**READ ME**

This is my attempt at solving SPACERYDE Software intern design challenge.

**Some background into my design**

Chart, radar chart

Description automatically generated

The picture above represents my theory behind my design. The planes will spawn at the outer edges of the black circle, having a radius of 10 km to the center (where I am assuming the air traffic control is located.

The airport circle is created by taking the maxium length between the widths of the runways (100 m +500 meter spacing) and the length of the run way (500m ). The maxium of these two would form the radius, encapsulating the airport area.

In between the red circle and the green circle (the airport) is where the holding points (purple circle) can be, which is where the planes must wait if the runway is full. The way I have tackled this problem is that first I have generated holding zones (orange circle). The holding zones will then generate holding points (purple circle) , which is where the plane will perform the holding pattern. The points will have a 900-meter distance between them due to the fact that each holding point has a radius of 500 meters and planes need to have a distance of 100 meters each and if a plane wanted to travel between the holes this would mean that an extra 200 meters (better illustration below)

Diagram

Description automatically generated

This will mean that the distance between each holding zone will also be 900 meters to satisfy the condition above. Each holding zone will generate its holding points by taking the circumference of holding zone and dividing by the 900 meters to determine the number of points.

The position of the center points of each point will be generated by first establishing the first point in the zone to be 900 meters away from the last holding point from the last holding zone. For example:

Diagram

Description automatically generated

The rest of the points will then be generated 900 meters away from the previous point. This will be done repeatedly and the order of creating each point will represent the position of the queue that the plane will need to move to.

Once there is a plane has been created, the Air traffic control will check if there are any empty runways. If there are the plane will be allowed to fly directly to the runway.

However, if there are no empty runways, the Air traffic control will find holding point in the list of holding points that would have the highest priority and send that position of the point to the plane for the plane to travel to.

Once a plane has gone to the end of the runway, freeing up the runway, the plane in holding point 1 will be allowed to enter the runway. Planes in other holding points will then move to the next holding point, e.g. 2-> 1 , 3->2 , 4->3 ..etc. (where the number represent the numbered holding points in diagram above showing and the arrow represents the plane traveling from its previous point to the next point). This would mean there would be a spiral pattern of how the holding points are stored and how the planes move from one onto the next.

My rationale for this design is that once the all the planes got to their holding points there would be no collisions as they would methodically follow a spiral pattern. The extra 200 meter gap as mentioned before would allow planes to travel between points to get to their own assigned holding point (unfortunately I did not have time to program this) . Furthermore, the time taken to travelling between each holding point will be constant (900 meters between each one) and the time limit factor would be the plane travelling from holding point 1 to end of runway which would be 1400 meters( 900 meter + 500 meter (length of runway ) ).

**UML diagram**

Diagram

Description automatically generated

Please note that the attributes and methods inside the classes are not accurate as I did not have time to update it but the relations between the classes are accurate