MineSweeper 3D

Isaiah Martinez
CSUN
Computer Science Department
isaiah.martinez.891@my.csun.edu

03/08/2024

Contents

1 Change History

Version: 0.09
Modifier: Isaiah Martinez
Date: 2/19/24
Description of Change: Discussed High Level Architecture of the project: Unity for modeling, C#/Python for helper script, Python for ML training, and JS for API connectivity. Added template to follow for documentation. Structured git repo directories. First, we will be working on a set amount of locations with small amount of available traffic data. Later, we hope to implement API connectivity to obtain traffic info and map data with more locations.

Version: 0.05
Modifier: Isaiah Martinez
Date: 2/12/24
Description of Change: Made Git repository. Looked at scholarly articles for related works. Uploaded sample scholarly article to view. Laid out big ideas for project. Began work on models to be used in Unity.

2 Introduction

2.1 Background Information

This document describes the architecture, design, and implementation for the popular game MineSweeper. See Here for more information. The project was done in Unity and utilized the usage of several scripts as well as assets which are acknowledged within the Acknowledgements section at the end of this document.

The title for this project: **MineSweeper 3D** is named so due to the project utilizing 3D elements to construct the game. Each cell within the visible board is a 3D element which casts shadows and reflects light. Not all elements, however, within the game utilize 3D structures, e.g., the menus.

This project offers players a new take on the classic game of MineSweeper by expanding into the 3rd dimension.

2.2 Project Information

This project addresses the interests of the major stakeholders including:

- The professor — wants a project that utilizes modern software development techniques, game design strategies, and smart application of the ideas discussed in class. Additonally, there is the desire for quality code design, precise implementation, and unambiguous documentation leading into the fulfillment of Software Development Tasks such as usability, reliability, etc.

- Players of the game — they want a functioning product with thoughtful insight into the design and playability of the game. The game should be as bug-free as possible. Furthermore, the game should be fun.

- The Developer — wants the code to be properly structured, commented, and legible. On top of maintaining high quality code, the elements within the various parts of the project not in code (those within the Unity Design Window) should also be held to the same caliber.

The design of this project is highly complicated and necessitates the documentation to descrbe the project from a variety of perspectives:

1. Logical View — components accompanied by their operations and attributes. Additionally, the relationships between the components amongst other components.
2. Process View — the processes that handle execution for components described in the Logical View
3. Development View — the larger scale view of the project including those described in the Process and Logical Views.
4. Use Case View — describe the end user interactions with the system. These serve as both a guideline for project construction as well as defining the user requirements.

2.3 Game Information

The Rules to play this game can be seen below in Figure 1. These are locationed on the publicly visible Github Repository seen here. Also located on the Github Repository is a ReadMe file which describes the process for downloading and playing the game for oneself.

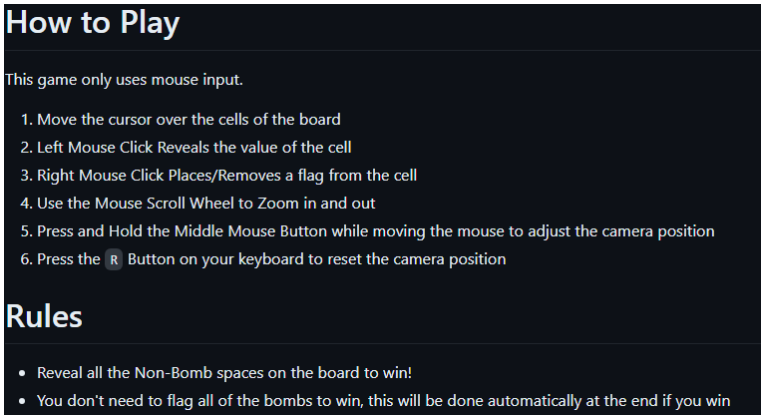

Figure 1: The rules of the game and how to play.