

Summary

This project explores the foundations of topological data analysis by creating an interface in virtual reality to let users interact with the data. As the points move around, the initial configuration changes to another configuration, causing these simple shapes to change. By altering the radii of spheres (points) in this multi-dimensional space to a diameter of the *simplex* that is less than the chosen threshold value, the spheres should connect and create edges allowing for the construction of different shapes.

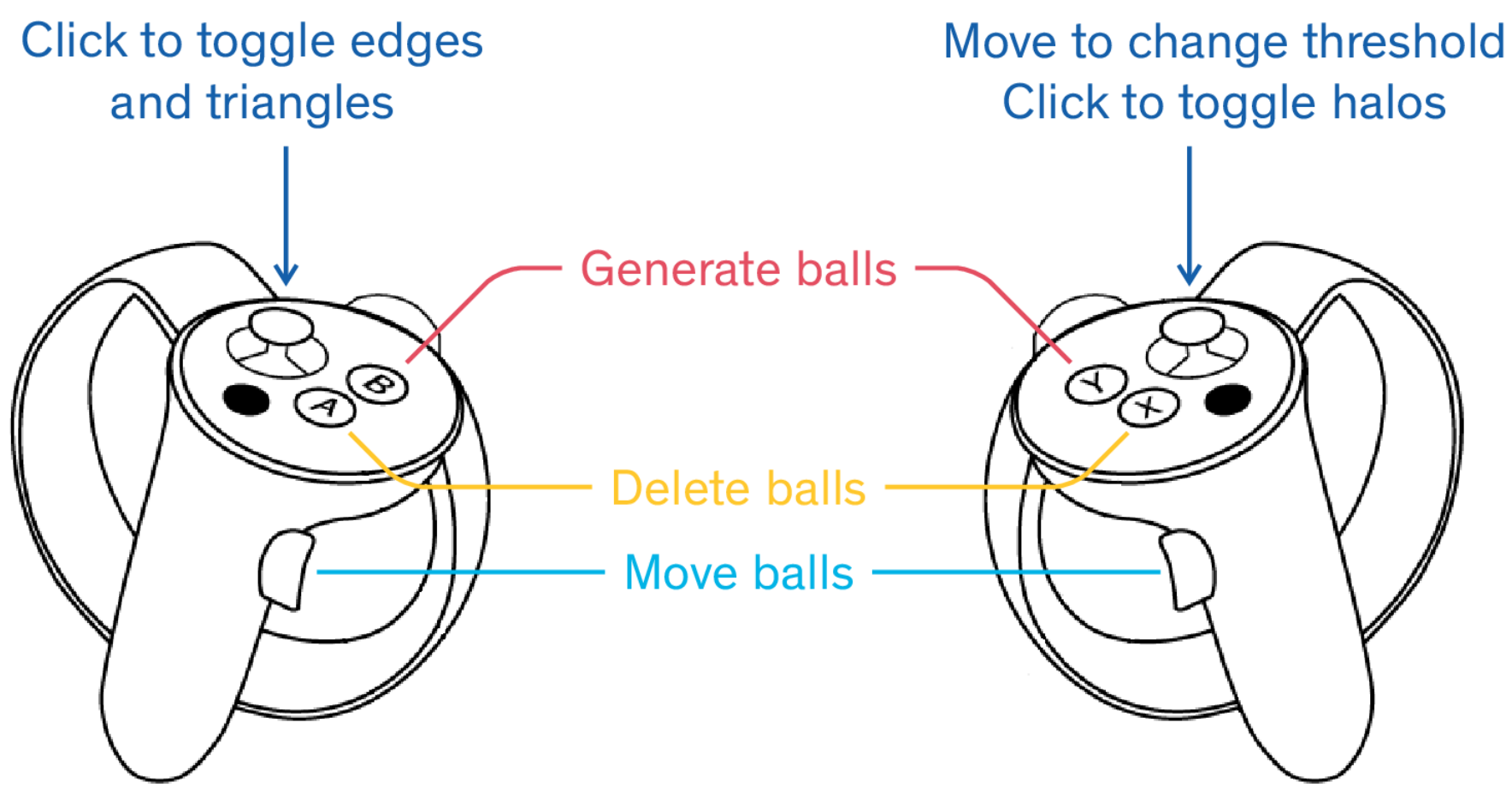
Motivation

We show how to make simple shapes as they need at least 3 edges and *vertices*. By using Unity, we can see how these shapes are changed in the space and how this can be applied to the real world.

Methods

In order to create our virtual application we used the game engine Unity, which supports the Oculus virtual reality head console and touch controllers. Creating the objects within Unity were possible through the implementations of scripts written in C#. These scripts were attached to objects in the Unity as a medium for interaction between the objects at a given run-time.

Instructions:



Simplices

Simplices are the simplest possible *polytopes*, geometric objects generalizing a tetrahedral region of space to n dimensions. The boundary of a simplex consists of sides called *faces*.

Examples of Simplices :

0-simplex: Points
1-simplex: Line segments
2-simplex: Triangles
3-simplex: Tetrahedrons

Simplicial Complexes

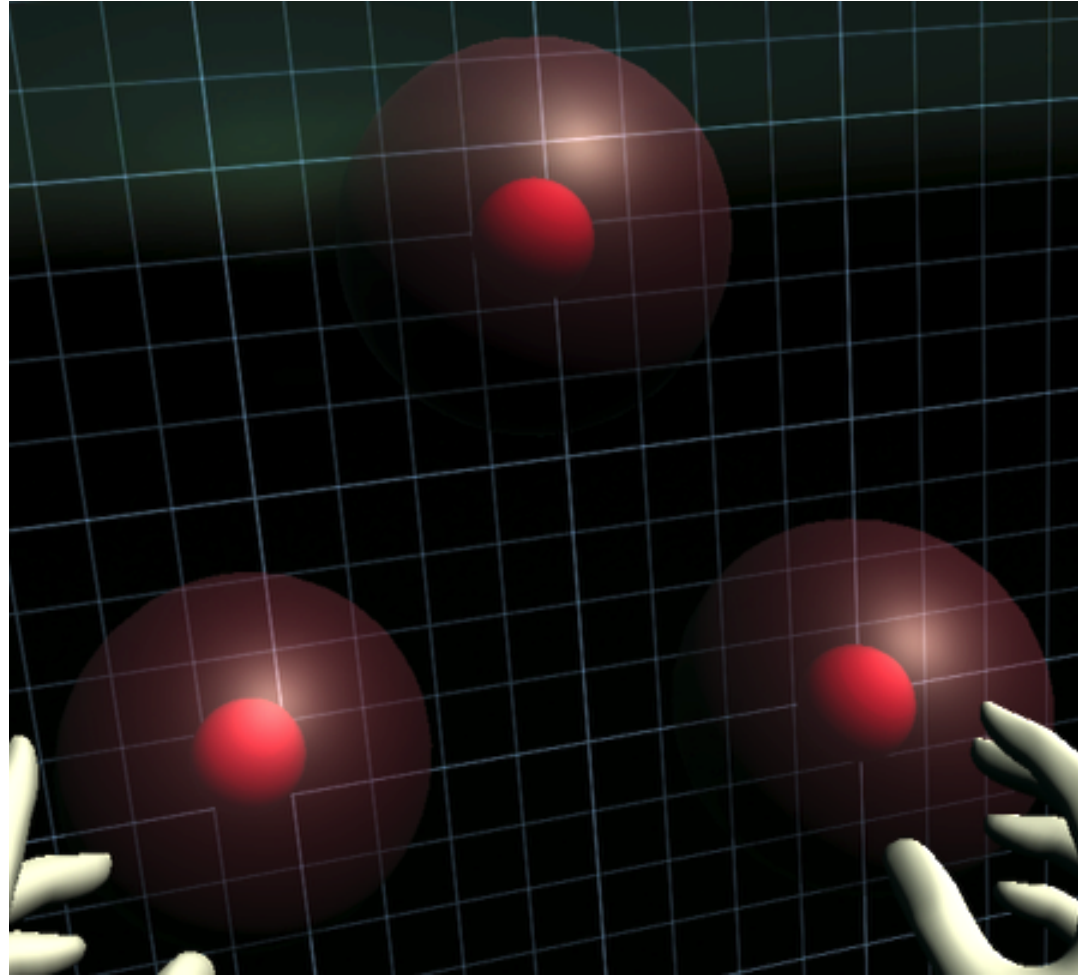
Simplicial complex are spaces with *triangulations*, the process of determining the location of a point by forming triangles to it from known points. Formally, a simplicial complex K in \mathbb{R}^n is a collection of simplices in \mathbb{R}^n such that the following apply:

1. Every face of a simplex of K is in K ,
2. The intersection of any two given simplices of K is a face of each of them

Main Objects

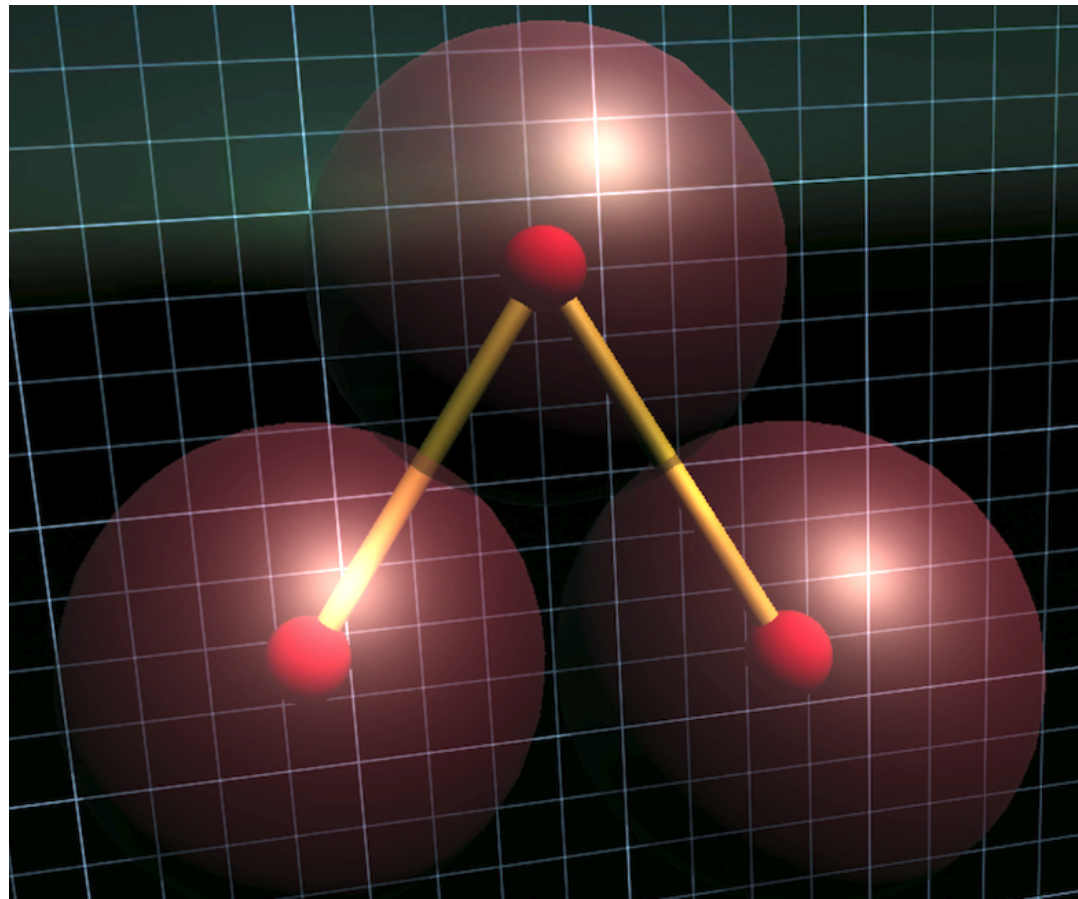
- Spheres & Halos : 0-dimensional simplices
- Cylinders : 1-dimensional simplices
- Triangles : 2-dimensional simplices

Spheres & Halos



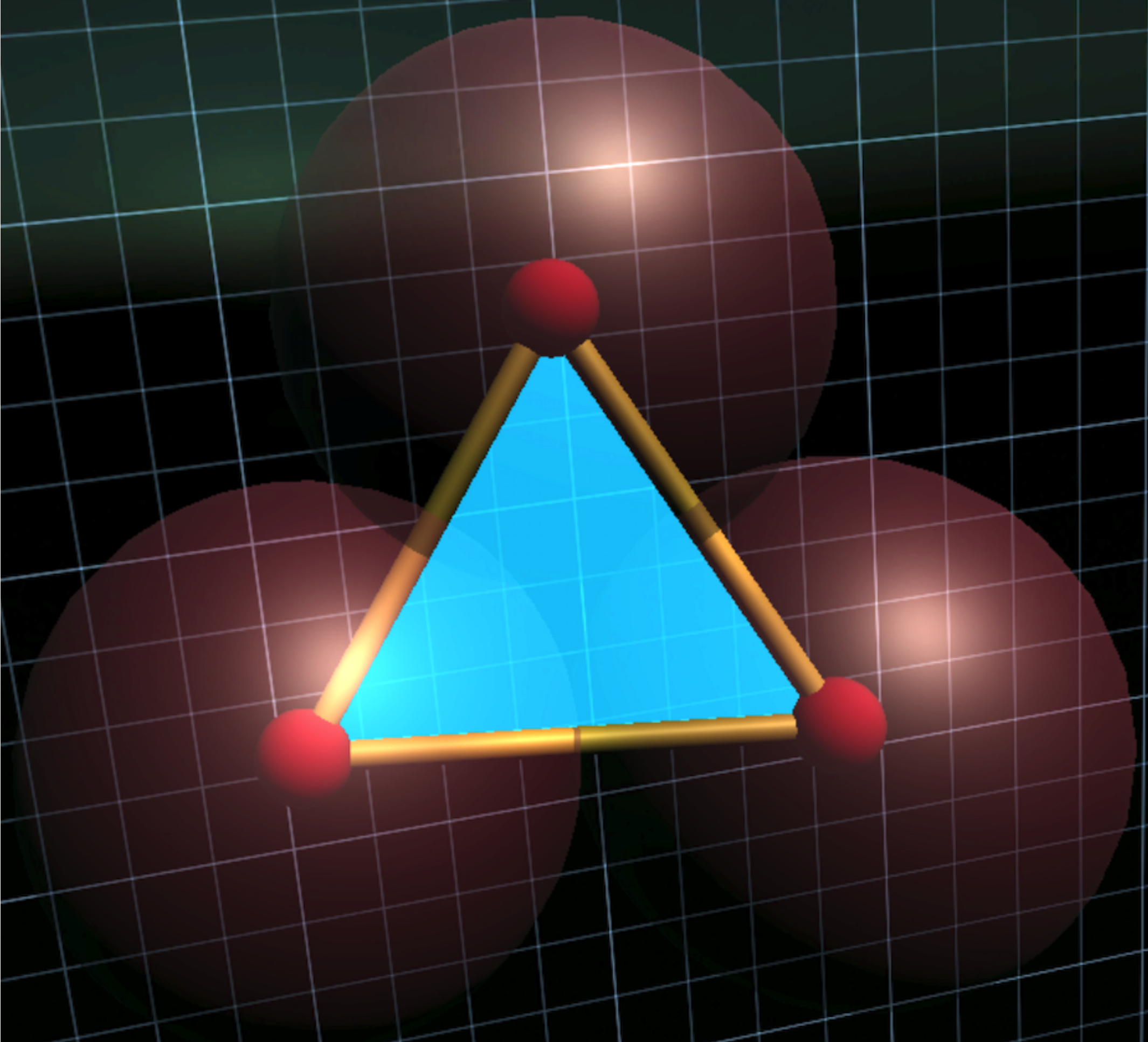
The spheres and halos are considered one entity here because they both originate at the same point. We are able to move the spheres freely in the virtual console along with having the ability to increase and decrease the radii of the halos.

Cylinders



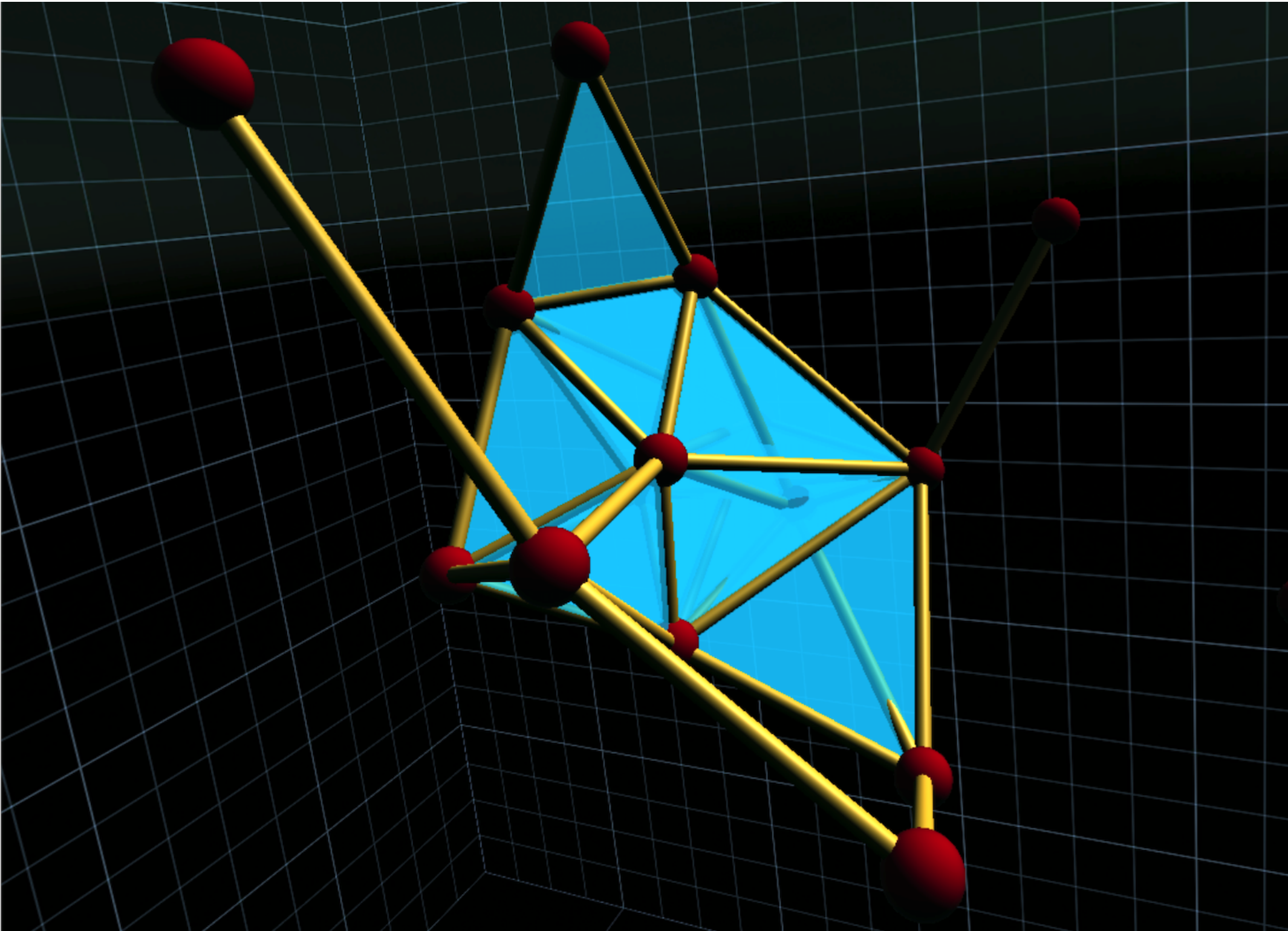
The cylinders appear when the halos' radii are increased large enough for them to intersect each other. If we choose to move the sphere and halos, the cylinder disappear if they are no longer close enough to intersect.

Triangles



Triangles appear when more than one sphere and halo intersect each other, thus forming a minimum of 3 cylinders. When this is done, we have successfully created a simplicial complex.

Additional Geometry



We are able to further explore more complex and interesting shapes when we add multiple spheres and increase their halos again to the point of intersection. The user also has the option of making the halos invisible so that they can examine shapes more clearly.

Math

By letting r be the radius chosen by the user, and P_1, P_2, P_3 , three points placed by the user in the scene. An edge is then drawn between P_1 and P_2 when $2r > \text{distance}(P_1, P_2)$. A triangle (*face*) is drawn among P_1, P_2 , and P_3 , when $2r > \text{distance}(P_1, P_2)$, $2r > \text{distance}(P_1, P_3)$, and $2r > \text{distance}(P_2, P_3)$.

Challenges

One of the biggest challenges we faced in the project was working with Unity. There were points where we had to restart Unity completely and we had to re-download the project again to make it run smoothly. There were points in the project where the hands or halos of the points would not show up. Also, cylinders appeared whenever two points came on the scene instead of the cylinders appearing whenever the halos intersect. Next, there was a part whenever the points are removed the cylinders disappeared in a weird way in the sense that when the second to last point was deleted, the last point still had a cylinder and that cylinder was dragged to the bottom of the scene. We have faced our challenges by debugging out code step by step to find the error and act accordingly to the project which was done by adding more source code classes to support the scene of the project and we made a instruction set for re-installing unity for the scene to work accordingly.

Future Goals

Our future goals are as follows:

- Make the scene more user-friendly. One of these goals would be to make the prefabs a better experience by adjusting the colors and size of the prefabs. Another one would be to add the visual of controls to the scene to show the controls
- Runtime performance when more points are added onto the scene.
- Change the construction from the Vietoris-Rips Construction to the Čech Construction.

References

Weisstein, Eric W. "*Simplicial Complex*." From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/SimplicialComplex.html>

Weisstein, Eric W. "*Simplex*." From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/Simplex.html>

Weisstein, Eric W. "*Polytope*." From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/Polytope.html>

Weisstein, Eric W. "*Topology*." From Encyclopedia of Mathematics. <https://www.encyclopediaofmath.org/index.php/Topology>