

Unity test for Plunge Interactive

This test requires you to develop a small **Tower Defense** in **Unity** in 3D using **C#**. The main objective of it is to evaluate the quality of the code you produce, so the aspects that we'll take into consideration are: code **correction**, **scalability**, **readability** and **encapsulation**, as well as the written English level.

As this test doesn't include any asset, so you'll have to **use primitives** to represent all the game's objects.

The scenario will be divided on a **grid that starts empty**, where the player will be able to place **different kind of turrets**. It must be **generated dynamically from code and support any size**.

The **enemies** will spawn from defined cells of the grid, which will be on a side of the map. The enemy objective is to **get the player's crystals**, which **will be on some cells of the grid**, **at the other side of the map**, while they resist turret damage. Enemies do **not** shoot back.

There will be **different types of turrets** that must behave differently. **Mandatory** types are:

- **Cannon turret**: Shoots balls that do instant damage on collision. Color **blue**.
- **Laser turret**: Shoots a laser beam that consumes enemy's life every frame. The beam will be represented with a line that goes from the turret to the enemy. Use Unity's **Line Renderer** component to draw it. Color **red**.

Enemies must be able to find a valid path to get from their spawn point to the crystals. They cannot pass through the player's turrets, so they **can walk through any empty cells** on the grid. To do the pathfinding you **must** use the **BFS** algorithm, which returns an **optimal path**. Any other algorithm or library will **not** be accepted. Enemies will have a **life bar**, so you can see how many hit points they have left.

You'll also have to implement camera **rotation** and **movement**. Cameras used for rendering 3D assets must be in **perspective mode**.

It's required that you write a **small document (between 2-4 pages)** in English explaining **what** is done in the code you're sending us, **how** is it done and **why** you did it like that. We would also appreciate a brief comment on what were the easier and harder parts of the test.