**Analysis**

**Background & Identification of the Problem**

[Research on Games similar to this] Look up civ5 the board game, use this as the predecessor

[Explain what the game is about and how a new player actually plays the game]

Firstly, the player logs into the system with an ID and password or creates a new account in order to track progress and allow them to access and create/delete save files related to their ID. After logging in, the system takes them to a form which asks them would they like to create a new game, or load a previous save.

Within the new game form, the player is prompted to choose what faction they would like to be and what size map would they like. Each faction has the same 3 win conditions (world domination, technological breakthrough and divine intervention) to try to achieve throughout the game and the 1st faction to achieve one of the three wins the game. The amount of factions in a game is dependent on the size of the map.

Within the load game form, the player is asked which save state they would like to load, if they choose 1, 2 or 3 the system will give information about the current save to ensure that the player that they are selecting the correct one. If anything but 1, 2 or 3 is entered the player is sent back to the previous form.

**The Current System**

[Look at the examples in class]

**Specific Requirements of the User and Acceptable Limitations**

[What are the key aspects of the game and what cannot be done due to limitations (with time, software, expertise)]

**Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Validation** | **Example Data** | **Comment** |
| AccountID | AUTONUMBER |  | 1 | Primary Key |
| Username | STRING | Must be between 5 and 25 characters long. | BobSmith152 |  |
| Password | STRING | Must be between 5 and 25 characters long. | UniquePass |  |
| Wins | NUMBER |  | 5 |  |
| Losses | NUMBER |  | 7 |  |
| Playtime | NUMBER |  | 60 | In Hours |
|  |  |  |  |  |

**Entity Relationship Model for Existing System**

**Data Volumes**

The system will not have to store vast amounts of data in its database. There can be up to 20 accounts within the system each allowed up to 3 save states each. Therefore the system must be able to handle up to 60 save states maximum for up to 20 accounts. There will be regular updates to information within the save states as at the end of each turn the save state is updated with new information regarding to the current game.

**Data Flow Diagrams for Existing System**

**Numbered Objectives of the Project**

1. *The system creates a new database when run for the first time, or if the database can no longer be found.*

The system must be able to identify if a database is present.

If the database cannot be located, a new database must be created correct to the database designed.

1. *The player must be able to create or login with an account on start-up of the system.*

First the player is prompted that they will need to create or login to access the system.

After this, the player is able to click one of two buttons to then enter details of an existing account or a new account.

If there are more than 20 accounts currently within the system, the player is asked would they like the system to delete accounts and their save states that have not been accessed for more than 6 months.

If it is a new account and there is less than 20 accounts, the database stores this information for future use in order to track a player’s progress so that they can continue where they left off in a previous start-up.

If it is an existing account, the details entered must be equal to the current Account ID as well as the Password before they are able to enter the system.

1. *Ability to delete saved games attached to only their account.*

Due to referential integrity, all the tile records linked to that save state must be deleted first

1. *Ability to delete an account after all saves have been deleted.*
2. *The player has the ability to create a maximum of 3 new games.*

After the player is assigned to an account ID, they are able to press a button in order to link them to a form which creates the game.

The player should be able to choose which faction they want to play as, the map size and how many turns would they like the game to last for.

When the player has finished customizing the settings to their liking they are able to press a button which saves the new game to their chosen save state (1, 2, or 3) and loads them into that game.

If the chosen save state is invalid, the user is asked repeatedly until they choose 1, 2 or 3.

If the chosen save state already contains information, the user is warned that if they continue that they will wipe an existing save state.

1. *The player has the ability to load an existing game.*

Through use of the tile table (e.g. select \* from tile where AccountID=AccountID and SaveID=SaveID into an array of records)

1. *At the end of each turn, the system automatically saves the game to the correct save state.*

Information about the game will be regularly updated to SaveState and Tile which exists inside of the database.

1. *On the first turn of a game, if the player did not tick the expert field when creating their account, they will be prompted with information about the rules of the game as well as how to win whenever they create a new game until they win 1 game.*
2. *Random and Pre-set Terrain Generation with iteration and Randomize procedure.*

Terrain is displayed through a rectangle of TImage’s named corresponding to their co-ordinates

These TImages are updated through the use of a for loop as all maps are of different size and with different tile features ( Island, Desert, Jungle, Tundra) with some being made up of randomly selected tiles

1. *Prevention of Faction Collision upon Game Creation through SQL queries.*
2. *Compile list of playing factions randomly with use of a dynamic array and Randomize procedure.*
3. *Place list of playing factions onto map and update the correct tile record in Tile table.*
4. *Squad Creation*

If the player has enough resources (food, gold and happiness) and they own the selected tile they can create a squad on their tile, which is able to move between tiles (1 tile per turn) and perform actions that have been given e.g. attack tilex1y2

1. *Settlement Advancement.*

When the player has enough resources to upgrade to a new paradigm the program should prompt the user if they would like to do so, and they also have the ability to see how much this will cost and choose whether they would like to or not.

1. *Tile Selection to Display Information.*

When a tile is clicked, display information about tile and what is on the tile (if a squad is present maybe new group box?)

1. *Turn Mechanic.*

Start off simple with end turn just increasing tile’s food, gold and happiness per turn when end turn and save map when turn is ended and when form is closed.

1. *Automatic SaveID in FmCreateNewGame (Friendly UI / Ease of Access).*

Should automatically change to 1, 2 or 3 if a save file is currently occupied

And not allow them to create more than 3 saves

1. Animations
2. Sound Effects
3. Win / lose conditions
4. Button to print off tile information at any given time

Extension:

Get a better logo/name

Get a decent background picture instead of light blue

**Custom Game** - Maybe make a new way to create a game with custom user parameters so that they can specify map length, each individual tile terrain, how many factions and what specific factions.

**Prospective User(s)**

**Proposed Method of Solution**

[Delphi combined with Microsoft Access + Reasons]

**Sources**

* d