

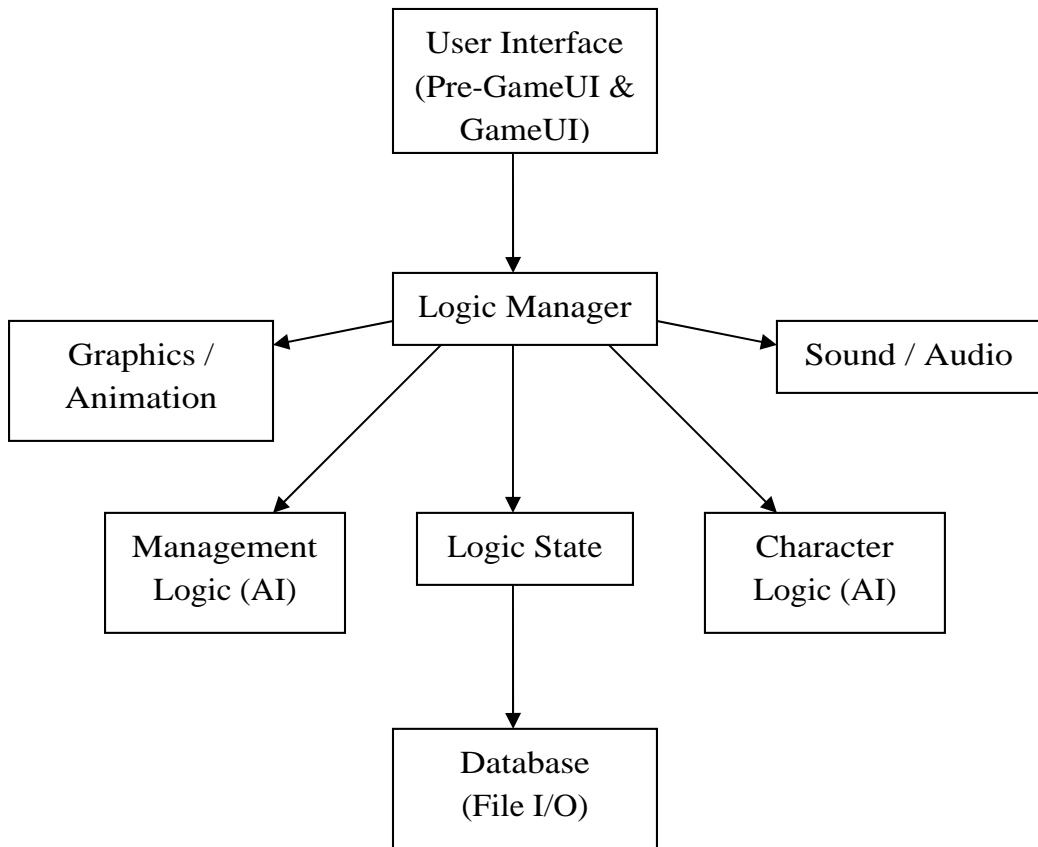
SimsU

Top-Level Design
By Dan Shue (dshue)

0 Overview

SimsU is a stripped-down combination of the popular Maxis games “Sims” and “SimCity”. Instead of building a city, the user builds a university (Brown) with students and professors, who will then determine the university’s success.

1 Levelized High-Level Component Diagram



User Interface

This component includes the Pre-Game UI and the GameUI. The Pre-GameUI prompts the user to initialize the university settings (either from scratch or from a previous game). The GameUI serves as the main user interface of the game; this is what the user sees on screen during the university construction process. Here, the user utilizes the mouse and keyboard to create / modify buildings and characters. Menus and other functions aid the user in carrying out these actions.

Logic Manager

This is where “everything comes together”. The Logic Manager gathers input information from the User Interface, and in turn, updates its current Logic State. When the user wishes to create a building / character, the Logic Manager will also tell the Management / Character Logic components to start the AI; it will also ask the Graphics component for the appropriate graphics to reflect the user’s actions. Once the Logic Manager finishes all the updates, it’ll return the modified Logic State back to the User Interface, which will display the modifications on screen.

Graphics / Animation

This component represents all the graphics (2D and 3D) that’ll be displayed in the User Interface. Examples include buildings, characters, and maps.

Sound / Audio

This component represents all the sound effects that’ll be heard during the game, as well as background music that matches the current “mood” of the game.

Management Logic (AI)

This component performs all the non-character functionality of the game, such as creating buildings, hiring professors, and financing the university budget.

Character Logic (AI)

This component performs all the character functionality of the game, such as running daily errands and social interaction between characters.

Logic State

This component represents the current state of the game, and is updated accordingly whenever the user decides to change something about the game.

Database (File I/O)

This component loads and saves games (via the current Logic State).

2 External Dependencies

I suspect that a large portion of the graphics and audio components will be taken from the libraries used in the original versions of “Sims” and “SimCity”. It is unrealistic to expect a bunch of college students to come up with the graphics and audio components from scratch in a span of 2-3 weeks.

3 Task Breakdown

Project Manager

The project manager is the “big boss”. He/she is responsible for the overall process of the project, and makes sure that all group members are up-to-date with their components. The project manager also handles all administrative details, such as group meetings and project organization. Finally, the project manager ought to have the most knowledge, experience, and enthusiasm for the project (out of all group members).

Librarian

The librarian writes up all the documentation for the project, including revised requirements, specifications, designs, and interfaces. In addition, he/she writes up a user manual so that people will know how to play the game. This requires the librarian to be in close contact with all the coders, since changes in the user interface (for example) will have to be reflected in the specifications and user manual.

Tester

The tester assures the proper functionality of each component before and after integration. He/she tests the components by writing “dummy” testing interfaces that check for inconsistencies within the program. It’ll be important

for the tester to start testing EARLY, so that bugs can be reported and fixed on time.

Core Programmers

The coders make up the majority of the group, and focus on the actual coding of the program. I propose the following components to be split-up among the remaining members of the group, with the number of coders in parentheses:

- User Interface (2)
- Graphics (1)
- Logic Manager (1)
- Management Logic (1-2)
- Character Logic (1-2)
- Database (1)

4 Group Organization

Note that the following assignments are somewhat arbitrary, since the groups haven't been finalized yet. I'm solely basing this section on the student bios on the CS190 webpage (in particular, the "role you would like to play in a project and skills you bring" question).

Project Manager: rhsieh

Librarian: dshue

Tester: rmckenzi

Core Programmers:

- User Interface: msh, bpeng
- Graphics: awhull
- Logic Manager: kfardig
- Management Logic: nige, tyoon
- Character Logic: jkim21, tneal
- Database: mplin

5 Schedule

- 2/28 **Group Selection & Delegation:** Group members are finalized; group roles are assigned
- 3/7 **Final Top-Level Designs:** Revised designs should be coordinated by the librarian.
- 3/12 **Interface Proposals:** Each core programmer proposes external interfaces for their specific component. Proposals should be coordinated by the librarian.
- 3/21 **Final Interfaces:** Revised proposals (after receiving feedback/comments) should be approved by project manager.
- 4/4 **Detailed Designs:** By now, all components should be detailed enough for hardcore coding in the next 2 weeks.
- 4/16 **Initial System Integration:** Components start to integrate.
- 4/28 **In-Class Demo:** The program should be mostly done at this point (a few minor bugs are OK). It should be functional enough for a demo. After this date, no major changes in the code should be taking place.
- 5/9 **Final Demo and Hand-in:** We're done!

6 Assumptions

The only assumption I made was that the UI and graphics components were separate from each other. It may very well be that those two parts are better off when they're treated together as one component, but I guess that's more of a design issue.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.