

ZEUS  
Zombie Epidemic Universe Simulator

Czar Ian Echavez

(24008064)

BSc Computer Science

Project ID: 370

Module: SE3IP11

Supervisor: Professor Atta Badii

# Abstract

ZEUS is a simulator that can be used to create a visual model of a theoretical zombie infestation. The simulator has a dual use as it can be used to model the aforementioned zombie infestation, or the simulation can be used to model a realistic spread of a pathogen. Users can set up a simulation which contain the parameters of the zombie infestation (or pathogen) as well as the scenarios which may be involved. The simulation may also save these simulations as well as load and use the simulations, to allow for the simulations to be passed across different users.

# Acknowledgements

I would like to thank Professor Atta Badii and Tom Thorne for supervising the project through its design and development as well as the documentation of the project.

Contents

[Abstract 2](#_Toc509860952)

[Acknowledgements 3](#_Toc509860953)

[1. Introduction 5](#_Toc509860954)

[2. Requirements and Analysis 6](#_Toc509860955)

[2.1 Case Studies (Existing examples) 6](#_Toc509860956)

[2.1.1 GLEAM Simulator 6](#_Toc509860957)

[2.1.2 Zombietown USA 7](#_Toc509860958)

[2.2 Simulation Mechanisms 7](#_Toc509860959)

[2.3 Use Case 7](#_Toc509860960)

[3. Design Specification 9](#_Toc509860961)

[4. System Development 10](#_Toc509860962)

[5. Sub-system Conformance Testing System Integration 11](#_Toc509860963)

[6. Usability evaluation 12](#_Toc509860964)

[7. Project outcomes (lessons learned) 13](#_Toc509860965)

[8. Conclusions and Evaluation 14](#_Toc509860966)

[9. Future of the project 15](#_Toc509860967)

[10. Appendices and References 16](#_Toc509860968)

# Introduction

# Requirements and Analysis

## Case Studies (Existing examples)

### GLEAM Simulator

The GLEAM Simulator [[[1]](#endnote-1)] is a two-part system composed of a Client and server application. The system uses the server to run the simulations, none of the computations are done on the user’s computer. The GLEAM system uses the client application to interact with the server. The client application itself is also split into smaller modules. The client application is split into a simulation builder, simulation manager and the simulation visualiser.

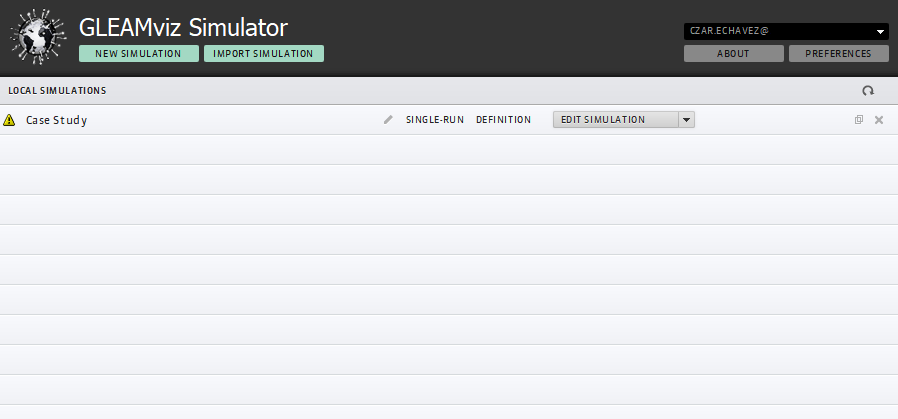


Figure 1 Simulation Manager

A user can track his/her simulations using the simulation manager. The simulation manager holds the simulations that are both complete and incomplete. Complete simulations can be submitted to the GLEAM server for the simulation to be run. Figure 1 shows the simulation manager’s user interface.

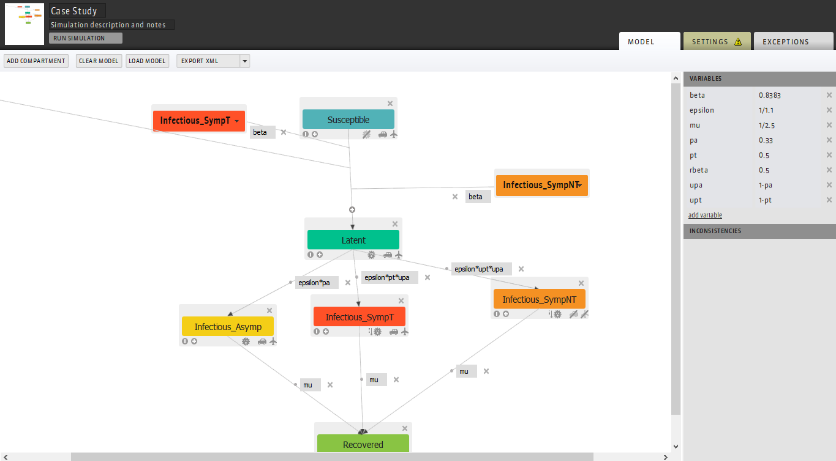


Figure 2 Simulation Builder

The simulation builder is where a user can create the spreading logic of the simulation i.e. how the disease propagates during the simulation as well as defining the starting point of the simulation. Figure 2 shows the user interface for the simulator builder.

The simulation builder works on the basis that each object on screen is a compartment and that connections between the compartments are transitions, each with a variable name (with set values) which are used to calculate the spread of the disease.

The simulation visualiser allows the user to see the results of the simulation. The visualiser has multiple settings and widgets to visualise the spread of disease in multiple ways. The world map remains the same however can be reskinned to highlight certain parts such as visualising which country is most susceptible to a pandemic. As shown in Figure 3, the visualiser also displays a graph of the new infections per day; other graphs can also be displayed.

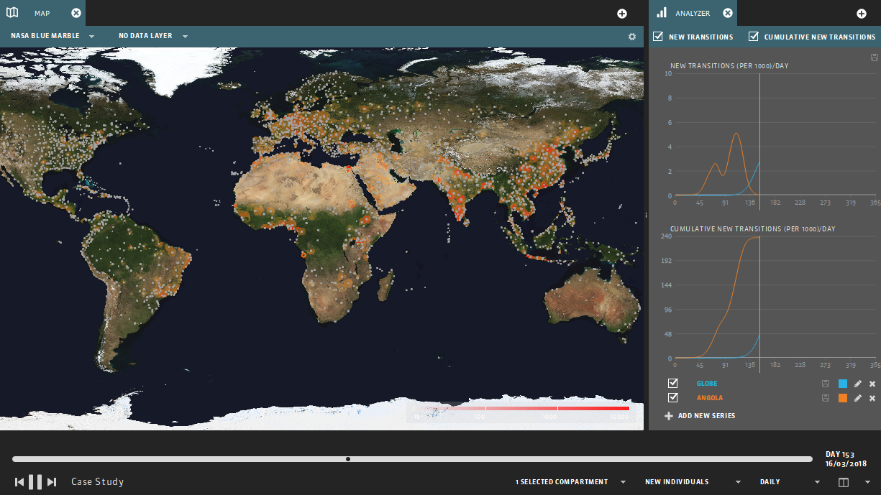


Figure 3 Simulation Visualiser

#### Evaluation of GLEAM

GLEAM has a great user interface design, however, some aspects of the software require the user to refer to the manual [[[2]](#endnote-2)]. The simulator builder has on screen help, however is not clear enough for a user to be able to make sense of what should be done in order for a simulation to be deemed runnable. Only after referring to the GLEAM user manual is it possible to understand how to create a simulation that the system deemed complete to be run. GLEAM breaking down its system into multiple pieces (i.e. the manager, builder and visualiser) also means that the software is not cluttered with a large amount of settings, ensuring that only the settings available in the current application is relevant to the action that the user needs to do; the simulation manager only allows the user to create and manage existing simulations, the builder only allows the user to create the disease mechanism and the simulation scenario, etc.

GLEAM is a simulator for realistic diseases, however, the goal of this project is also to create a zombie simulator. ZEUS still requires the use of disease spread, as it is assumed that the process of being turned into a zombie is that a person is first infected before becoming a zombie (assuming that the time for an infected to become a zombie is not the same as the incubation period)

Taking note of GLEAM’s flaws would help in ensuring that the development of ZEUS has an easily understandable user interface as well as provide a good user experience when using ZEUS. ZEUS needs to be set to only one simulation logic to simplify the simulator, therefore this is taken into account during the development process.

### Zombietown USA

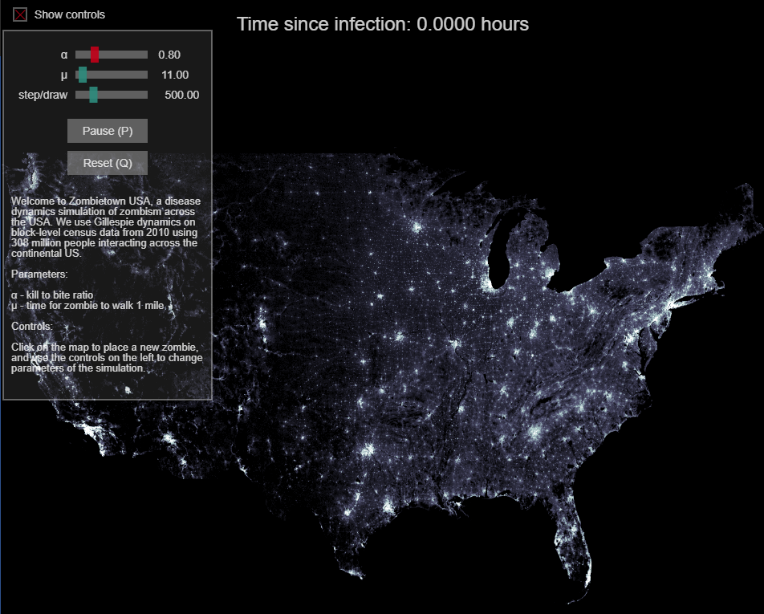


Figure 4 Zombietown USA demo

Zombietown USA [[[3]](#endnote-3)][[[4]](#endnote-4)] is JavaScript based application created by Matt Bierbaum and Alex Alemi.

The application is a simple simulation of how the zombies propagate through a population; the areas of high population are the lighter parts of the map, shown in Figure 4. The application only simulates the spread of zombies within the US and does not spread around the world.

The user interface for the application is also very simple, requiring only 4 different inputs from a user; the 3 variables the user can set, and the location(s) where the zombies can spread from.

Figure 5 shows the simulation after a few steps to show the spread of the zombies. The zombies are shown in red while healthy populations are still white (and grey). The simulator emphasises that the higher the population, the faster the zombies propagate through the region, hence the spread of zombies showing bumps on regions of high populations.

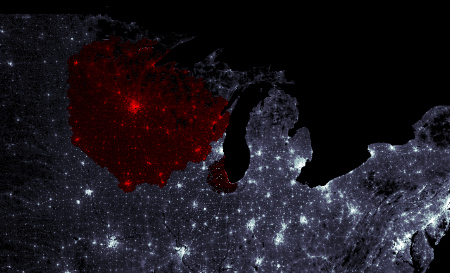


Figure 5 Infection Spread

## Simulation Mechanisms

## Use Case

|  |  |
| --- | --- |
| **Element:** ZEUS Simulator GUI | **Use Case ID:** 1 |
| **Stakeholders/Interested Parties:**   * Epidemiologists * Pathogen Researchers * Hobbyists | |
| **Primary Actor:** Software User | |
| **Description:**  The user of the software will need a way to interact with the software. A Graphical User Interface will provide a user with proper prompts as well as output the correct responses depending on the input(s). | |
| **Trigger(s):**   * User gives mouse inputs via clicking or moving move around the screen * User gives key inputs via the keyboard or on screen keyboard | |
| **Conditions:**   * User needs to have a keyboard * User needs to have a mouse * Keyboard and mouse need a way to communicate with the computer running the software, either via physical wire or wireless connection | |
| **Event flow:**   * User gives input into keyboard or mouse * Keyboard or mouse transforms input into string/numerical values the computer can process * The input is passed into the software * The software check if the mouse or keyboard input is relevant and performs actions based on the input | |
| **Alternate flow:**   * User inputs values to keyboard and mouse but neither are connected to the computer, therefore the input is not detected * User inputs values but the software is not running therefore inputs are not detected * User inputs values but the inputs are irrelevant to the software | |

# Design Specification

# System Development

# Sub-system Conformance Testing System Integration

# Usability evaluation

# Project outcomes (lessons learned)

# Conclusions and Evaluation

# Future of the project

# Appendices and References

References will be in the format:

<Object being referred to> [<Reference type: Book, Journal, Online, etc>] [<Access date>]

<If online, the link to the reference || otherwise the book ISBN or journal ID>

1. [] GLEAM Simulator [Online] [Accessed 26 March 2018 09:33]

   <http://www.gleamviz.org/simulator/> [↑](#endnote-ref-1)
2. [] GLEAM Simulator version 6.8 Manual [Online] [Accessed 26 March 2018 20:10]

   <http://www.gleamviz.org/simulator/GLEAMviz_client_manual_v6.8.pdf> [↑](#endnote-ref-2)
3. [] Zombietown USA Source (Github page) [Online] [Accessed 26 March 2018 20:43]

   <https://github.com/mattbierbaum/zombies-usa> [↑](#endnote-ref-3)
4. [] Zombietown USA Browser Demo [Online] [Accessed 26 March 2018 20:45]

   <http://mattbierbaum.github.io/zombies-usa/>

   [↑](#endnote-ref-4)