Asti Maera

ClickTacToe Project

Contents

[Game Design 3](#_Toc108777355)

[Game Description 3](#_Toc108777356)

[Storyboards 3](#_Toc108777357)

[New Player 3](#_Toc108777358)

[Returning Player 4](#_Toc108777360)

[Administrative 4](#_Toc108777361)

[CRUD Matrix 0](#_Toc108777362)

[Entity Relationship Diagram 0](#_Toc108777363)

# Game Design

## Game Description

The game to be created is a click-and-point adventure-horror game. The player begins in a room where they must pick the right item in order to proceed to the best next position on the map, points are added to their score for making it through a room. Bonus points can be applied for picking the best item. The item chosen determines the arrows that will be generated, allowing the player to move on to the next position of the map. Reaching the end of the map ends the game. Alternatively, going below a certain score or getting stuck in a room with no items, ends the game and the player “dies”. Players can take any leftover items to their next game if they win.

## Storyboards

### New Player

## Graphical user interface, application Description automatically generated

### Returning PlayerGraphical user interface, application Description automatically generated

Graphical user interface, application

Description automatically generated

### AdministrativeGraphical user interface, application Description automatically generatedGraphical user interface, application Description automatically generated

# CRUD Matrix

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entity |  Process | User Account | User Record | Location\_Usr | Tile | Tile\_Location | Item | Location\_Item | Inventory | Character | Structure |
| **Login**  Player clicks login with new records | C – ID  C – Username  C – Password | C - ID  C – login\_time  C – status  C – Account Type |  |  |  |  |  |  |  |  |
| Player clicks login button with known records and active game | R - Username  R- Password | R – ID  R – User\_ID C – login\_time | R – Coordinates |  |  |  |  |  |  |  |
| **Create Character**  Player creates a new character for the account | R – User\_ID |  |  |  |  |  |  |  | C – ID | R – ID  R – Style  R - Type |
| **Select Item**  Player adds an item to their inventory | U – inventory | U – Score |  |  |  | R – ID  R – point\_score  R – type  R – Structure\_ID |  | R – Item\_ID  U – Capacity  R – Account\_ID |  | R – ID  R - Style |
| **New Game**  Player enters new game | R – ID  R – Character\_ID | C - Score | R – Tile\_ID | R- ID  R – Coordinates  R – Structure\_ID | R – Coordinates  R – Tile\_ID |  |  | R – Account\_ID  C – Capacity | R – ID | R – ID  R – Style |
| **New Game**  Player moves to next tile | R – ID  R – Character\_ID | U - Score | U – Tile\_ID  U – Co-ordinates | R- ID  R – Co-ordinates  R – Structure\_ID | R – Coordinates  R – Tile\_ID | R – Location\_ID (Coordinates) | R – Coordinates |  | R – ID | R - Style |
| Player reaches the end of the game | R – ID | U – Score | D – Co-ordinates  D – Tile\_ID  D – UserAccount\_ID |  |  |  |  |  |  |  |
| **Logout**  Player leaves game |  | U – logout\_time  U – Status |  |  |  |  |  |  |  |  |
| **Admin**  Admin gives administrative status | R – ID | R – ID  R – account\_type  R – ID  U – account\_type |  |  |  |  |  |  |  |  |
| Admin deletes player | R – ID  D – User Account | R – ID  R – account\_type  R – ID  D – User Record | D – Location\_Usr |  |  |  |  | D – Inventory | D – Character |  |
| Admin changes a player’s status | R – ID | R – ID  R – account\_type  R – ID  U – status |  |  |  | U - Occupied |  |  |  |  |

# Entity Relationship Diagram

The diagram I’ve ended up with so far is quite basic but should hopefully cover most of the data requirements for the game. It may need to be revised in the next milestone, as there may be a need for one or more extra tables to cover score records or other data. There are two weak entity tables for location of entities in order to prevent the conflicts that I was finding with having many items and potentially many users tied to a tile (home tile). Potentially location could be reduced to one table as I’ve done with the Structure table. Diagram

Description automatically generated

User Account holds player identifying details. This links to their inventory, their game record, their location, and their character.

User Record holds all account details including score and status, potentially could be merged to user account.

Location\_Usr contains the whereabouts of the player at last update, this could be either upon logout or when a player dies.

Character identifies a player’s character. Players create a character for the game, each character has a base structure that can have style variations. A player can only create one character, but a character is able to be edited.

Structure holds all details for creation of items, characters and tiles. Tiles only have one set structure available to them. Items and characters have many available structures.

Item holds all item details including their expected location in the map.

Inventory holds any items that are able to be picked up and stored by the player.

Location\_Item holds the coordinates for any item on an identified tile.

Tile holds the identifying details for a tile as well as whether or not a tile is occupied, this is important for all gameplay tiles as it will allow for a player to access the tile if unoccupied.

## Test Data

When creating the database, I inserted multiple rows of test data in order to create queries and procedures for the next steps of the project. I continued to run into errors with foreign key requirements for my two weak entity tables so this would need further consideration on how best to remedy in the future.

This data was generated using an online tool, given more time I would like to apply different techniques for inserting the test data such as using php. Because the data has been automatically generated this has made some potential errors for gameplay such as login\_time and logout\_time being inordinately out of synch with realistic timeframes, co-ordinates being meaningless, and player status being a random phrase.

## References

https://github.com/astizm/DAT602