

PART 1

what you did

For this I created the framework for what I used throughout the task. I began by creating a point class which contained 3 doubles an X, Y and Z. I also created a class that managed most of my flight information. I had two functions in there to begin with. The first of which created an ArrayList and the 2nd of which added a single point to the ArrayList. This 2nd method could be used multiple times to add more and more points to the array.

how well it worked

This worked quite well. The points were also within the bounds and never ended being the same between two runs.

what its pros and cons are

pros: Simple and quick

cons: some hardcoded values, points can have extreme angles

what you might improve

Modify it so that the points chosen are at less extreme angles than those that can happen.

PART 2

what you did

For this part I added two additional functions into my birdFlight class (hence flight manager). The first of these is recursive and takes in two points and the ArrayList. It will then check the distance between these two points. A function was created in the point class that will calculate the distance between two points one being the point itself. If this distance is lower than a distance specified by a global variable in the flight manager then the two points are added to the ArrayList with the left point added first and then the right point. If the distance is outside the desired range then it would call a function to find a point that was between the two given points. From there the left point and the middle are used to call this function again. Once everything is done with them the middle point and right point is used. A helper function is also called before adding a point to prevent a point from being added two times in a row.

how well it worked

This worked well. It gave me points that were a nice distance apart. Additionally I was able to easily play around with how far apart the points were without having to change a huge amount of code.

what its pros and cons are

The pros of this are that it is able to divide a segment easily into a length of ones choosing without duplication.

The cons are that the segments are not constantly sized but rather are just shorter than a certain length. Depending on the length of the line it could possibly overload as well and cause the program to crash.

what you might improve

I'd want to make it so that each segment is the same length. Since at the moment there can be segments that are super short next to ones that are very fast.

PART 3

what you did

For this part I began using the point class for a different purpose. I created a function within the flight manager that would take in two points: the current point and the next point in the Array. With these it would then change the 2nd point from a point to a vector with its end on the Origin. To do this properly the first point (the actual end of the vector) was moved to the origin by subtracting its values from itself and thus this was done to the 2nd one as well. With the point on the origin rotating it became a matter of two calculations. The first used a projection of the point to rotate it such that the vector had 0 X. It was then rotated again using this new vector so that it had 0 Y. These angles were recorded and then used to rotate the bird such that it pointed the correct way since the rotation of the vector lined up with the rotation of the bird.

how well it worked

This worked out fairly well. There were a few issues where the rotation would be incorrect when the X value was negative to begin with, but that was a minor issue. Additionally rotating around the Y was an issue at first due to neglecting that the angle values being used to calculate the angle had changed. Beyond these issues it worked excellently and had no issues.

what its pros and cons are

The pros of this system is that it calculates the rotation of the vector with a high degree of accuracy.

The cons of this system is that it is unable to calculate the roll if there is a roll that needs to be calculated

what you might improve

I would probably improve the overall system so that the angle isn't calculated every frame but rather immediately upon creation

PART 4

what you did

For this part I created a function called vary point. This function is used during the division of the line segment. Each time the midpoint is calculated it is then sent to the vary point function to have its position slightly shifted.

how well it worked

This worked out decently most of the time. With how the function is set up I can easily change the variation range of the points without having to change many lines of code.

what its pros and cons are

The pros are that each point is randomly shifted within a range from its current position.

The cons of this are that the amount of shift is random so it's possible that a point shifts so little that it isn't noticeable or that two points are so shifted the change looks bad.

what you might improve

I would improve is such that the amount that is being shifted is a set pattern such that it doesn't look extreme and doesn't look non-existent.

PART 5

what you did

Well for most of the weekend working on this part I wanted to rip out my hair (good thing I already shave my head). My first attempt at this part was to curve the line between points so that the bird would fly through the point and then turn to the next point. This was my first mistake. The amount of math to do this was rather crazy. The 2nd mistake after figuring out the math was forgetting that the distance between two points was increasing each time I rotated a point to have the desired angle. The result of this and how I kept deciding to handle it resulted in headaches and infinite loops.

The way I ended up doing it was similar to the suggestion made in class of chopping off the corner if it was too big/small of an angle and then do the same for the resulting corners if they were too big, until the angles all met the required size. This had a few issues getting it to work, but it eventually came together.

how well it worked

It works okay-ish. The bird has a smooth turn and gradually turns at each corner. The problem is that due to how the function ended up working out the bird will slow down and seem to rotate in place if the angle was too small. This short coming is more easily noticed in tighter turns and when the distance between the points is either too small or too big. As it creates an obvious speed difference between the states.

what its pros and cons are

The pros are that it smoothly turns when it has to make a turn. It also is a bit faster than other methods

The cons are that as it is the bird will slow down as it is flying even though it's making a smooth turn.

what you might improve

I'd make this such that the bird didn't slow down when it made the turns while still keeping it smooth.