

Unity Intern Test

This is a technical test that requires a working solution in Unity3D, with C#.

Do note that while you're not expected to complete every user story listed below, each completed user story will contribute towards our final consideration. If possible, what you've done here should also reflect your skills in the listed preferences.

If you do want to copy-paste someone else's code, please be sure to include comments so we can see that you have understood what you have copy-pasted.

Screen Tapper Game

You're tasked with creating a game similar to OSU, where players will tap on the screen within a limited time for points. At the end of the task you'll be required to present whatever you've completed, and provide us the source code in a zip file, along with a `Readme.txt` file that explains how to run your application.

No art is required for this game, just use the minimal required, you can use text or raw sprites. However, you're not allowed to use Unity Assets for this test.

Technical Goals

1. There should be two scenes. "Splash_scene", and "Game_scene".
 1. The Splash_scene should be the first scene the player loads into. It will have the minimum needed items required, and it will load the Game_scene asynchronously in the background.
 2. The Game_scene will have all the items required to play the game.
2. Your game should follow a "Model-View-Update" or "Model-View-Controller" pattern, and there should be a clear differentiation between the model, view and update/controller layers.

User stories

1. As a player, I want to play this game on a desktop computer, whether it is a Windows or a Mac.
2. As a player, I want to see text on the splash screen that shows the loading progress.
3. As a player, after the loading is complete, I want to press the screen before I start the game.
4. As a player, when I play the game, the game will show one to three buttons on the screen for me to tap.
 1. The number of buttons is random.
 2. The location of it is random, but its should be confined within the screen. No button should overlap with the edges.
5. As a player, if I press all the buttons within 3 seconds, I will receive points, and the buttons will be regenerated.
6. As a player, after I've received points, I should see it on the top of the screen.
7. As a player, I want to see the timer on the screen to show how long I have to press the buttons.
8. As a player, I will play this game for 15 seconds, and then the game will end.
9. As a player, after the game ends, I want to see my high score.

10. As a player, after the game ends, I want to press a button to return to the splash screen.

Bonus points if you've completed everything:

1. Use Unirx and the ReactiveX paradigm.
2. Add tweening animations to the button on click, written using C#.
3. Move the buttons around the screen as it spawns, increasing with speed as the game progresses.