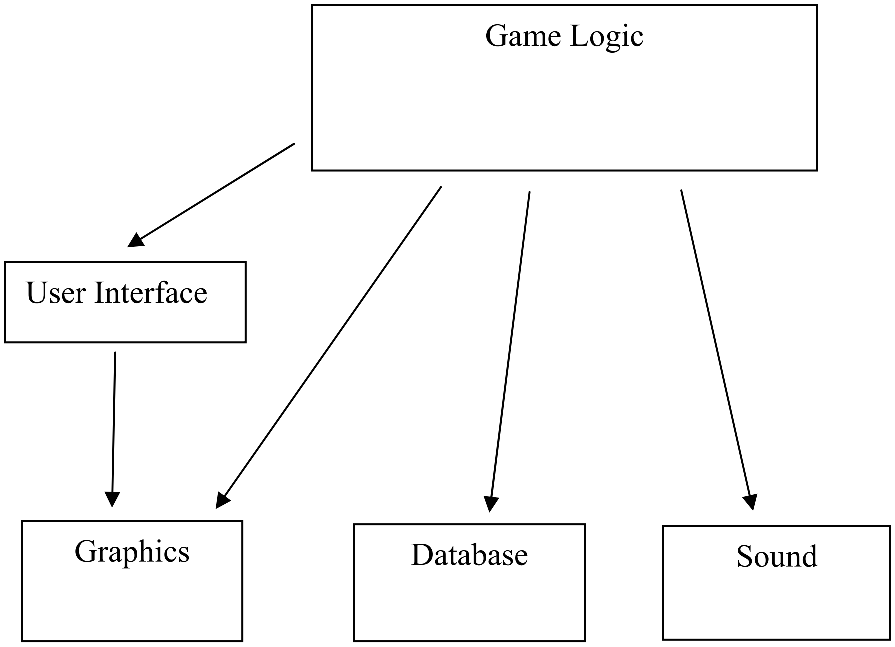
Bernard Peng
CS190
SimsU
Top Level Design

Overview

SimsU is a game that combines SimCity with the Sims. You run a college campus and in doing so, you can construct buildings for your students and hire professors. The object of the game is to take care of your students, and earn money to further improve your university.

Levelized High-Level Component Diagram



Description of Components

Game Logic – Based on what the player does, this component determines what the consequences of the player's actions are. This has two main components:

Management Logic – this part of the logic determines where buildings can be placed, and how the placement/availability of buildings affects the game play. This also determines which professors can be hired, and how they affect the university. The school's economy is also calculated in here, determining how the budget affects game play.

Student Logic – this part of the logic determines how a student responds to the situation he/she is placed in. Factors that may determine this include the student's class schedule, jobs, housing, and extracurricular activities. This component also determines the student's happiness level, and the overall happiness of the student body.

User Interface – This component controls how the user interacts with the game. This primarily involves the use of the mouse to select characters/buildings and choose options. It will also include keyboard shortcuts for various actions. There will be a menu always present for the use to perform the "main" actions such as building new buildings or managing the budget. When something on the screen is selected, there will also be additional menus for modifying that object (such as changing the buildings options).

Graphics – This component includes the tile-based engine used to display the actual campus and the activities going on there. It will display different buildings, students walking around as well as other things to show what's happening on campus.

Sound – This component will be responsible for providing background music if the player chooses, as well as sound effects to enhance the players experience (such as reactions to the players actions, or game events).

Database – This component will be able to take all the current game information and store it into a save game. It will also be able to load past save games to let the player resume from where they left off previously.

External Dependencies

The external dependencies in this program include:

Graphics Library – if we decide to use a pre-written graphics library, we would need to learn how to use it properly and adapt it to our needs.

Sound Library – if we decide to use a pre-written sound library, we would also need to learn how to use it and adapt it to our needs.

Real world university data – we would need to model the simulation of the university based on actual data, so research into how a real university works and how budgets look would need to be done.

Task Breakdown

Manager – This person would be in charge of keeping track of what everyone is doing, and that things will get done on time. The manager will also be in charge of setting up meetings and keeping good communication between the team members. (1 person) (cphartne, vk, itang)

Tester – This person would be in charge of testing the program and finding problems with it. The tester might write drivers for various components of the program and test those parts too. (1 person) (kfardig, rmckenzi)

Researcher – This person would have to look into how universities are actually run. They would have to find sample budgets, how different budget allocations may affect a university, and what the main things that typically concern a university. These are all factors that will need to be modeled in the game. After this is complete, this person can also handle the documentation duties (internal use and user manual) (1 person) (dshue)

Logic Coder – This person would code the main logic portion of the game play, which is the simulation of the university and its population. This person would also have to keep in communication with the researcher since the researcher's results heavily impact what the final game logic will be. The database part would probably be put in here too, since that part is small, and depends so much on the data of the game (which the logic controls) (3 people) (jkim21, mlin, dshue, tyoon, nige, tneal, bpeng, rhsieh)

Graphics Coder – This person will write code for loading various models for the buildings and people, and displaying them on the screen. (1 person) (bpeng, rhsieh, tyoon)

User Interface Coder – This person will be in charge of writing the GUI code such as the menus and the user interface. He will probably have to keep in touch with the Graphics Coder since much of their work will be used together. (1 person) (tmohamed, hyazawa)

Sound Coder – This person will be responsible for writing code to load up background music as well as sound effects for the game. (1 person) (rhsieh)

Artist – This person will have to draw/model all the buildings available in the game, and characters who will be students or professors. The end result will have to be in a format compatible with the graphics coder's code. (1 person) (rhsieh, bpeng)

Schedule

3/7 – Final Design Document finished.
-determine group members' roles
-determine final schedule
3/12 – Interface proposals, testing strategies, research begins
3/15 – Interface comments.
3/19 – Finalize interfaces, final research results
3/21 – Propose individual component designs, go over it with the Manager
3/31 – Finalized component design
4/14 – Code should be functional at a minimal level, begin integration, minimum art complete
4/21 – Code should meet requirements, no more "new code", integration should be complete
4/28 – Tested for a week, eliminate bugs, demo, user manual complete
5/9 – Final demo