Senior Game Developer - Take Home Test

For this step of the interview process, you are given the choice between the following 2 options:

Option 1: Complete the test outlined below - Block Breaker in Unity.

Option 2: Share a personal project built in Unity.

Option 1: Block Breaker in Unity

Estimated duration: 10h

Submission: 1 week

Summary

Implement a simple block breaker style game written in C# using Unity 2022 LTS.

This project is intended as a measurement of your problem solving, organizational, and game development skills.

Assignment

Create a playable Unity project with the goal of showing how you would implement a simple block breaker style game using only Unity built-in components.

Specifications

The game takes place in a single screen.

The game is composed of:

- A player controlled turret
- Multiple breakable blocks
- Unbreakable walls that limit the play area

You control a static turret that can shoot projectiles.

Goal:

- Destroy all the breakable blocks in the room by shooting at them.

Turret:

- The turret is static. You can only control the direction in which it is shooting.
- Your mouse indicates the direction in which the turret is shooting.
- Press "Space" to shoot a projectile. Note that you can shoot as many projectile as you want.

Projectile:

- Projectiles bounce off walls and blocks like in a block breaker game
- Projectiles disappear after a set amount of time

Breakable Blocks:

- Blocks are static.
- They take a set amount of hits before being destroyed.

UI:

- There is a single console style log on the left side of the screen.
- It shows the following events:
  - A projectile was shot
  - A projectile has hit a block
  - A block was destroyed
- It should auto-scroll, displaying no more than the last N messages that can be visible onscreen.

Game Over:

- The game ends when all the blocks have been destroyed.
- You have a restart button that relaunches the level with a different block pattern.

Deliverable

- Use Unity and C# to develop the game.
- Focus on writing clean, maintainable, and efficient code.
- Please do not use 3rd party assets.
- Include a README file that provides clear explanations of your design choices and process.
- Explain any challenges you faced, and potential improvements.
- Please create a GitHub repository for your game submission.

Option 2: Personal project in Unity

Submission: 1 week

Deliverable

- Make sure the project is built in Unity and C#.
- Make sure the project is in a GitHub repository.
- Include a README file that provides clear explanations of your project, design choices and process.
- Make sure to highlight any particular area we should focus on when reviewing the project.
- Explain any challenges you faced, and potential improvements.

Have fun completing the test and good luck!