CS 319 - Object-Oriented Software Engineering

Analysis Report

IQPuzzler Pro

Group 2J

Burak Alaydin
Kaan Altinay
Muhammad Umair Ahmed
Aliyu Saifullah Vandana
Skerd Xhafa
Supervisor: Eray Tuzun
TA: Mohammed Cavusoglu

Contents

1. **Introduction**

As a group we decided for this term's project to pick a popular mind game named IQPuzzler Pro. We will develop a similar digital version of this learning board game also packed with lots of improvements and added features that hopefully will make the game more enjoyable and fun to play.

The game itself consists of a set of ball-and-stick pieces of different colors and arrangements that the player has to arrange on the board in order to fill the board completely. If the player is able to fill the board using the remaining pieces with the starting pieces already on the board, the player wins that challenge.

There are 3 different board socket patterns in the game: standard rectangular and dual-diamond-shaped, both of which are used for the style of challenges mentioned above.

Taking these core principles, we have set out to create a digital version of the game and adding our own twist to it.

The game will be developed using Java since it is the more viable choice we can make keeping in view our current level of skill. It will also make it easier to build and the results would be more reliable and beautiful design wise.

2. Overview

This section provides a brief look at the components and mechanics of the game that we are looking to build.

2.1 Gameplay Components

The components of the game involved in active gameplay will be:

Board: The board can be one of these 3 types: rectangular grid of sockets, the dimensions of which will be flexible, a diamond-shaped grid of sockets, the dimensions of which will be fixed or a square grid of sockets, the dimensions of which will either be fixed or flexible depending on the game mode.

2.1.1 Ball-and-Stick Puzzle Pieces: An inventory of a variety of puzzle pieces that can be placed on the grid to solve challenges.

2.1.2 Countdown Timer: This timer will keep track of how much time is left when a challenge ends or if it does not end within the allotted time.

2.1.3 Grid: The grid is where all the puzzle pieces will be placed. The player's objective is to fill out the grid completely with the given pieces.

2.2 Gameplay Modes

We have two main game modes in our digital version of the game. 2D Challenges and Story Mode.

2.2.1 2D Challenge: In this mode, the player will initially have the choice to choose the style of board they want to play with, between the rectangular, the diamond and the square boards. Once this choice has been made, the player can choose the difficulty they want (please refer to section 2.1.3 for information on difficulty and how it will be governed). A randomly generated pattern will then be given to the player, along with an inventory of puzzle pieces, as the "challenge" and their task will be to recreate it the pattern given from the given puzzle pieces.

2.2.2 Story Mode

This mode will involve the player progressing through a story, level by level, from a selection of children's fairy-tales. Progressing through each level will reveal the subsequent part of the story. The said progression will be performed by solving a 2D puzzle related to the last-revealed part of the story. So, for example, if the player chose to play Rapunzel's story, they would get a "challenge" to recreate a pattern representing Rapunzel's hair to unlock the part of the story where Rapunzel lets her hair down the tower. The puzzles in this mode will be predefined and will generally follow the rules of the 2D game mode except that the grid size will be fixed in the story-mode challenges (Please refer to 2.1.3. Difficulty for more information on grid-size). All stories will have 3 challenges.

2.2.1.1. Beauty and the Beast: This story begins with an arrogant, rich and handsome prince who lives in a great castle in the forest. The arrogant young prince and his castle's servants fall under the spell of a wicked enchantress, who turns him into the hideous Beast until he learns to love and be loved in return. The spirited, headstrong village girl Belle enters the Beast's castle after he imprisons her father Maurice. With the help of his enchanted servants, including the matronly Mrs. Potts, Belle begins to draw the cold-hearted Beast out of his isolation. The levels in this story mode will progress in such a way that, as the player is able to solve more puzzles the village girl inches closer to saving the Prince-turned-Beast.

2.2.1.2. Peter Pan: This is the story of a boy who never grew up. Living a bleak existence at a London orphanage, 12-year-old Peter finds himself whisked away to the fantastical world of Neverland. Adventure awaits as he meets new friend James Hook and the warrior Tiger Lily. They must band together to save Neverland from the ruthless pirate Blackbeard. Along the way, the rebellious and mischievous boy discovers his true destiny, becoming the hero forever known as Peter Pan. The levels will progress with the journey of the boy to his becoming of Peter Pan.

2.2.1.3. Spiderman: This story centers on student named Peter Parker who, after being bitten by a genetically-altered spider, gains superhuman strength and the spider-like ability to cling to any surface. He vows to use his abilities to fight crime, coming to understand the words of his beloved Uncle Ben: "With great power comes great responsibility". The levels will progress with the journey of Peter to fight crimes in his city.

2.2.1.4. Cinderella: After her father unexpectedly dies, young Ella finds herself at the mercy of her cruel stepmother and stepsisters, who reduce her to scullery maid. Despite her circumstances, she refuses to despair. An invitation to a palace ball gives Ella hope that she might reunite with the dashing stranger she met in the woods, but her stepmother prevents her from going. Help arrives in the form of a kindly beggar woman who has a magic touch for ordinary things. The levels will progress with the character of Ella trying to free herself from the shackles of her stepmother.

2.3 Difficulty

Difficulty of the puzzles will be governed by 2 parameters in 2D mode: the size of the grid and the amount of time the player chooses on the countdown timer to start the game with. In Story mode, the player can only choose the amount of time on the countdown timer that they get to start with. Ideally, the amount of starting time on the countdown-

timer would be changed in steps (Infinite Time >Very Easy >Easy >Fair >Hard >Very Hard >Impossible)

2.4 Scoring and Bonuses

Scoring on the 2D and 3D mode challenges will be in terms of points. There will be no scoring in the Story Mode challenges BUT they will include time bonuses. A set number of points will be awarded for each correctly positioned puzzle piece in either mode. Once a challenge is successfully ended, all the points will be totaled up to give the player their score. As far as bonuses are concerned, there will be 3 main types of bonuses:

2.4.1 Score Multiplier

A score multiplier will multiply the points that you get for each correctly placed puzzle piece by a set factor. The player will be able to trigger multipliers by maintaining streaks (consecutively placing puzzle pieces in the correct place without fail) on the grid.

2.4.2 Leftover Time Bonus

In the challenge mode the player is given some time to complete a challenge. If the player is able to finish earlier than the allotted time, the leftover time will be multiplied by the score multiplier.

3. Functional Requirements

This section briefly describes the functionalities that our game MUST have in order to be completely functional.

3.1 Play Game

The player should be able to play a challenge with the grid size,

difficulty level and time limit of their choice. During gameplay the user should be able to place pieces on the grid and move pieces that are already on the grid.

3.2 Save Game

The player should be able to save the state of a game while it is in progress.

3.3 Load Game

The player should be able to load the state of a saved game from the save-game file and continue from where they left off.

3.4 Read Instructions

The player should be able to consult the instruction manual during the game in order to get help in making their moves.

3.5 Change Preferences

The player should be able to change various game-related preferences such as: Turn menu music on or off, turn in-game on or off, change sound volume and change the current user.

3.6 Check Stats

The player should be able to check statistics from all the previous games that have been played up till now, in order to see previous scores and compare them with their own.

3.7 Read Credits

The player should be able to see the credits roll for this game with all personnel (developers, supervisors and teacher assistants) listed.

3.8 Exit Game

Finally, the player should be able to exit the game to go back to their operating system.

4. Non-Functional Requirements

This section lists and explains the non-functional requirements (related to quality of user experience) that we aspire to implement in our game.

4.1 Usability Requirements

Inexperienced players should be able to start the game within 60 seconds. The main menu will not be overwhelmed with too much items we will keep only the most important ones like, play game, options, credit and exit. Game mechanics will be adjusted properly for each background and trait options in order to provide balanced gameplay. Difficulty of the game will be adjusted properly to avoid long stretch of boring easy levels or frustrating hard levels. When the user is placing a piece, the shape of the piece will be highlighted on the board to simplify and enhance gameplay. To rotate a piece, the player can use keyboard shortcuts instead of the mouse which could prove difficult sometimes.

4.2 Game Performance

The game should be able to run at a high frame rate(about 60 frames /second) in order to provide smooth frame transition. Extra components will be kept minimal so as not to trade off quality game performance. The in-game graphics will be clear, well designed with

appealing look in order to create a nicer atmosphere and as such, a nice gaming experience for the player. Control mechanism of the game will have low response time which will allow the player to play without intermittent delays.

### 4.3 Implementation Requirements

The game has a fixed number of puzzle pieces (12). For developers, adding more items should be a trivial process without much coding required. Our system will be able to recognize added items and respond properly.

4.4 Portability

The game will be designed such that the resolution will be suitable for all monitor types. The user should be able to play the game and get same experience on different platforms such as Windows, Linux and Macintosh systems.

5. System Models

5.1 Use-Case Model
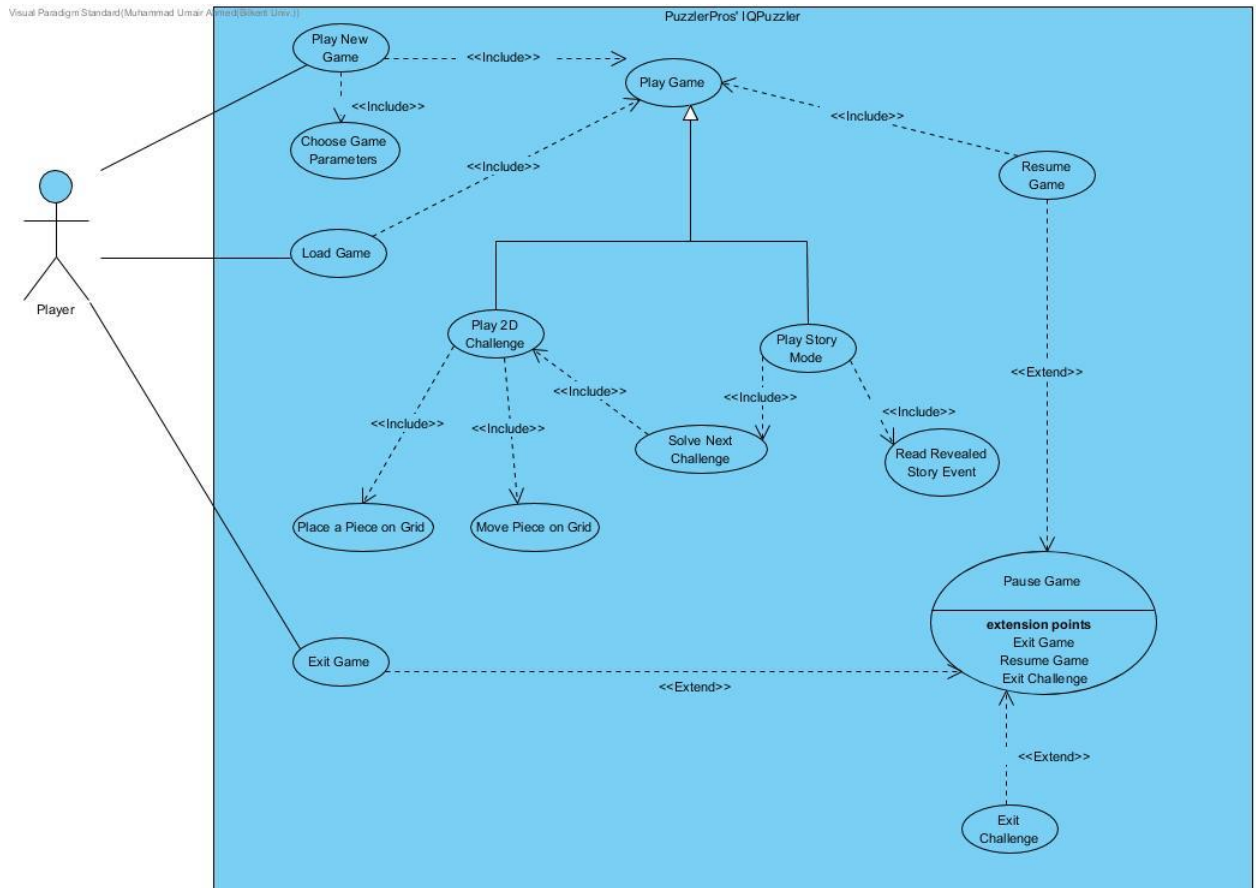
Use Case Diagram Textual descriptions

Play New Game

Use Case 1

Use Case : Play New Game

Primary Actor: Player

Entry Condition:

    1.   Player must be in the main menu

Exit Condition:

    2.   Player exits the game

Flow of events:

1. Player selects "Play New Game" button from the main menu

2. Player is directed to Choose Game Parameters(UC2).

Choose Game Parameters:

Use Case 2

Use Case: Choose Game Parameters

Primary Actor: Player

Entry Condition:

1.   Player selects the "Start New Game" option from the MainMenu

Exit Condition:

1.   Player chooses all the parameters and decides to proceed to Play Game

2.   Player navigates back to the MainMenu.

Flow of events:

1.   The Player chooses the Game Type they want to play, from 2D or Story Mode

2.   The Player chooses the difficulty level they want to play at, which governs the amount of time they start the game with on the CountdownTimer.

3.   The Player chooses the type of Grid they want to play on, from Square, Rectangular and Diamond, and also the GridSize.

Play Game:

Use Case 3

Use Case: Play Game

Primary Actor: Player

Entry Condition:

1. Player has successfully completed UC1(Play New Game) or UC10(Load Game)

Exit Condition:

1. Player chooses to Pause the game

2. Player finishes the game by completing the challenge

Flow of events (success scenario):

1. The Player is given an Inventory of puzzle pieces and the Pattern they have to recreate using Pieces from that Inventory.

2. When the user is ready, the CountdownTimer starts counting down. The Player's objective is to recreate the target Pattern before this CountdownTimer reaches zero.

3. The Player triggers either Place Piece on Grid (UC) or Move Piece on Grid(UC) use-cases repeatedly until the board is full.

4. If the Player manages to place 5 consecutive Pieces correctly on the Grid, the Streak Bonus is activated. Every activation of the Streak bonus multiplies the increment in the Player's Score for subsequent correctly placed Pieces by two. The Streak Bonus is deactivated at the first incorrectly placed Piece after being triggered.

5. Streak Bonuses get stacked if the user gets 10, 15 or greater multiples of 5 Pieces placed correctly on the Grid.

6. The Player recreates the pattern before the CountdownTimer reaches zero and the challenge is completed. The game then gives the Player the total score for that round, by adding the score obtained during the game and then adding 5 points to the Player's score for each second that's left on the Countdown Timer.

   • (Alternative Flow): The Player fails to recreate the Pattern in the given time and gets a fixed score for attempting the challenge.

7. The challenge is ended, and the user can now either progress to the next Level or go back to the MainMenu.

Play 2D Challenge:

Use Case 4

Use Case: Play 2D Challenge

Primary Actor: Player

Entry Condition:

- Inherited from UC3 (Play Game) with GameType 2D chosen by the Player in UC2 (Choose Game Parameters)

Exit Condition:

- Inherited from UC3 (Play Game)

Flow of events (success):

1. Flow of events inherited from UC3 (Play Game) with the following specifications: The target Pattern can only be 2-dimensional, the Grid can also only be 2 dimensional, either Square, Rectangular or Diamond, depending on the choice made in made in UC2 (Choose Game Parameters). The GridSize is also decided by choices made in UC2.

Play Story Mode:

Use Case 5

Use Case: Play Story Mode

Primary Actor: Player

Entry Condition:

- Inherited from UC3 (Play Game) with GameType Story Mode chosen by the Player in UC2(Choose Game Parameters)

Exit Condition:

- Inherited from UC3 (Play Game)

Flow of events:

1. The Player chooses one of the 4 Stories that are provided.

2. Beginning part of the story is revealed to the player

3. The Player then resumes the selected Story from the plot development that was revealed after solving the last solved challenge.

4. The Player is then given a new 2D Challenge which they have to solve in order for the next plot development in the Story to be revealed. The target Pattern in this 2D Challenge will be related to the next plot development. UC4 (2D Challenge) is then triggered.

5. The Player successfully completes UC4 and the next plot event in the Story revealed.

   a. Alternative Flow: UC4 is completed without success in recreation of the given Pattern as outlined in its parent UC3, flow event 6a. In this case the plot event is not revealed. The Player can retry the challenge or navigate back to a different Story, GameType or back to the MainMenu.

6. The Player can move on to the next challenge to reveal the next plot event or navigate back up the menus for a different Story, GameType or MainMenu.

Pause Game:

Use Case 7

Use Case: Pause Game

Primary Actor: Player

Pre-Conditions:

- Player must be playing the game Entry Condition:

Entry Condition:

- Player selects "Pause" button from game screen

- Player presses CTRL + P

Exit Condition:

- Player selects "Resume" button from pause menu

- Player presses CTRL + R

- Player selects "Exit Challenge" button from pause menu

Flow of events(success):

1. Player selects "Pause" button from game screen or presses CTRL + P

2. The system displays pause menu

Resume Game:

Use Case 8

Use Case: Resume Game

Primary Actor: Player

Entry Condition:

- Player selects "Resume" button from pause menu

- Player presses CTRL + R

Exit Condition:

Flow of events:

1. Player selects "Resume" button from pause menu

2. Player enters the Play Game use case again without their progress being affected

Exit Game:

Use Case 9

Use Case: Exit Game

Primary Actor: Player

Entry Condition:

- Player selects "Exit Challenge" button from pause menu

Exit Condition:

Flow of events:

1. Player selects "Exit Challenge" button from pause menu

2. The challenge game is stopped

Load Game:

Use Case 10

Use Case: Load Game

Primary Actor: Player

Entry Condition:

- Player selects "Load Game" Button from the main menu

Exit Condition:

- Player returns to main menu

- Player chooses game mode

Flow of events:

1. Player selects "Load Game" Button from the main menu

2. The player then selects the save-game that they want to resume their game from.

3. The game is loaded and the Play Game(UC3) use case is triggered.

Place a piece on grid:

Use Case 11

Use Case: Place a piece on grid

Primary Actor: Player

Entry Condition:

- Player selects a piece

Exit Condition:

- Player places the piece on the game board

- Player drops the piece back to change

Flow of events:

1. Player chooses a piece from the Inventory

2. Player drags the piece to drop on the Board

3. The piece is successfully placed in the location that the Player desired to place it on

- Alternative Flow: the location at which the Player tried to place a piece is unavailable because another piece is already occupying it. In this case, the Player's piece is simply returned to their inventory and they have to trigger this or the Place Piece on Board use case again.

Move piece on the grid:

Use Case 12

Use Case: Move piece on the grid

Primary Actor: Player

Entry Condition:

- Player selects a piece from the given pieces

Exit Condition:

- Player places the piece on the game board

- Player drops the piece back to change

Flow of events:

1. Player chooses a piece from the pieces already on the Board

2. Player drags the piece over gameboard which indicates available slots to place the piece

3. The piece is then successfully placed on the board.

   - Alternative Flow: the location at which the Player tried to place a piece is unavailable because another piece is already occupying it. In this case, the Player's piece is simply returned to their inventory and they have to trigger this or the Place Piece on Board use case again.

Solve Next Challenge:

Use Case 13

Use Case: Solve Next Challenge

Primary Actor: Player

Entry Condition:

- Player is playing Story Mode and the next story event is locked.

Exit Condition:

- Player starts the next challenge
- Player goes back to main menu

Flow of events:

1. Player is taken to a 2DChallenge. The 2D Challenge use case is triggered.

Read Revealed Story Event:

Use Case 14

Use Case: Read Revealed Story Event

Primary Actor: Player

Entry Condition:

- The Player selects the Exit Game option from the MainMenu.

Exit Condition:

1. The Game application is successfully terminated.

Flow of events:

1. Player chooses to play the story mode challenge

23

2. Beginning part of the story is revealed to the player before starting the game

Alternative Flow of events:

1. Player completes a level challenge in story mode

2. The player is given an option to solve the next challenge

3. Before the next challenge begins another part of the story is revealed to the player.
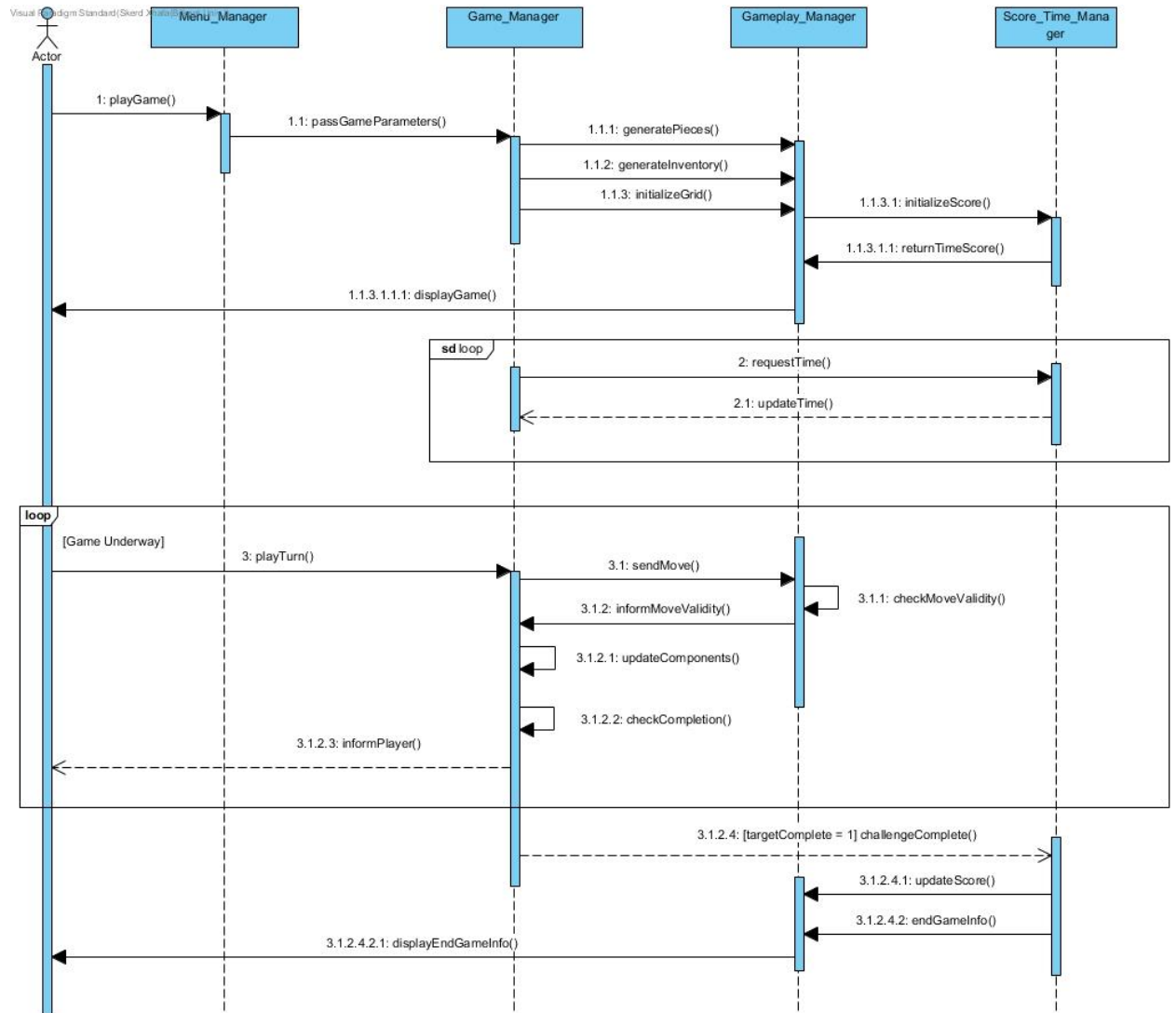
5.2 Activity Diagram

This activity diagram describes the overall flow of control that is available to the user to navigate through, starting from the main menu of the game, through all of the game modes and the actions associated with them and back to the main menu.
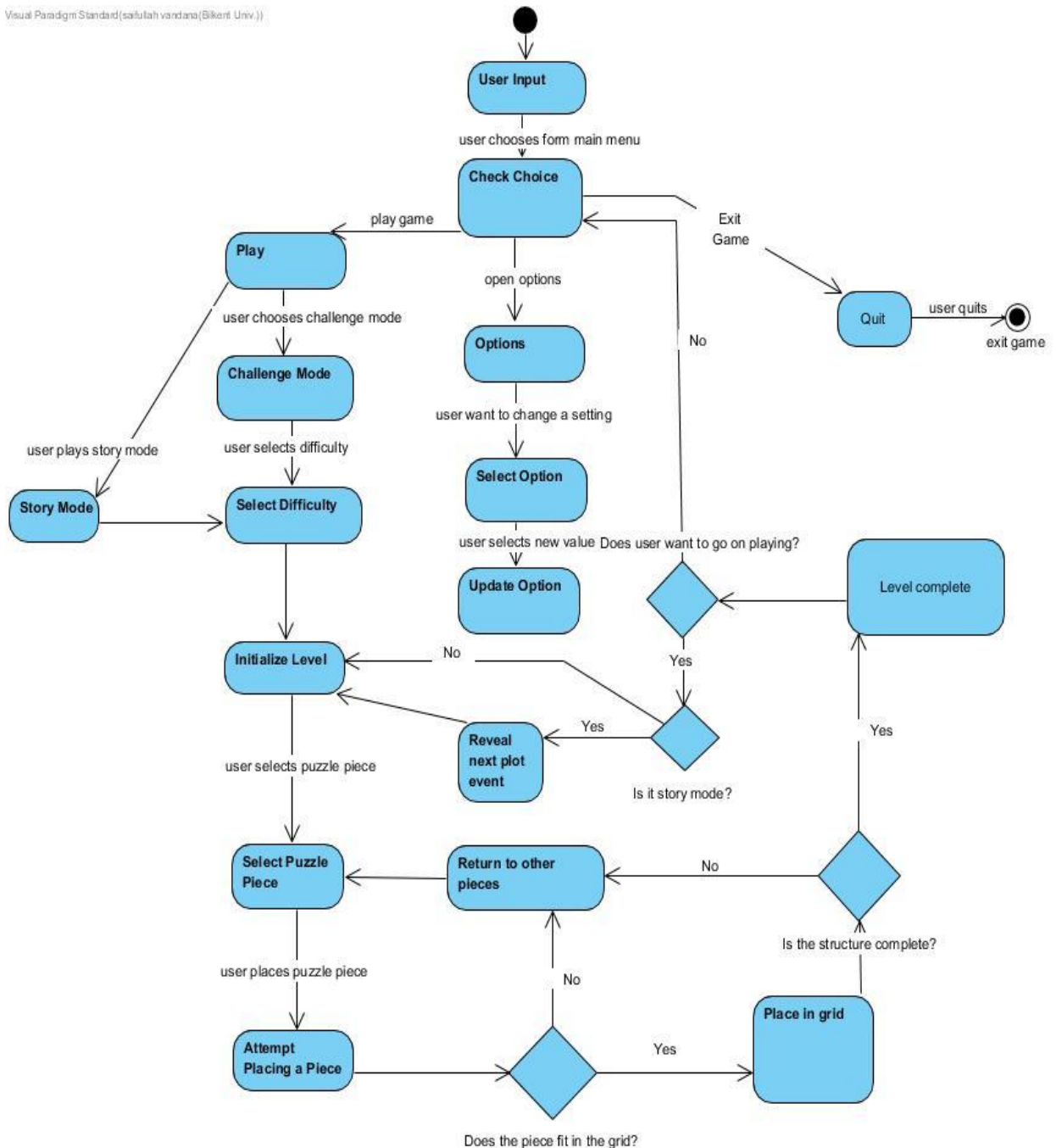
5.3 Sequence Diagrams

Play Game sequence

Firstly, a player is at the main menu. In here one can select many
options, and one of them is play game. After clicking this, game_manager
will send parameters to the gameplay_manager in order for the
gameplay_manager to initiate game puzzle pieces, game grid and
inventory. After all the initialization is done, the gameplay_manager will

start the time that will start counting as the game would be ready to start. At this point the player can play the game normally, by picking puzzle pieces to place on the grid. At each player turn, the game_manager will check if a piece is eligible to be put at the position the player dropped the piece. If it is the game_manager will place the piece on the grid, else it will notify the player that the piece was not placed on the grid. This can be done many times, until the grid is completed and full of pieces. At this time the game_manager will send a signal to the PieceAndTimeManager to calculate the points the player won for the completed puzzle and add these points and any eligible bonuses to the players account. After this is done, a post-game information will be shown to the user indicating points, time completed and an option to continue to the next level.

Options sequence
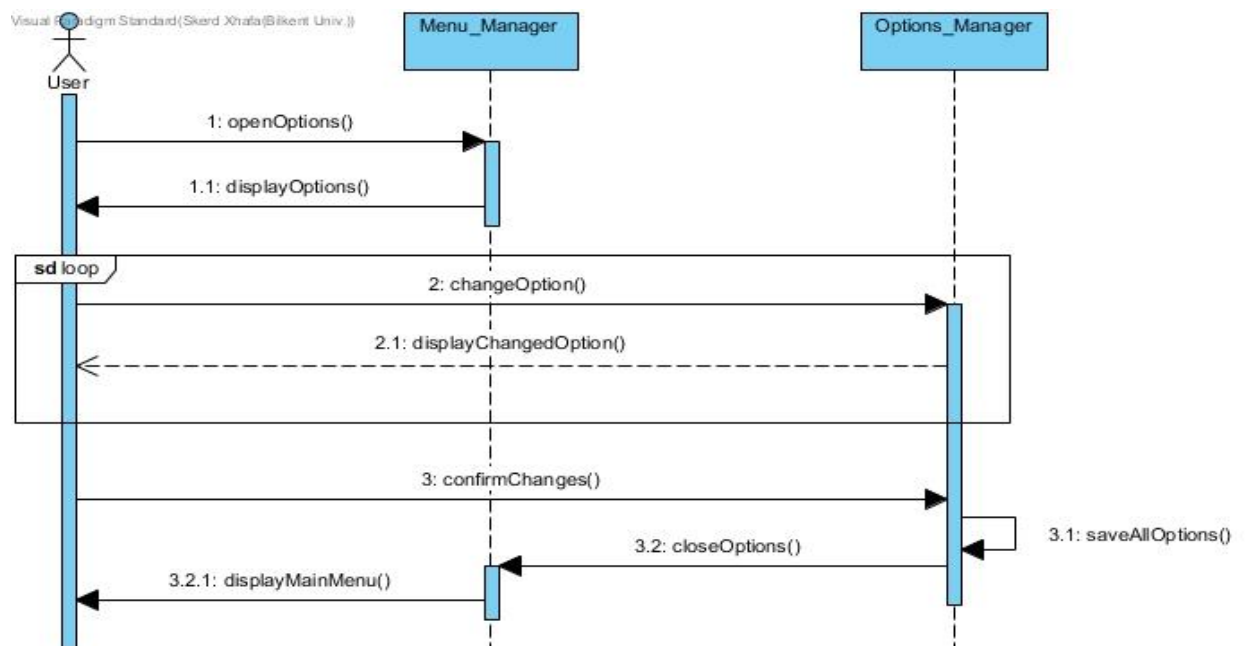
When the user is on the main screen, one can select many options. One

of the is the Options button. After clicking a message will be sent to the

menu_manager and all available options that can be tweaked are

shown to the user. The user can now change settings and after each

change, the option_manager will display these changes to the user.

This can be done as many times as the user wants, until the confirm

button is pressed. At this point, a message will be sent to the

option_manager that will save the settings in order to reuse them.

After this a signal will be sent to menu_manager, and it will remove the

options from the screen and show the main menu to the user.

Exit sequence

When the user is on the main menu, one can chose to exit the game. If

the Exit button is pressed the menu_manager will save everything and

then exit the game.
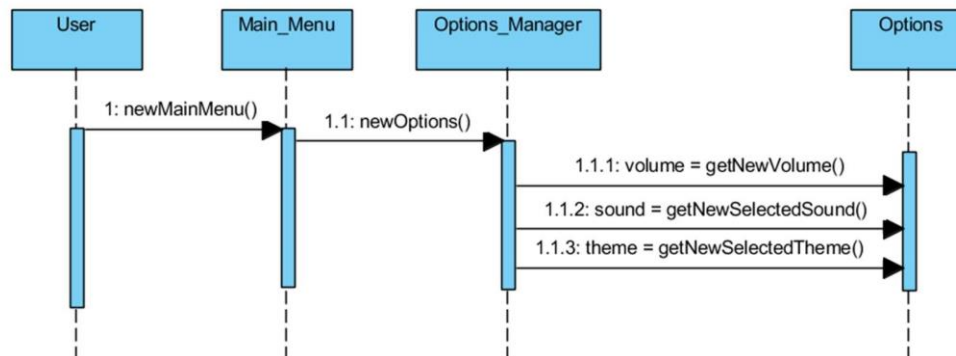
Stats Sequence Diagram:
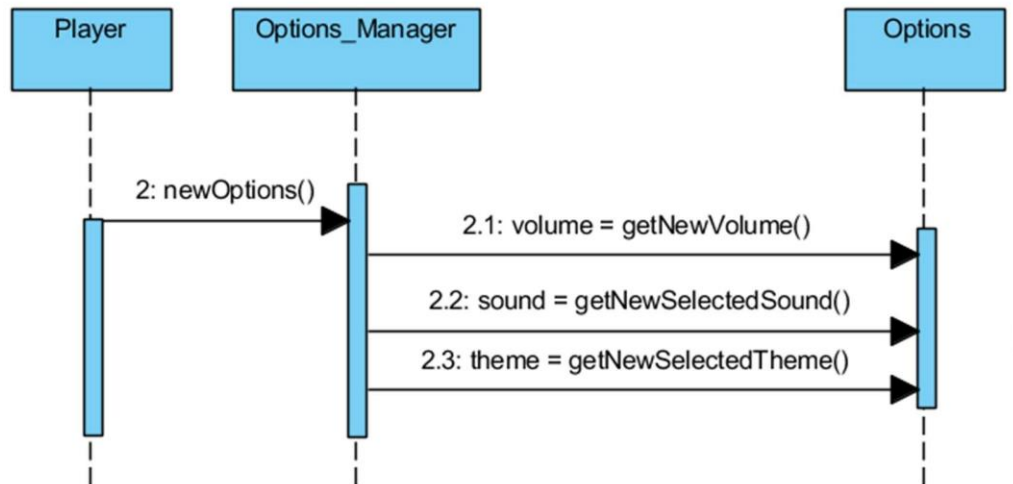

How to Play Sequence Diagram:
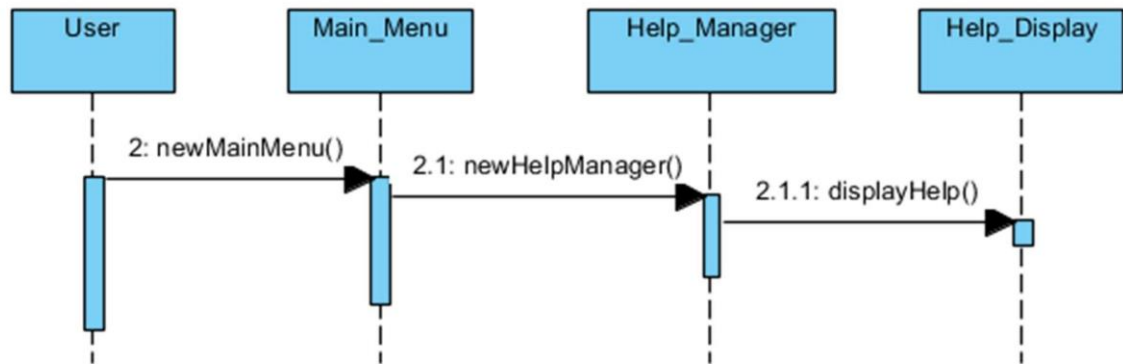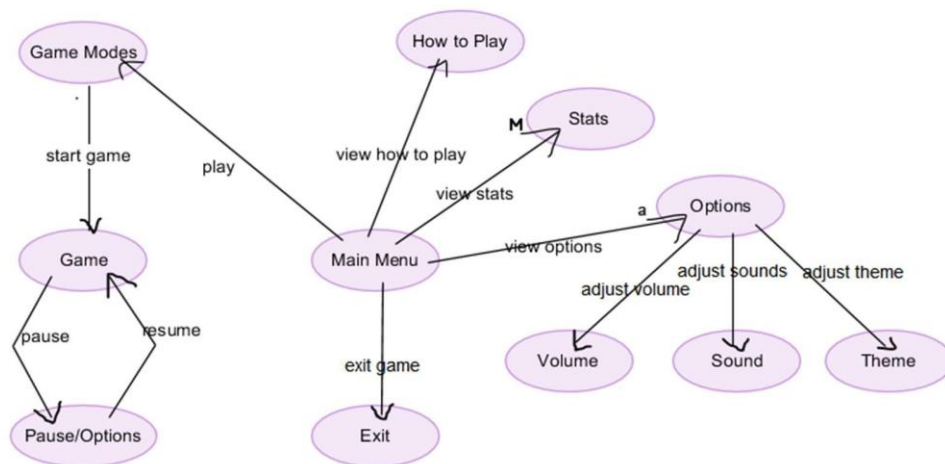

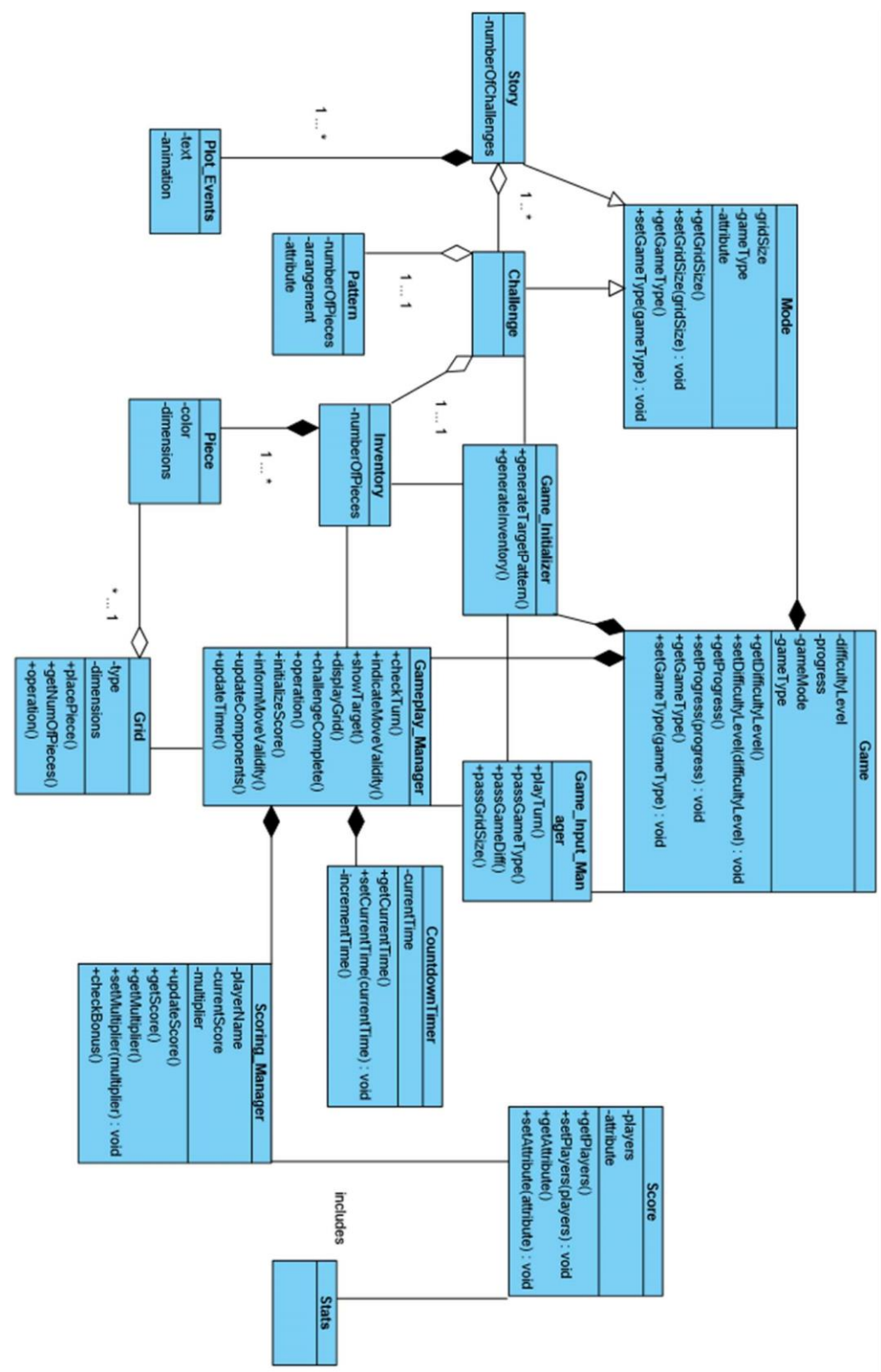**Options Sequence Diagram:**

Main menu:

In-game:

5.4 Navigational Path

5.5 Class Diagram

The above diagram describes our preliminary idea of what the relations will be between models of the game components of IQPuzzler Pro