MicroProject for Lectures 09-10

Jarek Rossignac
www.gvu.gatech.edu/~jarek

1 - Objective

Learn how to implement collision between a moving point and a polygon.

2 - Deadline

It should be submitted on T-Square before the beginning of Lecture 11. No extensions.

3 - Deliverable

You need only to upload a video. No report. No code.

You may work in a team of 2 or individually if you prefer.

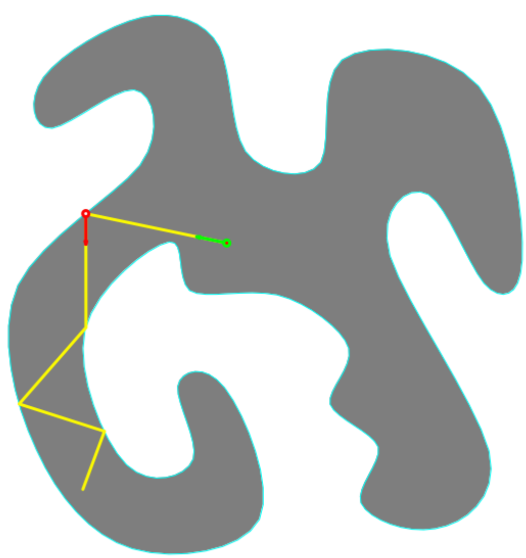
All members of the team must contribute to the video by speaking.

The faces and names of team members, the title of the video, and the class name ("CS3451) and date should be shown as the first frame of the video.

4 - Assignment details

4.1 Requirements

You should produce a 3-to-5 minutes video where you explain how to produce an image, such as the one shown below, where we have a polygon P bounded by a single loop of edges (the standard pts object) and a finite length ray R defined by the green point Q and by the green velocity vector V. You are also given a distance d: the length of the ray. The ray travels straight inside the polygon and is reflected as in a perfect mirror reflection when it hits an edge. But its total length remains d. Hence, the ray traces a broken path as shown in yellow.



The video must clearly explain the following points, using drawings or animations, showing the math formulae, and explaining the correct algorithms for computing what is needed:

- How do you test whether R intersects an edge [A,B]?
- How do you select the edge of P that R intersects first?
- How do you compute the corresponding collision point X?
- How do you compute the direction of the reflected portion of the ray?
- What is the overall algorithm for tracing the yellow path?

After these fundamentals, your video should explain how such a tool can be used to produce an animation of a particle moving inside the polygon (starting at Q and moving with velocity V).

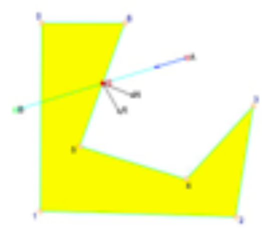
> Hint: set d to be a small distance that is travelled by the particle during the time (1/30 of a second) between two consecutive frames shown by your application.

At the end of the video, you should show the animation of the particle inside the polygon, as it bounces off the boundary edges.

4.2 Base code

It is strongly suggested that you verify your math and algorithm experimentally.

Also, I have posted an applet called Ray at http://www.cc.gatech.edu/~jarek/demos/



This old code does not have any comments and may be using geometry functions with different names than the ones used in the base code provided for this class. Still, some of you may be happy to use it as a starting point or a reference. That code does not have any video production facilities. So, I strongly suggest that you implement your own version of this code using the more recent base code. We recommend Project 06 (polyloop).

4.3 Extra credit suggestions

- Make a miniGolf game: Place the red ball and a green hole in the polygon. Let the user click and drag a line from the red ball to the mouse that defines the direction and magnitude of the initial velocity. Implement a friction model that reduces the instantaneous velocity by a fixed (constant) amount (not ratio) until the ball stops. Let the player play again. Detect when the ball falls in the hole.
- Make a racing game: Let the user control an acceleration vector with the mouse while the particle is moving. Update the velocity and the position of the particle so that it follows the parabolic path , although the path will change if the player changes the acceleration.
- Replace the point particle with a disk of a given radius r. This means that you need to detect collisions between a disk and the boundary of the polygon: it requires a different treatment for collision with edges and for collisions with concave vertices.
- Feel free to come up with your own ideas, but they should have similar complexity to the ones proposed above to warrant 20% extra credit.

4.4 Submission

This time, you are not required to submit any source code or a report, only the video, which should contain the names and faces of the two authors in the first frame. Compress the video, and name it P09-10_ParticipantsNames.zip.