**Analysis**

**Background & Identification of the Problem**

The system’s purpose is to provide entertainment for everyone. The game is turn-based strategy, where the objective is to outlast your opponents or defeat them in battle to win and if you are destroyed or are weaker by the end of the game you will be the loser.

Examples of this type of game can be seen by a game called Sid Meier’s Civilization which is a strategy and turn-based with very similar objectives to what was previously was mentioned. This example however is not very user-friendly and can be seen as very complex and hard to learn which is one of the problems I wish to solve within my system.



To play the game:

Firstly, the player is presented with a login form of which if they have an account they can type in their username and password to login to the system and track their progress by saving, creating or loading games. If they do not have an account they can navigate to the create new account form in order to setup an account and get started back on the login form.

After logging into the system, the user is presented with the menu form which gives the user the option to create a new game, load an existing game, delete their account if they are done with the game or just log out of the system.

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.

Finally In the current game form, the player actually plays the game by seeing the whole map and upgrades their settlement; it will be turn-based and will have the ability to save games so that you can load games if you exit the system for another time.

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

The key aspects of the game of which need to be fulfilled is that:

* Must be ability to play multiple factions
* Must have the ability to choose different maps
* Must be user friendly (unlike current system)
* The User must also be able to save and load in order for the player to play the game at their luxury.
* The player must be able to upgrade their settlement for game progression

Some aspects of the game which I would like to fulfil however due to limitations of time and expertise I will not be able to complete, some of them are:

* The player needs to be able to create squads to attack other settlements in order to win the game
* Introduce better graphics as well as animation to make the system look more attractive
* Ability to create a custom game where every detail about the game can be modified to the user’s preference

**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 |
| Last Login | DATE |  | 15/02/17 |  |

**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.

**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. *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. Sound Effects
2. Lose conditions
3. Button to print off tile information at any given time
4. Win Conditions

**Proposed Method of Solution**

To Conclude, I have chosen Delphi combined with Microsoft Access to create the solution. I plan to store all data such as variables and constants within Access and write the program to change this data as well as use it effectively in Delphi. The tables will be created upon first start-up of the program by using DDL in Delphi. Due to writing the program in Delphi, I feel I will have a large amount of control over how the human to computer interface will look because of the surplus of visual components in Delphi such as buttons and form structure. Storing information in Access tables instead of files means that I can utilize the structure and built-in functions of Access as some information will need to be constantly manipulated and updated. An example of this would be using the built in sort routine to sort how many times it took for a player to complete a level, shortest first.