CS 319

Object- Oriented Software Engineering

Final Report

IQ Puzzler Pro

Unknown - Group 1-I

Derviş Mehmed Barutçu

Elif Beril Şaylı

Giray Baha Kezer

Zeynep Ayça Çam

Zeynep Hande Arpakuş

Table of Contents

1. Implementation Process

Implementation have been continuing since just after the first implementation reports finished.Thorough project ,reports,diagrams and game logic and features was decided by all group members.In first iteration,all team members focused on game logic which is about mainly, blackboard and pieces.After demo, we realize that we need new features and our some features are needed to change for extensibility.Without reorganization,to move forward will be very hard. Then all connected to repository, which Zeynep created with the name IQPuzzlerPro at beginning of semester. Its advantage is directly connection to project source code and work neatly but unfortunately we did not take its favor so much. It means we only update Github at the end of each iteration and share our files by other platforms thereby Github was on the back seat. We understood its logic on pull push and if necessary to merge it to other pieces at the end but we can't use it efficiently for that project. When we are not in campus or nearby to other group members we used Skype to communicate and work cooperatively on report submissions.

We took the point of course which was related code implementation related design and analysis reports, these are important points of conducting project, these should be in connection and taken similar effort, so we focused every requirements. On the code parts we focused 2D and 3D because our board game is a builded by mind game Our game had some levels to be fancy for different groups beginner to experts, we tried to create some levels from every stages.

Generally, on the conducting project we focused our fundamental logic and did our promised works except for them we added some other additional features as challenges, leaderboard, 3D attempts for pyramid trials and data management for saving players' last checkpoints. Rest of our final report will focus on mostly some installation requirements and these points were mentioned above.

1.1 System Requirements

IQ Puzzler Pro game will require Java Virtual Machine and Java Setup Development Kit, any computer with these can easily run the project.

1.2 User's Install Guide

Our game project can be found on that link
https://github.com/ZeynepAycaCam/IQPuzzlerPro

After project cloned or bash from that link as a zip file, any Java Compiler can run it easily.

Also, gson library must be downloaded and included to the project from project properties. It should be downloaded as jar file. The instructions for this step are as follows:

-gson jar file can be downloaded from: **https://github.com/google/gson**

Step1:

Figure 1: Gson downloads

Step2:

Figure 2: GSON jar file downloads

2.Changes & Improvements

2.1 Design Changes & Improvements

Our first implementation report and code samples were not neat, that is why we changed our steps and configured codes to MVC model and report decomposition into more efficient style, also we had to change our packages about changed decomposition on second iteration.

In our design report we planned to have MenuViewer class, which is the façade class of our User Interface part, also planned to have DataManager class which manages communication with other classes and database.

We wrote some changes on MVC design pattern for design report, we evaluated that changes and changed our Menu and Panel classes into that direction in order to have better project design pattern.

In the first iteration, we implement shape drag, rotate and flip functionalities; however, after we do some changes in order to implement other boards and extra features, we realized that we can't continue to use same implementation to accomplish these functionalities. Therefore, we have to change these functions implementations to better fit all game modes in the second iteration.

For intersection of pieces and board while playing the game, we used the center logic, which comparing every rectangle centers and pieces' circles' centers, whichever is close to what rectangle system sets them together. Then the other pieces and choices could be translated in through logic.

In the first iteration, in our design report, we mentioned that we will use MS database to hold our values. However, in the second iteration, we thought that using file system for data management is better idea for our project. We also get help from JSON object to hold our data in the file, so putting data to file and getting data from it will become much more easier. To use JSON, we get help from GSON Google library, since it offers us ease of coding and gain time for the other parts without not much effort on data storing.

Deployment diagram according to our design changes can be seen above:

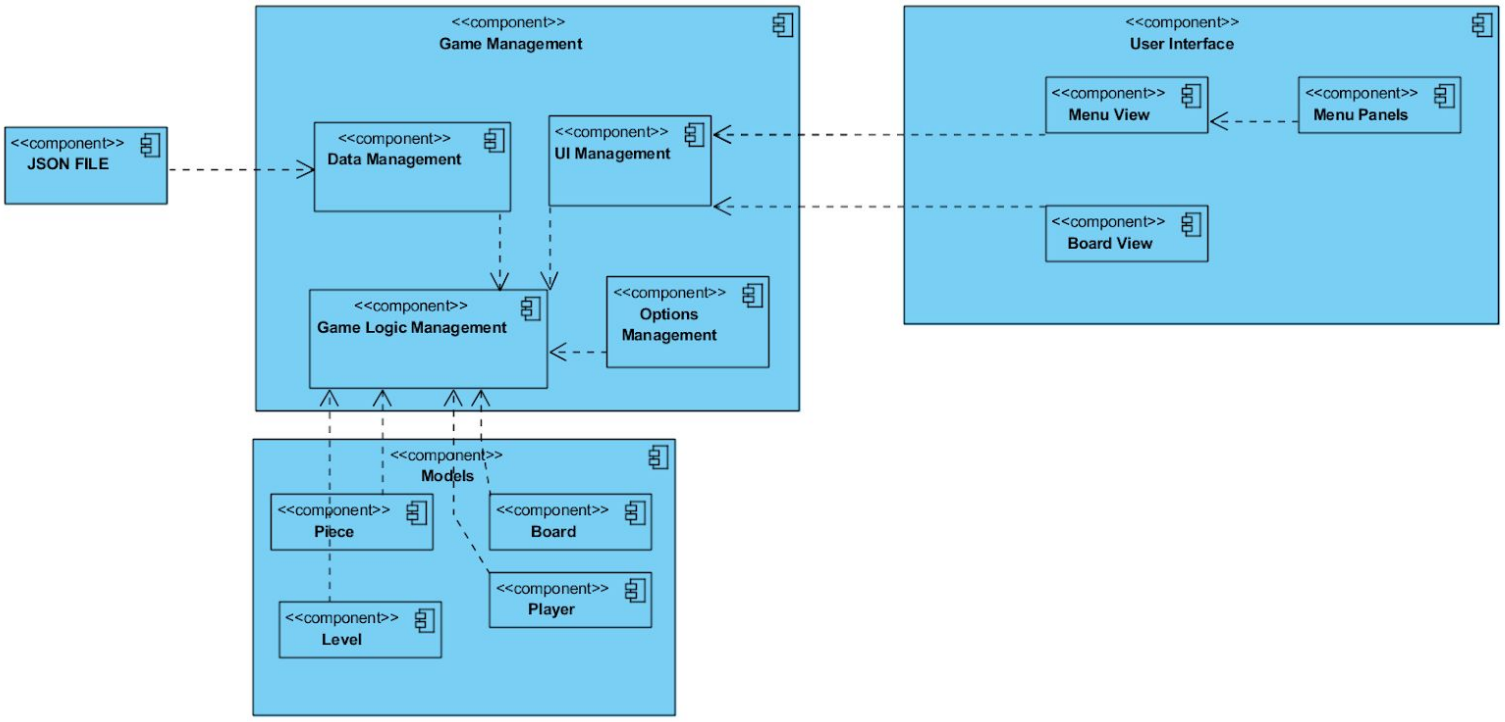

Figure 3: Deployment diagram

2.1.1 BlackBoard

As we promised for project features black board can be used for any solution. Any combination of all twelve pieces can be put on that board rotated or normally they are taken on their centers to rectangles of our board. We used GridPane to create black boards and a pattern to create related rectangles on it. 5 columns and 11 rows are created on that board as its original game. Its features can be seen below from the picture.

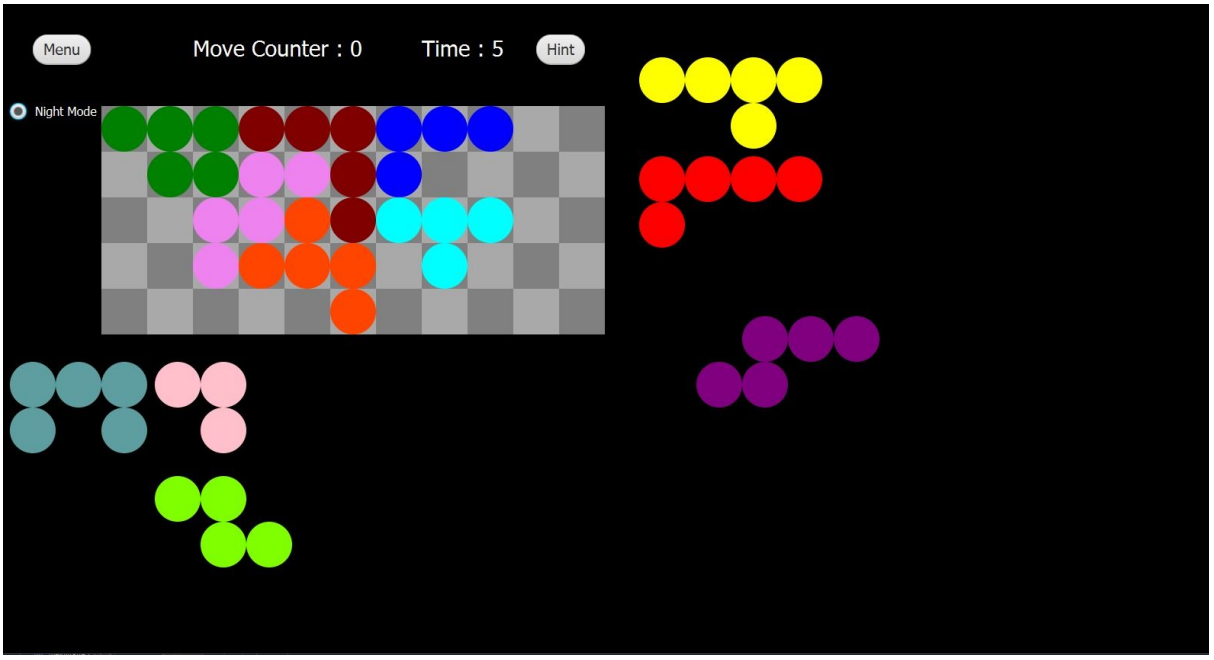

Figure 4: Black board Night Shift Mode

2.1.2 Grayboard

As we promised for project features grayboard has 55 rectangles on its characteristic pattern as its original. Its pattern rectangles as 4 - 5 - 7 - 7 - 9 - 7 - 7 - 5 - 4 in order. Twelve same pieces can be used on board for any combination for rotated or normal. This board was existed so as to be used for higher difficulty levels, therefore game can be friendly for different type of groups, children to adults. Its basic features and pieces can be seen below.

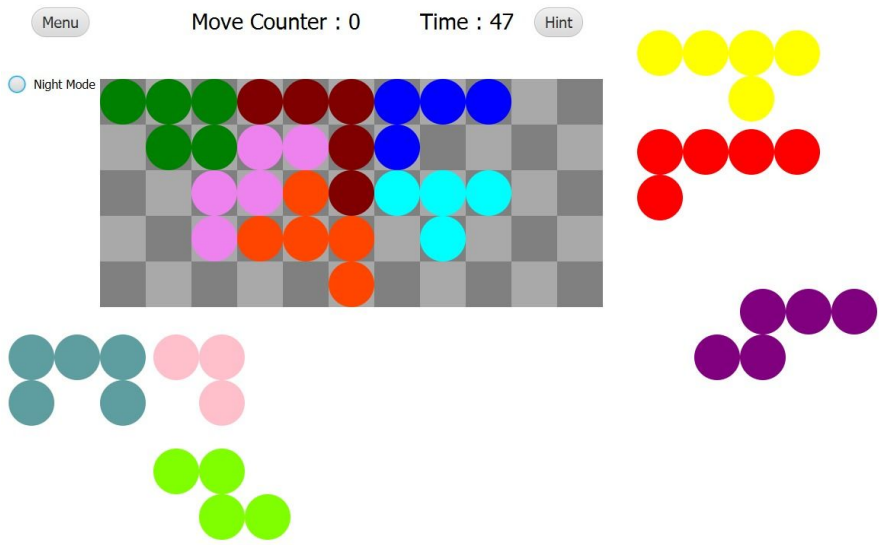

Figure 5: Black board

2.1.3 3D Improvements

Our game 3D improvements are mostly in progress, original game only offers pyramid shape for players. We created all pieces and did some logical calculations on intersection and combination but for game logic,to make pyramid and check the game is finished or not,we need more time.Also, movable camera is hard to implement with Javafx.

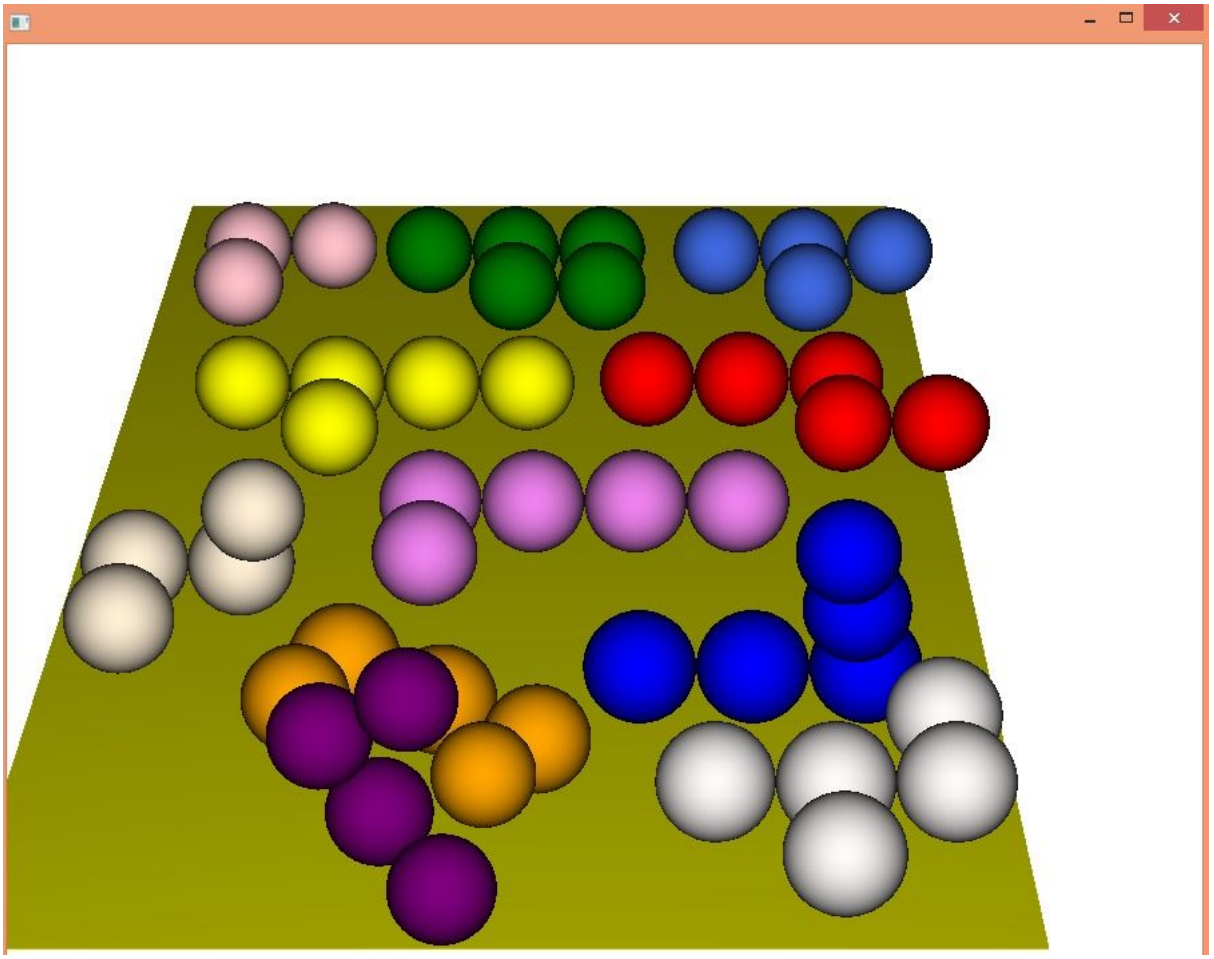

Figure 6: 3D extension screenshot

3.Extra Features

3.1 Hint

We offers users hints to give better user experience .Our game is hard and especially for children , we want to make game more basic.Hints have time limitations.

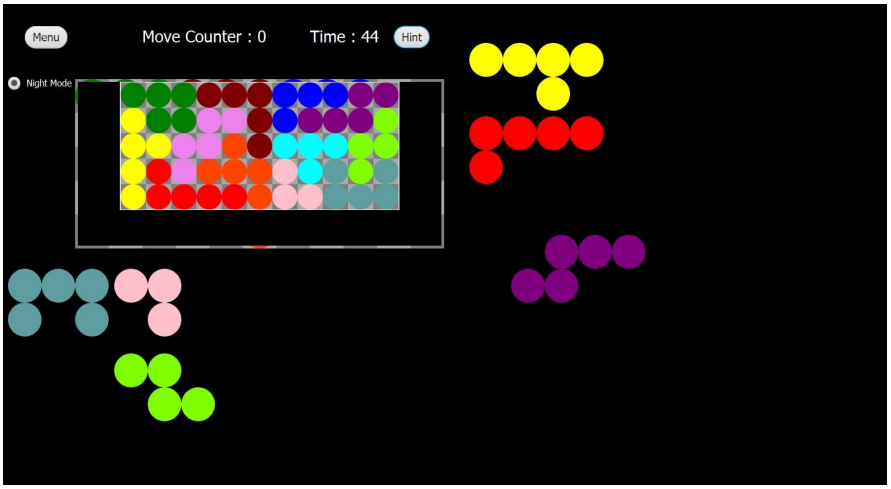

Figure 7: Black Board Night Mode Hint

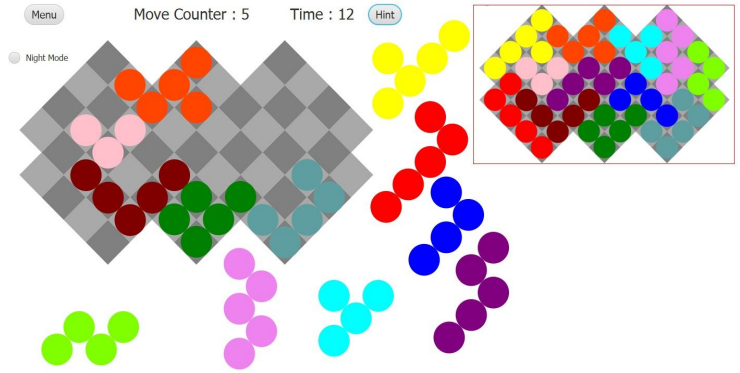

Figure 8: Gray board Hint

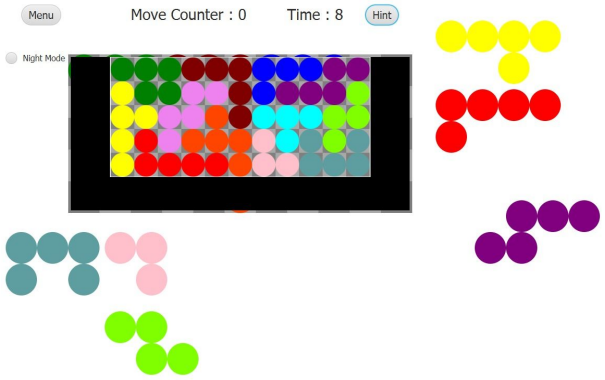

Figure 9: Black board Hint

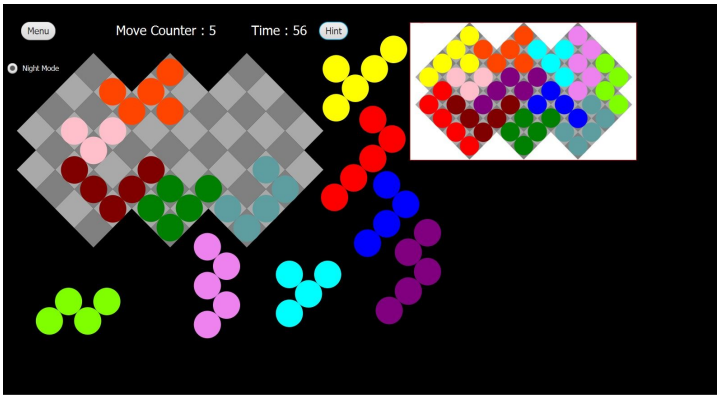

Figure 10: Gray board Night Mode Hint

3.2 Time and Move Count Challenge

Time and move challenge offers experienced users to challenge with time and piece's movement.

3.3 Night Mode

To protect user's eye health and make playable in the night time,we create night mode.

3.4 Save Game

Save game offers user to save their game with their nicknames..Player's private data stored with our data stores.To compare players each other and demonstrate in leaderboard we store scores of players.Also,in game we have levels and users have to pass basic levels to reach others.We store level information which is which user pass the which level with data store.Therefore , user can save the game and continue with his/her level which succeed before.

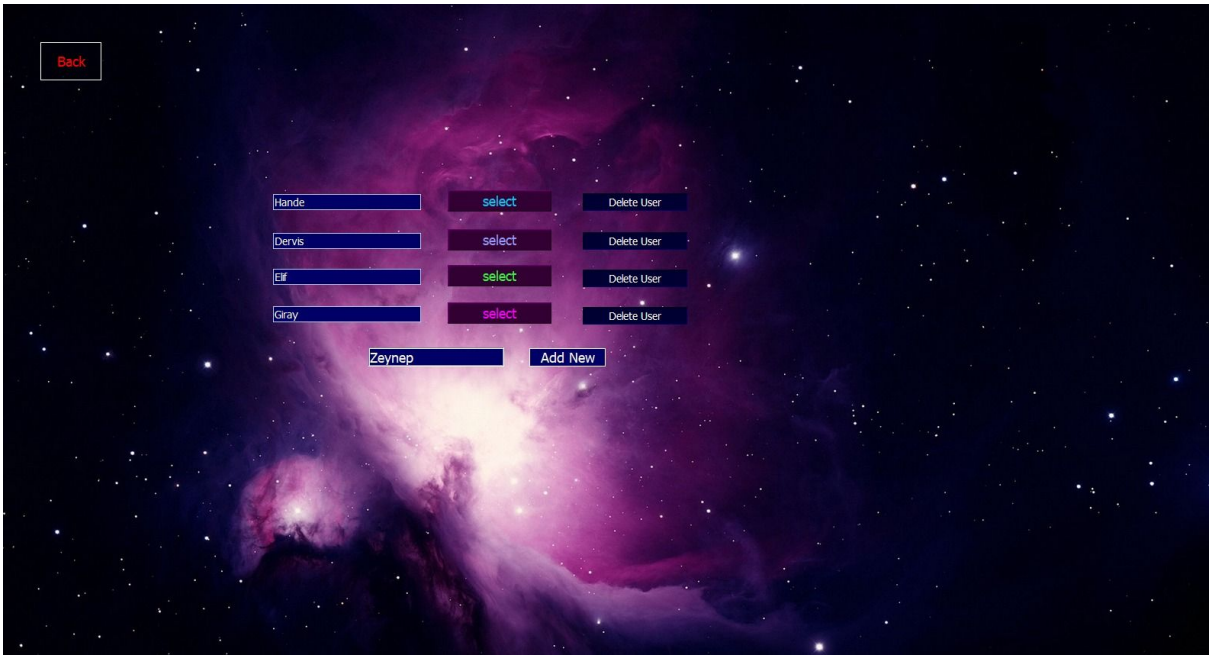

Figure 11: User Selection Screen

4.What is next and missing parts

At the end of the second iteration, we mostly manage implement the basic game logic. For 2D, we implement all boards with missing and extra features. For 3D, we have some progress on the implementation such as creating 3D pieces, add basic functionalities such as rotate and drag, combining these pieces as one union piece. We also implement 3D board, however we can't combine board and pieces; therefore, can't create pyramid shape which is aim of the 3D part. Rotating camera in 3D builds with JavaFx is problematic but with another IDE platform and another language,game's 3D part can be more possible.

Multiplayer part is another missing part. We can't implement this part because we are lack of time to implement multiplayer game mode. We decide to focus on single player mode which is more important for our game logic.

We added some challenge extensions for scoring them and putting into leaderboard so as to make our game more challenging. We can put them in order related their scores but there is no trophies for leaders, maybe in next step would be putting trophies for leader players.

Our future aims is firstly complete missing parts of our implementation, especially 3D part. After complete missing parts, we want to add extra levels for both black and gray board. Additionally, there could be some extra features such as a shop. This shop can provide users to buy features such as different board and piece colors. Users can buy these

features by gain scores from levels. In addition, we had a local database via downloading the program thanks to JSON object from GSON library, so there could be online game mode in next step to progress our game.

5.Conclusion

For the general perspective, we tried to complete what we are required to do to code original IQ Puzzler Pro game. We tried to do our best and had an efficient team-work to have a better project. We have learnt many things to get use to real life projects because main target of that lecture is to give us some experience on working in a group of people as we will experience in future.

5.1 Experiences that we had through project:

Sleepless nights, we improve our team-work features.Also,we experienced that we have entirely one project with all sides.We work like a test engineer, requirement engineer , developer and so on.

5.2 Planning, Designing the Project

We understood the importance of planning and designing while conducting the project, it provides us a neat work program. Thereby, we could easily saw some problems might occur without having a program and design related tough assignments, quizzes and exams conditions.

5.3 Time Management

Due to harsh Bilkent student conditions we had to be conscious about our schedule, we need to study external assignments, quizzes, exams and participate almost every hours while conducting our lecture project.

5.4 Some related topics of project we worked on which were not aware of them before

We just heard about JavaFX but we progressed a project on that and it was a good experience for us.

Also, 3D extension of JavaFX was one of the things we did not know but we are in progress in coding 3D JavaFX now on.

We were not aware about data storing objects before but we learned something about JSON object and we used it. Also while we were using it we used GSON Google libraries and thanks to them we calculated our equations easily.

6 . Allocation the work done by peers

Analysis , design and final reports are done by all team members.Throughout all project , all design choices are decided together.All problems about implementation and other things also handle with together.

Elif Beril Şaylı:

She implemented data management class, which was using json format to save info about players to json file with other member Zeynep Ayça Çam. This class is our data management class for our project. She created Player class which is about players , scores and player's level with Zeynep Ayça Çam. In 2D ,for first iteration , she created blackboard logic. She also wrote Menu Controller class with other members.She interested with pieces,mostly.She wrote piece's creation, rotation and movement logic.She wrote PieceUI logic and design her responsible pieces. In 3D, she created main logic especially, creating methods of intersection and unifying with other pieces . She determined to whole pieces control and creation logic for 2D and 3D. She created time management events with Derviş Mehmet Barutçu.

Zeynep Hande Arpakuş:

She implemented detect intersection method for 2D part of the game with Zeynep Ayça Çam. This method is used for fit pieces to the right places in the board. Also, they implemented the blackboard of the game. Also she wrote the logic of levels of two boards and calculated the right coordinates of pieces to create the game scene. She created pieces for game scene with other team members. Also she played a role in implementation of method of isBordFull with her old methods and contributed to implementation of MenuController. In 3D part of the game she wrote the codes of cameras to show the game scene as 3D, also implemented 3D board for the game scene. And she calculated the coordinates for creating 3D pieces.

Zeynep Ayça Çam:

She implemented data management class, which is using json format to save info about players to json file with other member Elif Beril Şaylı. This class is our data management class for our project. She contributed some of the menu panels UI's, their view mostly. She also wrote Menu Controller class with other members. Also, she implemented

the black board of the game with Zeynep Hande Arpakuş. She implemented detect intersection method for 2D part of the game with Zeynep Ayça Çam. This method is used for fit pieces to the right places in the board. She created Player class with Elif Beril Şaylı. She created some of the UI's of piece. In the second iteration, she organized some of the piece control logic again in order to better fit. She also implement isBoardFull method, which also get help from detect intersection method. Is game over method also written by her, which is again get help of isBoardFull method.

Giray Baha Kezer:

He implemented the sound managements' "wav" extended sounds to the main class IQPuzzlerPro,which click sound and playing sound. He is responsible for sound logic in game.He implemented main logic of Grey board rectangles and gridpane rotation on viewer He wrote the Hint class that could be taken through a button on menu. He wrote some pieces, contributed the MenuController class also. He contributed some Menu UI with logos and textures.

Derviş Mehmed Barutçu:

He implemented the Menu UI parts, especially most view of panels. He wrote some of the UI part of the black board and some logic of the black board with other members. He implemented some of the piece UI's with other members. He created time management with Elif Beril Şayli. He also implemented challenge modes of game, both time and move count challenge. He created Night Mode of the game. He connect games hint with timer class. He also had contributions on piece control class which are rotation, flip and drag functions of pieces with Elif Beril Şaylı. He also wrote Menu Controller class with other members. He implemented isBoardFull method which also help logic of ending the game.

We all help each others part, especially with our problems.