DAT602 – Assessment 1

Assignment 1 – Database Driven prototype game application

Alex Herbert

27/07/2024

## Introduction

This project is to design and implement a “point and click” game application that allows two or more players to play against each other at the same time from different devices via a network. Players turns are to be stored in the database in such a way that the database can be used to implement real-time gameplay. All rules, updates of moves, kills, picking up of items, or similar inventory functions during game play are to be implemented using TSQL procedures and functions. The main core functionality of the game application will be controlled using the database and the GUI is to merely provide the users with a platform to view the game and perform actions. The system that is to be implemented must support multiple copies of the game to be ran simultaneously. When a player is logged into their user account, the system will display a list of other online players and their highest score. Play is started by selecting an opponent that is currently not in a game, send a request, once both parties have accepted, they will start a new game.

## Game Concept

**Game Title:** Mouse Trap

Players control a mouse that must collect wedges of cheese that are scattered throughout the map, the map will be a basic maze grid, so players must navigate the maze to find the cheese. Along the way there will be mouse traps that spawn in random locations, these will block pathways so the player will either back track and try a different path, risk triggering the trap (this will have a chance of killing the player or taking of a large portion of health) or use an item that destroys the mouse trap clearing the path. The items that can spawn will be paper clips, which are used to trigger mouse traps to clear paths, and peanuts that can be used to regain health. If a player is killed by a trap, they will respawn at the starting tile of the map and will have to start again, any items/cheese they had collected before dying will be removed and redistributed into new tiles across the map.

The winner of the game will be the player that collects the most amount of cheese and makes their way to the centre of the map symbolised as a Mouse House.

## Game mechanics and rules

**Players and Movement**

* Players control a mouse character and navigate a maze to collect wedges of cheese.
* Players can move in four directions, up, down, left, right.

**Items and Obstacles**

* **Cheese**: Collected to score points.
* **Mouse Traps**: Randomly spawned to block pathways on the map. Can be destroyed with paper clips or the player can risk triggering the trap which may kill the mouse or deal a significant amount of damage.
* **Paper Clips:** Collect these items to be able to disable mouse traps.
* **Peanuts:** Collect to regain health points.

**Respawn Mechanics**

* On death, due to a mouse trap, players respawn at the starting tile, all their progress is reset and the items they have collected are redistributed around the map.

**Winning Condition**

* Collect the most cheese and reach the centre of the map to win the game. Score is saved and stored on the players account if it is a new high score.

## TABLES

To implement a game application system that utilizes a database as the core system that contains the logic, functions, and stores player and game data, tables or entities need to be established that serve a specific purpose within the program. These tables will either be used to store Player information such as username, password, health and current game, or they will be used to store specific game logic and values, such as trap locations, item locations and values, and games that have been started or completed.

**Player Table**

**Attributes**

* **PlayerID:** Unique Identifier for individual users.
* **Username:** Players chosen username that is displayed whilst in game.
* **Password:** Players password for accessing their account.
* **High Score:** Players highest recorded score.
* **CurrentGameID:** Used to track the players current game they are connected to.
* **Health:** The current health values for the player.

**Game Table**

**Attributes**

* **GameID:** Unique Identifier for individual games in progress.
* **Status:** Status of a game (In progress, waiting, completed)
* **StartTime:** Time that a game was started.
* **EndTime:** Time that a game was completed.

**Item Table**

**Attributes**

* **ItemID:** Unique Identifier for individual game items.
* **ItemType:** The type of item (Cheese, Paper Clip, Peanut)
* **PositionX:** The X position or co-ordinate of the item on the game board.
* **PositionY:** The Y position or co-ordinate of the item on the game board.