

Requirements and Analysis Document for MemoryShape

Edenia Isaac
Filip Linde

Nils Johnsson
Kevin Svensson

Date: 2020-10-01

Version: 1

Memory Shape

Contents

Requirements and Analysis Document for MemoryShape

Contents

Introduction	3
1.1 Purpose of application	3
1.2 General characteristics of application	3
1.3 Scope of Application	3
1.4 Definitions, acronyms and abbreviations	4
Requirements	4
2.1 User Stories	4
2.2 Definition of Done	5
2.3 User Interface	6
Domain model	10
3.1 Class Responsibilities	10
3.1.1 Model	10
References	

1. Introduction

In this document we will analyze the project requirements and define what the users can expect from our application.

1.1 Purpose of application

The purpose of MemoryShape is to provide a fun, engaging and challenging single player game. The application aims not only to entertain the user, but also to exercise her memory. MemoryShape targets users of all ages, with its fun, colourful shapes and various levels of difficulty.

1.2 General characteristics of application

MemoryShape is a desktop application available for Windows, Mac and Linux. The application's title screen features the MemoryShape logo and functional buttons for starting a game, customising the gamemode, checking out the leaderboard, changing the settings and closing the application.

Starting a game will generate a board with a grid of cards, each featuring a shape with a specific colour. The user will get the chance to memorise the sequence of cards before they are flipped to face the board. One by one each of the combinations of shape and colour on the board will be displayed on the screen above the board. It is then up to the player to remember and match each of the cards being displayed with its counterpart in the grid. Choosing the wrong card will result in losing one of your lives. If the player gets all of the cards right before losing her last life, she will move onto the next, more difficult level.

The leaderboard consists of a list of the user's best runs, each spot featuring the player's name, score and reached level in order for the user to track her progress and motivate her to improve her memory. The leaderboard is also saved between sessions.

1.3 Scope of Application

MemoryShape is played using only the mouse cursor, with either a mouse or a trackpad. The application supports use by one player at a time, although it is possible to challenge your friends for the top spots on the leaderboards by changing the nickname of the player between runs. You can also team up and work together by simply cooperating during a run.

MemoryShape includes several options for adjusting difficulty, so that users of ages and capability can enjoy a challenge that suits them. The leaderboard and scoring system provide the user with motivation to keep trying and to improve. The sense of accomplishment experienced by the user for each level she clears brings excitement for the next one. MemoryShape is perfect for those trying to improve their memory or who just want to have fun killing some time.

1.4 Definitions, acronyms and abbreviations

- **Card** - A clickable figure that contains a shape
- **CardSelector** - A display of the next selected card for the player
- **Board** - The playfield with all the clickable cards and a CardSelector
- **MVC** - Model View Controller: A design pattern to structure the code on a large scale.
- **GUI** - Graphical User Interface: The visible part of the program for the user.
- **UML** - Unified Modeling Language: A diagram describing how different parts of the program work together.
- **Domain model** - A diagram describing the parts of the programs model.
- **User Stories** - A small story that describes the type of user, what they want and why.
- **Definition of done** - An agreement on when a user story is completed.

2. Requirements

2.1 User Stories

Functional Requirements

Implemented:

The player needs to be able to:

1 -*Start a new game*

- b . Be presented with a board with cards for him to remember
- c . Be presented with a Display card above the board that can be match with the board

cards

2 -*Play the game*

- a .Pick a card from the board
 - ii. see a new display card if its was a match
- b .Get points
 - i. score when it's a match
 - ii. see my remaining lives when i get it wrong
- d .Check the leaderboard
 - i. be presented with the 10 higher scores

Non Implemented:

1-

- a . Register himself as a player / change the current player

2-

a .

- i. see the cards content for a few seconds if it wasn't a match

c .Go to next level

- i. be presented with my current score, my highest score and leading player

score

ii. be presented with a new/bigger board (a more challenging version) when i have match all the cards

e .Customise the game

i. choose a difficulty level

ii. choose a game mode

f .Change settings

i. change the game music volume

3 -Exit the game

a.Go back to main menu

Non Functional Requirements

f.

ii. change the game effects volume

2.2 Definition of Done

Acceptance criteria for a user story to be defined as done :

- The implemented functionality should be successfully tested
- Possible bugs should had been fixed
- Possible exceptions should be handled
- Code has been reviewed by another development team member
- Possible merge conflicts should be resolved
- Finally the functionality should be push to our master branch

2.3 User Interface

Upon running the application the start screen will appear (as seen in figure 1). The view features several interactable buttons each of which will show a different user interface.

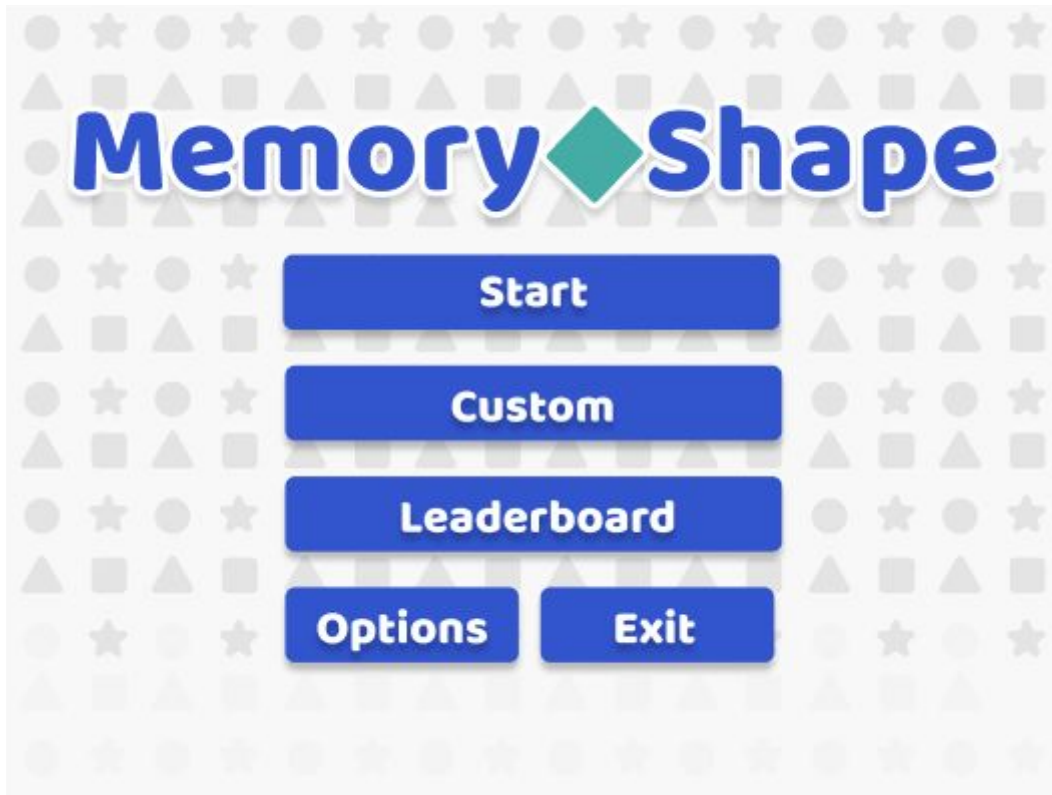


Figure 1: Sketch of the MemoryShape start screen.

Pressing the button labeled “Start” will start a new game, switching to a view with a board of cards (see figure 2). Clicking a face down card will flip it, revealing its face. The menu button in the top right corner brings out a popup window with menu options.



Figure 2: Sketch of the MemoryShape board screen.

Failing to get all the cards right before losing all of your lives will result in the game over menu popping up (as seen in figure 3). The menu features navigational buttons for going to a new game, the leaderboard and the main menu.



Figure 3: Sketch of the MemoryShape game over screen.

Pressing the button in the main menu labeled “Custom” will replace the view with a menu of game settings (as shown in figure 4). The button labeled “Back” will take the user back to the start screen, whereas the button labeled “Start” will start the game showing the board interface.

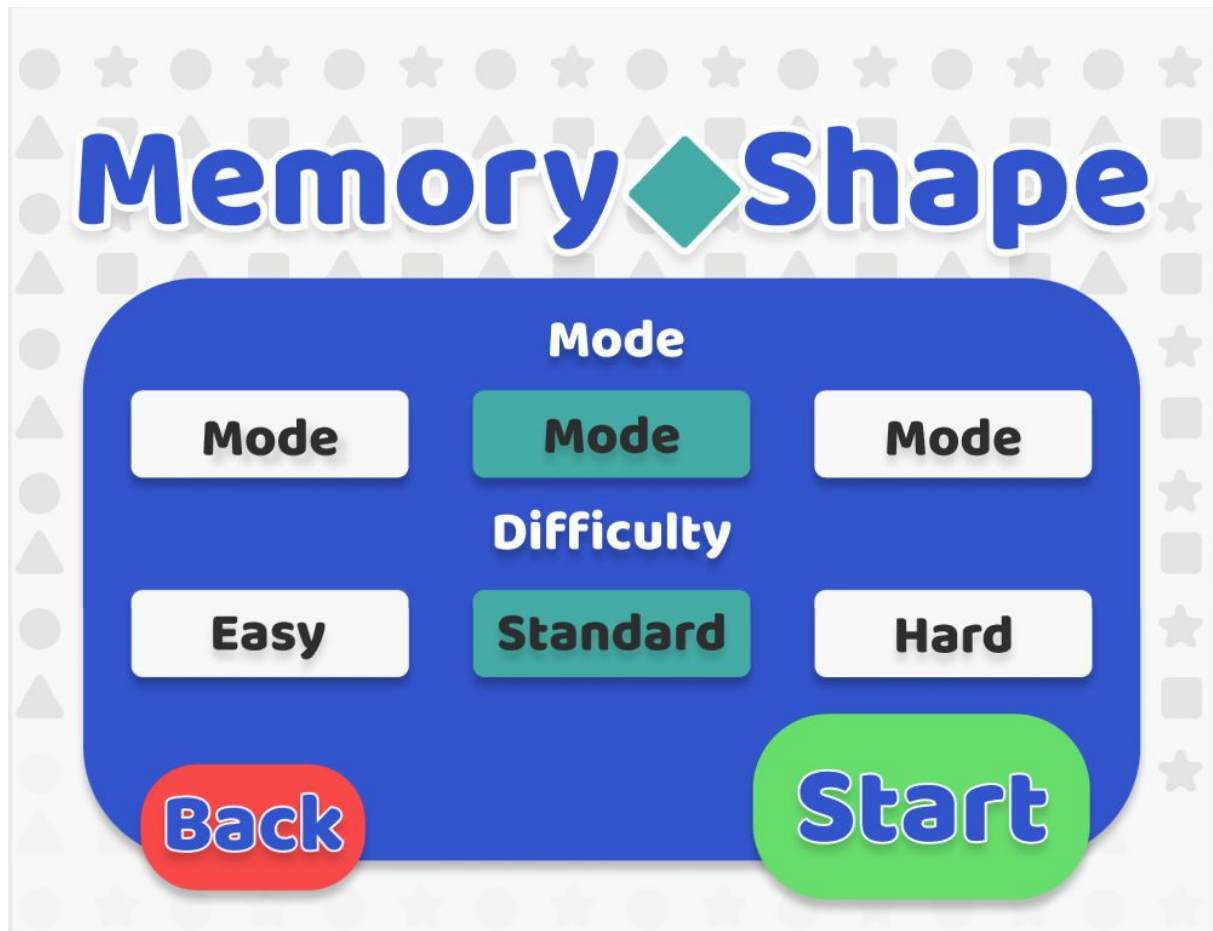


Figure 4: Sketch of the MemoryShape custom game menu.

Pressing the leaderboard button in the main menu will take the user to a list view of the best scores that have been achieved on the device. The button labeled “Exit” closes the application upon being clicked.

3. Domain model

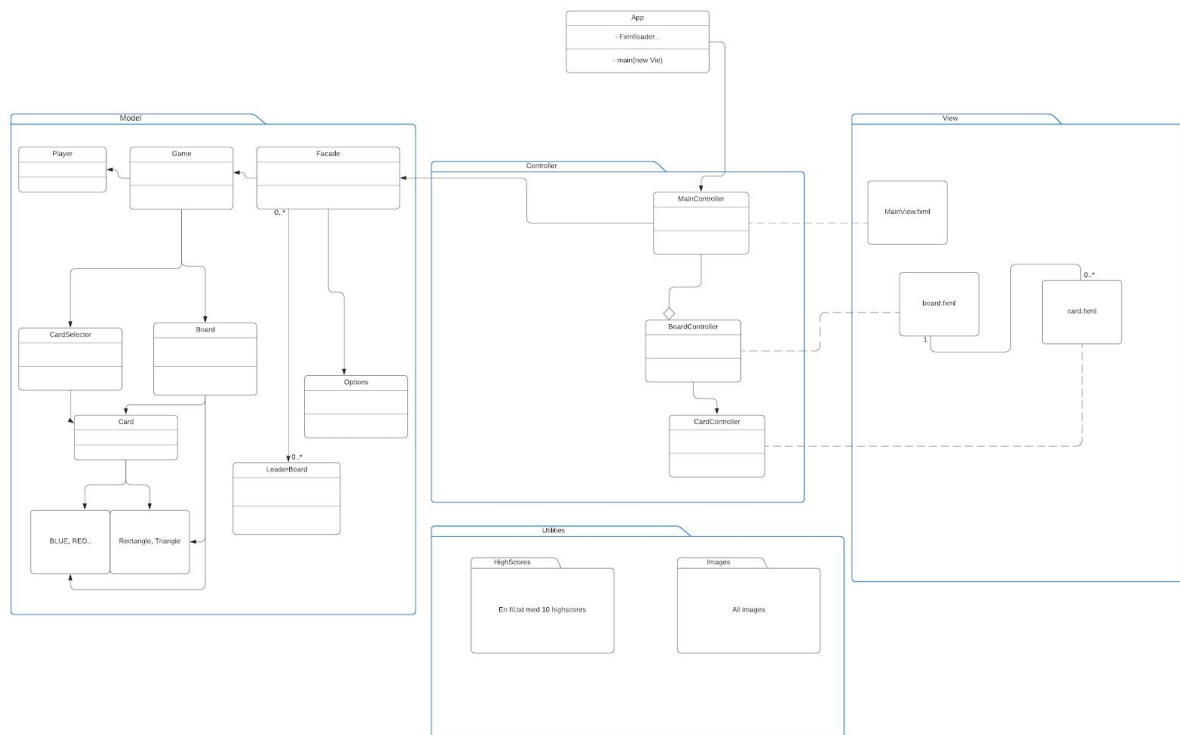


Figure 5: A UML diagram with the different packages and their contents.

3.1 Class Responsibilities

3.1.1 Model

Game - Contains all the parts required to run the gameplay-loop

Player - Contains the player related data.

Card - A clickable figure that contains a shape

CardSelector - A display of the next selected card for the player

Board - The playfield with all the clickable cards and a CardSelector

Leaderboard - Contains a list of saved highscores from previous runs.

Options - Holds information of the selected game settings.

Facade - A class that hides the complexity of the model and communicates with other packages.

MainController - Communicates information from the user to the model and is also a controller class to the mainView GUI

BoardController - Controller class to board.fxml

CardController - Controller class to card.fxml

App - Starts and runs the entire application.

4. References