CPSC 314: Programming Assignment 1

Due 4pm, Friday September, 23 2011

This is an exercise to get you started on OpenGL and experiment with both drawing and callback commands. The goal of the program is to provide a simple game where the user moves a paddle to deflect falling balls.

- Download and untar a1.tar.gz (from http://www.ugrad.cs.ubc.ca/~cs314/Vsep2011/a1/a1.tar.gz) It includes a source template (files a1.cpp, ball.hpp/cpp, paddle.hpp/cpp) and an executable a1_sol which runs on the department Linux machines.
- The department Linux machines should have all the libraries you need for the assignment. If you work at home (Linux or windows) you need to install glut libraries and headers on your machine. Check

  `http://www.opengl.org/resources/libraries/glut/glut_downloads.html`

  for installation instructions.
- Run 'a1_sol' to get a sense of what a possible assignment solution should look like.
- Build the template executable. In Unix use the provided makefile. In Microsoft VS use the provided project files. Depending on your home environment you might need to do other changes to make the code run.
- Make the following changes:
  - Draw a proper moving paddle: Modify 'Paddle::draw' function to draw a rectangular instead of square (cube) paddle. Modify the 'mouse_move' callback and the 'Paddle::draw' function to position the paddle centered at the 'x' coordinate of the mouse (ignore the 'y' coordinate). Hint: look at the way ball drawing (size and position) is controlled and use the same framework.
  - Allow changes in paddle size: Have pressing the '>' key increase paddle width and the '<' key decrease it (make sure your paddle doesn't end up having negative width).

- Generate colorful balls: There is a color variable associated with each ball, your task is to set this variable to a (semi-random) valid color value and modify the ball drawing function to use this color. Hint: look at how colors are set right now for balls and other elements (RGBA).
- Let there be more light: Use a random initialization to set the 'isLight' variable for a small subset of balls to true (note OpenGL cannot handle above 20 or so lights). For balls where the 'isLight' variable is true generate a light at the center of the ball. To set the light modify the 'Ball::setupLight' function. Hint: Look at 'setup_lighting()' function in 'a1.cpp' and use a similar setup. Note that you need to use $GL\_LIGHT0 + lightNo$ as the identifier for the light.

The required changes so far will earn you 85% of the grade. To earn the remaining 15% as well as bonus marks you need to add other improvements (the size of the bonus will be at the marker's discretion). For example, you could add the randomized ball sizes and the ball explosion (on mouse click) in the 'a1_sol' example, or add bricks that disappear whenever a ball collides with them (as in classical arcanoid), or make the game a multiplayer one (two paddles on opposite sides of the scene, two extra keys to control); you can make any other change as long as you focus on tasks involving OpenGL knowledge.

Document any changes you do in the README file you submit with the assignment. Advice - implement and test all the required tasks first before starting the free-form part.

Hand-in Instructions

- Create a root directory for our course in your account, called cs314. Later all the assignment handin files should be put in this directory.
- For assignment 1, create a folder called assn1 under cs314 and put all the source files that you want to handin in it, including the "makefile". Don't use subdirectories – these will be deleted. NOTE: we only accept README, makefile and files ending in cpp, hpp, c, h, txt.
- The assignment should be handed in with the exact command:

```
handin cs314 assn1
```

  This will handin your entire assn1 directory tree by making a copy of your assn1 directory, and deleting all subdirectories! (If you want to know more about this handin command, use: man handin.)