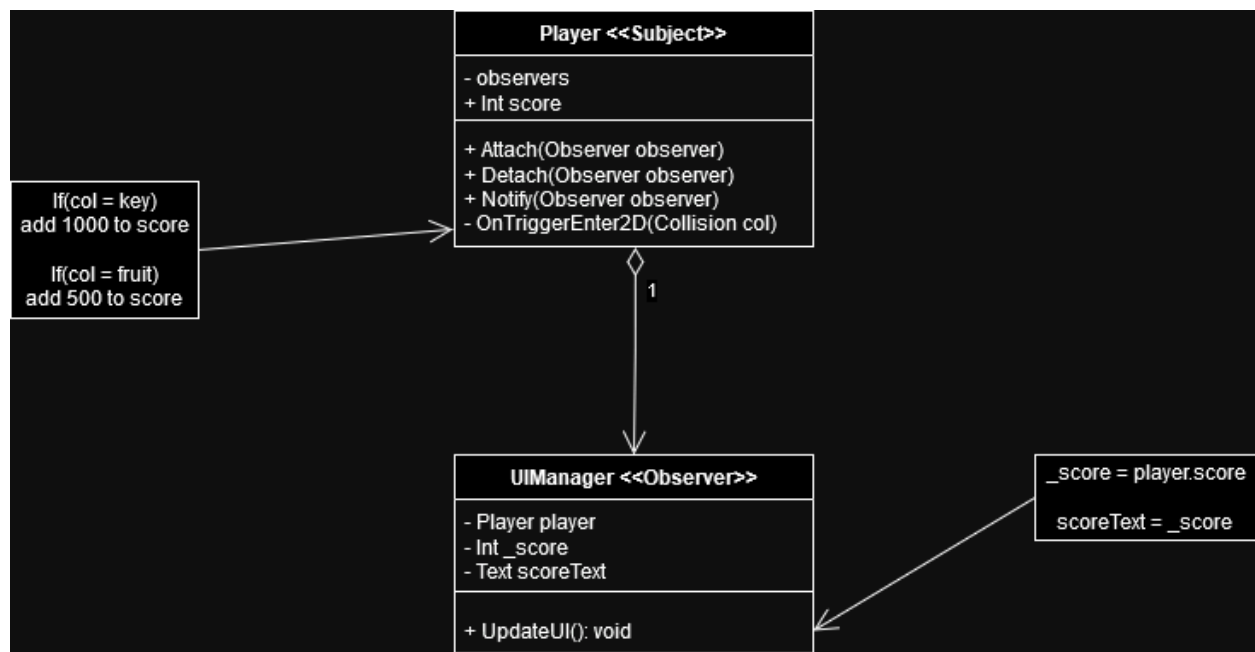


Review Challenge Mark Proof

Saad Khan - 100829159

Jeffry Lai - 100698915

Design Pattern Diagram and Explanation



For the design pattern, we decided to use the observer pattern to drive the UI of the game. The player (subject) will collect various items like fruits and the key at the end of the level (See `OnTriggerEnter2D` function in diagram), an integer keeping track of the score will update every time these items are collected within the Player script. The UIManager (observer) is listening to the Player script to see if any changes are made to the score, if the score has changed, the UIManager will update the UI (See `UpdateUI` function in diagram) and change the score accordingly.

DLL Explanation

For the DLL, we created a class that will handle the objectives of the game. In our case, it was to handle the score distribution for keys and fruits, which both have the same logic with the differences being the amount of score given. Once imported into Unity, the DLL provides the class to be attached to all objectives and rewards the player with points with the amount of points being modifiable in the inspector.

Reasoning behind submitting past the original timeline

We believe the reason we weren't able to submit before the original deadline was that we didn't have a solid plan. Our planning phase lacked proper communication and understanding, which led to our time being mostly allocated to figuring out a plan on how we should tackle the question.

Saad:

I used the rest of my time building the functionality of the game first before creating the design pattern diagram or implementing it into the game properly. Because of this, I used the remaining time I had to make the game functional rather than initially starting with what design pattern should be implemented. For the future, I will first create a quick plan, collect the design patterns needed, create their diagrams, implement them into the project, and then worry about game functionality later.

Jeffry:

The idea of what the DLL has changed multiple times due to errors creating or writing the DLL. Initially we planned on making a DLL version for the PlayerMovement script, however the PlayerMovement relied on too many other scripts to be compiled as a DLL (Observer.cs, Subject.cs, and UIManager.cs). We later changed to try to make the Observer.cs and Subject.cs into DLLs as they made more sense to be ones but we ran into issues causing the modified PlayerMovement.cs and UIManager.cs to not compile at all. We ended up making a DLL for the objectives in the game (keys and fruits) which is what is presented in the DLL Explanation above and in our project/build.

Contributions

Saad Khan - created a unity project with the environment, coded player movement functionality, and included observer design pattern, and design pattern diagram and reasoning.

Jeffry Lai - created a DLL, modified player movement to handle working with the DLL, and built the project and posted the release.