Solution Answers and Documentation

# Answers & Documentation

## Question1.

How would you implement this scenario in a modular and generic way with seamless transitions between the scenes?

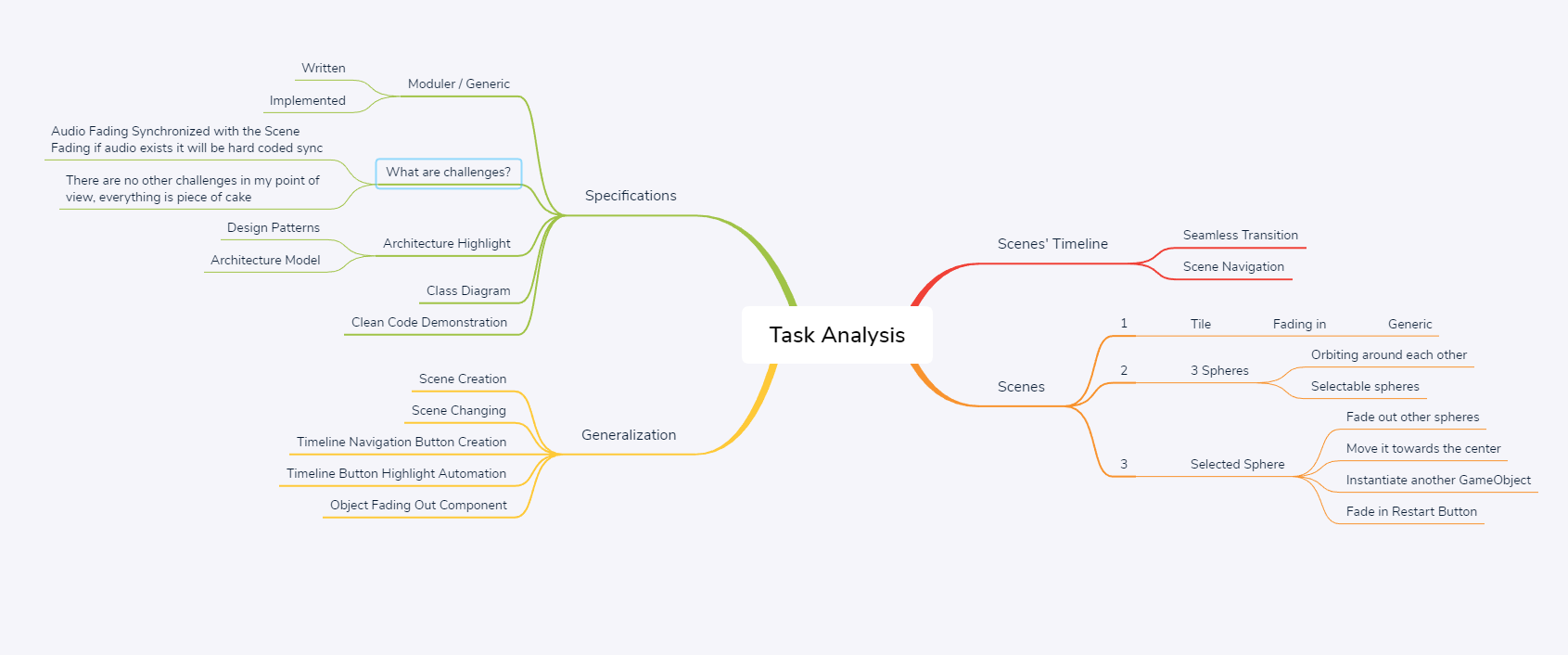
It is very simple to handle the transition between scenes with different fading out/in transitions.

I would implement it through using the animation system which is built in unity and will make a blank white/black image which will be a child of canvas that has highest layer priority on the screen that has an animator which at any time we are going to change the scene we will play the fade-out animation and wait until it finishes (fully black or white) then change the scene and at the On Loaded Scene event will play the fade-in animation of the same image. Using the same technique we can fade out specific object(s) which uses animation to fade out, but instead of changing the opacity of an image we can change the color or even the alpha of the color of the material if it was set to fade render mode.

## Question2.

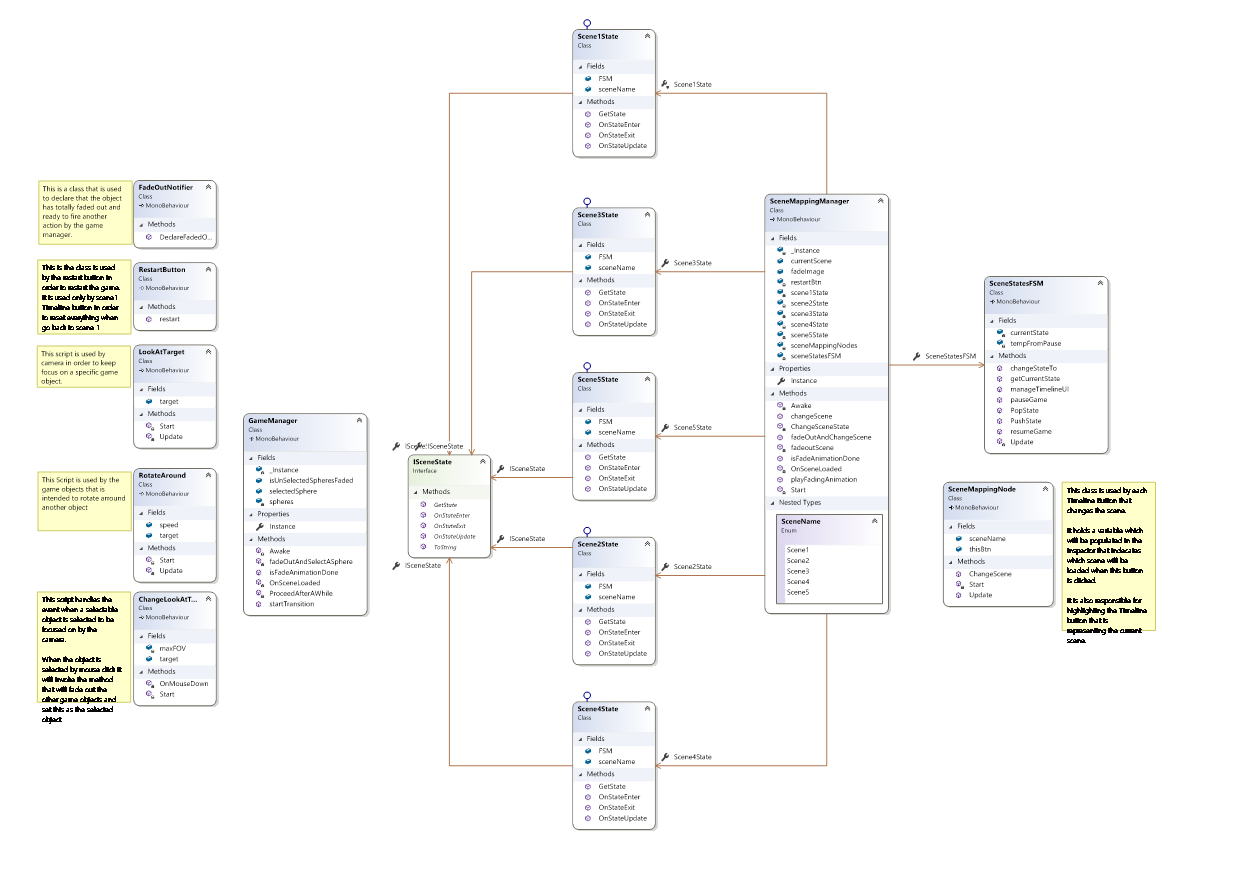
• Block out the basic architecture of your solution in code.

Task Analysis:

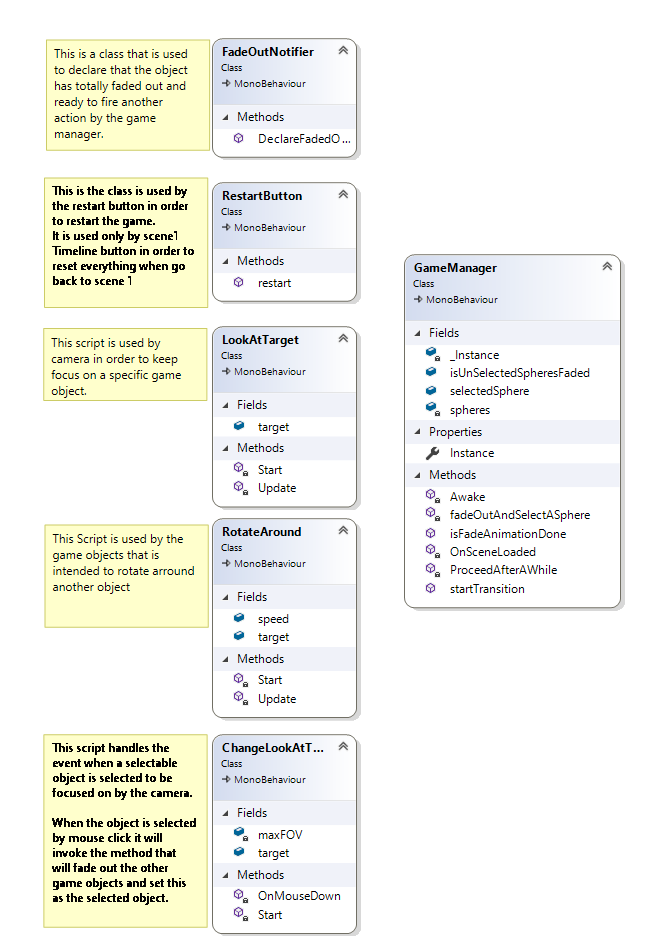


### Solution Basic Architecture

An Overview:



Game Management and independent scripts:



Game Management and independent scripts:

* GameManager: Doing general gameplay processes and holds the shared data members that needs to be shared.
* FadeOutNotifier: Notifies the game manager that specific game objects need to be faded out is completely faded.
* RestartButton: A script that is used by the Restart button to restart the game and the Timeline button of scene1 to reset the logic.
* LookAtTarget: Used by the camera to keep focus on a specific gameObject.
* RotateAround: Used by an object that is needed to rotate around another object.
* ChangeLookAtTarget: Reflects the mouse clicks on an object and perform a specific action including changing the Look at target to be the pressed object.

Scene Management & Scene States Finite State Machine

