

Project Report

Amazing Adventures

AA

DAT602

By Travis Byrman

Contents

Milestone One.....	4
Game description.....	4
Idea	4
Gameplay	4
Login and Registration	4
Lobby.....	5
Administration Interface.....	5
Storyboard	6
Storyboard 1: Gameplay	6
Storyboard 2: Login.....	7
Storyboard 3: Login – Invalid Attempt.....	8
Storyboard 4: Login – Account locked	8
Storyboard 5: Login – New Username Detected	9
Storyboard 6: Register Form	9
Storyboard 7: Lobby.....	10
Storyboard 8: Administrative settings	11
Storyboard 9: Administrative player settings	12
Storyboard 10: Player settings.....	13
Storyboard 11: Global chat	14
Storyboard 12: Character creation	14
Storyboard 13: New game form	15
Storyboard 14: Error prompt	15
Design Choices	16
Logical Entity Relationship Diagram	17
ERD Analysis.....	18
tbl_tileID → tbl_tileAsset	18
tbl_tile → tbl_character.....	18
tbl_character → tbl_backpack.....	18
tbl_character → tbl_game.....	18
tbl_game → tbl_tileAsset.....	18
tbl_player → tbl_chat.....	18
tbl_player → tbl_game.....	18
tbl_game → tbl_character	18
tbl_item → tbl_backpack.....	18

tbl_item → tbl_tileAsset.....	18
CRUD Diagram.....	19
CRUD Analysis	20
SQL Script	22
DDL Script.....	22
tbl_tile	22
tbl_tileAsset	22
tbl_item.....	22
tbl_character.....	22
tbl_backpack	23
tbl_player	23
tbl_chat	23
tbl_game	23
Test Queries/Data	24

Milestone One

Game description

Idea

I plan to create a single player multicharacter game where characters move around the map via tiles, collecting items and competing with other characters. To achieve this, I will be using a MySQL database along with C# to create an intuitive frontend and efficient backend.

Gameplay

- The aim of the game is to collect as many items as possible by moving around a tiled map. While doing this, characters must cautiously move around the map, avoiding trapped tiles. If a character moves onto a trapped tile the character will die and lose the game.
- Upon entering into the game, the character will start on the home tile (alternative tile if taken by an existing character).
- Each tile has a maximum of one item or trap. Items and traps are spawned randomly onto tiles.
- The overall game will end when all the characters have quit or died.
- Characters can move up, down, left, or right using arrow keystrokes or the buttons provided.
- If a character leaves the game and the game is still open, their progress will be saved, and they can resume.
- If a tile is occupied by a character, another character cannot be on the same tile.
- When a player tries to join a game, they will be asked to configure their character before joining.
- The game has a maximum time duration. If the game has not been completed before the time runs out, the character scores are calculated, the character with the highest score wins the game.
- Certain players can have administrator abilities. These abilities do not give the player a gameplay advantage, but give the player the ability to manage/monitor other players.

Login and Registration

- Upon logging in, if a player tries to log in with a username that is not established in the database, they will be given the option to register.
- If the player does have an existing account, they will be given the option to try to login again with the correct credentials.

- If the player attempts to login with a valid username, but the password is incorrect, they will be given 5 attempts before the account will be locked.
- If a player account is locked the player will have to ask an Administrator to unlock the account.

Lobby

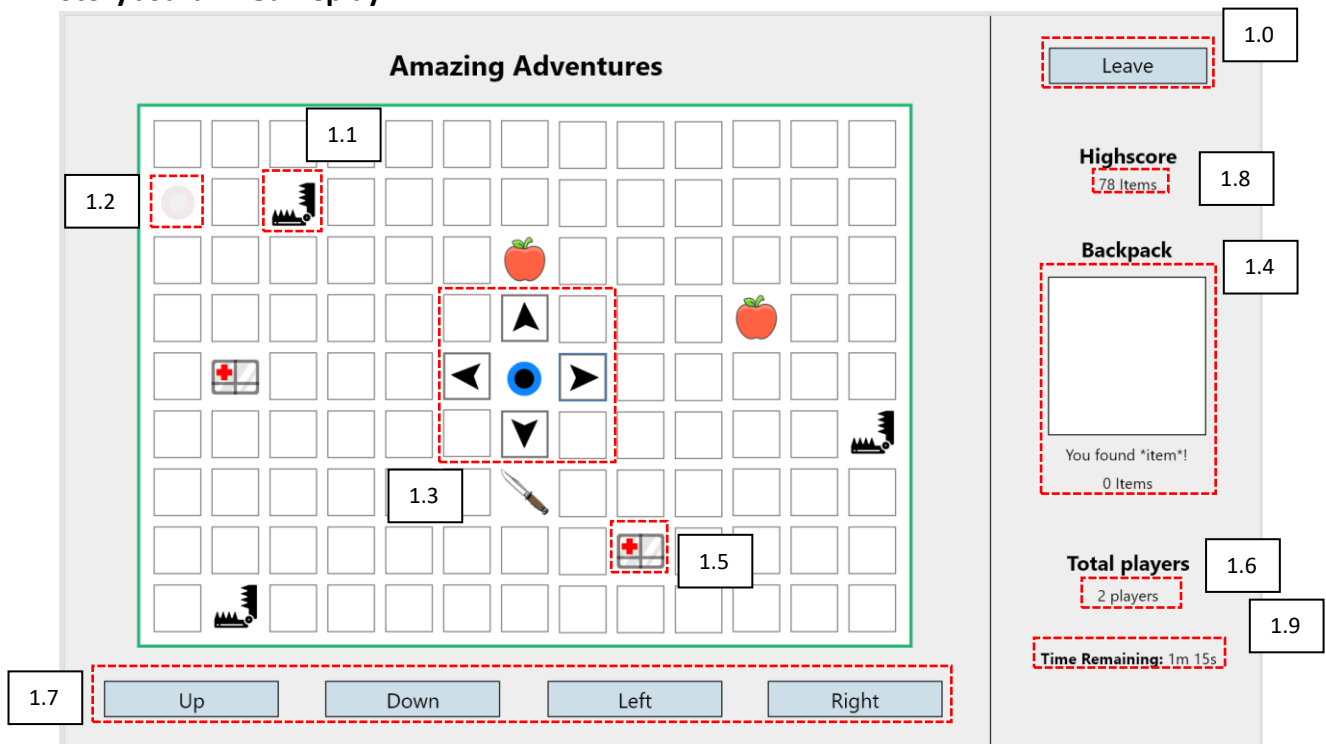
- Inside the lobby, players will be shown the total amount of players and available games they can join.
- Global chat is also available for any player to use.
- Players can also create a game where other players can join.
- If a player is an administrator, they will have access to admin settings.
- Players also have access to their settings, where they can change their username, password etc.

Administration Interface

- After selecting the admin console, administrators will be shown a total list of players and active games.
- From here administrators can lock and change player accounts and delete active games.
- Only certain players will be given administrator abilities.

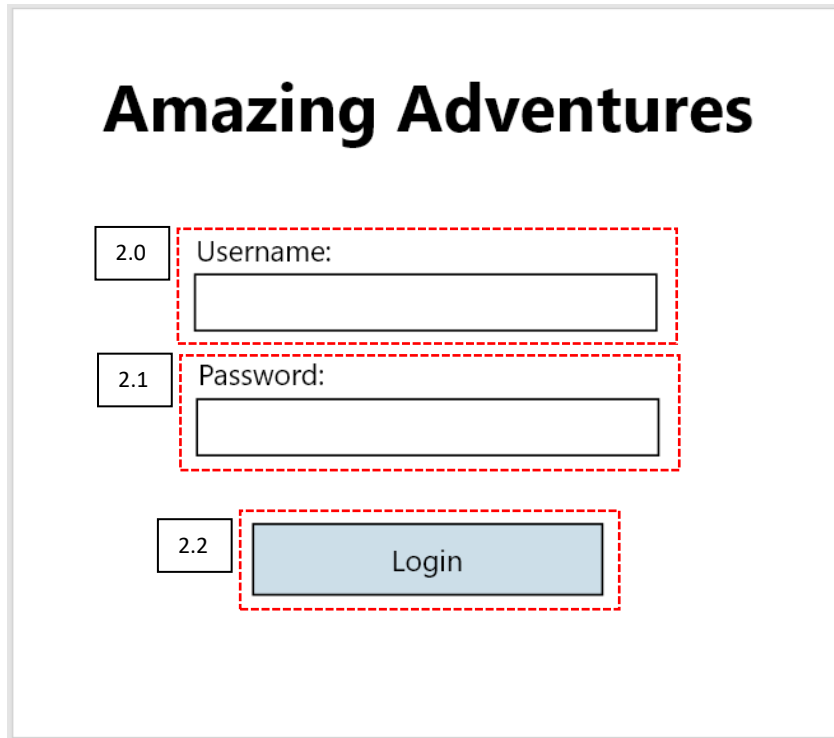
Storyboard

Storyboard 1: Gameplay



- 1.0 **Leave button** - Click to exit the game. Player is returned to lobby.
- 1.1 **Trap** – If a character moves to this tile they die and can no longer play.
- 1.2 **Other character** – This is another player.
- 1.3 **Current character** – This is the character that the player is in control of.
- 1.4 **Latest item** – This shows your total items as well as the latest item you have found.
- 1.5 **Item** – This is an item, if a character lands on this tile, they collect the item.
- 1.6 **Total player count** – This shows the total players in the game.
- 1.7 **Character movement** – These buttons can be used to move the character.
- 1.8 **Highscore** – This shows the user their highest score that have received.
- 1.9 **Time remaining** – This shows the player how much time is left.

Storyboard 2: Login



The storyboard shows a login interface for 'Amazing Adventures'. It features a title at the top, followed by three elements: a username input field (2.0), a password input field (2.1), and a login button (2.2). Each element is enclosed in a red dashed box. The login button is a light blue rectangle with the text 'Login' centered inside.

Amazing Adventures

2.0 Username:

2.1 Password:

2.2

2.0 Username field – Input field where a user inputs their player account username.

2.1 Password field – Input field where a user inputs their player account password.

2.2 Login button – User clicks this button to login. If username or password is invalid login will be unsuccessful. If username is not a valid player account username, user is given the option to register. User has 5 login attempts before account is locked.

Storyboard 3: Login – Invalid Attempt

Amazing Adventures

Username:

Password:

Login

3.0 Password Incorrect
You have 5 attempts remaining

3.0 **Prompt** – Indicates user how many login attempts remaining.

Storyboard 4: Login – Account locked

Amazing Adventures

Username:

Password:

Login

4.0 Password Incorrect
This account 'Useraccount123' is **locked**

4.0 **Prompt** – Indicates to user that the account they tried to log into has now been locked.

Storyboard 5: Login – New Username Detected

Amazing Adventures

Username:

Password:

Login

Register

5.0 New username detected
Please register above

5.0 **Prompt** – Indicates to user that the username they entered is not recognised in the database. Registration button is shown, given the user the option to register a player with that username.

Amazing Adventures

6.0 Email:

6.1 Username:

6.2 Password:

6.3 Register

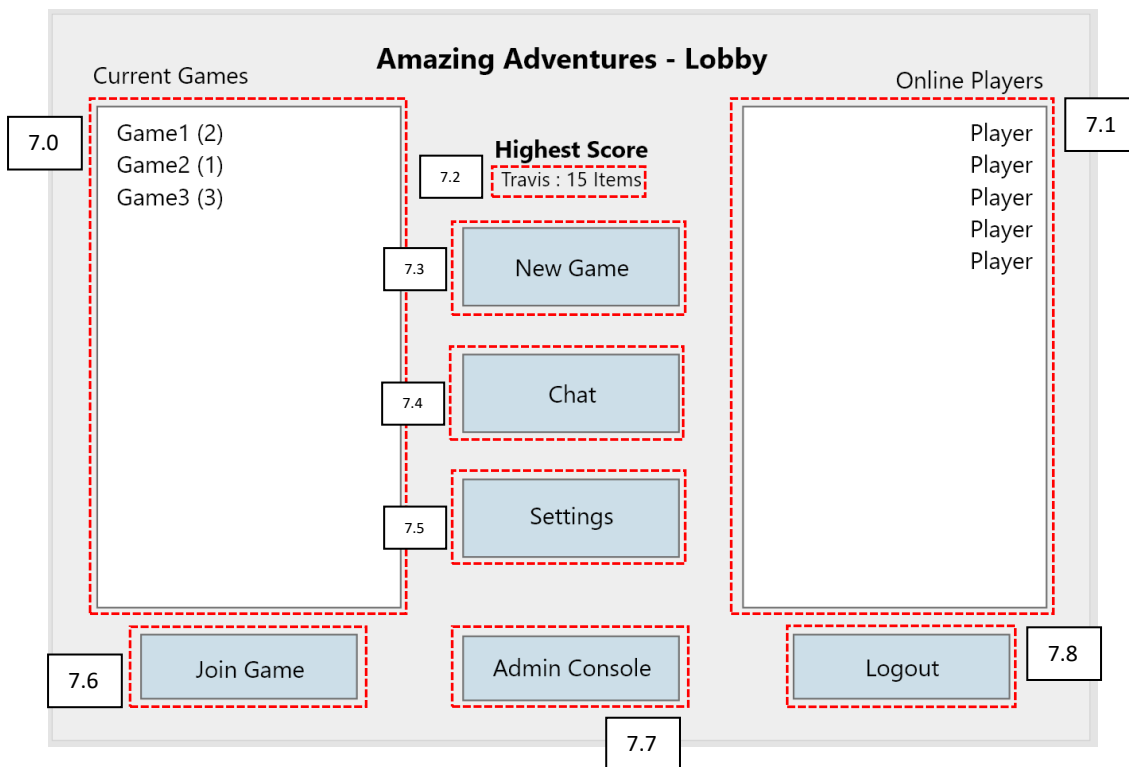
6.0 **Email field** – Input field where a user inputs their player account email address.

6.1 **Username field** - Input field where a user inputs their player account username.

6.2 **Password field** - Input field where a user inputs their player account password.

6.3 **Register button** – User clicks this button to register the account. If username, password, or email address is invalid, registration will be unsuccessful.

Storyboard 7: Lobby



7.0 **Current games list** – This shows the current games that players can join.

7.1 **Online players list** – This shows the list of players that are online.

7.2 **High-score** - This shows the players high-score (most items they have collected in a game).

7.3 **New game buttons** – Players can press this button to create a new game.

7.4 **Chat button** – Players can press this button to open the global chat.

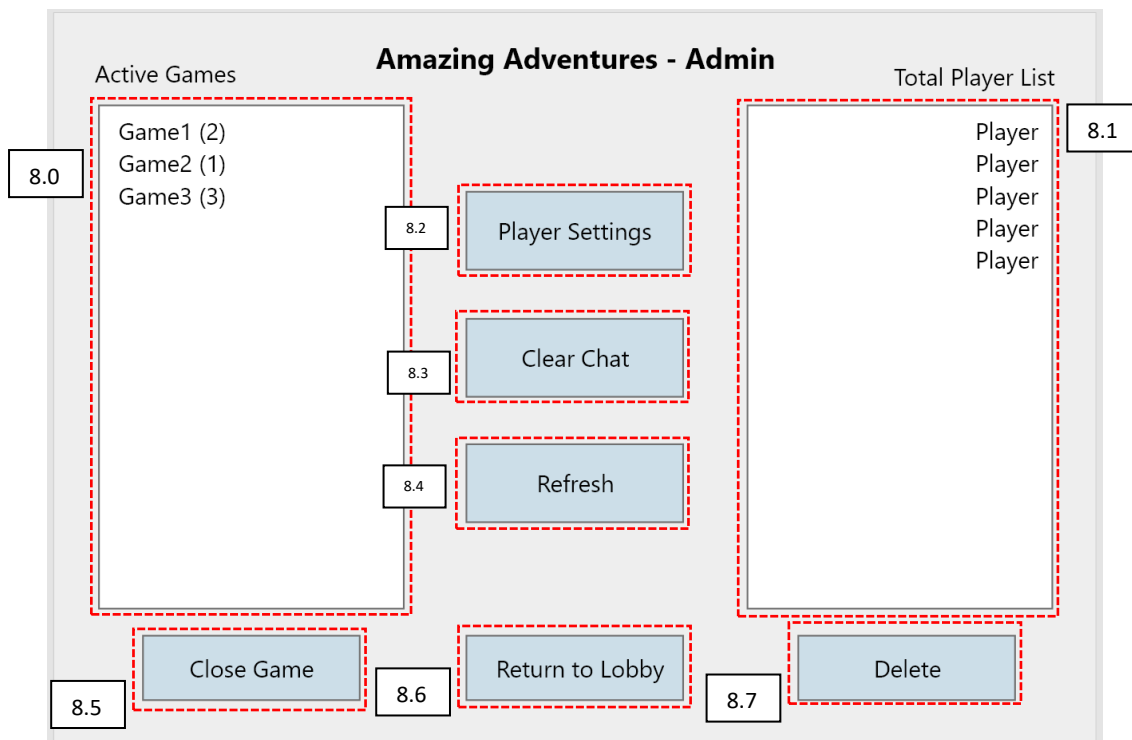
7.5 **Setting buttons** – Players can press this button to view their player settings.

7.6 **Join game button** – Players select a game from the current games list and press this button to join it.

7.7 **Admin console button** – Players can press this button open administrator settings. Only players with administrative access can see this button.

7.8 **Logout button** - Players can press this button logout of their account. User is returned to login screen.

Storyboard 8: Administrative settings



8.0 **Active games list** – This shows the current games that players can join, as well as the total player count to the left.

8.1 **Total player list** – This shows the list of player accounts created.

8.2 **Player settings button** – Administrators can press this button after selecting a player from the player list to view that players account settings.

8.3 **Clear chat button** – Administrators can press this button to remove all messages from the global chat.

8.4 **Refresh button** – Administrators can press this button to refresh the total player and active games lists.

8.5 **Close game button** – Administrators can press this button after selecting an active game from the list to close/remove the game.

Storyboard 9: Administrative player settings

Amazing Adventures - Manage Player

Username: 9.0

Password: 9.1

Email: 9.2

Highscore: 9.3

Admin: ☐ 9.4

Locked: ☐ 9.5

9.6 9.7

9.0 **Username field** – Input field where a user inputs their player account username.

9.1 **Password field** – Input field where a user inputs their player account password.

9.2 **Email address field** – Input field where a user inputs their player account email address.

9.3 **Highscore field** – Input field where a user inputs their player account highscore.

9.4 **Admin tick box** – Tick box field where a tick represents the user has administrative access.

9.5 **Locked tick box** – Tick box field where a tick represents the players account is locked.

9.6 **Save button** – Administrators click this button to save the players details. If details are incorrectly added, administrator is prompted.

9.7 **Cancel button** – Administrators click this button to cancel the players detail form. This will close the form and return the admin to the administrative settings.

Storyboard 10: Player settings

The storyboard shows a form titled "Amazing Adventures - Player Settings". It contains three input fields: "Username:", "Password:", and "Email:". Each input field is enclosed in a red dashed border and has a corresponding label box to its right (10.0, 10.1, and 10.2 respectively). Below the input fields are two buttons: "Save" and "Cancel". The "Save" button is enclosed in a red dashed border and has a label box to its left (10.3). The "Cancel" button is also enclosed in a red dashed border and has a label box to its right (10.4).

10.0 **Username field** – Input field where a user inputs their player account username.

10.1 **Password field** – Input field where a user inputs their player account password.

10.2 **Email address field** – Input field where a user inputs their player account email address.

10.3 **Save button** – Players click this button to save their players details. If details are incorrectly added, player is prompted.

10.4 **Cancel button** – Players click this button to cancel their players detail form. This will close the form and return the player to the lobby.

Storyboard 11: Global chat

The storyboard shows a chat window titled "Amazing Adventures - Chat". Inside the window, there is a text area labeled "Global Chat" containing four lines of test text: "User1: Testetstst", "User2: Testetstst", "User3: Testetstst", and "User1: Testetstst". Below the text area is an input field with a greater-than sign (>) as a placeholder. At the bottom right of the window is a button labeled "Close Chat". Red dashed boxes highlight the text area (11.0), the input field (11.1), and the "Close Chat" button (11.2).

11.0 **Global chat text** – Players will see every player's text chat and date/time of publish.

11.1 **Global chat input box** – Input field where a user inputs the text they wish to add to the chat. Once finished, press keystroke enter to submit.

11.2 **Close chat button** – Players click this button to close the global chat form. This will close the form and return the player to the lobby.

Storyboard 12: Character creation

The storyboard shows a character creation form titled "Character Select". It contains three input fields: "Character Name" (12.0), "Character Colour" (12.1) with the value "Blue", and two buttons: "Join" (12.2) and "Cancel" (12.3). Red dashed boxes highlight the "Character Name" input field (12.0), the "Character Colour" input field (12.1), the "Join" button (12.2), and the "Cancel" button (12.3).

12.0 **Character name input box** – Input field where a player inputs their character name.

12.1 **Character input colour** – Input field where a player chooses their character colour.

12.2 **Join button** – Players click this button to join the game. This will close the form and will add the player to the game.

12.3 **Cancel button** – Players click this button to close character creation form. This will close the form and return the user to the lobby.

Storyboard 13: New game form

New Game

Name:

Duration:

13.0 **Game name input box** – Input field where a player inputs their game name.

13.1 **Game duration input box** – Input field where a player chooses their games time/duration.

13.2 **Create button** – The player clicks this button to create the game. This will close the form and the player will be asked to create a character.

13.3 **Cancel button** – Players click this button to close the new game form. This will close the form and return the user to the lobby.

Storyboard 14: Error prompt

Error!

An error has occurred:
Invalid Input

Please try again.

13.0 **Ok button** – Players click this button to acknowledge the error.

When an error occurs, this form will pop up notifying the user of their error. The message in red will change depending on the error.

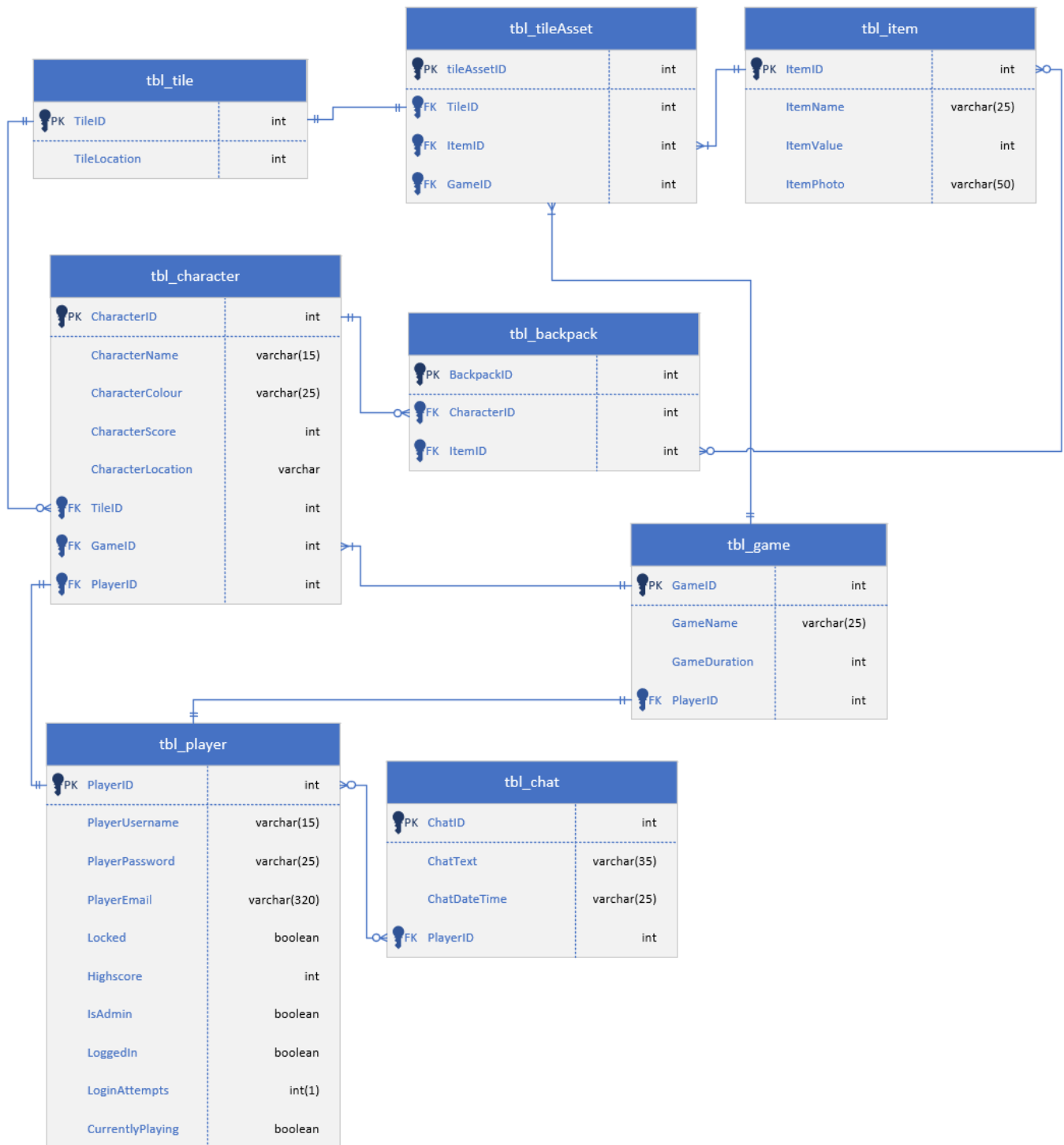
Design Choices

I choose to have a colour theme that mainly uses the colours grey, white and black. I decided to do this as I found the colours work well with each other and make the features stand out. I found that by using the colour blue and red to show errors and buttons make those features stand out against the grey background. These colours draw the user's attention.

As shown in the gameplay screenshot above, I decided to add buttons that move the player around the map. I chose to do this as it allows users who have a touch screen device to play the game. I also made the text and buttons large enough so that people with disabilities and can easily view them. It also allows a large variety of devices to be able to play this game without any difficulties due to resolution.

I have also decided to use combo box when necessary (such as for character colours) as it limits the users range of how many colours they can choose.

Logical Entity Relationship Diagram



ERD Analysis

tbl_tileID → tbl_tileAsset

The tile table contains the location of each tile. Each tile asset needs to be on a tile. One tile can only have one asset, one asset can only be on one tile.

tbl_tile → tbl_character

The tile table contains the location of each tile. A character needs to be on a tile. One tile can only have one character, one character can only be on a tile.

tbl_character → tbl_backpack

The character table contains information about the character, such as character name, colour, and score. A character can have multiple items inside the backpack table. A character can have only one item at a move, only one item can be claimed by a character on a tile.

tbl_character → tbl_game

The character table contains information about the character, such as character name, colour, and score. A character can only be in one game, one game can have many characters.

tbl_game → tbl_tileAsset

The game table contains information about the game, such as who created the game, game duration and name of the game. One game can have many tile assets, many tile assets can be in one game.

tbl_player → tbl_chat

The player table contains information about the player, such as username, password, email is it an admin etc. Many players can make submit many texts in the chat table. Many texts can be submitted by many players.

tbl_player → tbl_game

The player table contains information about the player, such as username, password, email is it an admin etc. Only one player can create a game, the same game can only have one creator (player who creates that exact game).

tbl_game → tbl_character

The game table contains information about the game, such as who created the game, game duration and name of the game. One game can have many characters, many characters can only be in one game.

tbl_item → tbl_backpack

The item table contains information about the items, such as item name, item value and item photo. Many items can be in many backpacks, many backpacks can have many items.

tbl_item → tbl_tileAsset

The item table contains information about the items, such as item name, item value and item photo. The same item can be many tile assets, many tile assets can have the same item.

CRUD Diagram

Table:	tbl_player	Process:																		
		Register Player	Check Username/Password	Lock Account	Logout Player	Remove Player	Edit Player	Create Character	Start New Game	Join Existing Game	Game Ends	Character Moves	Character Collect's item	Player Adds Chat Message	Player Changes Settings	Admin Creates Player	Admin Removes Player	Admin Locks Player	Admin Edits Player Info	Admin Ends Game
	PlayerID	C			R	R	R	R	R	R	R			R	R	RC	RD	R	R	R
	PlayerUsername	C	R				RU							R	C				U	
	PlayerPassword	C	R				RU							R	C				U	
	PlayerEmail	C					RU							R	C				U	
	Locked	C		U													U	U		
	Highscore	C			RU															
	IsAdmin	C				R	R		R						RC	R		RU	R	
	LoggedIn	C	RU		U	R	R									R		R		
	LoginAttempts	C	U																	
	CurrentlyPlaying	C			RU		R	U		U	U								R	
	tbl_character																			
	CharacterID				R			C		C	D	R	R							D
	CharacterName							C		C	D									D
	CharacterColour							C		C	D									D
	CharacterScore				R			U		U	D									D
	TileID				R			U		U	D									D
	GameID				R			U		U	D									D
	PlayerID				R			U		U	D									D
	tbl_backpack																			
	BackpackID										D		C							D
	CharacterID										D		U							D
	ItemID										D		U							D
	tbl_chat																			
	ChatID													C						
	ChatText													C						
	ChateDateTime													C						
	PlayerID													U						
	tbl_game																			
	GameID								C		D									D
	GameName								C		D									D
	Duration								C		D									D
	PlayerID								U		D									D
	tbl_item																			
	ItemID																			
	ItemName																			
	ItemValue																			
	ItemPhoto																			
	tbl_tile																			
	TileID																			
	TileLocation																			
	tbl_tileAsset																			
	tileAssetID								C		D		D							D
	TileID								U		D		D							D
	ItemID								U		D		D							D
	GameID								U		D		D							D

CRUD Analysis

By creating the logical ERD diagram first, I was able to use it as a reference to create the tables and fields for my CRUD diagram. After analysing the assessment requirements and schedule, as well as my storyboards, I would be able to create the processes needed. Here is what I found.

Register Player

When a player enters a username into the login fields, that is not recognised as existing in the database, the user will be prompted. A register button will appear, as well as a label explaining to the user that they can create an account if they do not have an existing account.

After clicking the register button, a registration form will show, getting the user's information to create an account.

Check username/password

When a user submits the login forms, checks will be made with select statements to decide whether the username and password that has been entered is correct. If not correct and the username is recognised, the user will be prompted, as well as the amount of login attempts, they have remaining to access the account.

Lock account

If the user has tried 5 times to log into the same account, the account will become locked. This means that any user will be unable to log into this account without having an administrator unlock it. Once unlocked, users will have another 5 login attempts until the same thing occurs again.

Logout player

If a user logs out of their account, by clicking the logout button in the lobby, the lobby form will close, and the user will be presented with the login form. The account will no longer be shown to be logged in.

Remove player

If an administrator decides to delete another user's player account, they will be asked to confirm this action, then the account will be removed. If the user is currently logged in, they will be logged out immediately. If a user wants to delete their player account they will be prompted to confirm, then they will be logged out.

Edit player

If an administrator wishes to change the settings of another player, they can do so by choosing a player from the player list in the administrator settings. A button will only be visible to administrators on the lobby form. After an admin selects the user, they can press player settings and change the player's settings.

Create character

When a player wants to join a game, they will be prompted to create a character. Here they can choose their character's name and colour before entering. Once entered, the game will be shown, and their character will have the corresponding features.

Start new game

When a player chooses to create a new game, they will be prompted with a form and will need to decide on the games name and duration. If incorrectly submitted, the user will be promoted with the corresponding error message.

Join an existing game

When a player joins an existing game, they will be prompted to create a character (explained above). Then they will have three seconds before the game starts for themselves.

Game ends

When the game ends the player will be given their score. If the score is greater than their highscore, then this new score will be their highscore. Once the game ends, all characters are deleted, and players are returned to the lobby. The game will no longer exist, and players will not be able to join.

Character moves

A player can move their character by using the keystroke arrows or buttons provided on the game play. When moved the database will save the players current location and add any items to their backpack if an item was on the tile.

Character collects item

When a character moves to a tile with an item on it, the item will be removed from the tile and added to their backpack. Depending on the item, the number of points will be determined by it.

Player adds chat message

When a player creates a chat message, it will be shown in the global chat where every player can read it. The players name, chat text and date/time will be provided. Chat messages can be cleared at admins choice in the administration settings.

Player changes settings

If a user wishes to edit their player account details, they can do this by clicking the settings button in the lobby. Here they will be given access to edit the details of their player account. After clicking the save button, the player details will be saved. If incorrectly inputted, the user will be prompted, and the corresponding error message will be shown.

Admin creates player

If an admin wishes to create a player account, they can do so in the administrator settings. After clicking the button, they will be shown a form where they can enter the details. If the details are incorrect the admin will be promoted the account will not be created until errors are corrected.

Admin removes player

If an admin wishes to delete a player than can in the administrator settings by selecting the player and clicking delete.

Admin locks player

If an admin wishes to lock another player account, they can do so in the administrator settings. This will make the account unavailable to any user.

Admin ends game

If an admin wishes to delete/end a game they can do so in the administrator settings by selecting the active game and clicking close game. When they do this, all characters in the game will be deleted and the players will be prompted, that the game ended by an admin. The scores will be shown, and they will be returned to the lobby.

SQL Script

Separate file attached titled AssessmentD DL.sql

DDL Script

tbl_tile

TileID – Primary key that is an auto-incremental number. This is used to link the tiles to the tileAsset table, and character table.

TileLocation – Integer, used to hold the corresponding value in which the tile can be found in an array.

tbl_tileAsset

tileAssetID – Primary key that is an auto-incremental number.

This also table consists of three foreign keys, TileID, ItemID and GameID. This table is used identify where and item is, what it is, in what game.

tbl_item

ItemID - Primary key that is an auto-incremental number. This is used to uniquely identify each item and to link the item to the tileAsset table. This is also used to link the items to the backpack table.

ItemName – Varchar field, used to identify each item by name.

ItemValue – Integer field, used to hold the value of each item.

ItemPhoto – Varchar field, used to hold the link to the item photo in the database.

tbl_character

CharacterID - Primary key that is an auto-incremental number. This is used to identify each character. This is also used to link the character table to the backpack.

CharacterName – Varchar field, used to identify each character by name.

CharacterColour – Varchar field, used to give each character a colour.

CharacterScore – Int field, used to hold the characters score.

CharacterLocation – Varchar field, used to store the characters current location in the database.

There are also three foreign keys, TileID, GameID and PlayerID. These are used to link the tile, game and player tables to the character.

tbl_backpack

BackpackID - Primary key that is an auto-incremental number.

CharacterID – Used to link the character to a backpack that is held by an item.

ItemID – Used to link an item to the backpack that is held by a character.

tbl_player

PlayerID - Primary key that is an auto-incremental number. This is also used to link the player to the character that they create per game. It is also used to link the player to a game they create. Chat messages from the chat table are also linked to the PlayerID

PlayerUsername – Varchar field, this is used for users to log into their player account.

PlayerPassword Varchar field, this is used for users to log into their player account securely.

PlayerEmail – Varchar field, this is used for users to associate themselves to their player account securely.

Locked – Boolean, used to identify when the player account is locked or not.

Highscore – Int, used to store the highest score a player gets using their character.

IsAdmin – Boolean, used to identify whether the player has administrator access/powers.

LoggedIn – Boolean, used to identify whether the player is online.

LoginAttempts – Int, used to identify how many times the user has attempted to log into their account.

CurrentlyPlaying – Boolean, used to identify whether the player is in a game or not.

tbl_chat

ChatID - Primary key that is an auto-incremental number. Used to uniquely identify each chat message.

ChatText – Varchar, used to store the players chat message.

ChatDateTime – Varchar, used to store the date and time that the message was submitted.

PlayerID – Used to link the chat message to the player. Makes the chat message identifiable and traceable back to the player.

tbl_game

GameID - Primary key that is an auto-incremental number. Used to link the game table with the tileAsset and character table.

GameName – Varchar, used to identify each game by the name given by the user who created the game.

GameDuration – Int, used to store the duration of the game, how long it needs to run for, and how long the duration is left.

PlayerID – Used to identify which player created the game.

Test Queries/Data

As shown in my SQL script, I used a variety of select, update, delete and insert statements to test the functionality of my database.

I used 3 insert statements per table, to give myself enough data to experiment with.

Over top of this I used at least 1 of each select, update, and delete statements for each table. This gave me enough data and statements to work with to test the functionality. From this I found that I needed to add cascade delete statements to my foreign keys to effectively delete records. After adding cascade delete statements, I found the functionality of my database to be great.