Rural Cultivation & Atmospheric Emulation Application (RCAEA) Design Document

|  |  |
| --- | --- |
|  |  |
| **Document:** | Design |
|  |  |
| **Authors:** | Zisis Damianidis  Richard Dyer  Tsanko Hadzhiev  Mihail Hadzhinikolov  Al Al-Mohaiminul Islam Khan  Raima Khan |
|  | Tsanko Hadzhiev |
|  | Richard Dyer |
|  |  |
|  |  |
|  |  |
|  |  |
| **Creation Date:** | 21/11/16 |
| **Last Revised:** | 22/11/16 |
| **Group Name:** | Tanks & Co.™ |

INTRODUCTION

Purpose of Design Document

The Design Documents describes and shows the details of the classes, methods and variables involved in building the application. The Class Description provides a description for all the classes. The Method Description clarifies some of the methods. In the interaction section some of the more abstract methods are shown how they will interact with each other.

Class Description

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Contains | Returns |
| Simulation | Simulation is a core class, it contains all the necessary data for the GUI to display. The Simulation is in charge of playing, stopping, and restarting the simulation. It is also in charge of getting User input on how much fertilizer, water, soil type to be used in the simulation, and determining which province the application will use. | FileHeader, DataBase, Plot, Statistics |  |
| SimulationStorage | Uses the FileHeader to connect to the database and save or load a simulation. | FileHeader |  |
| File Header | Connects directly to the database and provides read and write functionality. |  |  |
| DataBase | On creation the DataBase object connects to the Applications database to specifically retrieve weather, and crop information. It will hold all information of crop and weather objects stored in the Applications Database. This object can always be used to retrieve a crop or weather object. |  | Weather, Crops |
| Weather | The weather object is created from the database and contains weather data properties that pertains to a particular province and month. |  |  |
| Plot | A plot is the center point where all the conditions and factors for a crops status are expressed. It keeps tracks of itself as the user moves through the timeline by creating and holding the necessary amount of PlotWeek objects. A plot can also calculate a crops status at any given time and return the details in a CropData Object. A plot can add and remove a crop the necessary week(s). In order to determine the status of a crop The Plot utilizes the database object from its base class the Simulation to retrieve weather and crop details located in the database. A Plot will also draw itself and determine when it needs to be redrawn based on the current date then simulation is on. | PlotWeek | CropData |
| PlotWeek | Represents the status of a plot in a given week. | Crop,Weather |  |
| Crop | Crops are originally created by the database. A Crop contains all its requirements to survive. It is also capable of creating and managing CropWeeks that is stored inside. | CropWeeks |  |
| CropWeek | Crop Week houses data of the crops status that pertains to a particular time in its growth. |  |  |
| CropData | CropData is created by the plot, it is a summary of information that useful to the user. It information that was created based on a particular crop, time, plot values, crop values, weather values, and user input such as watering options. |  |  |
| Statistics | This class generates a summary or specific information and interacts directly with the GUI to display information. |  |  |
| Report | The Report generates a text file based on the simulation(s) object it will make use of the statistics object to create a useful detailed overview for the user. | Simulation |  |

Method Description

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Returns | Class |
| + run( ) | The simulation runs from the start date until the end date. Run will update the currentDate variable based on the elapsed time from the start date. Unless paused, the plots will be updated based on the currentDate. |  | Simulation |
| +saveSimulation(string fileName) | Save Simulation checks to see if the simulation has made chances since its last save and then turns the simulation into a binary file,uses the FileHandler class to go into the database and save the current simulations name and object. |  | SimulationStorage |
| +LoadSimulation( string fileName) | Enters into the database using the FileHandler class and searchs for the given the FileName and turns the binary file into a simulation object and subsequently returns that object. | Simulation | SimulationStorage |
| +getCurrentCropData() | Calculates the crop details and returns the information as a CropData object. This method however will take into account the currentDate from the Simulation base class. | CropData | Plot |
| -drawSelf( ) | Each plot will be in charge of drawing itself. As the simulation moves back and forth between weeks drawSelf will be called ONLY when the current image displayed needs to be changed based. |  | Plot |
| +deleteAllCrops() | Will firstly look at the surrounding Plots and if the surrounds plots hold the same type of crops then deleteAllCrops will also be called in the other plot. Until no crop is found. Then it will delete the crop in the plot all the weeks until its |  | Plot |
| -manageWeeks( ) | Monitors based on the start and end date how many PlotWeek objects are needed and creates/delete them as necessary. |  | Plots |

Interaction

Concussion