Project Report - Project 3 Phase 2
Computer Graphics - CS 6491
Vinayak Gargya

Objective

The objective is to move a variable radius and height cylinder in the mesh given in the basecode. Each vertex of the mesh is a pillar of variable radius and we must move the cylinder in such a way that it always touches at least two pillars.The cylinder should move towards the current position of the mouse. While the cylinder moves around, its volume should stay constant, i.e. as its radius increases, its height should decrease to preserve its volume and vice versa.

Approach

The prime problem in this project was to be able to create a circle that is tangent to two other circles, i.e. a circle that touches two other circles but does not enclose them. The two other circles in this case are the two closest 'pillars'. The next problem was to move the circle such that it maintains this constraint. This involved figuring out the locus of the centers of such circles that are tangent to two other circles.

Circle tangent to two circles

In order to find such a circle, we must find its center and its radius. The center would be a point that would be equidistant to the closest points on the other circles and the radius would be the distance between them. I started off with the simplest case which is to assume that the center of the circle lies on the line connecting the two other circles' centers.

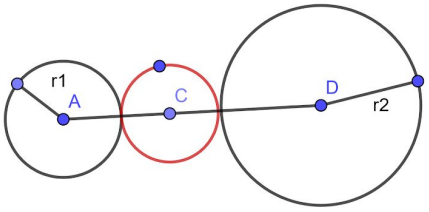


Figure1

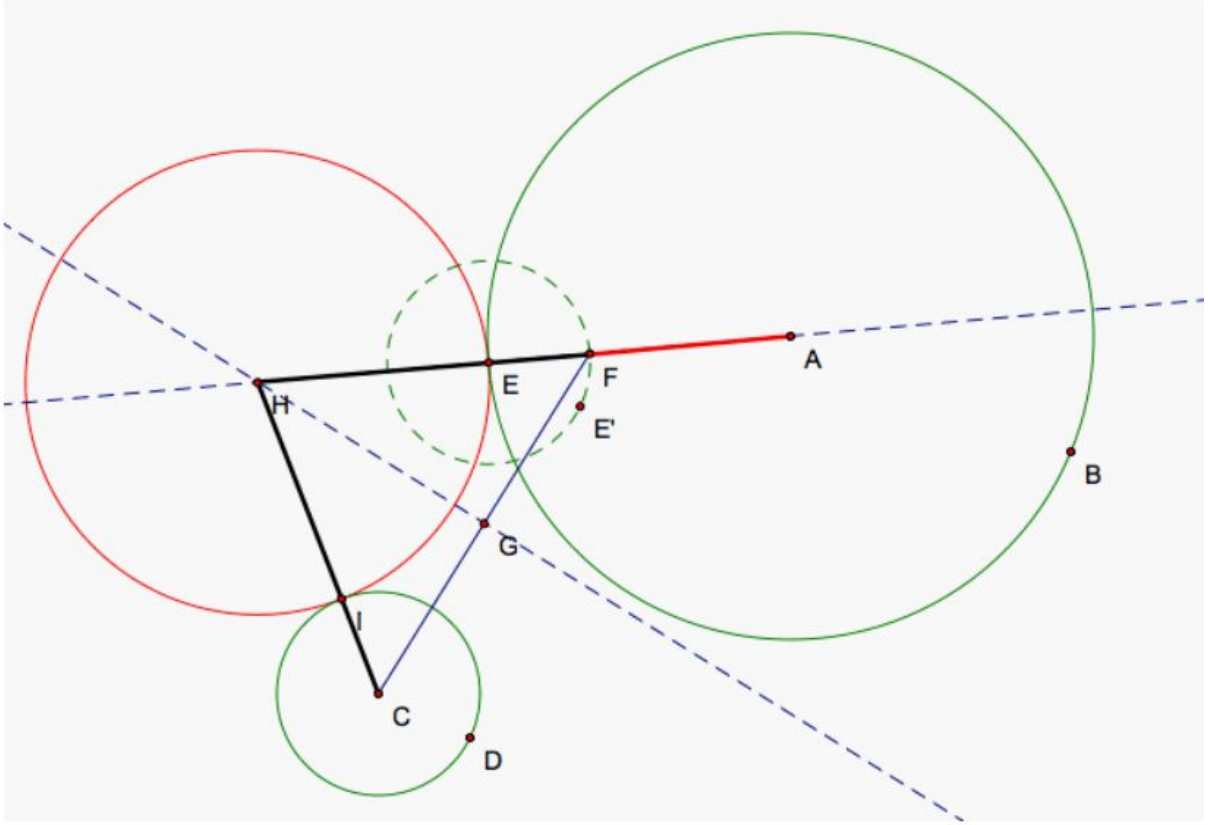
Center of circle 1 = A, center of circle 2 = D
Then the radius of circle 3

$$R = [AD - (r_1 + r_2)] / 2$$

And it's center

$$C = A + (r_1 + R)\overline{AD}$$

This is only one such possible circle, there can be many more such circles that touch the two circles.



The locus of the centers of such a circle is a hyperbola with the focus as the centers of the two circles. The constant difference between the distances from the hyperbola to its focus is the difference in the radius of the two circles r_2 - r_1 . This can be confirmed from the figure above since HF = HC and AE - EF = r_2 - r_1
Therefore, distance HC - HA = AF = AE - EF = r_2 - r_1
Once we have the foci and the constant difference in the distances for the hyperbola, we can figure out different points on the hyperbola using parameterized equations.
For a hyperbola,

$$x(t) = a\,sec(t)$$
$$y(t) = b\,tan(t)$$

Here, we assume that the hyperbola is horizontal and these points lie on AD and its perpendicular from point C in figure 1. We can project these points on the real world axis to obtain the world coordinates of these points. A and b are constants for the hyperbola with

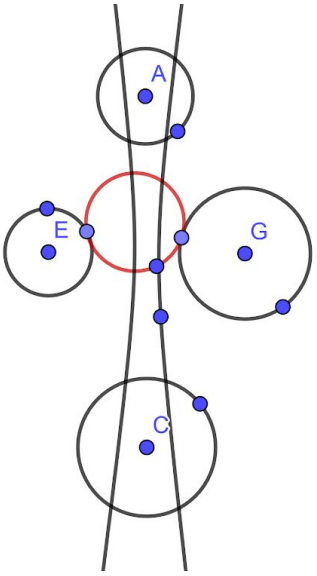
$$a = 1/2 * (|r_1 - r_2|)$$
$$b = c^2 - a^2$$
$$c = ||A - (A+H)/2||$$

We parameterize t uniformly to get a nearly uniform speed for the circle.

Circle tangent to three circles

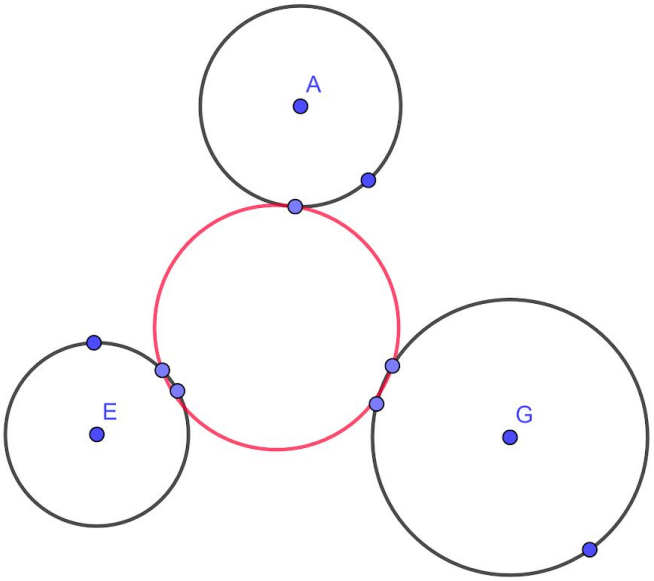
While the tangent circle is moving in between two pillars, it is likely that at some point it will hit another pillar and be tangent to three pillars. Once this happens, we don't need to expand the circle anymore and move it along the hyperbola for the first two pillars. In every iteration, I perform a collision check between the circle and two other pillars. For example, consider

the case given below. The circle in red is the tangent circle between pillars E and G. While it is growing, it can possibly collide with pillars A and C only. Once this collision happens, we would have a circle tangent to 3 circles.



Shifting to a new pair of pillars

Consider the case where the circle is tangent to 3 other circles. This is an Apollonius circle. At this point, in order to move the circle towards the mouse pointer, we must choose a pair of pillars in between which we should move our tangent circle. The choice of the new pillars depends on the relative position of the mouse. I calculate which edge's midpoint is closer to the mouse and choose that one.



However, at this point, we have a problem. We also need to calculate the parameter value t for this hyperbola which would give the same center and radius of the tangent circle as the one before it. For example, consider that the original pair of pillars were E and G and we would calculate the center and radius of the tangent circle using the hyperbola between

them. Let's say at parameter t = 0.5, the tangent circle collides with another pillar. Now we choose a new pair of pillars, let's say AE. For this hyperbola, using t = 0.5 would give us an entirely different point and the circle will 'jump' to a new location. We don't want this to happen, hence, we need to calculate a new value of t, which would give approximately the same center and radius for this new pillar pair AE as we would get for the original pair EG with t = 0.5. This is easily done by projecting the current center and radius on the axes of hyperbola of AE and then calculating the value of t using arctan2. This process is repeated until the tangent circle reaches the mouse cursor's position.

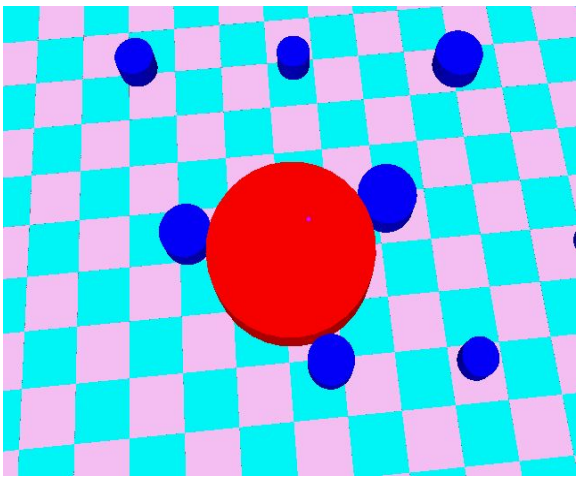
Preserving the Volume

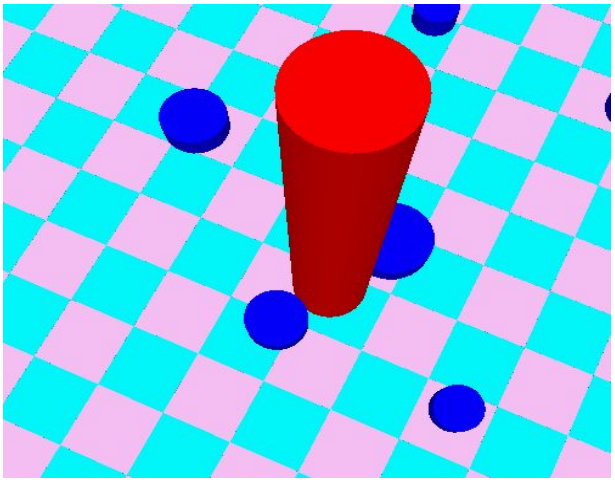
The initial volume of the pillar is stored and the new height of the pillar is calculated such that the volume remains same whenever the radius changes using the formula:

$$Height = VOLUME / (radius * radius)$$

Results

The result is a cylinder moving between the pillars until it reaches the mouse position, while staying tangent to at least two pillars and preserving its volume





Limitations/ Future work ideas

The path finding approach that the circle uses right now is 'greedy'. In case there is a 'hole' in the mesh or it is concave' at some points, the pillar may never reach the desired point. There could be a better approach to path finding. The pillar also does not go out of bounds of the mesh, i.e it will not expand to a point outside the mesh. At this point, new pathfinding only happens when the pillar collides with a third pillar. This could be changed to make it realtime

How to Run

1. Click on the blue 'pillars' to move them around the mesh and position as desired
2. If you want to change the radius of a pillar, click on the pillar to select it and then press 's' and move the mouse along the x-axis to increase or decrease the size of the pillar
3. If you want to increase the volume of the cylinder, press 'v' and move the mouse to increase or decrease it.
4. Press '9' to start the simulation

5. The cylinder will follow the mouse and stop when it comes in contact with it

References

1. http://jwilson.coe.uga.edu/EMAT6680Su06/Swanagan/Assignment7/BSAssignment7.html
2. https://www.ck12.org/book/CK-12-Calculus-Concepts/section/10.3/