**Project: Short Puzzle Game/ One Short Puzzle**

**1. UML**

UML diagram is created on StarUml Software with the help of classes. I divided UML diagrams in two parts.

In 1st uml there is a script of BlockManager in the association of classes which is used for the game blocks. BlockManager taking the data from the following scripts/classes: Block, ScenesManager and ScoreCalculator. You can see the snapshot by clicking here: [UML-1](https://github.com/HafizTalha/ShotPuzzleGame/blob/master/Documentation/UML%20Diagrams/UML-1.JPG)

In 2nd uml there is a main script of PlayerController which managing the controlling part of the game in the association of scripts/class PlayerVFX and InputData. You can see the snapshot by clicking here: [UML-2](https://github.com/HafizTalha/ShotPuzzleGame/blob/master/Documentation/UML%20Diagrams/UML-2.JPG)

**2. DDD**

N/A

**3. Metrics**

I used sonar cube for finding the code metrics. First, I established a local server for sonarcube then configure my github project with it. The code is passed as per the sonar cube results. The snapshot for the results of the sonar cube where it is showing the code is passed submitted. It can be checked also in the below link. The code has passed the quality gate of sonar cube.

[Sonarcube Results](https://github.com/HafizTalha/ShotPuzzleGame/tree/master/Documentation/Metrics-Sonarqube)

**4. Clean code**

I have written a clean code. Every method has headers which are commented which tells the functionality of every function. Code is also tested from sonar cube and the quality gate is passed from sonar cube as well. The clean code scripts can be found under ShortPuzzleGame\Assets\Scripts. There is total eight available scripts which are also managed in folder structure code and resources.

**5. Build**

I used the build management system from unity itself because it has its own build management system that’s why no other build management system was required.

**6. Unit Test**

I Integrated a Unit test about score calculation which checks either score is updated or not and performed it in unity test runner. Snapshots can be seen in following link.

[Unit Test Snaps](https://github.com/HafizTalha/ShotPuzzleGame/tree/master/Documentation/Unit%20Test)

**7. Continuous Delivery**

Jenkins run on a local server and all of the screenshots are already in the documentation folder. It checks after periodically (time can be set according to our requirements) the repository on GitHub if there is any changing or not? If there is any changing it will build automatically which is done by command line arguments. The command line argument used here is:

-nographics -batchmode -quit -executeMethod BuildScript.PerformBuild -logFile "$WORKSPACE/unity3d\_editor.log"

**8. IDE**

The IDE used in the project is Visual studio, Mono can also be used with unity but I used visual studio in my project. The favorite shortcut keys are following: Find all refrences Shift+F12 Go to implementation Ctrl+F12 For selecting all Ctrl+a For format Ctrl+k For comment Ctrl+k+c For uncomment Ctrl+k+u

**9. DSL**

N/A

**Functional Programming**

Used Functional Programming in PlayerController Script. By using system.Linq (Library) we can find the game object of the main camera, then we get the camera components. The PlayerController scripts for the subject game can be found at

[PlayerController Scripts](https://github.com/HafizTalha/ShotPuzzleGame/tree/master/Assets/Scripts/Player)