CS334 Computer Graphics

Problem Set 2
Plane Animation

Handed out: Tuesday, September 7
Due: Tuesday, September 14

The purpose of this assigment is to become familiar with basic OpenGl and glut programming. You will write an animation program for a circular ball in a polygonal room. Let $(x(t), y(t))$ denote the position of the ball after moving t seconds from initial position (x_0, y_0) with initial velocity $(\dot{x}_0, \dot{y}_0)$. The mass of the ball is 1. When the ball is in the air, it moves ballistically:

$$\begin{aligned} x(t) &= x_0 + \dot{x}_0 t \\ y(t) &= y_0 + \dot{y}_0 t - 0.5 g t^2 \end{aligned}$$

with $g = 9.8$ the gravitional constant. When the ball hits the polygon, an elastic collision occurs then ballistic motion resumes. If the pre-collision velocity is v and the outward normal of the ball at the point of collision is n, the post-collision velocity is $w = v - 2(v \cdot n)n$.

The animation program computes the state (position and velocity) s_i of the ball at a sequence of time steps t_i starting from $t_0 = 0$. Set the time step δ to $1/30$ of a second. State t_{i+1} is computed from state t_i as follows. Compute the state s_b at time $t_i + \delta$ under the assumption that the ball moves ballistically. If the ball is inside the polygon, set $t_{i+1} = t_i + \delta$ and $s_{i+1} = s_b$. If not, set t_{i+1} to the time t_c where the ball hits the polygon, compute the position and velocity at t_c under ballistic motion, and update the velocity using the elastic collision formula.

User Interface

The program opens a window and processes keyboard and mouse commands.

- p Create a new polygon $a_1, a_2, \ldots a_n$ and delete the previous polygon. Enter the first $n - 1$ vertices with left mouse clicks and the final vertex with a right mouse click. A middle mouse click deletes the latest vertex. Display the polygon unfilled.
- b Create a new ball and delete the previous ball. Press the left mouse key to specify the initial position of its center, drag the mouse to specify its radius, and release the mouse key to finish. Display the ball filled.
- a Animate the ball from its initial position with initial velocity $(0, 0)$. A mouse click restarts the animation from the mouse position with velocity $(0, 0)$. If the ball is not inside the polygon, it does not move.
- s Stop the animation.
- q Exit the program.

Hints

- Use a variable to record the current state: creating a polygon, creating a ball, animating, or neutral. Store the polygon, the ball and the state in global variables.
- Render the ball as a triangle mesh using problem 1 from homework 1.
- Use shader.h to construct shaders, as explained in class.
- Use vertexarray2.h to define vertex arrays and vertex index arrays. The vertices are 2D and have no color.
- Use shader2p.vs and shader1.fs to draw the polygon in a specified color. Create a second vertex shader for drawing the ball at its current position.
- Use glutTimerFunc and double buffering for animation.