FlyPort

Work Sample Test

Information

This is a work sample test that is aimed for us to see how familiar you are with coding in C# as well as working in Unity Engine. Please read the entire form before you start working.

You will be assigned two tasks that we wish for you to complete. The tasks should be made in unity and every task should be in a separate scene. The code used for each task should be in their own folders.

No public variables are allowed to be used in any of the scripts. Public properties are however allowed, but requires a reason for being used. Include the reason as a summery for that property.

When you code, you should follow the coding conventions listed here: https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/naming-guidelines

When looking over your test, the examiner will look att:

- How you have written your code
- How you have named your variables, methods and classes as well as gameobjects and prefabs.
- How your structure in unity hierarchy and project folder is.
- How well you have followed the coding conventions mentioned above.

There is no time limit for these tasks and no real correct answer, we wish to see how you solve it. You are free to use the internet for guidance. When you have completed your tasks we want you to upload it to GitHub as a public repository.

The repository should contain a unity project with your solutions. We would also like it if you could add a PDF to the repository, containing how you solved the tasks and if you encountered any problems. Try to use less than 200 words for each task. There is no minimum amount of words needed, since it's optional, but here you can show the examinator how you think while working.

If you used the internet to complete a task, then please add a short explanation of how you searched for answers and why you deemed the source to be reliable, in the PDF mentioned above. Same as for the explanations for the tasks, try to keep it under 200 words.

Once you are done, send us a mail informing us about it and we shall examine it and return with our answer as soon as we can.

Good luck!

Task 1 - Singelton

Create a singleton. The singleton should have a method/function that prints out "Singleton is being called" in the console window.

The singleton should not be located on a gameobject in the scene and should not be referenced as a variable in any other scripts.

Create a script that calls for the Singleton method when a specific key is being pressed down. The keycode should be changeable in the inspector to the script. This script should be located on a gameobject in the scene.

Task 2 - Polymorphism

Create a trigger object that should swap between two colors when interacted with. The interaction should be a OnTrigger.

The script should have a bool in the inspector that changes to either a OnTriggerEnter or a OnTriggerExit.

The script should hold two colors that can be changed in the inspector and they should have the base colors of red and blue. They should always start with the color of the first assigned color, in this instance red.

Create a prefab of the trigger object and set out three of Them in the scene.

Change the prefabs so that they follow:

Prefab 1	OnTriggerEnter	Blue	Red
Prefab 2	OnTriggerEnter	Yellow	Green
Prefab 3	OnTriggerExit	Black	White

Once that works, create two new trigger objects that inherit from the original button script.

- One of the new trigger objects should hold five colors that can be changed in the inspector and that loops through them when interacted with.
- The other one should when interacted change into a random color between the two initial colors when interacted.

Create a prefab for each of the new trigger objects and add them to the scene.