

Part 1

What I did- For this part I coded up two objects, one of them to represent the station that produces the signals and one to represent the signals sent out by the station. The station creates each signal and gives it a heading in a random direction, later changed to be aimed towards the blimp as well as locations in front of the blimp.

After a signal has traveled a certain distance it will signal that it is dead and be deleted.

how well did it work- For sending out the signals this worked fairly well; however, in the 2nd part it became an issue both in how effective it was and the framerate of the model.

pros and cons- The pros of this model is that it's able to aim a ray right at the blimp. The cons are that with how it is currently set up it is slow and very ineffective.

what would I improve- to improve this I would want to create a way for the signals to travel faster towards where I want to send them without having them be missed.

Part 2

What I did- For this part I created a new blimp object. This object will be passed the signals to see if it is in range of the blimp. If the blimp gets two signals from two station it will use the two points to triangulate its position from the station. Once it has a position it will adjust it's position on the corresponding axis so that it is heading towards it's destination. ie: if it's heading towards 200,-200 from -200,-200 it will only adjust its Z position while continuing to move along the axis at a constant speed.

how well did it work- When this worked it worked well; however there was a complication that arose from the rays that prevented it from working to the best of it's ability. This will be detailed more at the end of this report.

pros and cons- The pros are that this is able to move and determine a point using two points. The disadvantage with how it is currently designed is that it drastically moves to correct it's position rather than slowly moving.

what would I improve- There is a lot I would like to improve with this; however, that holds true to the entire task as I feel it didn't go well overall.

Problems I ran into many problems with this assignment. One being self inflicted as I attempted to have my blimp turn towards it's goal from it's believed location so that instead of jumping around it smoothly would move towards the goal. This however caused another problem to be hidden in that I was calculating the position using the wrong formula. This problem was realized Friday the 30th. After fixing the formula a problem was found with the rays that caused it to not work well. The rays in an attempt to reduce the amount of points missed and keep pulses constantly in the "air" moved at a rate to where the space between two points would be the

distance that the blimp could detect a point. The result of this was that there were many blank areas where the blimp didn't figure out it's location because it wasn't getting any points because they were moving far too slowly.

While writing this report I thought of ways to fix these problems and am going to attempt the fix and will write any changes below this, potentially redoing Part 2.

Part 2 take 2: My attempted solution to my problem here was to set the speed of the ray to the distance of the blimp such that after one frame the ray would be at the blimp's location reducing the number of rays on the screen overall, as well as allowing the blimp to keep updating its location every frame. This did not work as planned. Instead it failed to get a new triangulation after the first point despite the points being put almost right on top of it. As a result I have still submitted tracks using the above version despite its issues as it demonstrates the ability to triangulate even though it's not constant.