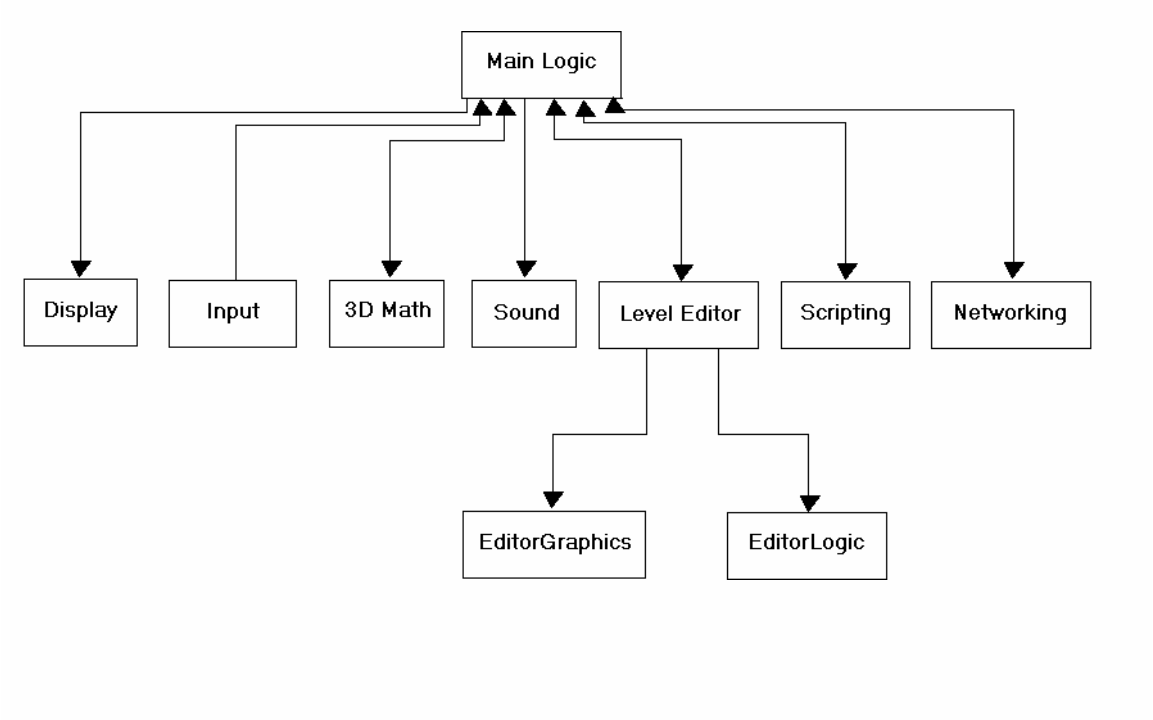
CS Contra

CS190
Top Level Design
By Tim Neal (tneal)

1.0 COMPONENT DIAGRAM



2.0 DESCRIPTION OF COMPONENTS

2.1 Main Logic

This component is the center of the entire project. It will keep track of everything that is going on in the game, and it interfaces with all the other parts of the program to keep communication going between the other components. This is where all of the game objects such as enemies, players, and backgrounds will be generated and stored in maps or vectors. It is also where all of the main game logic should take place. This component of the project is the most important because it needs to make sure that all the other components are able to communicate with one another through it.

2.2 Display

This component displays the action that is happening in the game and draws the objects on the screen. It will use OpenGL and GLUT in order to accomplish this. It will be important that this class run smoothly so that the animation is fluid in the game and so that objects do not seem to appear out of thin air.

2.3 Input

This component is responsible for handling all the input that is coming in from the user such as key presses, mouse movements, and commands. It will have to notify the main logic when input comes in, and the main logic will handle it from there.

2.4 3D Math

This component will contain the math needed to do collision detection and such calculations for the program.

2.5 Sound

This component will have two tasks. First off, it will have to play background music for the game. Secondly, it will also have to play sounds for certain actions in the game. Both of these parts should be able to draw from a library of music or sound and play it when called upon by the logic.

2.6 Level Editor

This component has two parts. The first part of the level editor is the graphical part; all the art that can be used to make a level must be made available in the level editor, as well as any additional art that could possibly be used. The second part of the level editor is the logic component of the editor. This component contains all the information needed to run the editor, and it makes sure that the user interface works correctly.

2.7 Scripting

This component will be used to log certain attributes such as information about an enemy or player. The logic can query the scripting component to check all sorts of information about the players or the enemies, and then it can make choices based on the information it has retrieved.

2.8 Networking

This component will need to find any other players on the network that are interested in playing a game. It will then need to send information back and forth between the two computers to make sure that both players are seeing the same images on

their screens at the same time. The server will handle all of the sending of the information, and will just handle the client computer as another input device.

3.0 EXTERNAL DEPENDENCIES

This program will have several external dependencies. The graphics will depend on both OpenGL and GLUT libraries in order to run correctly. The networking component will surely depend on some networking libraries. The sound portion of the program will depend upon some external library of sounds that it can call upon. There will also be a dependency upon the art designed for this program.

4.0 TASK BREAKDOWN

4.1 Project Leader/Documenter/CVS Organizer

This person will be in charge of several different parts. First of all, this person will be responsible for organizing meetings and making sure things are getting done on time. This person can also organize CVS, which should not take too much time. They should also be the one to do the documentation for the program once it is all done, and this also should not take too much time to do.

4.2 Main Logic Coder

I believe two people are needed on the main logic because this will be the center of the whole program. The people who code this part of the program really will be responsible for making sure that all of the other parts of the program come together smoothly, and that all the interfaces are working correctly. This should be more than enough code to write for two people.

4.3 User Interface

This person will be responsible for dealing with the user interface for the program. They will have to handle all user input and make sure that it goes to the main logic. This person must also make sure that the graphics are put together correctly to form the environment.

4.4 Graphics Design

This person will be responsible for designing all the artwork for the actual game. They will need to do the designs for the characters as well as the designs for all the components of the maps.

4.5 Scripting/3D Sound

This part has two tasks. The first task would be to handle all the scripting, which should not be too large, but it has the possibility to get quite big if extra time is available. The second task that this person would handle would be to make sure that 3D sounds were implemented and that they interfaced correctly with the main logic.

4.6 Networking

This person will be responsible for doing the networking part of the program. This will involve checking to see who is available to play online for a two-player game, and it will involve passing information back and forth between the two computers so that both players see the same images on their screens.

4.7 Math/Tester

This part has two responsibilities. First, this person will do the 3D math part of the program. This part really should not take that long, especially since most of that kind of stuff was covered in CS123, so this person's other responsibility is to be the tester. This person will run the code at different stages to search for bugs, to make sure that things are running smoothly, and to make sure that all requirements are met.

4.8 Level Editor

I think there should be two people that are working on the level editor. One person should be working on any graphics that the level editor may need in addition to the game graphics. The other person should be dealing with the main logic of the level editor, such as the user interface and saving and loading files.

5.0 GROUP ORGANIZATION

Project Leader/Documenter/CVS Organizer (1 person)
Main Logic Coder (2 people)
User Interface (1 person)
Graphics Design (1 person)
Scripting/3D Sound (1 person)
Networking (1 person)
Math/Tester (1 person)
Level Editor (2 people)

While this should cover most of the tasks needed to complete the project, any other miscellaneous tasks can be divided up among any of the group members who have some extra time. Anyone who finishes their part early can move on to help out with testing/debugging, and can work on implementing any extra features.

6.0 SCHEDULE

M 3/3 – Group Roles should be decided.
W 3/5 – Group meeting to discuss project.
F 3/7 – Final Design for project should be finished.
W 3/12 – Interface prototypes.
M 3/17 – Group meeting to discuss interfaces.
F 3/21 – Final Interfaces completed.
M 3/31 – Group meeting to discuss design.
F 4/4 – Detailed Design complete – Start coding.
F 4/11 – Group meeting to discuss progress.
W 4/16 – Start to integrate the code.
M 4/21 – Begin Testing/Debugging of code.
F 4/25 – Code optimization and complete documentation.
M 4/28 – In-class Demos
W 4/30 – Group meeting to discuss any final debugging/editing/optimization or any final add-ons to the program
F 5/2 – Public Demos

7.0 ASSUMPTIONS

No assumptions were necessary; I felt that the specifications documentation was clear enough for me to get a good enough picture of how this program should be split up.