Improving Sim City

Brigham Young University - Idaho

Larson Caldwell, Sam Graham, Haru McClellan

**Introduction:** As an instructor of several climate change courses, Brother Sargeant has created a simulation for students where they are able to model power plants. This simulation is wonderful for instructing these students, however the user experience is intimidating, and needs improvement. This project’s aim is to make the end-user’s encounter with climate change more friendly, fun, and novel.

**Program Definition:** The current implementation has three main features:

1. Allow students to access and modify their country’s parameters and save those changes to an online file. (Through the instructor physically typing things in)
2. Allows the course instructor to add and modify parameters as needed. (Instructor has control of the document)
3. Enable different groups to exchange resources with each other in a classroom environment. (This is done using a paper system)

Brother Sargeant currently uses a large excel program to do the number cruching for this simulation. This program is used to model climates (socioeconomic conditions). It has been found to scare students unfamiliar with Excel, and has grown quite large and unwieldy. This program is intended for students in that class, as a way of replacing this Excel file.

**Design Overview:** The design is based off of the current Excel file. Its function is very similar except it is much more user friendly. The application will pull data from online resources. This will enable Brother Sargeant (and others) to update the various parameters of the simulation. Currently this will be saved to files on Google Drive and used in the simulation during each stage of the simulation.

The project is visually held mainly in a single window. This shows the current round. There are several options for the various types of plants. User’s can select to remove or add to the plants as desired. Additionally, a window can be called up to enter in information regarding tradable commodities..

**How the current version works:** On the Excel spreadsheet that Brother Sargeant has created, the users compete in teams. The teams manage a fleet of power plants which are used to produce energy.

The groups must fulfill three requirements:

1. Supply their city’s energy demands
2. Maintain at least a 45% approval
3. Not exceed their budget.

These goals are met in a series of seven rounds, numbered zero to six. In each round players must make decisions for the resources they receive. Resources are traded among the groups in the class, and these resources modify various aspects of energy production and emissions output.

In addition to these requirements, the groups compete with each other to:

1. Minimize emissions
2. Maximize security
3. Maximize profit in order to get the best ranking.

There are four primary factors influencing the long term goals:

1. Nuclear plants,
2. Renewable plants,
3. Fossil fuel plants, and
4. Oil.

Each has trade-offs. For instance, Nuclear has low emissions, and low security.

This is done through several different methods.

First, the user can create and destroy power plants. Each type of plant has different statistics on how much it impacts approval, energy supply, emissions, profit and security, as well as how much it costs per unit.

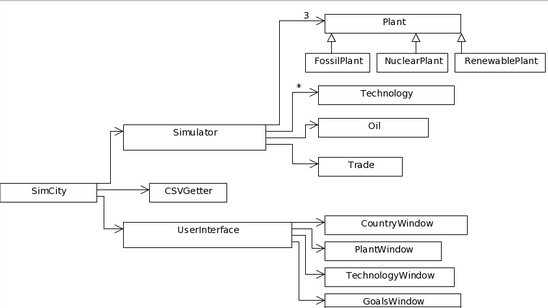
Another way to progress in the game is to use action points, technologies, emission credits and market shares to boost power plant statistics and both long and short term goals directly. It should also be noted that there is a fourth energy source: oil. While the player is unable to affect oil directly by creation or destruction of plants, he may alter its statistics through use of technologies and action points.

The end goal of the game is to get the highest profit, security, and lowest emissions. The team with the best score out of these three classes will be declared the winner.

**Design for this Program:** This program provides the same functionality of the old version, with better interaction with the user. Due to the steep learning curve associated with the Excel spreadsheet program, many users have problems with this assignment. One of the principal goals is to emulate the current version’s logic and data, but to have a new face for this program.

This will be done by a JavaFX application. It will include a window for viewing any given round. This data will be gathered from the internet as a comma separated values file, which can be edited by an instructor. Once this has been downloaded, the student group will be able to add their information for that round. The program executes based on input and forms a file which is then sent to the instructor. The rankings are then found from these input files, and the next round commences until completion.

The code will flow in the following manner according to this UML document. (See (Appendix A for a more detailed UML, See Appendix B for mockups of the GUI)



**Class Descriptions:**

**SimCity**

This is the class that uses the user interface to display options and results as calculated by the simulator. It will also use the CSV getter to access files and retrieve data to be used in the simulator class.

**Simulator**

The simulator will run the algorithms of the program. It will contain the three types of plants available and oil. It will also contain the statistics of the country and their manipulation when influenced by technologies, upgrades, emission credits, market shares, and trades.

**Plant**

This class contains information about a type of plant, such as the number, the cost to build each one, the energy that each plant produces, and so forth. This information is to be manipulated and retrieved using getters and setters.

**FossilPlant**

This is a specialized plant that holds data specific to fossil plants.

**NuclearPlant**

This is a specialized plant that holds data specific to nuclear plants.

**RenewablePlant**

This is a specialized plant that holds data specific to renewable plants.

**Oil**

This class, like the plant class, contains information concerning the country’s oil statistics. Specifically the daily consumption, growth, and emissions will be retrievable and mutatable through this class.

**Technologies**

This class contains the changes that will be made to different aspects of power production and cash when they are implemented.

**Trade**

This class contains the amount of cash, emission credits, market shares and technologies that will be traded, as specified by the user.

**CSVGetter**

This class is in charge of retrieving information about a specific country scenario and using it to set up a program based on that situation. It can also save any changes made and put them into files to be accessed later.

**UserInterface**

This class is a window and several buttons that will offer options to act and change the modifiable values. It contains other windows to handle specific tasks and views.

**CountryWindow**

This window displays the statistics of the country, as calculated in the simulator.

**PlantWindow**

This is a window that allows the user to view the statistics on a specific type of plant. Buttons on this window will swap between the different kinds of of plants. The user will be able to add plants, remove plants and purchase upgrades while watching the changes that take place from their actions.

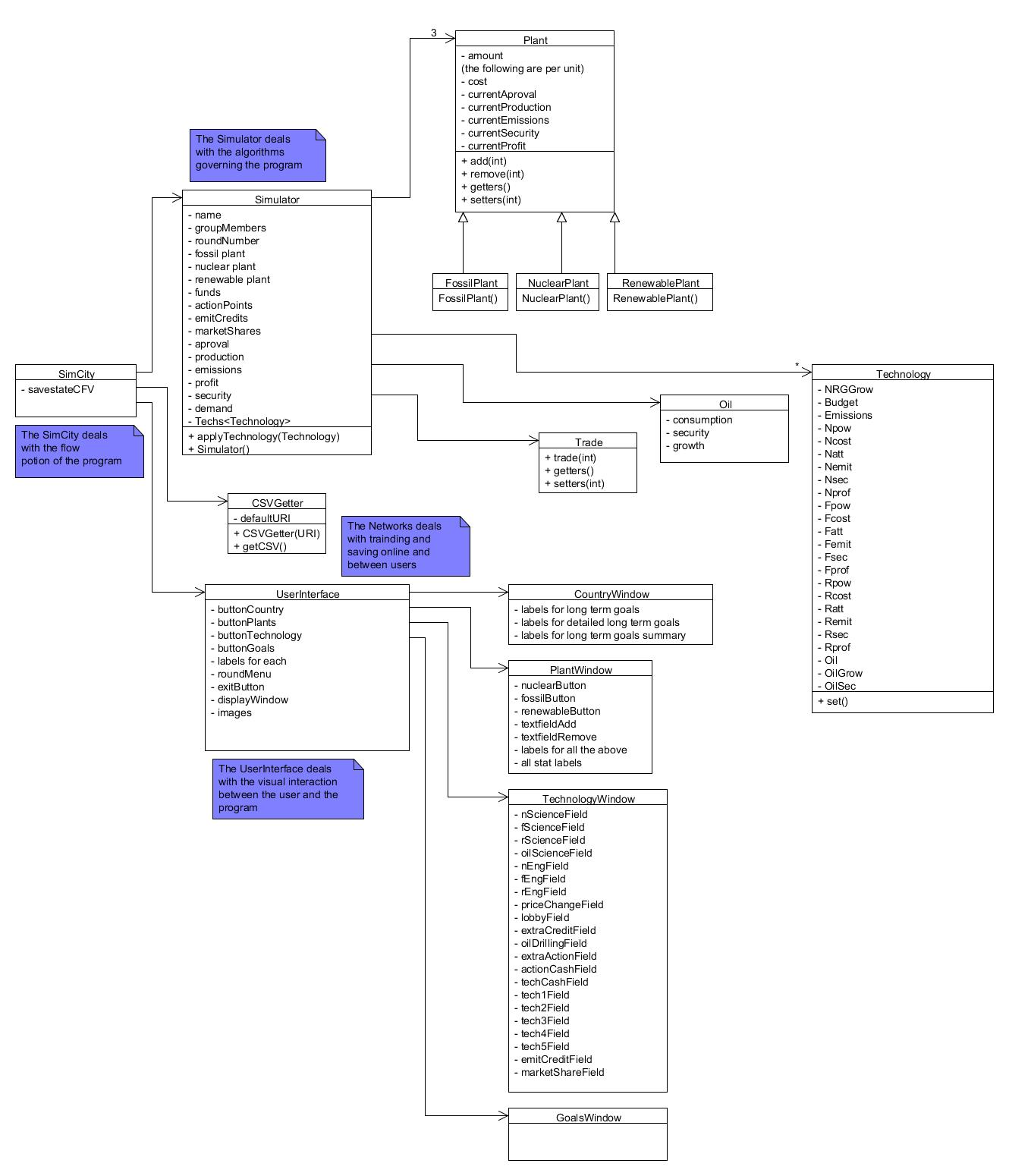
**TechnologyWindow**

This is a window that displays to the user the various technologies he currently has. He may implement them at this window and see what affect they will have on his country.

**GoalsWindow**

This window displays the goals of the country and whether or not they are being met. The user can toggle between long and short term goals. Graphs will be on both of these pages show the relative strengths of each statistic.

**Appendix A: More Details into the UML**



**Appendix B: GUI Mockups**



