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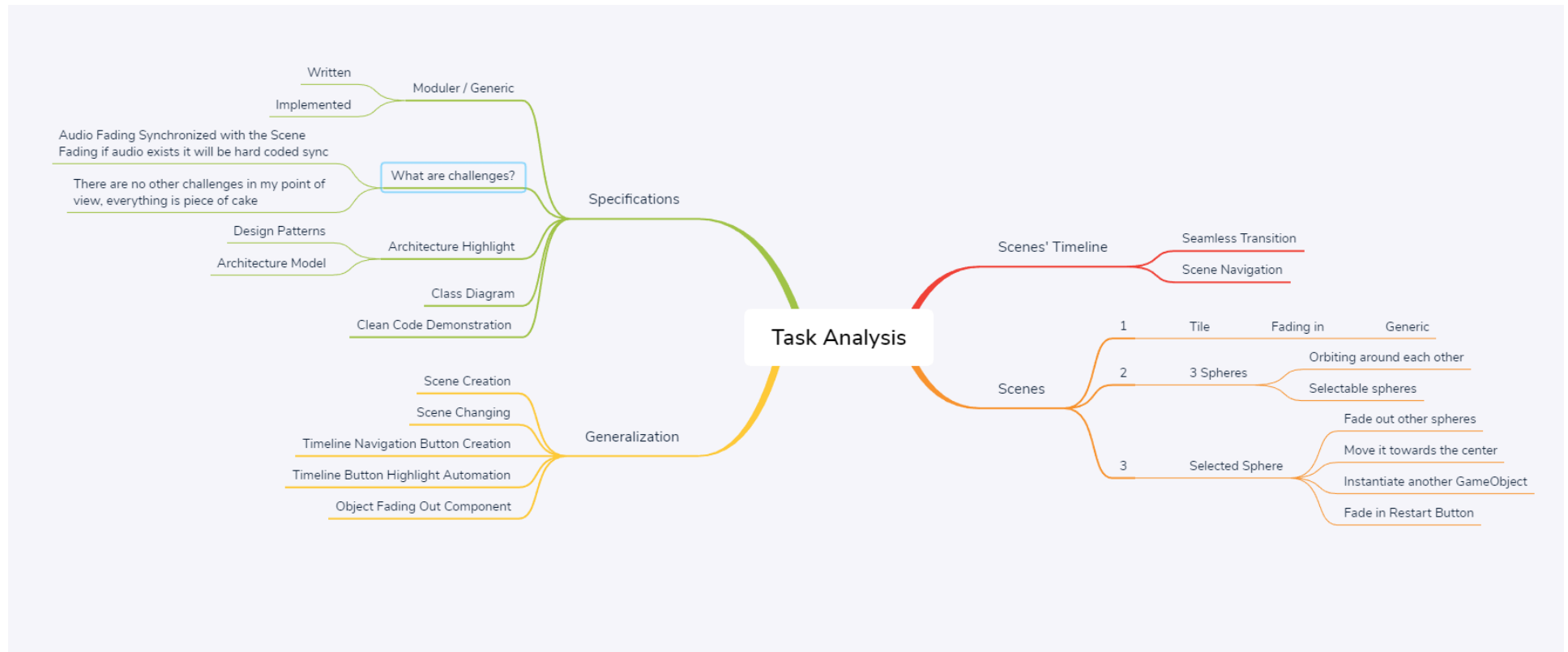
It is very simple to handle the transition between scenes with different fading out/in transitions.

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:

Written

Moduler / Generic

Implemented

Audio Fading Synchronized with the Scene
Fading if audio exists it will be hard coded sync

What are challenges?

There are no other challenges in my point of
view, everything is piece of cake

Specifications

Design Patterns

Architecture Highlight

Architecture Model

Class Diagram

Clean Code Demonstration

Scenes' Timeline

Seamless Transition

Scene Navigation

Task Analysis

1 | Tile | Fading in | Generic

Orbiting around each other

Scene Creation

2 | 3 Spheres

Scenes | Selectable spheres

Scene Changing

Generalization

Timeline Navigation Button Creation

Fade out other spheres

Move it towards the center

Timeline Button Highlight Automation

3 | Selected Sphere | Instantiate another GameObject

Object Fading Out Component

Fade in Restart Button

Solution Basic Architecture

An Overview:
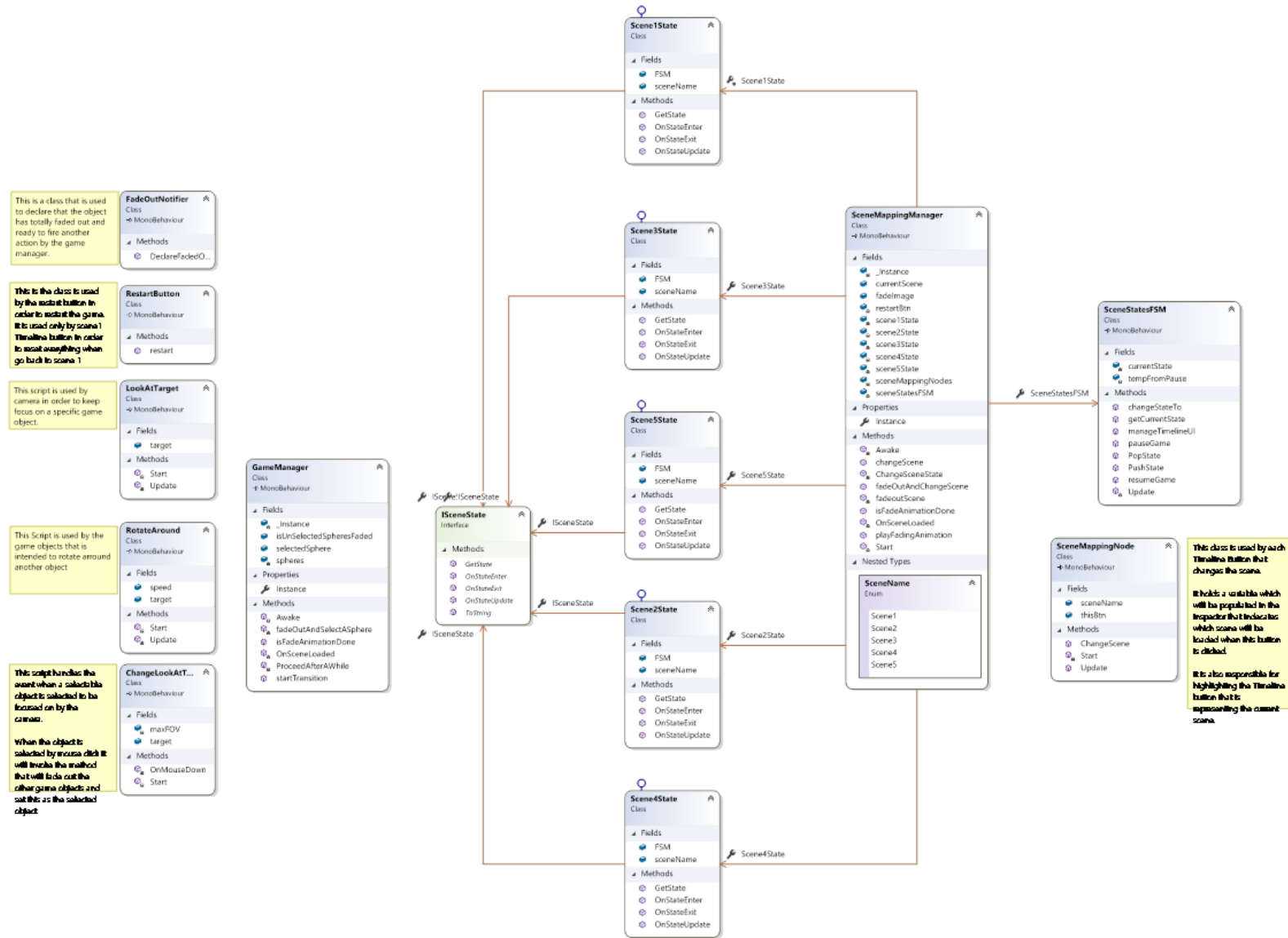
Game Management and Decoupled Scripts:

This is a class that is used to declare that the object has totally faded out and ready to fire another action by the game manager.

FadeOutNotifier
Class
→ MonoBehaviour

▲ Methods
- ◉ DeclareFadedO...

This is the class is used by the restart button in order to restart the game.
It is used only by scene1 Timeline button in order to reset everything when go back to scene 1

RestartButton
Class
→ MonoBehaviour

▲ Methods
- ◉ restart

This script is used by camera in order to keep focus on a specific game object.

LookAtTarget
Class
→ MonoBehaviour

▲ Fields
- ◆ target

▲ Methods
- ◉ Start
- ◉ Update

This Script is used by the game objects that is intended to rotate arround another object

RotateAround
Class
→ MonoBehaviour

▲ Fields
- ◆ speed
- ◆ target

▲ Methods
- ◉ Start
- ◉ Update

This script handles the event when a selectable object is selected to be focused on by the camera.

When the object is selected by mouse click it will invoke the method that will fade out the other game objects and set this as the selected object.

ChangeLookAtT...
Class
→ MonoBehaviour

▲ Fields
- ◆ maxFOV
- ◆ target

▲ Methods
- ◉ OnMouseDown
- ◉ Start

GameManager
Class
→ MonoBehaviour

▲ Fields
- ◆ _Instance
- ◆ isUnSelectedSpheresFaded
- ◆ selectedSphere
- ◆ spheres

▲ Properties
- 🔧 Instance

▲ Methods
- ◉ Awake
- ◉ fadeOutAndSelectASphere
- ◉ isFadeAnimationDone
- ◉ OnSceneLoaded
- ◉ ProceedAfterAWhile
- ◉ startTransition

Scene Management & Scene States Finite State Machine

SceneMappingManager
Class
→ MonoBehaviour

▲ Fields
- 🔒 _Instance
- ● currentScene
- ● fadeImage
- 🔒 restartBtn
- 🔒 scene1State
- 🔒 scene2State
- 🔒 scene3State
- 🔒 scene4State
- 🔒 scene5State
- 🔒 sceneMappingNodes
- 🔒 sceneStatesFSM

▲ Properties
- 🔧 Instance

▲ Methods
- 🔒 Awake
- ● changeScene
- 🔒 ChangeSceneState
- ● fadeOutAndChangeScene
- 🔒 fadeoutScene
- ● isFadeAnimationDone
- 🔒 OnSceneLoaded
- ● playFadingAnimation
- 🔒 Start

▲ Nested Types

SceneName
Enum

Scene1
Scene2
Scene3
Scene4
Scene5

SceneStatesFSM
Class
→ MonoBehaviour

▲ Fields
- 🔒 currentState
- 🔒 tempFromPause

▲ Methods
- ● changeStateTo
- ● getCurrentState
- ● manageTimelineUI
- ● pauseGame
- ● PopState
- ● PushState
- ● resumeGame
- 🔒 Update

SceneMappingNode
Class
→ MonoBehaviour

▲ Fields
- ● sceneName
- ● thisBtn

▲ Methods
- ● ChangeScene
- 🔒 Start
- ● Update

This class is used by each Timeline Button that changes the scene.

It holds a variable which will be populated in the inspector that indecates which scene will be loaded when this button is clicked.

It is also responsible for highlighting the Timeline button that is representing the current scene.

Scene1State
Class

▲ Fields
- ● FSM
- ● sceneName

▲ Methods
- ● GetState
- ● OnStateEnter
- ● OnStateExit
- ● OnStateUpdate

ISceneState
Interface

▲ Methods
- ● GetState
- ● OnStateEnter
- ● OnStateExit
- ● OnStateUpdate
- ● ToString

Scene3State
Class

Scene5State
Class

Scene2State
Class

Scene4State
Class

🔧 ISceneState
🔧 ISceneState
🔧 ISceneState
🔧 ISceneState
🔧 ISceneState
🔧 ISceneState

🔧 Scene1State
🔧 Scene3State
🔧 Scene5State
🔧 Scene2State
🔧 Scene4State

🔧 SceneStatesFSM