Scopely BCN

Client Engineer Technical Test

Tower Defense MVP

Unity version and project setup

Please use **Unity 2020.3.5** to perform this test. Inside the project we are providing, you will find the different assets needed. Remember to open the scene **MainScene** to see the battlefield with the placed Base and SpawnPoints.

Challenge Formulation

Your mission is to implement a basic tower defense game in Unity that will be used as a framework for the technical interview that we'll have later. During the interview, we will talk about the technical decisions and pros & cons of them, so focus on completeness and correctness with a codebase that is **clean** and properly **architected**. The architecture should accommodate growth, encourage high performance at runtime and make it easy for other team members to get involved with the code later.

Your game will be tested in the Editor, so do not worry about publishing it to any platform. It will need to support mouse and keyboard controls.

Also, you will need to provide **documentation** explaining the technical approach you followed during the development, and the reasons behind the decisions you made.

This is the list of requirements to implement:

1. In the center of the provided battlefield there is a **Base** for the player to defend. Create a system of spawnable “creeps”. Creeps spawn from the **SpawnPoints** already placed in the battlefield. When they spawn, they automatically move towards the Base **in a straight line (don’t implement or integrate any pathfinding, as NavMesh)**. Make the timing, number, and behavior of spawned creeps easy to tweak and tune.
2. When a number of creeps reach the player’s base, inform the player that has lost the game -- feel free to add a health bar to the creeps or to the base. When the game is lost, display the **LosePopup**.
3. Create a system of placeable **turrets**. The player can instantiate a turret anywhere on the battlefield.
4. Make the turrets shoot projectiles at creeps. Projectiles that hit creeps cause damage to them. Make the parameters for the amount of damage caused by a projectile and the amount of damage a creep can take easy to tune and tweak.
5. Implement a simple economy such as making each turret cost 5 coins to build and each creep giving you a coin when it dies.
6. Implement two types of turrets with different capabilities: a regular one and a freeze/slow-down effect on the other. (You have both assets provided).
7. Add different types of creeps with varying attributes such as speed and hit points. (You have two different creep assets provided).
8. Implement a system of waves. Once all the creeps of a particular wave are cleared, the next wave starts. If all the waves are cleared and the base is still alive, then display the **WinPopup**.

Deliverables

At the end of the technical test, you will return:

- The complete Unity project. In order for us to test, it should be as easy as opening it with Unity and clicking Play.
- Documentation. This has the form of a text explaining the technical design of the test, mentioning the decisions that you took and the reasons, possible refactors, etc.

Evaluation

This is what we will take into consideration when evaluating the test:

- The code and how it interacts with the scene is more important than what happens when you press play. **We don’t expect you to invest time in making it beautiful, and it won’t add any value to the test.**

- We will only evaluate the features mentioned in the requirements list. You don’t need to add any extra feature.
- Focus on scalability: the test should be implemented in a way that is easy to add new units or turrets without major refactors.
- The documentation is important: there you can justify the decisions that you made because of the lack of time. If there is something you would like to add but you don’t have the time to do so, do a proper explanation so we can evaluate it.

Test review interview

The **most important part of the evaluation** will be an interview in which you will walk us through the project. The ability to explain, reason & discuss how you have approached certain aspects of the game development is important, as well as providing a good implementation for them. This will be scheduled by a Talent representative after your technical test has been returned.

Assets

We are providing some assets for you to use:

- The **MainScene** containing:
    - Terrain
    - Base
    - SpawnPoints
- Turret-regular.
- Turret-freezing.
- Creep-small.
- Creep-big.
- WinPopup.
- LosePopup.