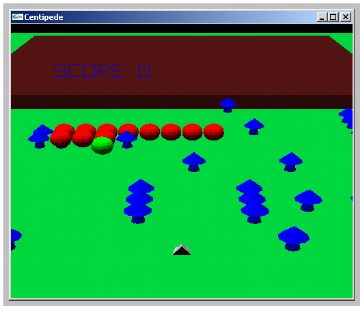
Dan Morris
CS148

Handout #16
Summer 2005



# Project 4
# Interactive 3D Game

CS148 PP5: 'AliBaba' by Jeremy Slade (jslade)
40 thieves / 0 killed / 0 escaped

## Due: Thursday, 8/11/05, 11:59 pm (max one late day)

## Project Goals

It's time to put together everything you learned in CS148 to make an interactive virtual world, most likely in the form of a game.

## Platforms

Unlike previous projects, you don't have to make this project run on Linux. Your program should run on Windows 2k/XP, Cygwin, Linux, *or* OS X. If you choose to use Linux, your program must run on the Sweet Hall Linux machines.

Easing up the usual platform restrictions should save you some time, and also opens up the whole world of platform-specific libraries to you. If you want to use OS-specific features for user input, sound, networking, etc., go wild.

You don't have to use GLUT, but you do need to use OpenGL. That means that if you use features from a game engine or game library, make sure you're still writing your own OpenGL code.

## Supplies

We won't be providing any new support code for this project. In fact, we won't provide sample solutions… we'll show some examples in class, but creating your game is up to you and your imagination.

We encourage you to use the mesh-loader and image-loader that we provided for pp3, to use the examples we've discussed in class or put online in the examples directory, and to use libraries that you find on the web.

One particularly useful library is the Simple Vector Library, which might help you if you need to matrix or vector operations:

http://www-2.cs.cmu.edu/~ajw/doc/svl.html

If you're working on Windows and you want to add music and/or sound to your game, you might find the BASS audio library helpful:

http://www.un4seen.com/

Please make sure to include *all* files required to compile and run your project, including any necessary libraries, header files, project files, images, meshes, etc.

If you are feeling particularly motivated and want to work on a networked game, speak to us and we'll help you get set up with some basic networking code.

## Groups

This is – optionally – a group project**.** You may work in groups of up to three. Your project is expected to be somewhat more complex for group projects, although two people don't have to submit twice as much code as one person. A group will get full credit for meeting the project requirements and doing just a bit of extra credit.

We do recommend working in groups, so that you can walk away from CS148 with a slightly larger project to show.

## The Assignment: 3-D Game (or something else)

Your assignment is to write a 3-D video game using OpenGL. You are free to design and implement any sort of game you like, as long as it incorporates the required functionality described below. Your game does not have to be original – in fact, since you only have a short time to write your game, it may be easiest to model your game after an existing game.

You are not required to develop a game, although some of the required features won't make sense in other types of application. If you want to work on something that's not a game, talk to us and we'll come up with an appropriate set of alternate requirements. Options might include some kind of data visualization tool, a snazzy 3D file browser for your computer, a 3D front-end for your favorite data analysis tool, etc. If you're going to stray far from the game theme, try to talk to us within a few days of the day this handout is going out, in case we really can't find a way to make your idea satisfy the requirements.

## Requirements

- A 3-D animated scene
- Smooth, time-based animation for some aspect of your game
- At least one change in your viewing transformation (i.e. the camera should move or multiple cameras should be available)
- Colored and shaded objects with lighting
- Multiple light sources

- At least two texture-mapped objects. At least one of these objects needs to be a polygonal mesh (i.e. not a GLUT object or OpenGL quadric). Feel free to reuse code or objects from PP3.
- At least one partially-transparent object. This can be a 3D object or a 2D overlay.
- OpenGL picking with the mouse (this could be a menu or it could be integrated into the game play)

Ideally, you'll be able to fulfill each of the requirements above in a way that adds to the experience of playing your game. In other words, instead of creating separate modes just to demonstrate each requirement, try to integrate all the requirements into your game play.

Since this is a graphics course, your grade will be primarily based on your game's graphics, which includes the user interface and overall look-and-feel. You will receive some extra credit for coming up with a brilliantly creative game, but this will not compensate for problems with graphics or with the basic requirements, so we encourage you to start with a simple game concept, so you can get the graphics down before adding complex game features. For instance, a simple 3-D conversion of a 2-D game like Centipede or Pac-Man would be a great place to start.

## Design Reviews

You aren't required to submit anything to us until the day the project is due, but we strongly encourage you to stop by and discuss design or implementation issues with us, so we can help you stay on track. We may be able to help you figure out early what aspects of your game will be especially time-consuming, or to give you feedback about how your graphics might be improved.

## In-Class Demo Day

On the final day of class, students will *optionally* present your final projects to the class. Participating groups will have 5-10 minutes to show off their project, ideally some of which should be time where other students are playing your game. Knowing that your project will be up on the big screen is good motivation to tighten up the little details and get it ready for prime time, so we encourage you to take advantage of this opportunity.

To further encourage you, we will give ten points of extra credit on the project *just for demo'ing*. We will also let the class and staff "rate" each demo, and the "winner" – the highest average rating – will receive 50 points (yes, out of 100) of extra credit for the project. Second place will receive 35 points, third place will receive 20 points (all on top of the ten points you get just for demo'ing). There is no maximum amount of extra credit for this project.

To even *further* encourage you to demo, if you demo in class *and we can see that you've included all of the required features*, we will grade your project based on your demo (unless you ask us to do otherwise). This is a huge win, since a well-run demo can make sure we only see the strengths of your game and not the… errr… minor problems that you *could* have fixed if you had more time.

Free extra credit seems like a good deal to us, so we figure almost all of you will want to take advantage of this opportunity. Please mail cs148staff@cs on or before Monday, August 8th if you *are* planning to demo in class on Thursday, August 11th.

We will bring laptops running Windows and Linux, but we recommend that – if possible – you demo on your own laptop to avoid last-minute machine-specific issues. Please let us know if you will need one of our machines.

If you are a remote student, we will be happy to demo your game for you, just tell us what platform you're using by August 8th, and send us your code and binaries by 10pm on the day *before* the morning of the demos (we won't use this as your handin, just as your demo; you should still hand in before the deadline). We can run the show for you, but ideally you could call in and guide us through the demo during the demo session.

All local students should plan to attend class that day, or you will make us sad, you'll be out ten points, and you'll miss all the fun.

## Deliverables

You must include a README. The README should document:

- who is in your group
- the platform(s) your program runs on
- any surprising steps we need to know about to compile and/or run your program
- how to play your game
- any bugs we should be aware of
- anything you added *after* your demo
- any additional information you think would aid us in grading your project

You should **submit your code and a program executable** (unlike previous projects) using the submit script. Be sure to remove any object files since they will take up a lot of space on the server. You should run…

```
/usr/class/cs148/bin/submit
```

…as always, but the submit script will not try to compile your project this time around, since projects are not required to compile under Linux.

## Motivational Blurb

This is a great chance for you to create a cool application that you can show off later, and to really put your graphics skills to the test. We hope you'll take advantage of this opportunity and put some time and creativity into the project…

CS148: Intro to Computer Graphics
summer 2005 edition