Analyse and design a MySQL C# based game. Prototype TSQL procedures for game functions. Revise and refactor the codebase and GUIs. Implement multiuser concurrency.

DAT602 Project

MySQL Game

13/11/2024

Jayden Houghton – 13524050

Contents

[Design (Milestone 1) 2](#_Toc175830815)

[Game Description 2](#_Toc175830816)

[Graphical Design 4](#_Toc175830817)

[Logical ERD 11](#_Toc175830818)

[CRUD Table 11](#_Toc175830819)

[DDL SQL 14](#_Toc175830820)

[Prototype Application 19](#_Toc175830821)

[SQL Procedures (Milestone 2) 22](#_Toc175830822)

[Multiuser Concurrency (Milestone 3) 22](#_Toc175830823)

[References 22](#_Toc175830824)

# Design (Milestone 1)

## Game Description

**Concept**

Survivor Island is a 2D point-and-click style game. Two or more players can verse each other, racing to collect items and stay alive the longest. Plays can click on tiles to move around the 2D tile map. The island is covered in an assortment of items, including food to heal health, treasure to increase the score. Fighting animals will award score points but reduce health.

**Items**

Once on a tile with an item, players can click again on the tile to interact with the item. Food items will be added to the players inventory and treasure items will be instantly converted to score. Items are spawned at random empty tiles throughout the game.

Picked up items are added to the inventory and stacked in designated slots, while treasure is automatically converted to score. Use buttons will use the related item and apply its effect.

|  |  |  |
| --- | --- | --- |
| Item | Type | Effect |
| Fruit | Food | Health +20 |
| Meat | Food | Health +50 |
| Coin | Treasure | Score +1 |
| Gem | Treasure | Score +5 |

**Hunting**

NPC animals will spawn at random tiles around the map. Moving to a tile with a creature and clicking the tile again will trigger the fight calculation. Successfully hunting an animal will award score points. Animals will wander randomly around the map with a simple algorithm.

|  |  |
| --- | --- |
| Animal | Effect |
| Spider | Health -10, Score +5 |
| Snake | Health -30, Score +10 |
| Boar | Health -80, Score +20 |

**Scoring and Winning**

Score points are gained through picking up treasure and hunting animals. The game ends when there are no items and animals left (as there is no way to gain score). The player with the highest score wins!

**Accounts**

Players can login to their personal account. If the account does not exist, it will automatically be created. Accounts can be banned by admins and will be locked if a user fails too many login attempts. Accounts store basic information about the player and all of their characters. Accounts can be deleted by the player.

**Admins**

Some accounts will have admin abilities. This includes killing running games, adding new players, updating existing player’s data, deleting player accounts and banning an account.

**Map**

The map is composed of 2d tiles, that may have other items, animals or obstacles placed on top. The map and item placement will be randomly generated. Player’s move by clicking on an adjacent tile (no diagonal). Some tiles will be filled with obstacles (e.g. rocks) which the player cannot move to.

**Lobby and Games**

Once signed in, each player will be able to see a list of all other online players and their highest score. Player’s can click on another player to send a game request. If the other player accepts, a new game is started.

**Chat**

Each game has its own text chat where players can message each other.

## Graphical Design

**Login and Register**

A screenshot of a login form

Description automatically generated

3

2

1

*The user loads up the game, entering their username and password, then clicking the submit button to login. New players enter the username and password they want to set for their account.*

1. Username is used for login and also how the player is named in the game.
2. Password is hidden for security.
3. Submit attempts to login the user.
   1. If the username is found but password is incorrect, display login failed popup (below).
   2. If username not found, display register popup (below). If clicked yes, go to lobby and create new account in database.
   3. If account is locked, display locked account. Account will be locked if 5 failed login attempts are made.

*The user entered incorrect login details.*

A screenshot of a computer error

Description automatically generated

*The username does not exist, so a new user is being made.*

A screenshot of a computer error

Description automatically generated

*The user has inputted incorrect passwords too many times and has been locked out.*

A screenshot of a computer error

Description automatically generated

**Lobby**

A screenshot of a computer

Description automatically generated

7

6

5

4

3

2

1

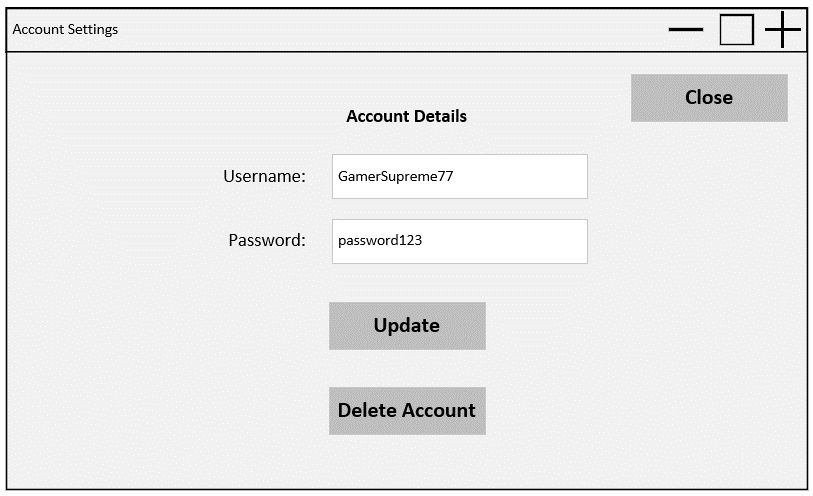
*The user has successfully logged on. Selecting the online player SammyNice from the Online Players list, the user clicks invite player. SammyNice will receive a game invite from the user, which will be listed user Game Invites. If SammyNice selects the game invite and clicks Accept Invite, the game will begin.*

*After playing a game, the user clicks Account Settings to update their password.*

*After updating their password and returning to the lobby. The user clicks Logout to log out and return to the login screen.*

1. Shows all currently online players with their highest score.
2. Select a player in the list then click invite player to send a game invite.
3. Shows all players that have invited you to a game.
4. Select an invite and click accept invite to begin a game.
5. Logs out and opens login page.
6. Opens account settings page with player data.
7. Only visible for accounts with admin. Opens admin window.

**Account Settings**



2

1

*The user wants to update their password. They replace the current password with their new password, click update, then click the close button to close the window and return to lobby.*

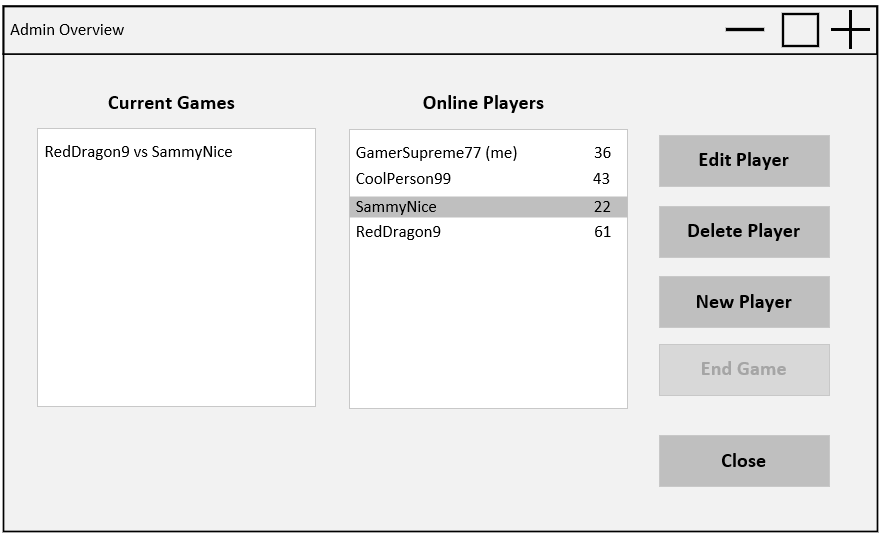
1. Updates the player info in the database.
2. Opens delete account confirmation popup. Accepting opens the login page and removes account from database.

A screenshot of a computer error

Description automatically generated

*The user clicks delete account in the account settings window then clicks OK to delete their account.*

**Admin Overview**



3

4

5

6

2

1

*After clicking Admin in the lobby, an admin user can view all online players, active games, and admin controls. Clicking edit player opens the account settings page (variant of above account settings page), where the admin user changes their password, unlocks their account, and saves changes, allowing a locked-out player to log in.*

1. Shows all current active games.
2. Shows all online players.
3. Selecting an online player and clicking edit player, opens up the account settings page (variant of above) with that player’s data.
4. Selecting a player and clicking delete player will open up the delete account confirmation popup. Clicking OK will delete this account.
5. Clicking new player will open up the make account screen (below). Clicking make account will create the account in the database.
6. Selecting an active game and clicking end game will stop and delete the game.

A screenshot of a confirmation message

Description automatically generated

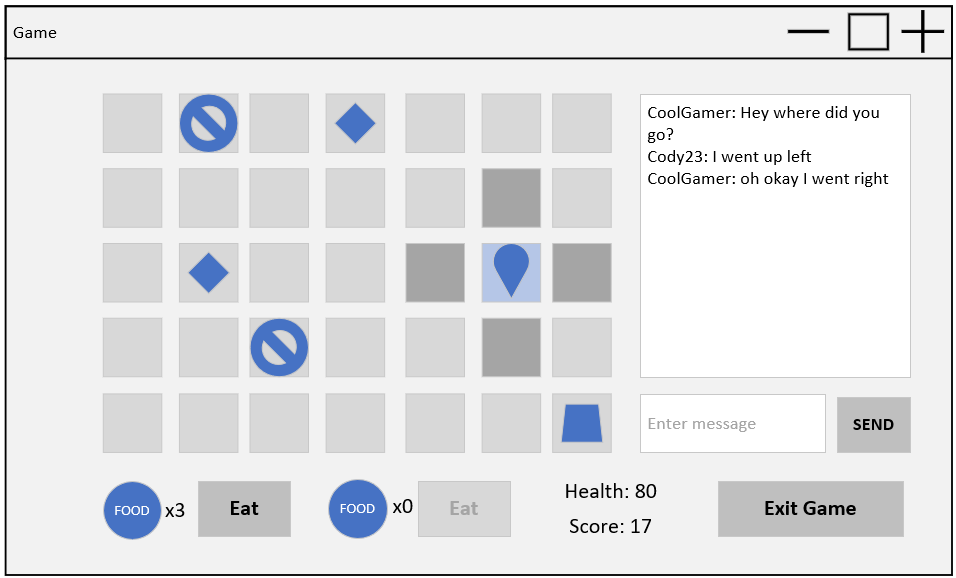
*Admin user selects an online account, clicks delete account, then clicks OK to delete user.*

A screenshot of a login box

Description automatically generated

*Admin clicks New Player from the admin page, enters username and password, then clicks Make Account to make the new user.*

**Game**



9

3

8

7

6

5

4

2

d

c

b

a

*The player clicks on adjacent tiles to move their character until they are on top of an item (e.g. gem). The player clicks the time again to pick up the item, which processes the items effects, in this case increasing score and setting the time to blank. The player moves again by clicking adjacent tiles until they are on the same time as an animal (e.g. snake). The animal moves to adjacent tiles at random intervals. Once on the same tile, clicking the tile again ‘hunts’ the snake, reducing the user’s health, increasing score and changing to a blank tile. The player clicks on adjacent tiles again to move across the board to a food item (e.g. fruit). Once on the time, the player clicks again to pick up the item, which is added to their inventory (shown bottom left of the screen). The player clicks the eat button nest to the related food item, increasing their health and reducing the quantity of that item in the inventory by one. The player then clicks on the chat message box, types out a message and clicks Send. This displays the message in the chat log, which all players in the game can see.*

Items are randomly placed when the board is loaded. The player starts on the home tile (1,1).

1. Items are randomly placed when the board is loaded. Mock-up key:
   1. Player
   2. Obstacle
   3. Item
   4. Animal
2. Exits the game back to lobby
3. Score is increased when collecting treasure or hunting. Highest score at the end of the game wins.
4. If at least one food item of the respective type is in the inventory, clicking eat will use it and apply its affects (only two food types).
5. Clicking an adjacent valid tile will move the player to that position. Valid means no player (however items or animals are excluded).
6. Animal
   1. When standing on a title with an animal or item, clicking the tile again will hunt the animal or pickup the item, applying the related effects to health, score and inventory.
   2. Animals will randomly move to adjacent times at intervals.
7. Recent chats by players in the game are displayed here
8. Any typed message will be displayed in the chat.
9. Displays health which is affected by hunting animals (reduces) or eating food (increases).

## Logical ERD

A diagram of a computer program

Description automatically generated with medium confidence

## CRUD Table

Key: C = Create, R = Read, U = Update, D = Delete

A screenshot of a computer

Description automatically generated

\* Cascade delete of all characters, games, maps, tiles, inventory items, etc.

Hote: item placement is done as part of creating a game. Each item has a related tile type, which has a percentage of being spawned at random when generating the map.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

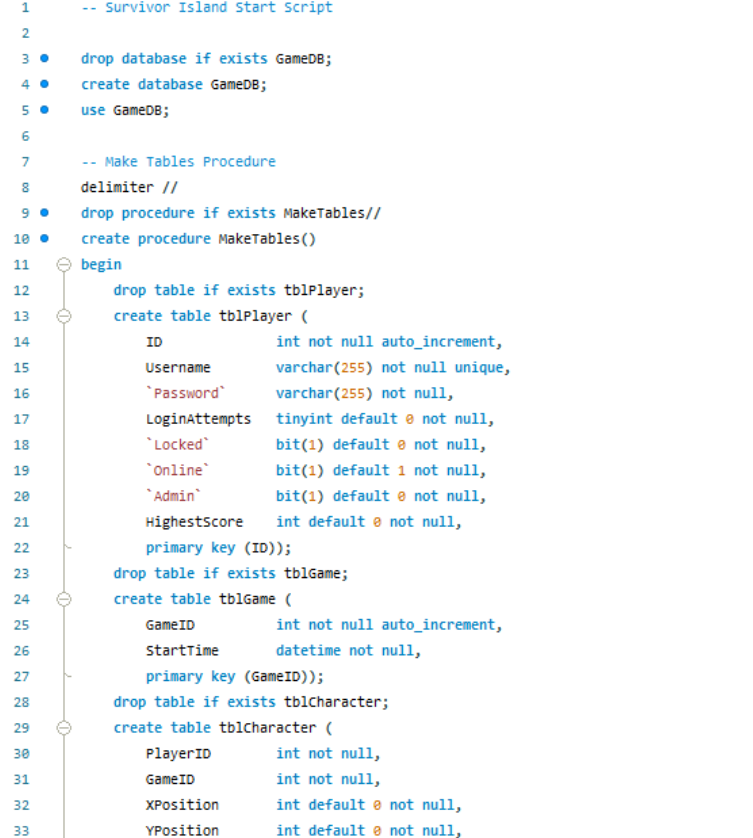
Description automatically generated

A screenshot of a computer

Description automatically generated

## DDL SQL

**Start.sql**



A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Test\_procedures.sql**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

## Prototype Application

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Note: input is the list box where admins can select players or games and click buttons to perform actions.

A screenshot of a computer

Description automatically generated

# SQL Procedures (Milestone 2)

To be done later.

# Multiuser Concurrency (Milestone 3)

To be done later.

# References

**There are no sources in the current document.**