

# CS174 Homework 2 - Spring 2010

Due 11:59pm on 5/21/2010

## Overview

In this assignment we will create an animation of a pinball machine. We will draw the pins directly (not using the glut or glu drawing functions), specifying their normals. Additionally, we will create camera view modes which move the camera around the scene and follow the pinball. Useful OpenGL functions for this part include: gluLookat to position the camera and glOrtho, glFrustum, or gluPerspective to setup the view.

## Template Code

The template code (pinball.cpp, Timer.cpp, Timer.h) provides convenience functions which draw axes, a ground plane, and provide some basic interaction. Pressing 'v' will toggle the camera viewpoint between a number of fixed positions. Pressing 'q' will quit. You should override 'v' as specified below and add the other options outlined in the requirements. The template code also includes a timer class to control your animation (by accessing a variable called Time). Do not modify Timer.cpp or Timer.h.

## Requirements

- There must be at least 10 pins and 4 walls for your pinball machine.
- The required pieces must be drawn using simple polygons (triangles or quads), specifying the normal for each. You may draw other things (including the pinball) using the convenience functions provided in assignment 1. Also, you may fill the top of your cylinder with a sphere, so that when viewed from the top it can be seen easily.
- When 'n' is pressed, drawing of the normals should turn on and off
- All normals should have unit length and be located on the center of the polygon they correspond to.

- The initial direction and location of the pinball should be chosen randomly at runtime (unless you specify otherwise in the extra credit).
- When the pinball runs into a pin or a wall, it should reflect off of it and continue along.
- Pressing 'v' should toggle between viewing modes.
- The first mode should be a static top-down orthographic view of the scene.
- The second mode should be an animated perspective flythrough which forms a complete loop and repeats. The path of the camera is up to you, but it must turn 360 degrees around the scene and zoom in or out at some point during the process.
- The third mode should be a first person view of the ball as it moves.
- Pressing 's' should draw a wireframe outline of the viewing volume used in the third mode (in other words, the viewing volume of the pinball).
- Pressing 'r' should reset the process, starting the pinball somewhere else in the scene (randomly chosen).
- When resized, the aspect ratio of the objects in your scene should stay the same (no stretching).
- All code should be in one file, called pinball.cpp

## Submitting

Use courseweb to submit a zip file containing pinball.cpp and screenshots of your simulator in action. Show at least one screenshot of each view, one with the normals enabled, and one with the viewing volume of the pinball drawn. The format of the screenshots should be jpg. Name your zip file [Last-Name][FirstLetterofFirstName].zip.

## Extra Credit

For up to 5% extra credit, do something extra. For example, add more realistic physics (friction, or rolling), more views (different fly-around modes), more complicated objects (why not robot pinball?), multiple balls, a working spring system to launch the ball, or anything else you can think of. Please note what you have done for extra credit in the comments of the source code with clear labeling (e.g. "Extra Credit:").

## Collaboration and Late Policy

You can team-up with another person for this homework. However, at the submission time add a README to the zip file you are submitting, and explicitly identify the contribution of the team members. You will also be allowed to submit this homework one day late. However, you will be evaluated over 80% of the total credit.