Software intern design challenge

Thank you for your interest in SpaceRyde. The following challenge is designed to assess your technical competency in a particular domain. Your response will determine whether you are selected for a follow-up interview with our team. Please return a complete solution within **7 days** of receipt. Do not spend more than 10 hours on this challenge.

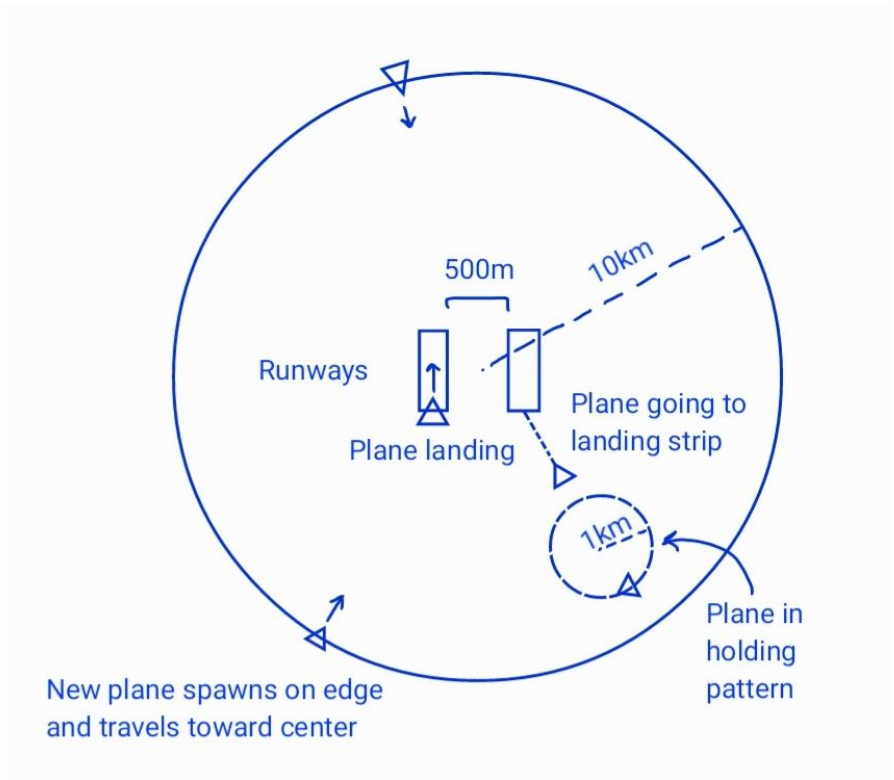
You have been selected as the software engineer for a new air traffic control (ATC) system. You may develop this system using either C++ or Python. Assume that the system is for a 2D world (i.e. don't account for the height of airplanes). The ATC system has the following features:

- Keeps track of all airplanes' positions (latitude, longitude) within the traffic control zone. The traffic control zone is a circle of radius 10km.
- Queues airplanes for landing based on time of arrival into the traffic control zone.
- Commands airplanes to go for landing (when it is their turn) to one of two runways.
- Otherwise commands airplanes to fly in a circular "holding pattern" which is a circle with a radius of 1km around a suitable point.
- The **ATC must never allow for planes to come within 100m of each other**, whether flying or landing.

An airplane has the following characteristics:

- Assume each plane is a point.
- When an airplane appears/'spawns' in the world, it does so at a random point on the edge of the traffic control zone, and it travels towards the center of the zone.
- Airplanes travel at a speed of 140 m/s while flying. Assume that they instantaneously stop and disappear when they reach the end of the landing strip.
- Airplanes automatically transmit their position (latitude, longitude) at a rate of 10 Hz to the ATC system while flying. They stop transmitting their position once landed.
- When given a command from ATC to land at a specific runway, an airplane travels to the base of the runway in a straight line. Once it reaches the base, it turns/changes angle instantaneously to follow the path of the runway till the runway's end. Its speed is the same (140 m/s) throughout.

Below is a diagram of the traffic control zone to aid in your visualization of the system.



The dimensions of the runways are 100m wide x 500m long. They are 500m apart from each other, equidistant from the center as shown in the diagram.

Account for the following considerations when designing the system:

1) An arbitrary number of airplanes should be able to be spawned (up to the maximum limit for the traffic control zone area, of course).
2) How would you allow for any of the above parameters to change without changing the code? (e.g. change number of runways, runway lengths or positions, or traffic control zone, etc without code changes).
3) What is the best way to transmit messages between the airplanes and the ATC?
4) What is the best way for the ATC to set holding patterns and paths for the planes such that they will not come too close to each other?

If you are familiar with ROS, you may consider using it to implement this system. However, it is not necessary and you can use any framework or architecture to implement it.

Submit your source code along with a brief, typed, well-formatted report explaining your implementation. Your solution will be judged based on its effectiveness, flexibility, modularity and the clarity of your explanations. **Keep in mind that we are not trying to assess your familiarity with any specific technology, but rather your understanding of engineering fundamentals and analysis.**