

Three spheres problem

Tasks/Questions

answered by Shabbir Rudro

1. Implement the scenario.

👤 The scenario is implemented at <https://github.com/shabbirrudro/threeSpheres>

2. How would you implement this scenario in a modular and generic way with seamless transitions between the scenes? (Answer in text). Seamless means that the objects don't suddenly disappear. The solution should be reusable for different scenarios/apps.

👤 Seamless transition of the scenes can be done by instantiating the current scene as a new scene (like a prefab). Before that, the state of the current scene has to be preserved. A C# script can be written like GameStateManager. With a static 'Instance' running at Awake(), the state can be preserved like this.

```
using UnityEngine;

public class GameStateManager : MonoBehaviour
{
    public static GameStateManager Instance;
    public Vector3 selectedSpherePosition;
    public Quaternion cameraRotation;
    void Awake()
    {
        if (Instance == null)
        {
            Instance = this;
            DontDestroyOnLoad(gameObject);
        }
        else
        {
            Destroy(gameObject);
        }
    }
}
```

```

    }
}
GameManager.Instance.selectedSpherePosition = transform.position;
GameManager.Instance.cameraRotation = mainCamera.transform.rotation;

```

At Start() of the new scene, everything can be instantiated like this:

```

void Start()
{
    GameObject selectedSphere = Instantiate(spherePrefab,
GameManager.Instance.selectedSpherePosition, Quaternion.identity);

    Camera.main.transform.rotation =
GameManager.Instance.cameraRotation;
}

```

Just like the camera position and the sphere location, anything else's transform can be preserved and then instantiated in the new scene, which makes the transition very smooth.

3. *What are the challenges?*

- 🧑‍🎓 The easier option would have been to create the fade out and fade in effects on the 2nd scene. In that way, there would be no need for a smooth transition. But I understand there may be usecases where loading a new scene would be required, and that proved to be a bit challenging.
- 🧑‍🎓 Reducing the GPU load can be an issue. Although this task had very few GameObjects, but in other cases with hundreds of objects, the running scripts need to be optimized to make sure unnecessary items are not running.
- 🧑‍🎓 Possible improvements include using prefabs for every objects and instantiating them in runtime when they are needed. With continuous improvements over time, the codes, the hierarchy in Unity UI, and the visuals of the objects can be improved.

4. What challenges do you see in additionally supporting an augmented reality modus for iOS/ Android and HoloLens? The user should be able to toggle between AR and Non-AR.

- 👤 The main challenge, in my opinion, would be optimizing the whole scenario to fit the different input method in AR/VR instances. Based on the gaze of the camera/eyes, a pointer has to be created to select the desired object, either by a tap on the screen or the dedicated button. There should also be an option to select the objects with just the hovering of the pointer and selection after a small delay. Providing all these options and also having the ability to run the game in a flat display could be difficult. The stability of the object tracking, and environmental mapping could also be challenging.
- 👤 The GPU load could be a bigger issue, especially in mobile devices with lower computing power and high definition VR headsets (e.g. Pimax 8k, it renders 4k footage for each eye. Without proper optimization, the users wearing VR headsets may experience motion sickness. Any other cases, lower framerate may produce a less than perfect experience. In mobile devices, the battery life management could also be an issue.
- 👤 The difference in platforms between iOS, Android, and HoloLens may be difficult with their different APIs and possible requirements of abstraction layers. The UI/UX design must also be different for different platform.
- 👤 Seamless transition between AR and non-AR may also be difficult.
- 👤 Learning curve of new technology, ensuring the users are willing to learn and use the new features safely and without distraction must be ensured.