PROJECT 2: HEATED EARTH

Design Study Report

Team 3
Faour, David (dfaour3)
Frank, Robin (rfrank8)
Poskevich, Robert (rposkevich3)
Sachdev, Gaurav (gsachdev3)
Slater, Joey (jslater3)

October 26th 2014

Georgia Institute of Technology
Atlanta, GA

Context

In this project, we explore the effects of certain implementation choices on the quality of programs that simulate and model heat transfer of planet Earth. The application calls for an initial temperature setting of 288 Kelvin degrees, modeling temperature heating and cooling while the earth rotates. The application requires a presentation layer. The study attempts to find the balance between performance and visualizations. The study also explores the impact of performance with multithreading.

Research Questions

This design study will answer the following 7 questions as follows:

1. What are the effects of initiative on the structure and elegance of an architecture?
2. What are costs and benefits of using threads to provide concurrent computation?
3. What is the effect of buffer length on idle time and throughput? Conversely, under which configurations does the buffer fill up?
4. What specific configuration would you recommend? Under what circumstances?
5. How long did it take for the simulation to stabilize (where stabilize means overcome the bias introduced by the initial conditions)? How did this depend on the initial conditions?
6. What effect did cell size have on the other factors?
7. How to determine the usability of your presentation and controls?

Subjects

The subjects of the design study are named EarthSim and ConsoleSim with the primary focus being on EarthSim.

- EarthSim contains the user presentation and is critical in the analysis of thread control and buffer settings.
- ConsoleSim was used to isolate the simulation effectiveness from the presentation to insure the model was working as intended.

Experimental Conditions

This design study was performed on a system with network connectivity turned off and all unnecessary background processes turned off. A summary of the specification of this system can be found in the system specification table below. The ConsoleSim was initiated via command line, while the EarthSim was initiated via Eclipse.

System Specification:

System Type	MacBook Pro 10.1
CPU Details	• Single Intel i7 2.4 GHz processor, quad core • 256 KB L2 Cache • 6 MB L3 Cache
RAM	8GB
Boot Firmware	Boot ROM MBP101.00EE.B05
System Firmware	SMC 2.3f36
OS	• OS X 10.10 (14A389) • Kernel Version: Darwin 14.0.0
Network Connectivity	Enabled
Execution Context	Eclipse v4.4.1 with JDK 1.6
Additional Notes	No other programs were running on the system. All known background applications were stopped.

Variables

This design study will use the following independent variables and values:

- PresentationThreaded Type: Boolean
- SimulationThreaded Type: Boolean
- BufferSize: 1, 10, 100, 1000, 10000
- Display Rate: 100
- Time: 1, 1000
- GRID Spacing: 1, 15

We explore the effect of the independent variables on the following dependent variables:

- Memory Usage (bytes)
- Average CPU (percentage)
- Threads (count)
- Number of Iterations (count)

Methods

This design study ran 9 test cases against the ConsoleSim component. This component focuses on the simulation of the heating and cooling algorithms and contains a stabilization criteria. The purpose of these tests is to determine how long it took for the simulation to stabilize across the various settings.

The second portion of the study is focused on the performance of the presentation layer and consists of 29 out of a possible 96 test scenarios. The study was stopped prior to running all 96 test scenarios due to time constraints and diminishing returns of the test results.

Analysis Techniques

Empirical evidence was captured along with subjective analysis of GUI display properties were used to analyze performance. Graphs are created to display the relationships between dependent variables and various dependent/independent variables.

Results

Given the GUI limitations on stopping criteria, it was not possible to determine the impacts of the variables on the how long it took to display the full modeled scenario, therefore results are focused on impacts to dependent variables. The design study does not believe this to be a major flaw as one could reasonably determine the appropriate design based upon interpretation of the results and refined designs in future iterations. It would be recommended that the analysis presented here would be used to refine the set of criteria in a follow up design session to determine the optimal presentation layer settings.

<u>Research Questions, Analysis and Answers:</u>

What are the effects of initiative on the structure and elegance of an architecture?

We evaluated both CPU and Memory usage in Master Controlled Threads, Presentation Controlled Threads and Simulation controlled threads. In all scenarios, the presentation control thread yielded better performance as shown in figure 1.0 and improved memory usage as shown in figure 2.0.

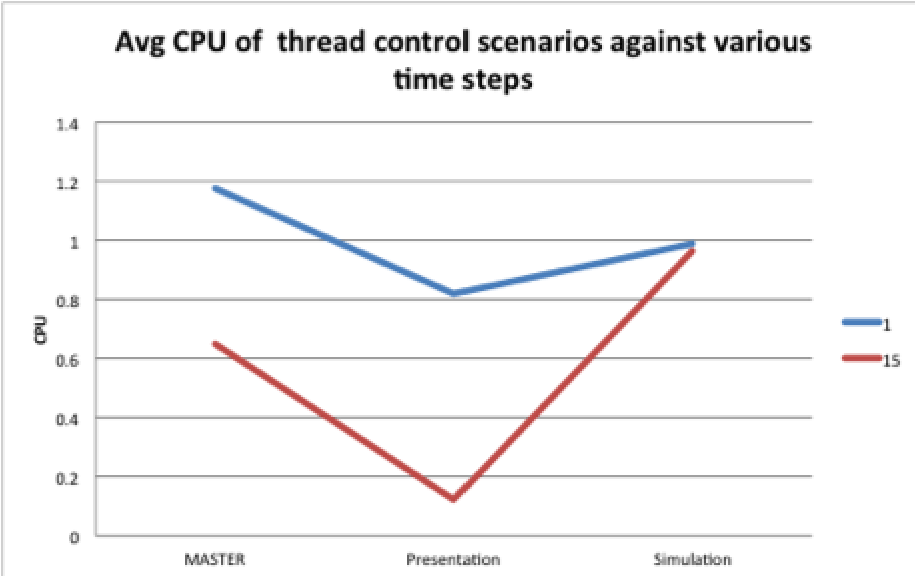

Figure 1.0: Comparison of average CPU between test cases with different lattice granularity settings

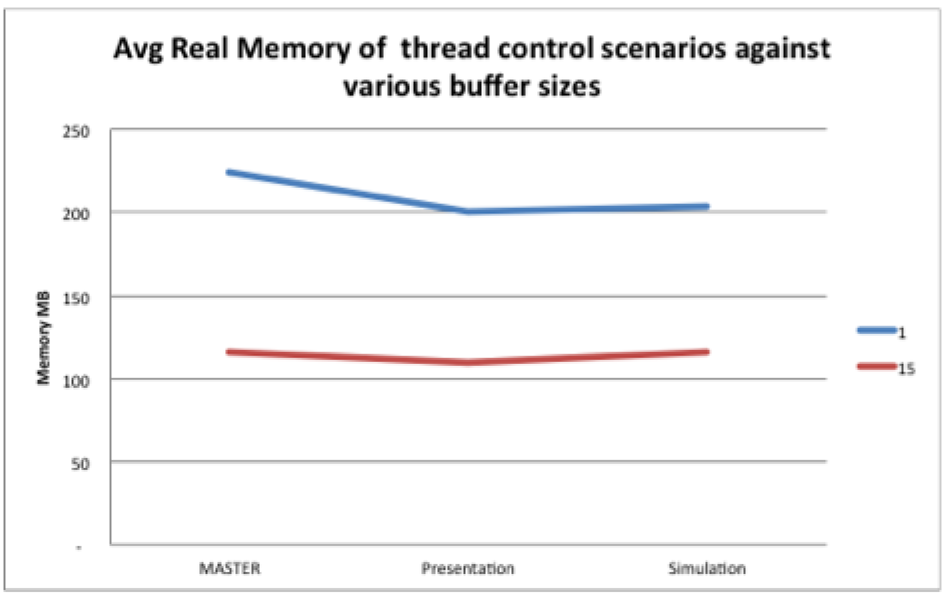

Figure 2.0: Comparison of average memory usage in test cases grouped by thread control session and various lattice granularity.

What is the effect of buffer length on idle time and throughput? Conversely, under which configurations does the buffer fill up?

Adjusting buffer sizes improved CPU performance and was inconclusive when looking at memory usage. The figure 1.0 illustrates decreased CPU when buffer sizes were increased for both Master Controlled and Presentation Controlled Initiates, however, Simulation Controlled Initiates indicated slightly degraded performance.

There were three conditions where the application crashed due to memory errors. In all 3 scenarios, buffer sizes were above 10,000 and lattice density was set to a depth of one with time steps set to ine as well. These settings were visually rich proving the design tradeoffs that visually rich design choices put additional stress on system resources.

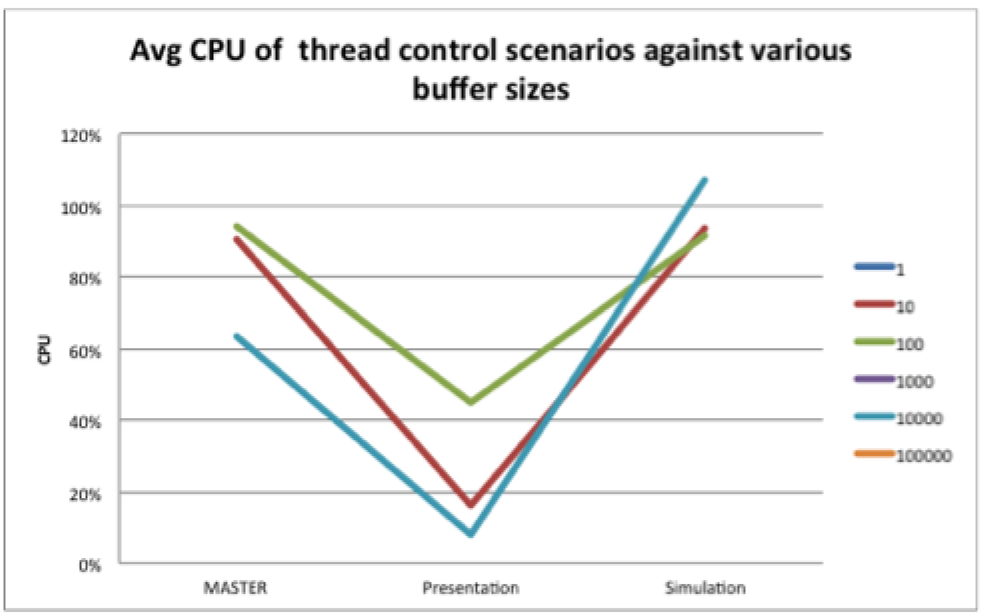

Figure 3.0: Comparison of average CPU in test cases grouped by thread scenario and various buffer sizes.

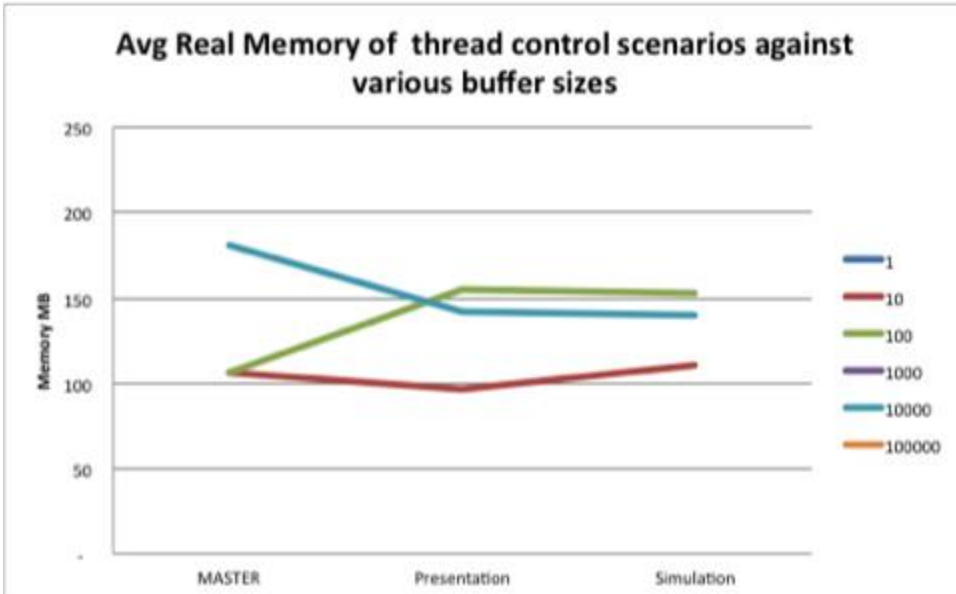

Figure 4.0: Comparison of average memory usage in test cases grouped by thread scenario and various buffer sizes.

What specific configuration would you recommend? Under what circumstances?

Based upon results of this design study, it would be recommended to have the Presentation Initiate control the threads, with a time step set to 10 and the lattice density set to 1. Balancing the frequency of the time step would offset the increased system needs of setting the lattice density to 1 (fig 6.) The graph below illustrates the change over the simulation across all of the latitudes with the poles being reflected at the outer boundaries and the equator in the middle of the graph.

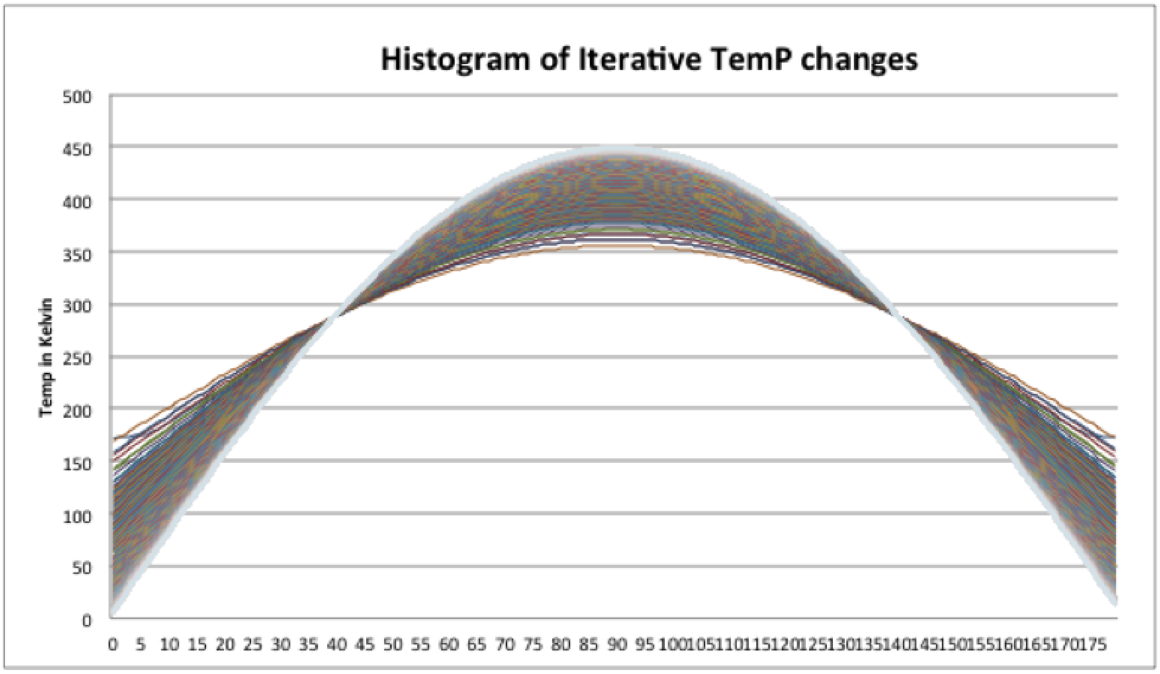

Figure 5.0: 255 of the simulation iterations with a lattice density of one and a time step of one, poles are on the ends and the equator is in the middle. Changes reflect how the poles stabilize over time while the equator heats up.

How long did it take for your simulation to stabilize (where stabilize means overcome the bias introduced by the initial conditions)? How did this depend on the initial conditions?

As shown in figure 6.0, the time steps and degrees of precision get more refined, the number of iterations increase. In extreme cases, such as 1000 time steps and 100 degree lattice definition, the results terminated immediately. The graph also clearly calls out that as the size of the cells decrease, it requires more iterations to stabilize. We can also see that in order to improve the speed of the simulation engine, the design can call for larger time steps in the model.

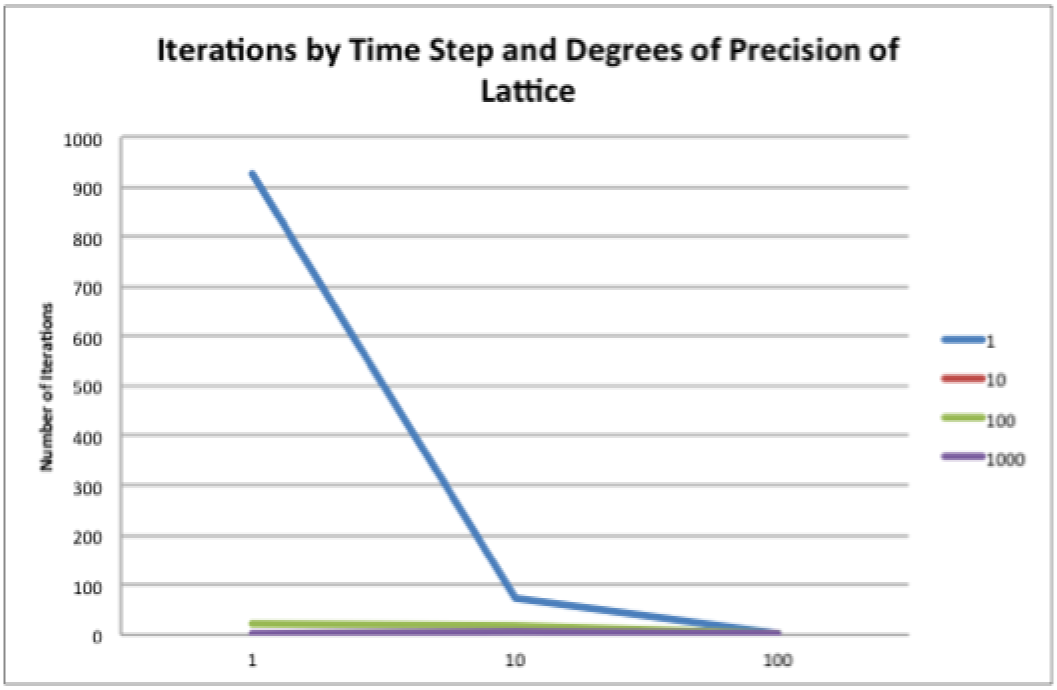

Figure 6.0 Charts the number of iterations by lattice granularity (x axis) with various time steps.

How you would determine the usability of your presentation and controls?

Overall, the presentation and controls were usable. The figure below shows the temperatures near the poles were cooler than the temperatures near the equator. The presentation requires further adjustment as the color gradients are difficult to understand. The rotational position is indicated in the simulation information along with current time. While both are updated in the simulation, their relative values in context of the picture lend little value to the experience.

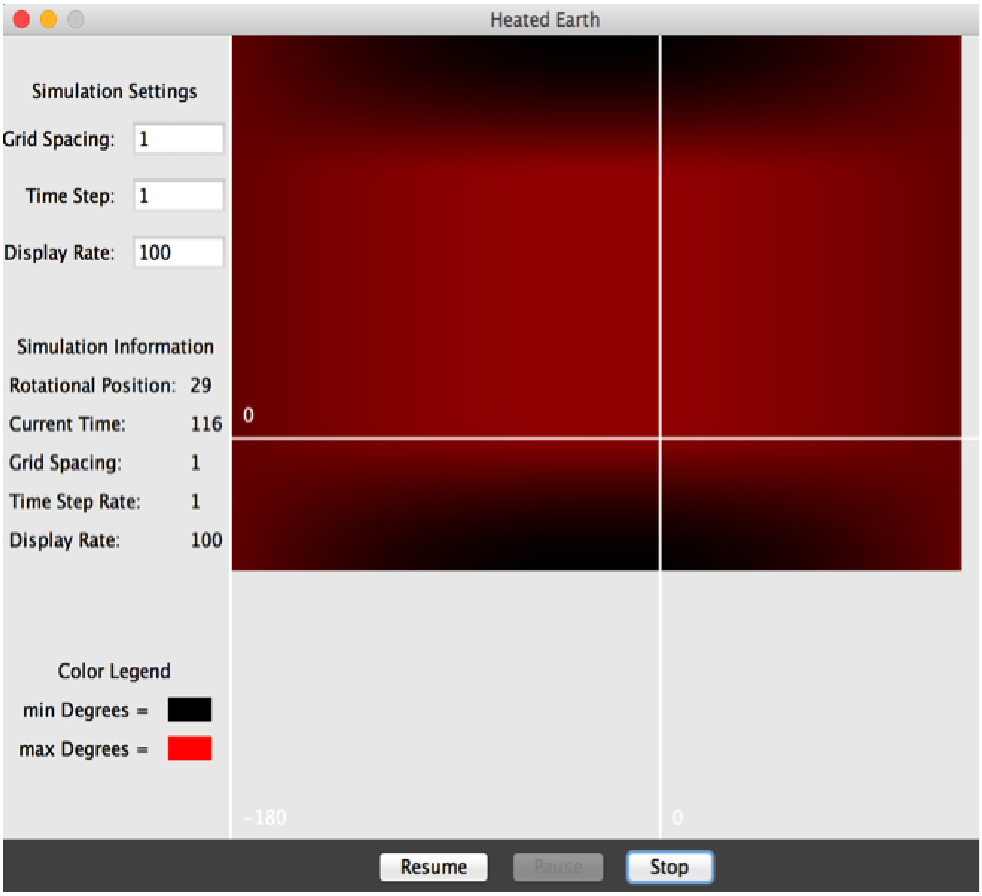

Figure 7.0: Screen Capture (GUI Representation) of Heated Earth user interface taken 116 time steps into the start of the simulation.

Discussion

The main driver of this application was to produce a GUI with the key focus area being the animation of the heating of planet Earth. The study clearly proves that setting visually rich parameters like fined grained grid size, small incremental time steps and fast display rates taxed system resources but provided the richest visualizations. Whereas, the command line running of the simulation model achieved quick results leveraging modest system resources. There are also several non-functional and functional enhancements that should addressed in future iterations of the program.

The lack of termination criteria or status within the GUI made the analysis of performance difficult. Most test cases were terminated before any end criteria was achieved, while this might be a bug in the code, it was unclear. The system also attempted to reflect the current time but there was an unclear relationship to what the model was attempting to display. Future designs must consider the inclusion of stopping criteria in the GUI to illustrate that the model has completed with a set of final results. As the stabilization criteria exists in the simulation engine, it is not believed to be a significant design impact by including it in the GUI model.

The visualization's use of color gradients was helpful in understanding how the temperatures fluctuated, however, the model would become unstable and develop irregular patterns over a period of time. Future considerations might include output to file formats for troubleshooting problems and/or understanding model behavior. The designers would have to incorporate some type of file system or database logging in order to capture and store results for offline analysis. This additional logic would add code and complexity and user requirements would need to be flushed out.

Another observation was made during the design study regarding the simulation of the earth rotation and the heating/cooling effect. It was clear in the simulation engine that the temperature at the poles decreased over the time in proportion to the heating near the equator (fig 5.0). This contrasted at times with the visualization in the presentation layer. The color gradient made it difficult to determine just how cold the poles were from the equator and which longitudes were being heated or cooled. Future design considerations should consider how best to represent the active columns facing the heat source in order to call attention to where the temperatures should be rising and where they should be falling. The designers would have to consider the additional tradeoffs in the application, as the animation would require additional resources and additional code complexity for the sake of user experience.

Much of this design is reusable for future enhancements to the model. Designers might want to consider adding density of heat transfer from each cell based upon various material make up (water and land transfer heat differently). This would require additional settings for each cell type to hold a material type. The decision would add code complexity along with computational complexity but add to the realism of the model.

Finally, designers might also include the ability to model across 3 dimensions, including heat transfer from within the globe and from the sun. This would also mean the heat transfer would happen on six planes as opposed to the current 4. Again, the trade off would be a more accurate model but require additional code complexity, computational requirements and presentation needs.

Adaptability

To explore the adaptability and flexibility of our design, we consider the following scenarios:

Change Scenario	Assessment
Different methods of driving the calculation could be used in the future, requiring different underlying class structure. This might be done in order to study different strategies of	In order to change the calculations, changes would only need to be done in SimulationCalculation class. If studying the effects of multiple calculations against each other, SimulationCalculation would be turned

containing the data or of simulating the temperature diffusion.	into an interface with different concrete calculations in the implementations. Furthermore, the calculations could be turned into a Strategy pattern switching the strategy on the differing calculations.
The user may want to change the temperature of an earth already in the process of being heated. They may wish to observe the effects of a lowered or increased temperature on the rest of the earth.	Calculating the diffusion of heat across three dimensions would require changes to the underlying algorithm, likely through a new implementation of IHeatableEarth or SimulationCalculations.
The user may wish to simulate temperature diffusion across a non-spherical planet.	In order to implement a non-spherical object several design problems would need to be addressed. First, the input of the non-regular object would have to be handled via a GUI where the user could draw their own shapes, or via the import of a file representing the shape. Then heat at all edges of the shape would need to be considered. Finally, using an array based system of data storage could lead to non-optimized memory usage, unless some sort of algorithm to map locations to array locations was added. Interestingly, using object

	references to access data in a non-rectangular object would be very similar to using them for rectangular objects.
The user may wish to limit the amount of memory that the temperature diffusion calculation uses.	If the user wishes to limit the amount of memory used by the system, the system should force the user to use calculations that are not memory intensive, namely using primitive floats with arrays. This functionality could easily be added to the program through manipulations of the GUI.
The user may wish to export the data from the calculation to use outside of the simulation. This may include simply the end state of the heated planet, or may include each iteration of temperature diffusion across the planet.	Exporting the end state of the heated planet would be a trivial matter, since all the data is displayed to the screen. Also each iteration is already captured for the animation, so the application would just need to store all of the iterations until completion for exporting.

Conclusions

This design study considered over 100 variations of the program and tested over 50 scenarios. As is the case in most design studies, given the time constraints of the project, some pieces of the architecture and design still require additional study in future iterations. There are also requirements and usability questions that need to be investigated, such as the ability to download data for additional analysis. In the end, the study successfully demonstrated that thread management controlled by the presentation layer improved performance. The study was unable to determine conclusively, which set of presentation values impacted the presentation run time of the simulation in the GUI due to the lack of exit criteria, however, the simulation engine code was able to determine which set of values was most accurate. The study succeeded in that it answered some critical design choices, such as recommending the presentation layer control the threads. The study also pointed out designs to avoid, low update frequency and large grid size due to simulation accuracies and presentation layer usability.

***