Bilkent University

Department of Computer Engineering

CS319 Design Project

IQPuzzler Pro:

Analysis Report

Burak Alaydin, Kaan Altinay, Muhammad Umair Ahmed, Saifullah Vandana and Skerd Xhafa

Supervisor: Prof. Eray Tuzun

Progress Report
October 21, 2018

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Object Oriented Software Engineering project CS319.

Content Table:

Analysis Report

IQPuzzler Pro

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 in a set pattern identical to the one of the predefined "challange" pattern on the board filled with spherical sockets waiting to be filled. If the player is able to recreate the predefined patter on the board from a challange, then they win that challange and continue to the next one.

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. The third pattern is a square of sockets which involves a very different challenge: constructing a pyramid with the given “pieces”.

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 JavaFX since it is the more viable choice we can make. It will also make it easier to build and the results would be more reliable and beautiful design wise.

2. Overview

This section provides an in-depth look at the game we have come up with. IQPuzzler Pro is a computer video game having the following characteristics.

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.

2.2 Gameplay Modes

Our version of the game will have 3 main modes:

2.2.1: 2D:

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: 3D:

This mode will only have the square grid and the player, after choosing their desired difficulty level, will be tasked with building a pyramid from a given inventory of puzzle pieces.

2.2.3: Story Mode (Tentative):

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).

2.2.3.1: Story Event 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.

2.2.3.2: Story Event 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.

2.2.3.3: Story Event 3: Spider Man:

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.

2.2.3.4: Story Event 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. 3D and Story modes, 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) and by incorporating a bonus **tricky** puzzle piece from the inventory in their 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.

2.4.3: Shapeshifter Puzzle Piece Bonus:

This bonus results from the user being able to maintain a streak. The shapeshifter puzzle piece as the name suggest can be made to fit into a puzzle bending its shape to fit the missing piece.

2.4.4: Tricky Puzzle Pieces:

This tricky piece got its name from the fact that it is shaped in a rather bizarre way. Its shape is a mix of different regular shapes which makes it complex. If a player is able to fit this piece to solve a puzzle the score doubles.

3 Functional Requirements

3.1: Play Game:

This can be found on the screen homepage. The user starts a new game by entering this screen. New games can be a normal game, challenge or story mode. This will be decided by the player. In a normal mode game, the player is directed to the main game screen with the default difficulty being easy unless the player decides to change that from the settings menu. The game is mouse controlled which the uses to drag and place a piece in its place. The pieces will be shown in two's (current

and next piece). With the help of the mouse the player can switch a current piece to the next piece until the desired piece appears.

3.2 Load Game:

This screen contains the saved games by a player. It is found on the screen homepage. On entering this screen, the player is able to view different saved games ranging from challenges to story mode types. The player can choose which one to play and then resume the game.

3.3 Challenge:

This screen directs the player to a challenge mode. It can be accessed directly from the homescreen. In the challenge mode the player is required to complete a level of the game within a specific time. The count-down timer keeps track of the time the player used to complete the game. The challenge mode is automatically programmed to progress with difficulty as the player advances to higher levels.

3.4 How to Play:

This aspect of the game can be accessed from the homescreen menus. It entails the description of the game, the proper controls to use and the explanation of the varying modes of play in the game. Some key features present on the game screen are also explained in this part. Some of these key features are undo button, redo button, hint button etc.

3.5 Options: The player can access this feature from the homescreen of game and also it is present as a sub-menu item list when the

player pauses a game. With this feature the player can be able to change some settings of the game.

Some of the changeable settings include:

- Volume level
- Theme
- Sound

3.6 Stats:

This feature will redirect users to a page in which the user will be able to see every achievement he/she has and very interesting statistics about all the games played like the number of puzzles solaced and much more.

3.7 Credits:

The player can access this feature from the home screen. This feature redirects to the screen containing some information of the developers of this game. This information includes the e-mail address and location of the developers which can be used by the player for feedback.

4 Non-functional requirements

4.1 Extendibility:

The game will be implemented with an extendible design which will allow room for future developments. By so doing, new features can be added to the already existing ones without making much changes on the original source code. This can be achieved using the Object-Oriented Design concepts which will make it easy for an extendible design to be made.

4.2 Reliability:

The game progress of a player will be stored in an online database which can be accessed from homescreen using the load game. It safe to say that the records, rewards and game progress of a player will not be lost due to a system failure or power outage.

4.3 Usability:

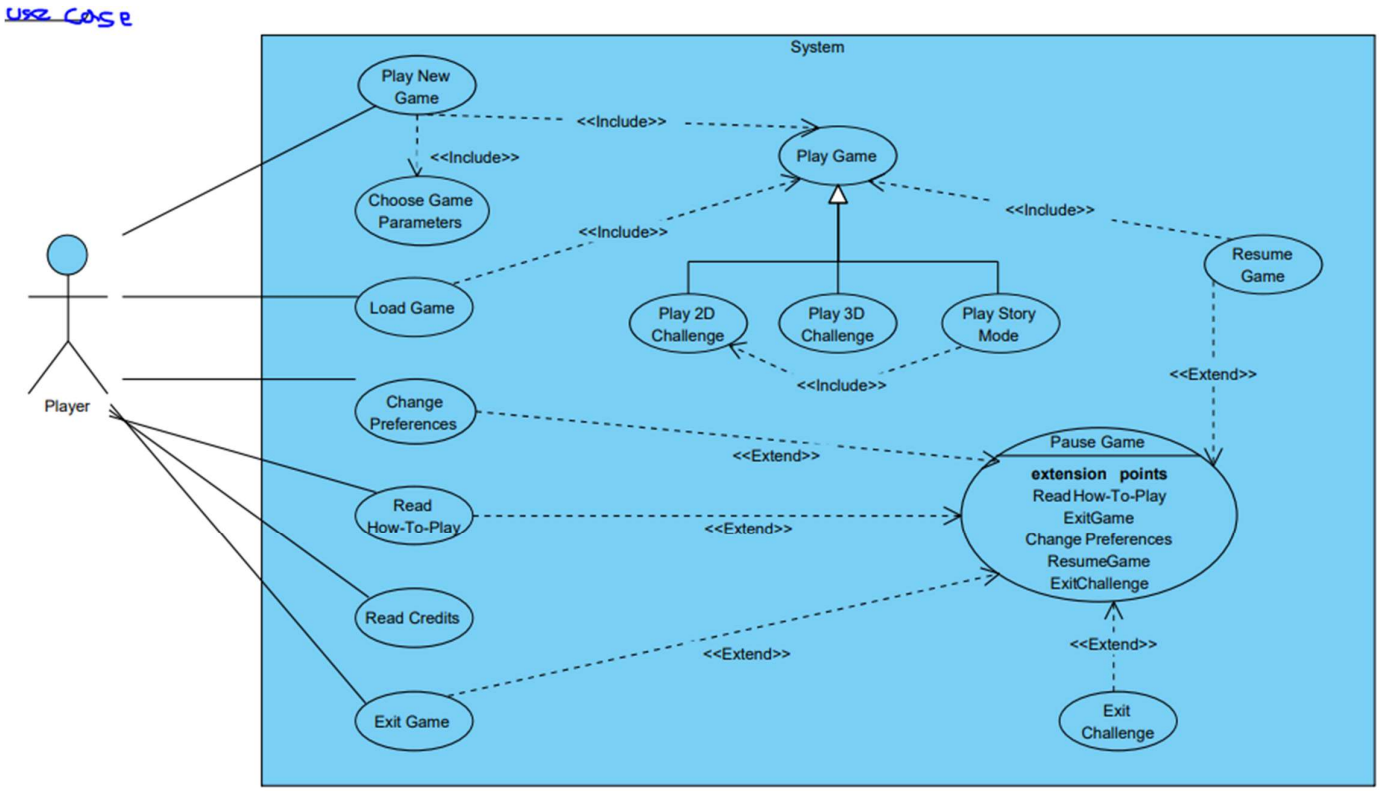
Considering that, the user interface plays an important role in making an impression upon the user, the game will be made to have a friendly and easily comprehensible user interface. The player can navigate their way throughout their play with ease and comfort. There will be icons present with captions of their functionality in clear writing. The player can access these icons on the home screen of the game.

4.4 Maintainability:

The game will be coded in a comprehensive and maintainable manner such that contributors who are new to the game will find it easy in helping improve the game. The codes will be commented in simple and articulate language with brevity which will explain the codes for easy understanding.

5. System Models:

5.1 Use Case Models:



Use case diagram textual descriptions

Play New Game
Use Case 1
Use Case : Play New Game
Primary Actor: Player

Entry Condition:
- Player must be in the main menu

Exit Condition:
- Player exits the game

Flow of events:
1. Player selects “Play New Game” button from the main menu
2. Player is directed to screen to choose the mode of game to play.

Choose Game Parameters:

Use Case 2
Use Case: Choose Game Parameters

Primary Actor: Player
Entry Condition:

- Player selects the “Start New Game” option from the MainMenu

Exit Condition:

- Player chooses all the parameters and decides to proceed to Play Game
- Player navigates back to the MainMenu.

Flow of events:

1. The Player chooses the GameType they want to play, from 2D, 3D 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.

    a. Alternative Flow: If the GameType is 3D, the player does not get to choose these parameters, they are fixed.

Play Game:

Use Case 3
Use Case: Play Game

Primary Actor: Player
Entry Condition:

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

Exit Condition:

- Player chooses to Pause the game
- 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. For every Piece that the Player places correctly on the Grid towards the recreation of the target Pattern, the Player’s Score is increased by one.

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 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.

    a. (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 UC2 (Choose Game Parameters). The GridSize is also decided by choices made in UC2.

Play 3D Challenge:

Use Case 5
Use Case: Play 3D Challenge

Primary Actor: Player
Entry Condition:

- Inherited from UC3 (Play Game) with GameType 3D chosen by Player in UC2 (Choose Game Parameters)

Exit Condition:

- Inherited from UC3 (Play Game)

Flow of events:

1. Basic flow of events is inherited from UC3(Play Game) with the following specifications: the GridType is Square, the target Pattern is always a 3-dimensional pyramid. GridSize is fixed.

Play Story Mode:

Use Case 6
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 3 Stories that are provided.

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

3. 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.

4. 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.

5. 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:

- 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 Challenge:

Use Case 9
Use Case: Exit Challenge

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. All the game modes are displayed from which the player can choose to resume the play

Change Preferences:

Use Case 11
Use Case: Change Preferences

Primary Actor: Player
Entry Condition:

- Player selects “Change Preferences” Button from the main menu
- Player selects “Change Preferences” Button from the pause menu

Exit Condition:

- Player goes back to main menu
- Player goes back to pause menu

Flow of events:

1. Player selects “Change Preferences” Button from the main menu
2. Player is directed to a new screen with containing items that can be modified to fit the player’s preference. Some of these items are sound, volume, theme etc.

Alternative Flow of events:

1. Player selects “Change Preferences” Button from the main menu
2. Player is directed to the change preference screen

Read How-To-Play:

Use Case 12
Use Case: Read How-To-Play

Primary Actor: Player
Entry Condition:

- Player selects “Read How-To-Play” Button from the main menu
- Player selects “Read How-To-Play” Button from the pause menu

Exit Condition:

- Player goes back to main menu
- Player goes back to pause menu

Flow of events:

3. Player selects “Read How-To-Play” Button from the main menu
4. Player is directed to a new screen with directions on how to play the game

Alternative Flow of events:

3. Player selects “Read How-To-Play” Button from the main menu
4. Player is directed to a new screen with directions on how to play the game

Read Credits:

Use Case 13
Use Case: Read Credits

Primary Actor: Player
Entry Condition:

- The Player chooses the Read Credits option from the PauseMenu
- The Player chooses the Read Credits option from the MainMenu

Exit Condition:

- The Player clicks the Back button
- The Player presses the Contact Developers button

Flow of events:

1. Credits roll on screen.

Exit Game:

Use Case 14
Use Case: Exit Game

Primary Actor: Player
Entry Condition:

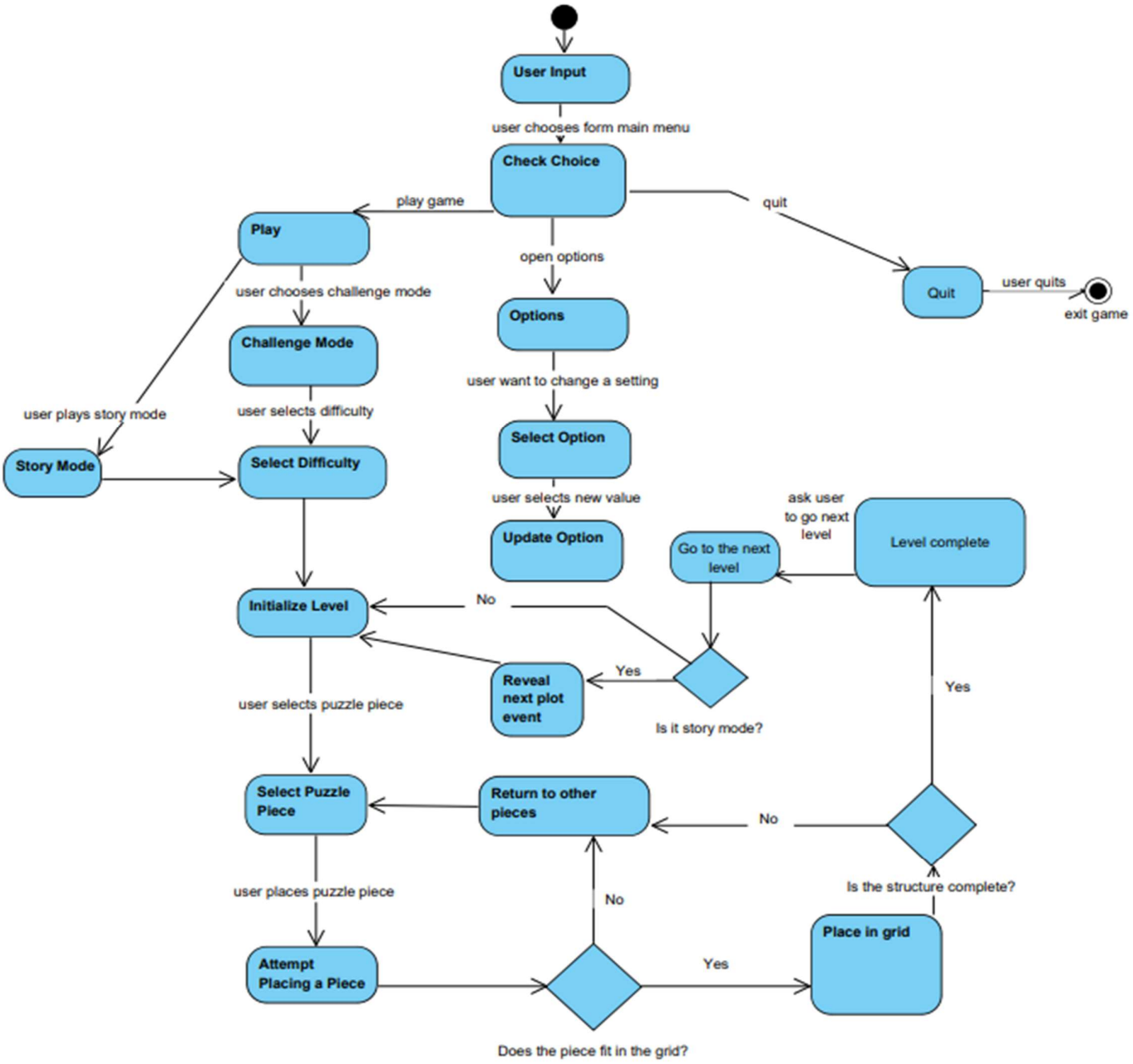
- The Player selects the Exit Game option from the MainMenu.

Exit Condition:

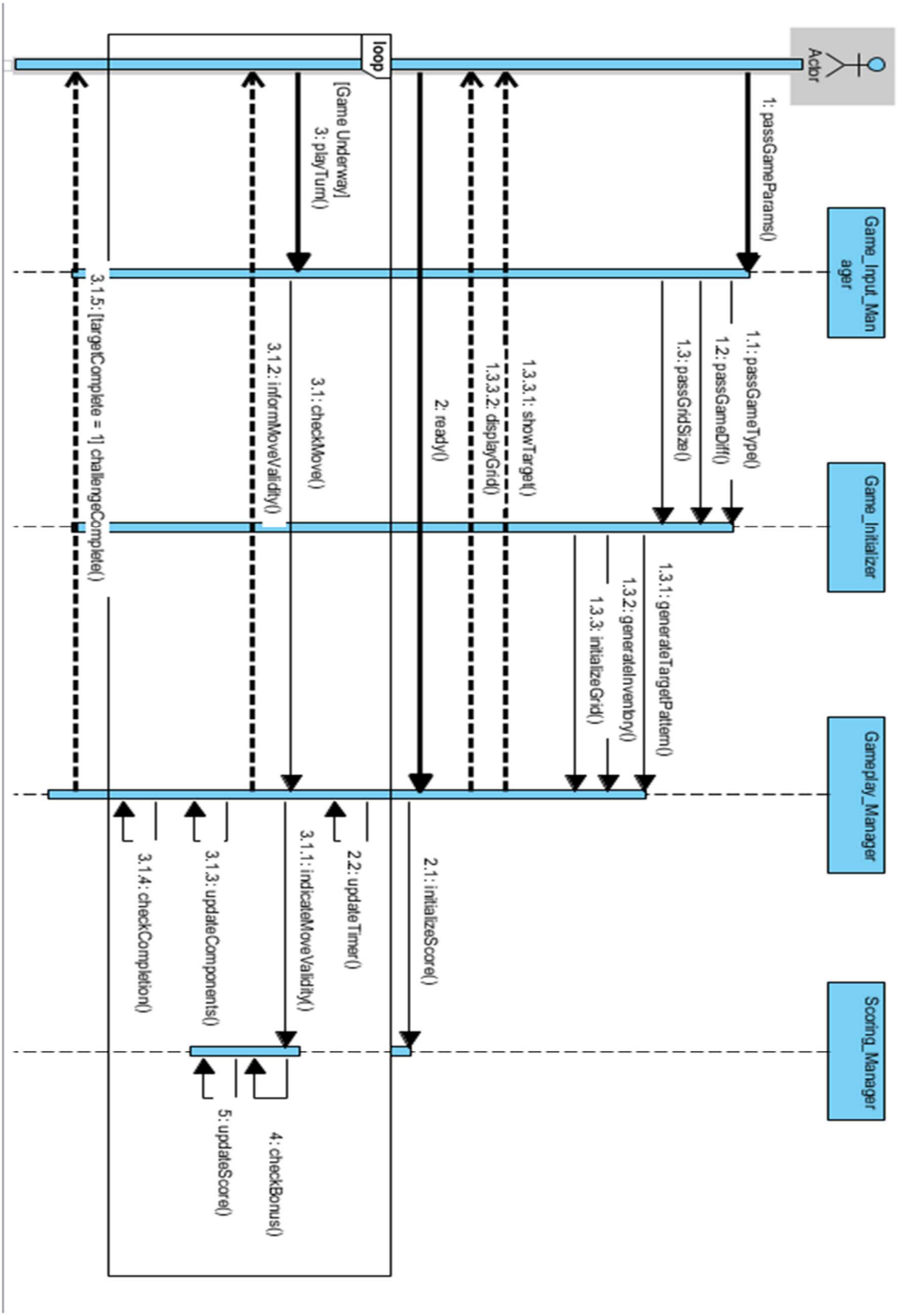
- The Game application is successfully terminated.

Flow of events:

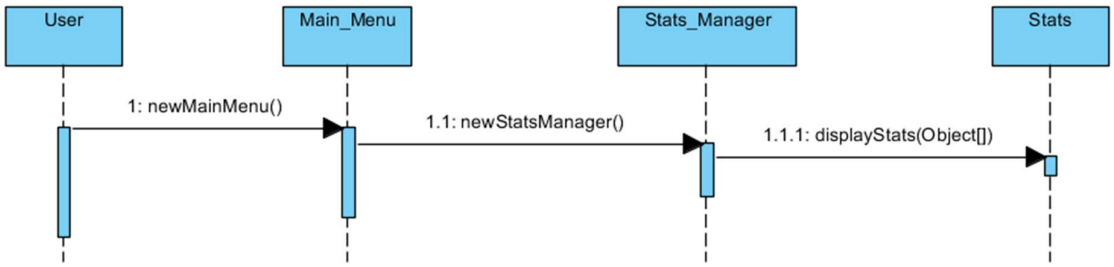
Activity Diagram:



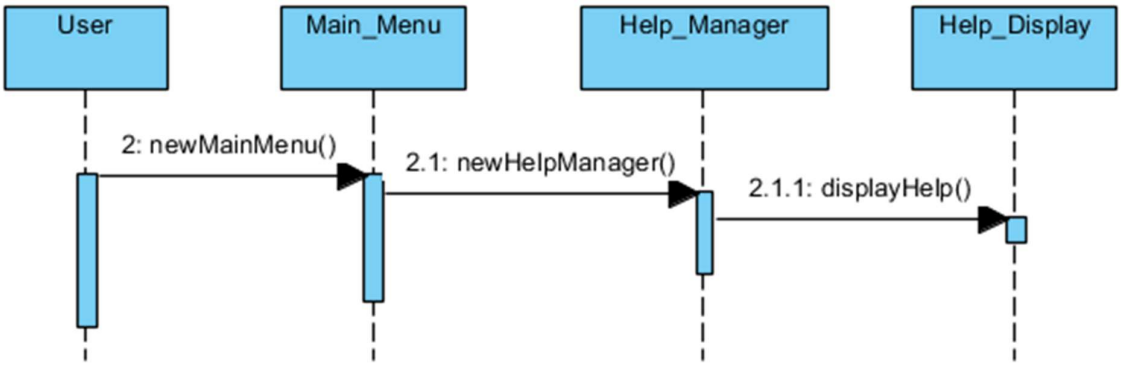
5.2 Sequences: Play Game Sequence Diagram:



Stats Sequence Diagram:

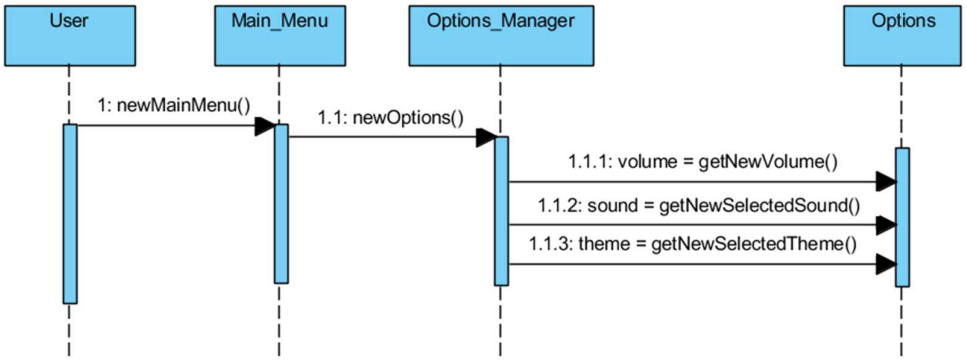


How to Play Sequence Diagram:

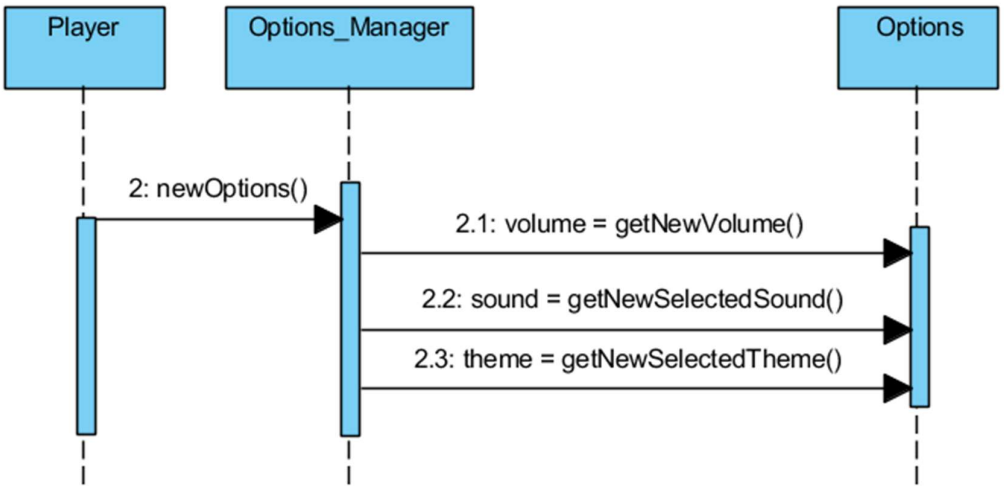


Options Sequence Diagram:

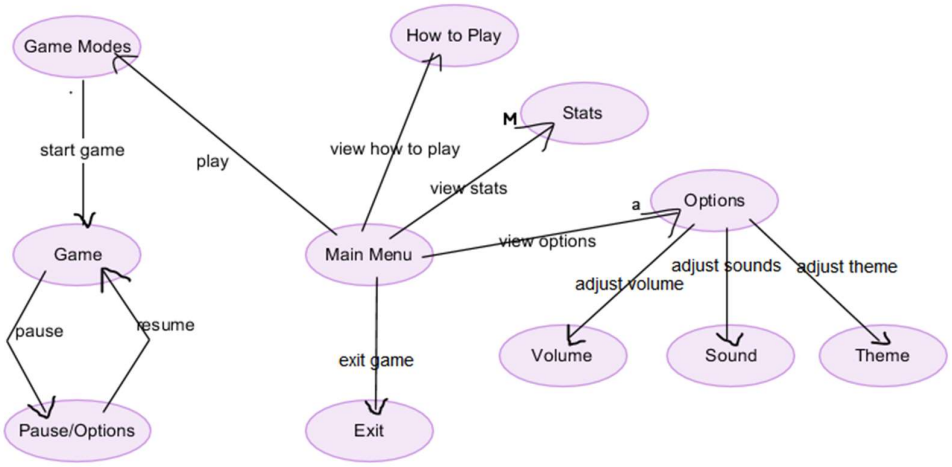
Main menu:



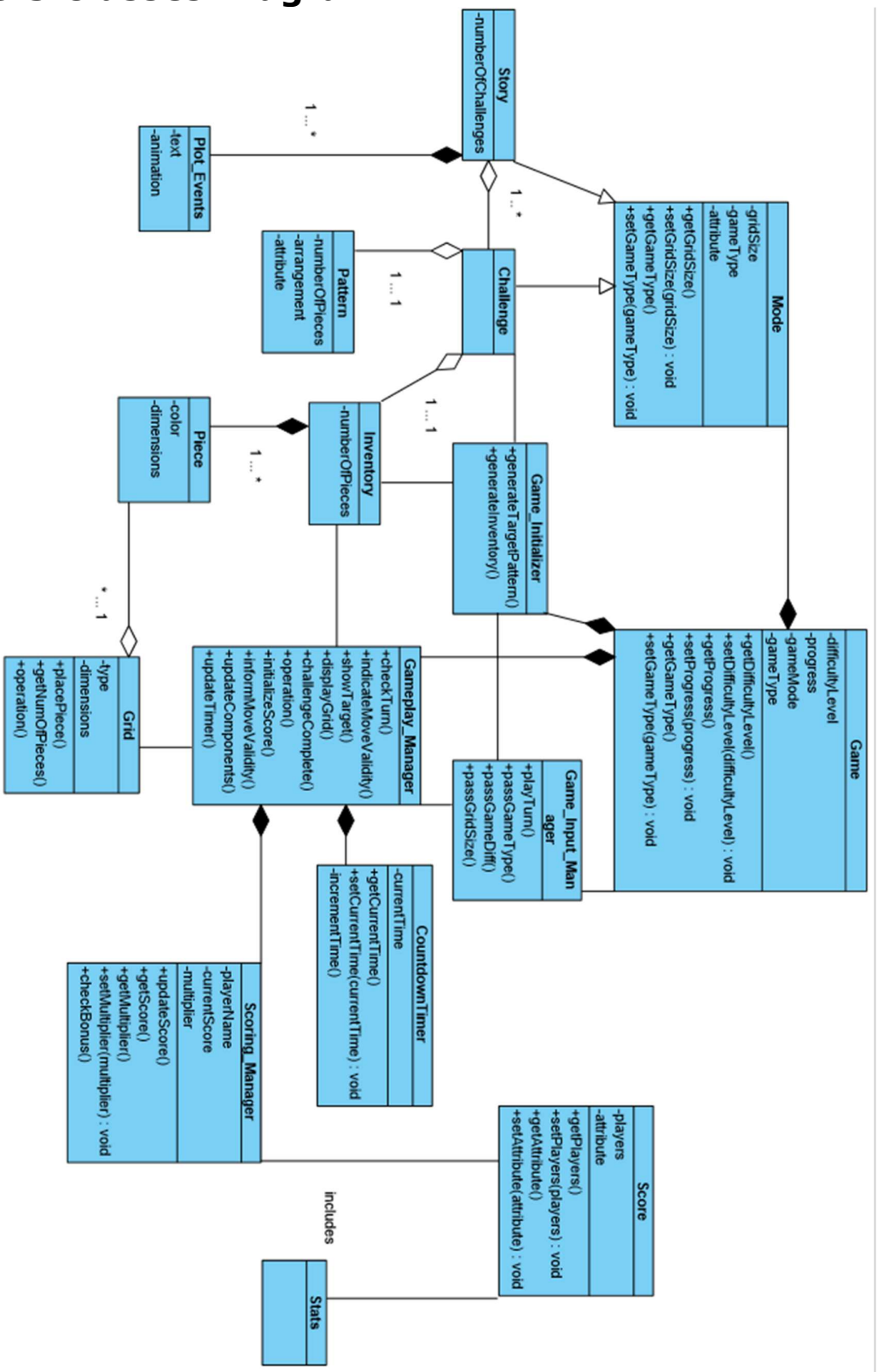
In-game:



Navigational Path:



5.3 Classes Diagram:



3 Glossary

Glossary for any domain-specific terms you use in your report.

4 References

[1] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.