How can your implementation be optimized?

Ans:

For the smooth transition I have called the distance calculation and nearest object calculation in Update() method.We can call the same in FixedUpdate() if we want to reduce the calls.

Also we can call the calculations only when the player is moving.If the player is stationary we don't need to calculate since the results will be the same.

I was not familiar with Dependency Injection(DI),but during this test I learned about DI and I think we can use DI in the controller,Bot,Item scripts.That will make it more scalable and easy for testing. Didn't get time to implement it though.Exploring Zenject package for the same.

How much time did you spend on your implementation?

Ans:

2-3 Hours for the Basic Exercise.Learned UnitTesting and applied in the project.That took another 3-4 Hours.

What was most challenging for you?

Ans:

Deciding the logic for the nearest object calculation.First i tried with individual classes and that was not giving me the desired output.Then i tried the abstract classes.

Also I was confused at first whether to use Raycast or vectors for the distance calculation.I think raycasting will be more expensive than arithmetic calculation, so I decided to go forward with vectors.

What else would you add to this exercise?

Ans:

As a game developer my thoughts are going in that direction.Like we can have a game from this basic concept.

Items can be collectables which can be collected by the Player as rewards and Bots can be enemies.Bots will be activated when the player reaches near the Bot and they will start firing at the Player.Player will have a health bar which will get reduced when hit by the firing of bots. GameOver when the health bar reaches 0.