



Specifications Document

BROWN UNIVERSITY EDITION Richard Hsieh (rhsieh) . 02.07.02

Description

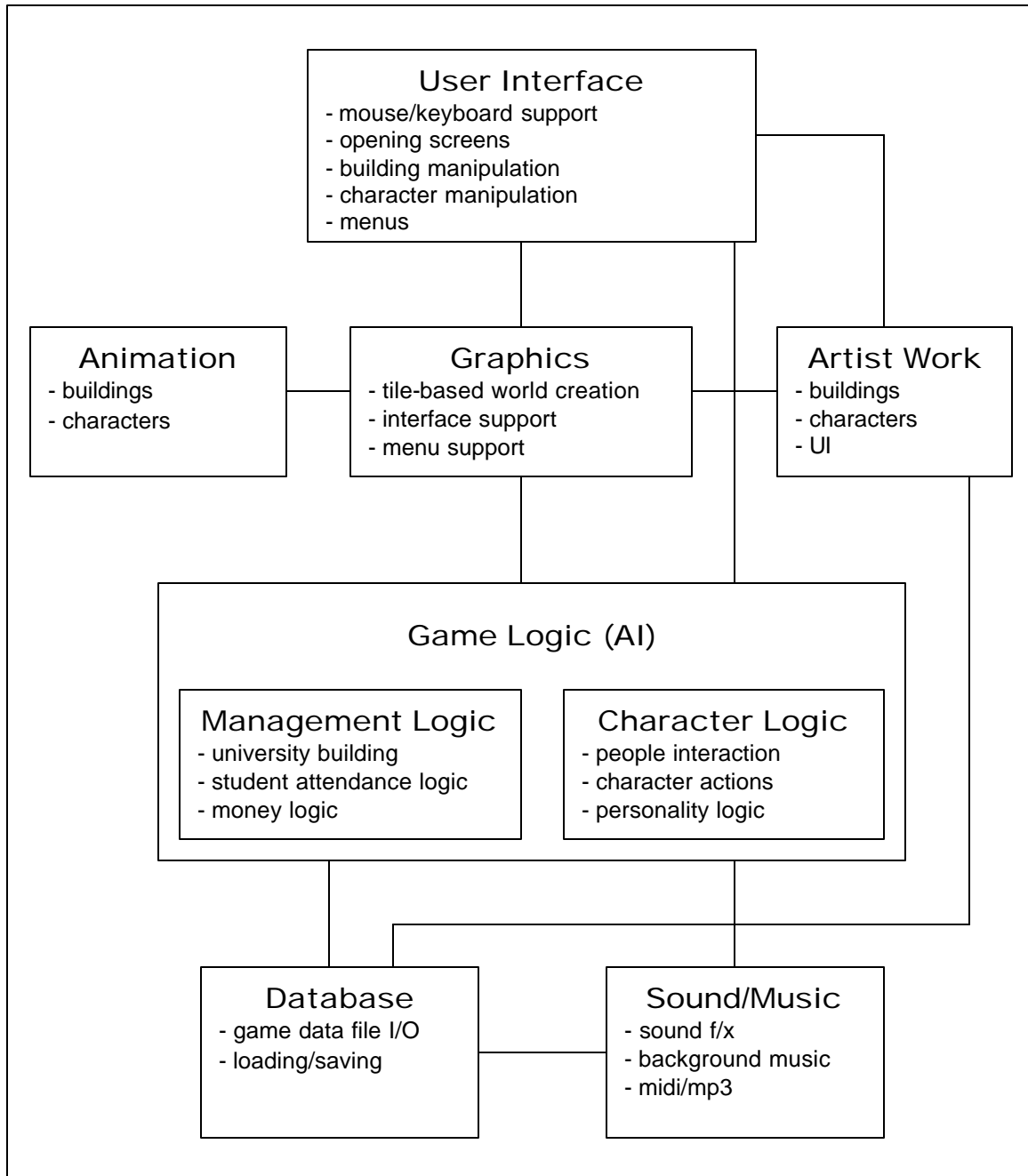
The CS190 version of SimsU is a stripped-down combination of the popular computer games The Sims by Maxis (see <http://www.thesims.com/>) and SimCity 4 by Maxis (see <http://www.simcity.com/>). However, instead of allowing the user to create and run a neighborhood with people performing the daily tasks of life, SimsU will allow the user to create and run a university with students and faculty who will go to classes, run errands, and socialize.

The object of The Sims and SimCity is to create a world and to ensure the well-being of its populace. In SimsU, the object will be to create a university that students will be interested in enrolling. When students enroll, they will pay tuition, which will give the player money to spend in upgrading the school. As the player builds better facilities and acquires better professors (which the player can also control), more people will want to attend. When students finally graduate, students which who are happy with their experiences at the university will give back to the university, giving the player even more money to spend.

To build a university, the player will be given the option to purchase building contracts to build dorms, libraries, greens, lecture halls, offices, dining halls, frat houses, athletic facilities, and campus bars. Because this is the Brown University version, these buildings will look similar to ones on the Brown University campus. The player can place these buildings (which will be pre-made so the user does not need to build one) anywhere he would like. Students will begin enrolling in the college once the user "opens" the university to the public.

On the more detailed level, the player will be able to control each of the students and professors. He will have the ability to control the people in it, seeing how they interact with their environment and others around them. This will affect the personalities of the students (happiness, study skills, smartness, social ability) and professors (teaching ability, likeability) and affect the money flow into the university. Because there can be too many students for the player to handle, an AI will handle daily tasks of students, such as going to classes, sleeping, and eating. However, to make sure students are having fun the player will need to specify what types of students to accept into the school (music types, jocks, nerds, etc) and modify their schedules to attend clubs and go out to parties. Based on this, the player creates a campus spirit which will characterize the university.

System Model Diagram



System Model Components

Game Logic (AI)

- Control the main aspects of player control in the game and decide the results of decisions the player makes.
- Management Logic & AI
 - Determines how building certain buildings and hiring professors will affect the how the game is played.
 - Manages where buildings can be placed and what can be built.
 - Controls how much money the player will get from student enrollment and how money is spent.
- Character Logic & AI
 - Determines how people will interact with another based on their personality ratings.
 - Controls how players will react if told to perform a certain action.
 - Decides what actions characters will take based on the schedule that the player creates for them (or that is automatically generated for them).
 - Determines where players will walk if told to walk to certain places on the campus, avoiding other players and buildings (collision detection).

Graphics

- Top-down view of the campus using an isometric tile-based map.
- Interface for manipulating buildings, people and the map.
- Menu support to allow user to set up game.

Animation

- Animate buildings to show activity going on inside of them.
- Animate people to have gestures, walking/running around campus.

User Interface

- Perform actions primarily by mouse clicks on graphical iconic menu on main map screen (like Warcraft III and many graphic adventure games) but provide keyboard shortcuts.
 - Clicking on people or buildings in game will bring up different features. When clicking on a building the user can select various options (upgrade, etc), or when clicking on a person options for modifying characters' lives will show up.
- Support for menus popping up during gameplay, and as a stand-alone screen.

Menus

- Initial Menu
 - Start Game
 - Load/Save Game
 - Quit
- In-Game Features
 - Create New Building
 - A list of buildings the user can create which the user can click and put into the map.
 - Hire Professor
 - Show list of professors with ratings and cost next to each.

- Management
 - Cash Flow
 - Student Enrollment
 - Professor Hiring
 - Manage Endowment
 - University Funding
 - Choose which departments to fund (CS and art!).
- Student Actions
 - Party It Up
 - Manage Schedule
 - Allow player to choose which classes the person attends.
- Much more...

Artist Work

- Designing buildings from an isometric top-down view.
- Drawing lots of different types of characters to show variety of students and to differentiate them from professors.

Database

- Method of storing data in a compact, efficient way.
- Creating load and save formats for storing campus information.

Sound/Music

- Creation of sound effects to play during the game.
- Support for background music which matches mood of game (mp3/midi).

User Interface



This is a screenshot from the newly-released game Simcity 4. SimsU will borrow heavily from this game as it uses the same concept. A simple menu interface with graphical icons will appear on the bottom left the screen always. When the user clicks on any of the buttons, more options will appear, showing character information, buildings available to build, or cash to spend.

Non-Functional Requirements

Performance

- Data needs to be efficient and the graphics engine needs to be solid.

Testing

- Each feature needs to be tested thoroughly. This means lots of playing of the game after the game is integrated of every aspect of the game.

Reliability

- Because SimsU is a game, SimsU should not crash and algorithms and logic should be solid and not cause anything weird to occur.

Ease of Use

- Like any game, it will take time for the user to become accustomed to the interface. However, there are many games that have a similar interfaces and it should not be anything new.

Portability

- This game could be ported to any other system that supports 3D graphics and C++.

Documentation

- Documentation will involve instructions on how to play the game.

Dependencies on Other Systems

- Choice of libraries for graphics and sound that fits the game's criteria will need to be decided before any programming will start.

Risky Parts

The greatest concern is that two weeks is not enough to program this game. People who are interested in joining with me to create this game will need to be ambitious and devoted for those two weeks. I am concerned that the scope of the game is too big, however, I have scaled it down greatly from my original plans from my original project proposal. I believe it can be done with 10 devoted programmers.