems described in the previous section has similarities to what I’m looking for, but they all have major differences. Empires of the Undergrowth is relatively similar in theme, being insects interacting with each other and their environment, however it is mechanically very different and does not have the same use, being a player-oriented strategy game. It is not suitable for simulating insect behaviour, as the player has limited view and is required to direct the actions of the insects. It is also not something a user can watch the development of, due to its reliance on player interaction. Orb Farm is a close match to what I want, but is entirely water-based. It meets most of the criteria, but instead of bees, beetles and ants it simulates the behaviour of fish, algae and other small marine life. Conway’s Game of Life is far too simple, simulating cellular automata with extremely simple behaviour instead of animals or plants.

## 1.3 Objectives

1. The user interface will include a view of the “terrarium” in the form of a 2D grid of square spaces occupied by “tiles” (simulated lifeforms or materials).
2. There will be at least 9 types of tile that can be added to the terrarium by the user and perform different functions
3. There will be tiles representing bees, which move across the terrarium by “flying” (moving through empty “air” tiles)
4. Bees will move towards flower tiles, and create honey tiles when they return to their nests afterwards
5. Bees will construct nests or hives by changing air tiles into hive tiles, arranged into multi-tile arrangements with an interior that is visibly separate from the outside
6. Bees will “attack” beetles in their nests by surrounding them so that no air tiles remain adjacent to the beetles
7. There will be tiles representing ants, which move by “walking” (moving through empty spaces while remaining adjacent to other “solid” tiles such as substrate)
8. Ants will attack beetles by surrounding them so that at lease three ants are adjacent to the beetle. This will cause the beetle tile to be replaced with either an air tile or an insect food tile, depending on what I think is more suitable when I get to that point
9. Ants will move substrate tiles to create burrows and anthills, within which they will cause ant egg tiles to appear
10. Ant eggs, bee eggs and beetle eggs will turn into ants, bees and beetles respectively, as long as they receive sufficient food
11. Some tiles, specifically those representing living things, will track how much food and water they need, and act in ways that allow them to get more if there is food and water available
12. The user interface will include a selection of checkboxes which allow the user to select which type of tile they wish to place
13. The user will be able to replace tiles within the terrarium by clicking on them, replacing them with the tile type they have selected
14. Selecting one tile type from the list of checkboxes will deselect all other tile types so that the user only has one type of tile selected at a time
15. There will be tiles representing plant stems, which create more plant stems (based on availability of water), share water with adjacent plant stems and create flower tiles
16. There will be tiles representing snails, which eat plant stems and flowers (turn them into air, and gain food) and walk like ants
17. There will be an option for the user to allow snails to randomly appear. This is for the sake of spontaneity, which some users may find pleasant. It is optional so that snails do not disrupt the experience of users who do not want that randomness
18. All tiles will perform their functions once before each time the visible grid updates, to show everything moving simultaneously
19. Tiles will not be able to move outside the boundaries of the terrarium
20. Tiles representing water will “fall” into empty spaces below them if there are any that aren’t obstructed by other solid tiles
21. There will be tiles representing beetles, which walk like ants, eat honey and eggs of other creatures and lay eggs

## 1.4 Data Modelling

## 1.4.1 Analysis Entity Relationship Diagram // Object Diagrams // Storyboard

Parent class Creature: child classes bee, ant, beetle, snail

Parent class Object: child classes substrate, ant egg, bee egg, water

# 2 Design (12 marks)

## 2.1 Hierarchy Chart

## 2.2 Normalised Entity Relationship Diagram.

### 2.2.1 Entity Description in standard notation

### 2.2.2 Design Data Dictionary

For each table in your database describe as shown below:

Table XXX

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |

Table YYY

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Description** |
|  |  |  |
|  |  |  |
|  |  |  |

## 2.3. Form and algorithm Design

### 2.3.1 Form1

#### 2.3.1.1 Form Screen Shot

#### 2.3.1.2 Structure Chart

#### 2.3.1.3 Class Diagram

#### 2.3.1.4 Pseudo Code

#### 2.3.1.5 Validation

### 2.3.2 Form2

#### 2.3.2.1 Form Screen Shot

#### 2.3.2.2 Structure Chart

#### 2.3.2.3 Class Diagram

#### 2.3.2.4 Pseudo Code

#### 2.3.2.5 Validation

### 2.3.1 Form3

#### 2.3.3.1 Form Screen Shot

#### 2.3.3.2 Structure Chart

#### 2.3.3.3 CLass Diagram

#### 2.3.3.4 Pseudo Code

#### 2.3.3.5 Validation

## 2.4 Report Design

### 2.4.1 Report 1

#### 2.4.1.1 Screen shot

#### 2.4.1.2 SQL and or Pseudo Code

### 2.4.2 Report 2

#### 2.4.2.1 Screen shot

#### 2.4.2.2 SQL and or Pseudo Code

# 3 Testing (8 marks)

## 3.1 Test Plan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Objective Number** | **Area tested** | **Test data** | **Description purpose** | **Expected outcome** | **Output reference** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## 3.2 Test Results

Screen shots or mention the video.

# 4 Evaluation (4 marks)

## 4.1 Objectives comparison

## 4.2 Improvements

## 4.3 Analysis of 3rd party feedback

# 5 Technical Solution (42 Marks)