Kostiantyn Makrasnov
Yuriy Fedas
CS273 – Spring 2018
May 17, 2018

Global Population Simulator

Requirement Specification:

This project involves utilizing researched rates of disasters and net continent population growth values to estimate how the globe's population will be impacted during a set period of time. The program should display all data as it is being simulated by utilizing an external graphics library. More specifically, a map of the globe needs to be displayed with various animation constructs showing the occurrence of various disasters and population state of each continent. All calculations should occur on a day by day basis, at the end of the simulation the user will be able to replay the entire simulation animation without the need of recalculating the individual days. The user should have the option to set a specific start date for the simulation and will need to have the ability to focus on one specific continent. Also, information on how much total deaths each disaster has caused after the start of simulation should be available upon user's request within the UI (mouse rollover). More advanced features such as prediction of changes to the net continent population growth due to human-induced and natural factors will need to be implemented as time allows.

Use Cases:

User edits simulation settings in the "Quick Start" tab

#	User's Actions	System's Response
1	User edits the value of a particular input widget on inside the "Quick Start" tab	
2		Using slots, system updates the appropriate input variable in the *sim_helper* class.

User clicks the "Begin/Pause Simulation" button

#	User's Actions	System's Response
1	User clicks the left most button at the bottom of the program labeled "Begin/Pause Simulation"	If button is labeled "Begin Simulation" and any "Quick Start" widgets have invalid inputs, the user receives bad input message dialog.
2		System renames button text to appropriate state. If the simulation was already running, further calculations are halted. If the simulation was never started, the system reads rate data from file and begins the simulation (**see "user begins simulation" use case)**. Finally, if simulation has just been paused the calculation and animations begin from the moment they are stopped.

User clicks the “Reset Simulation” button

#	User’s Actions	System’s Response
1	User clicks the right-most button at the bottom of the program labeled “Reset Simulation”	
2		System pauses the simulation if it’s not already and shows a dialog asking if user wants to save simulated data to file.
3	User clicks “yes”, “no”, or “cancel” in the dialog	Closing the dialog counts as a “cancel” response.
4		If user selects “yes”, system writes all final data calculations to a file names “Results.txt", if “cancel”, then simulation revamped to the condition it was in before pressing the button. Finally, if “yes” or “no” response then erase all recorded simulation data and blank out animation graphics state.

User replays the simulation after calculation end

#	User’s Actions	System’s Response
1	User changes the position of the slider that appeared after simulation calculation has been finished.	Slider is disabled while no simulation calculations concluded.
2		System updates the date of the selected simulation prediction (top-right) and the animation’s population values using the values pre-recorded during the simulation’s calculation

User rolls over a continent in the “Simulation” tab

#	User’s Actions	System’s Response
1	User drags mouse over a particular continent area in the “Simulation” tab	If simulation is currently running or was never started the system ignores this input
2		System shows a small text box with the current population of the continent and the amount of deaths caused by disasters. The top three killer disasters are shown in descending order with the total number of deaths they caused.

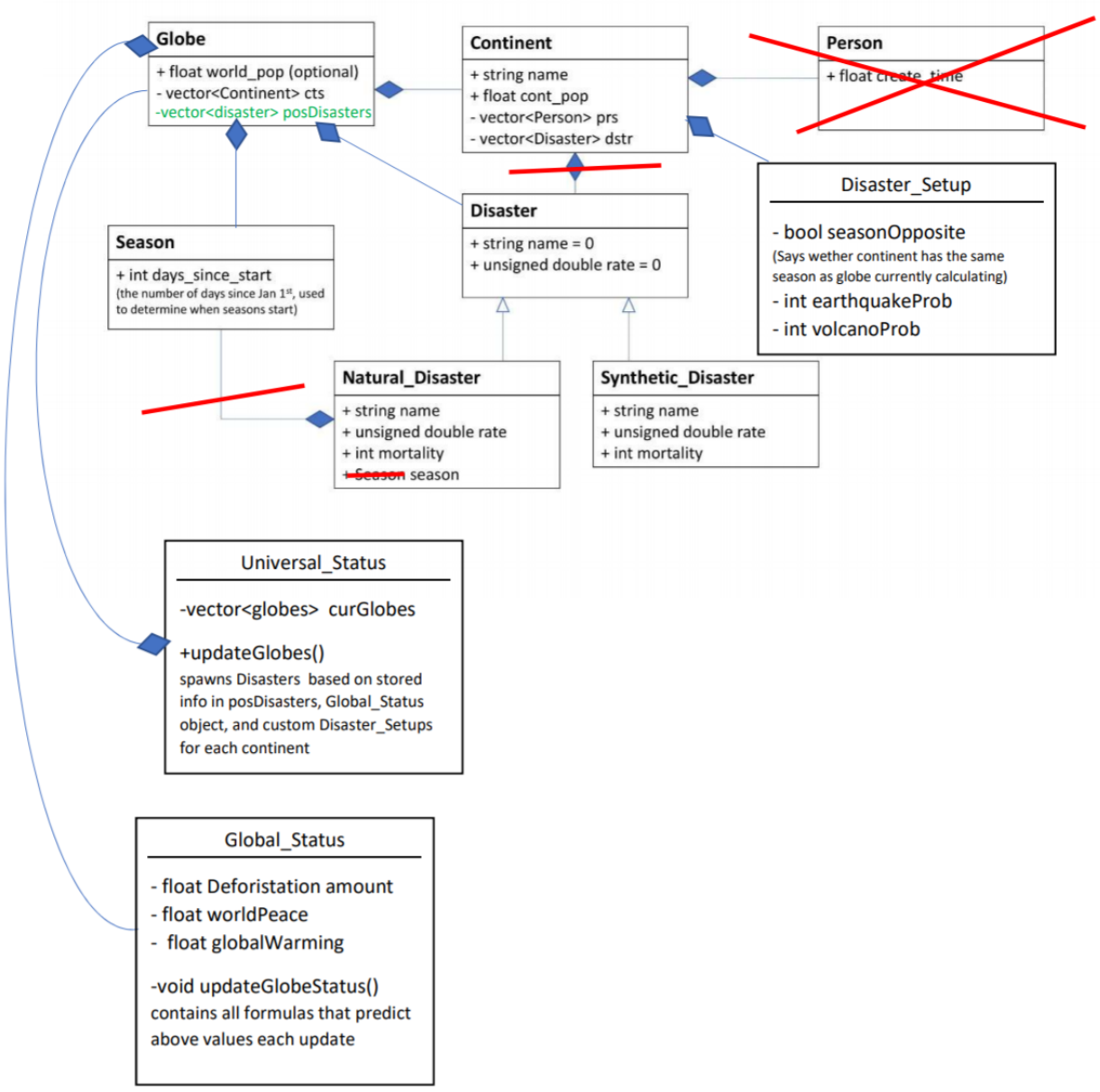
User want to exit the program

#	User’s Actions	System’s Response
1	User clicks any input whether it’s a button or window controls to exit the application	
2		System pauses the simulation if it’s not already and shows a dialog asking if user wants to save simulated data to file if any is present.

3	User clicks “yes”, “no”, or “cancel” in the dialog	Closing the dialog counts as a “cancel” response.
4		If user selects “yes”, system writes all final data calculations to a file names “Results.txt", if “cancel”, then simulation revamped to the condition it was in before pressing the button. Finally, if “yes” or “no” response then erase all recorded simulation data and application exits

UML Diagrams:

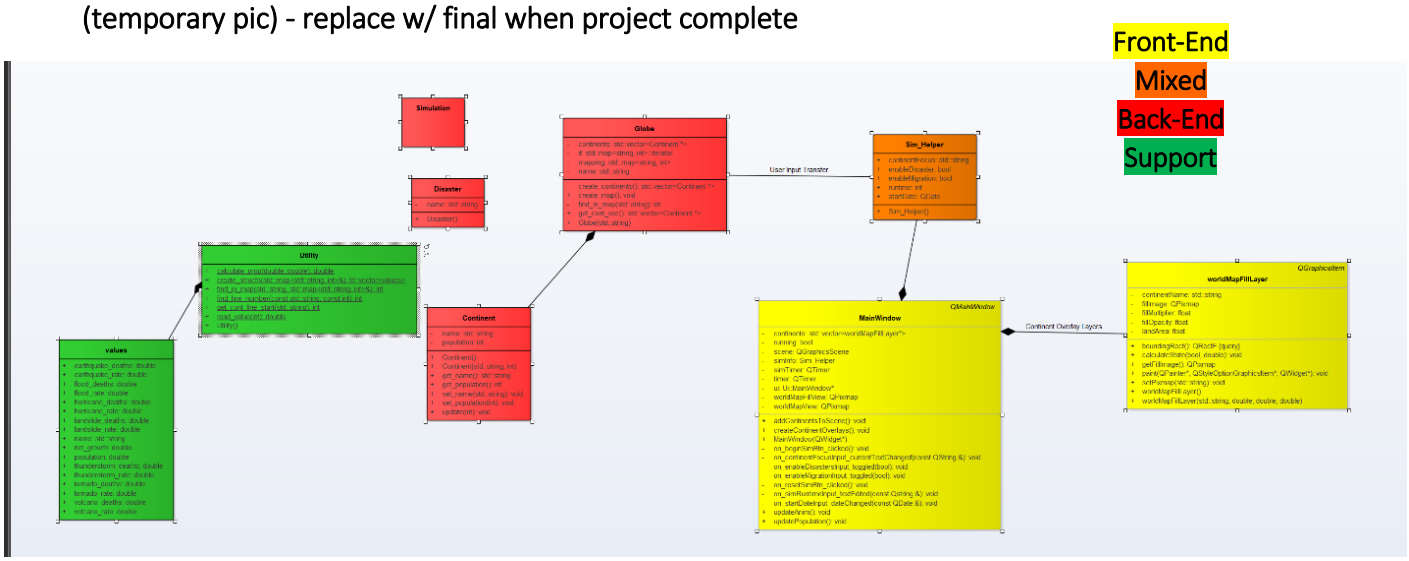
Initial Design



Kostiantyn Makrasnov
Yuriy Fedas
CS273 – Spring 2018
May 17, 2018

Final Design

(temporary pic) - replace w/ final when project complete



Pseudo Code:

Create Map Animation Setup (on Application Start - *MainWindow* constructor)

1. Create a new graphics scene to populate with items
2. Create a Sim_Helper object to store for user's simulation settings
3. Find the full map from file
4. **while** file was not found give an exception/message on screen
5. Prompt user to select their own map background file
6. Add background globe pixmap to scene
7. **Call createContinentOverlay function (with scene passed in):**
8. **for** each element in the vector of continent names
9. Create pixmaps for the files names specified
10. **if** formatted file name wasn't found add the name to recovery vector
11. Add all found pixmaps to scene w/ correct parameters
12. **for** each element in recovery vector
13. **while** recovery file replacement not found by user
14. prompt user in a file dialog to find the needed overlay png file
15. Add recovered pixmaps to scene w/ correct parameters
16. Add string item to show the date of the frame shown
17. Create timer that will act as the driver for the program's animation updates during calculations
18. Update initial animation state by calling animation update method once

Continent object creation (on Application Start)

1. Create line mappings using the codes in the appropriate namespace
2. **for** each continent to create

3. Create informational struct *values* by using the utility static methods
4. (Involves parsing a text file with appropriate checks & exceptions)
5. Use informational struct inside a continent constructor to create continent object
6. Add the pointer to continent object to the storage vector

Update Animation State (every 100ms - 10 fps)

1. for all continent overlay layers
2. Grab the current population amount from appropriate continent object in the current globe
3. Grab all disasters indicator objects (contain date) stored in disaster animation queue for each continent
4. while disaster animation queue for cur continent is not empty
5. Add a disaster indicator objects to the queue of indicators to paint and add random screen location
6. Calculate the opacity of overlay by comparing the ratio of land area to
7. Grab amount of days passed so far in animation
8. Alter the date string to display appropriate prediction date
9. Force repaint scene View to show new opacity settings and all the disaster indicators

Update Population State (clicks occur upon animation start, until simulation is complete)

1. for each continent in globe
2. call the continent's update function
3. Apply the appropriate net growth for the day to the current population of continent
4. Based on the probability of each disaster for the continent add disasters for the day into a queue
5. while disasters are left in daily queue
6. Calculate the death toll of disaster based on research
7. Apply the amount to the appropriate disaster death sums
8. Record the state of the continent by adding a key to a map of dates, continent info structs

User altering state of slider (following simulation calculation - replay mode)

1. Change slider position value to correct prediction date that was recorded in each continent memory map
2. for each continent overlay layer
3. Grab the display information (memory struct) for the chosen date from the correct continent object
4. Update population values as well as continent hover information
5. Update the date string graphics item with chosen date
6. Force repaint of the entire animation screen (all scene items)