Bilkent University

Department of Computer Engineering

CS 319 Term Project

Section 2

Group 2G, Oldies but Goldies

Q-bitz

Final Report

Project Group Members

1. Burak Kırımlı
2. Cansu Canan Ceyhan
3. Emre Keskin
4. Mert Çerçiler
5. Yağmur Özkök

Supervisor: Eray Tüzün

Table of Content

1. Introduction

We are working with the implementation since first iteration and finally our functional requirements are virtually completed. We divide functionalities at the very beginning yet during real implementation process everyone ended up helping one another and most of the classes has more than one person working on them. As a group, everyone is using eclipse and we are send each other what we have done in a certain class via e-mail.

As a result, we have fulfilled every promise we have decided on functional requirements phase. In addition to them, we have included some minor sound and image properties which makes our project a bit more desirable.

2. Design Changes

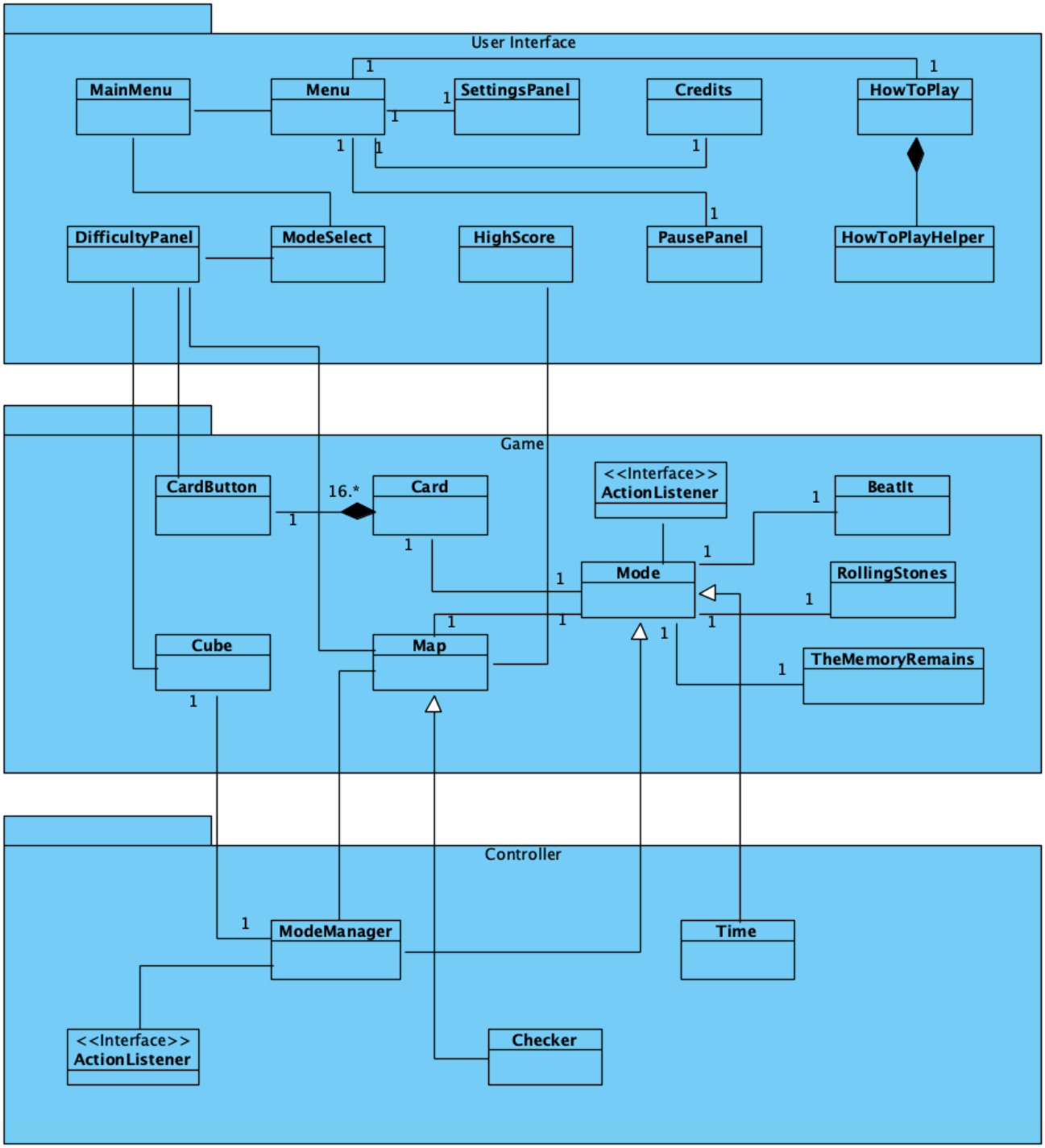
In the beginning of our implementation phase we have only focused on the functionalities that we have mentioned on the previous reports. As our project is finalizing we have decided to add certain gif effects when player make 3 or more accurate placements of cubes in our design report. Yet, when we added such gifs our screen looked a bit crowded and complicated therefore we decide to implement sound effects rather than gifs.

Now each move of the player leads a specific sound effect, right and wrong moves have different sound feedbacks as well as accurate moves in combos.

When we store combo value, we have decided to award combos. Normally with every right move player wins 1 points and if he/she makes 3 accurate moves in a row player wins 3 points for each right move until he/she makes a wrong move.

Moreover, in each puzzle there is a randomly selected bonus cube and when the player puts that cube accurately 20 seconds are added to remaining time.

In addition to these features, we have given player option to press keyboard button “p” instead of clicking put button every time he/she desired to put a cube in to map. With all these changes there are some modifications in our low-level design so we have updated the class diagram as given below. Our model view controller architecture(MVC) is not altered much, only sound Manager and file manager classes are gone since we do not need them in our projects final version. Instead we have time, menu, how to play, how to play helper and credits which were not mentioned in the design report.


User Interface
MainMenu
Menu
SettingsPanel
Credits
HowToPlay
DifficultyPanel
ModeSelect
HighScore
PausePanel
HowToPlayHelper
Game
CardButton
Card
16.*
<<Interface>>
ActionListener
BeatIt
Mode
RollingStones
Cube
Map
TheMemoryRemains
Controller
ModeManager
Time
<<Interface>>
ActionListener
Checker

3. Lessons Learnt

As a group before implementation process we have done requirements and analysis part regarding how we will manage this project. If it was not for this lecture, we would not have done this part conscientiously and probably face difficulties regarding details of implementation. In the end, we have experienced a complete lifecycle of an application.

During beginning of the implementation phase while we are trying to combine what everyone has written so far, we have once again realized the importance of object-oriented programming. Since there are lots of buttons and panel, names of such variables tend to confuse others and we need to ask each other what that particular piece of code accomplish. Therefore, we can easily say that our group cooperation and communication skills have increased after this project.

4. User's Guide

4.1 System Requirements & Installation

Graphical user interface (view part) of the Project will be implemented using JAVAFX libraries. Therefore, our software requirement is Java Development Kit(JDK) 8 or any newer version of JDK, so that our used libraries can be executed properly.

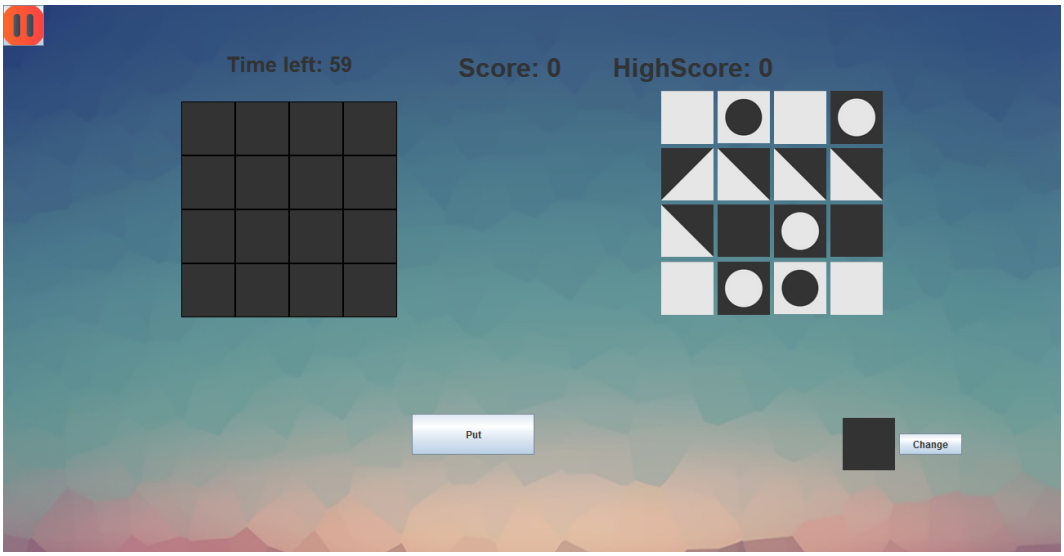
4.2 User's Install Guide

- Enter to the link below to see our reports and projects
- Download the project as a ZIP file and run it in any Java Compiler (for example: Eclipse etc.)
- In order to learn how to play the game, users can use "How to Play" option
- More information about the game, our second iteration reports will help.

4.2.1 Play Game

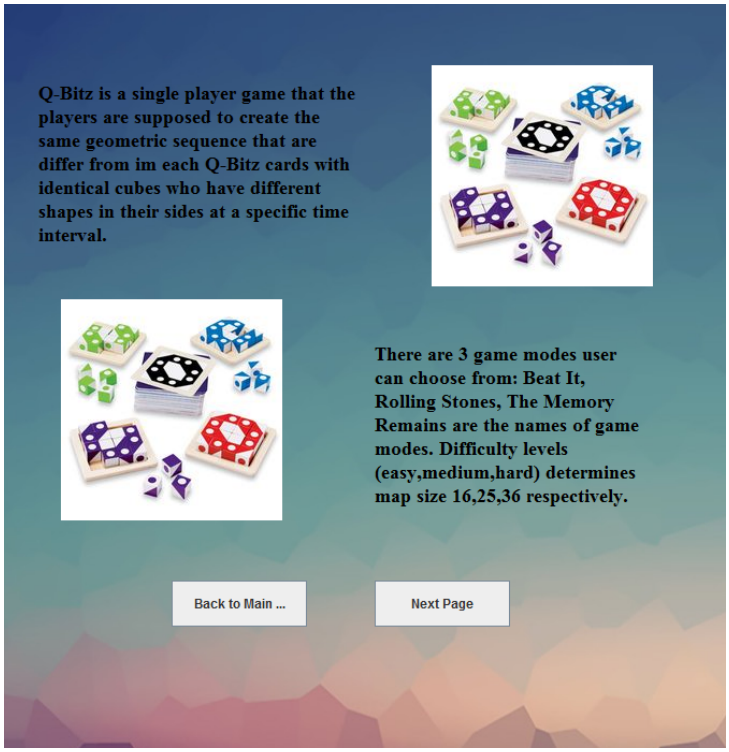
1. Player chooses "Play Game" button from main menu.
2. Player chooses mod which are "Beat it", "Rolling Stones" and "The Memory Remains".
3. Player chooses level among easy, medium and hard.
4. Player is represented a puzzle and a cube.
5. Player presses "Change" button if he/she desires to change side of given cube.
6. Player presses "Put" button then click desired space to put the cube in any place on map.
7. Player can know he/she is successful if the cube remains in place.
8. Player solves the puzzle.

9. Player is presented with another puzzle.
10. Steps 4-9 until time limit is up.
11. Player sees his/her score on screen.



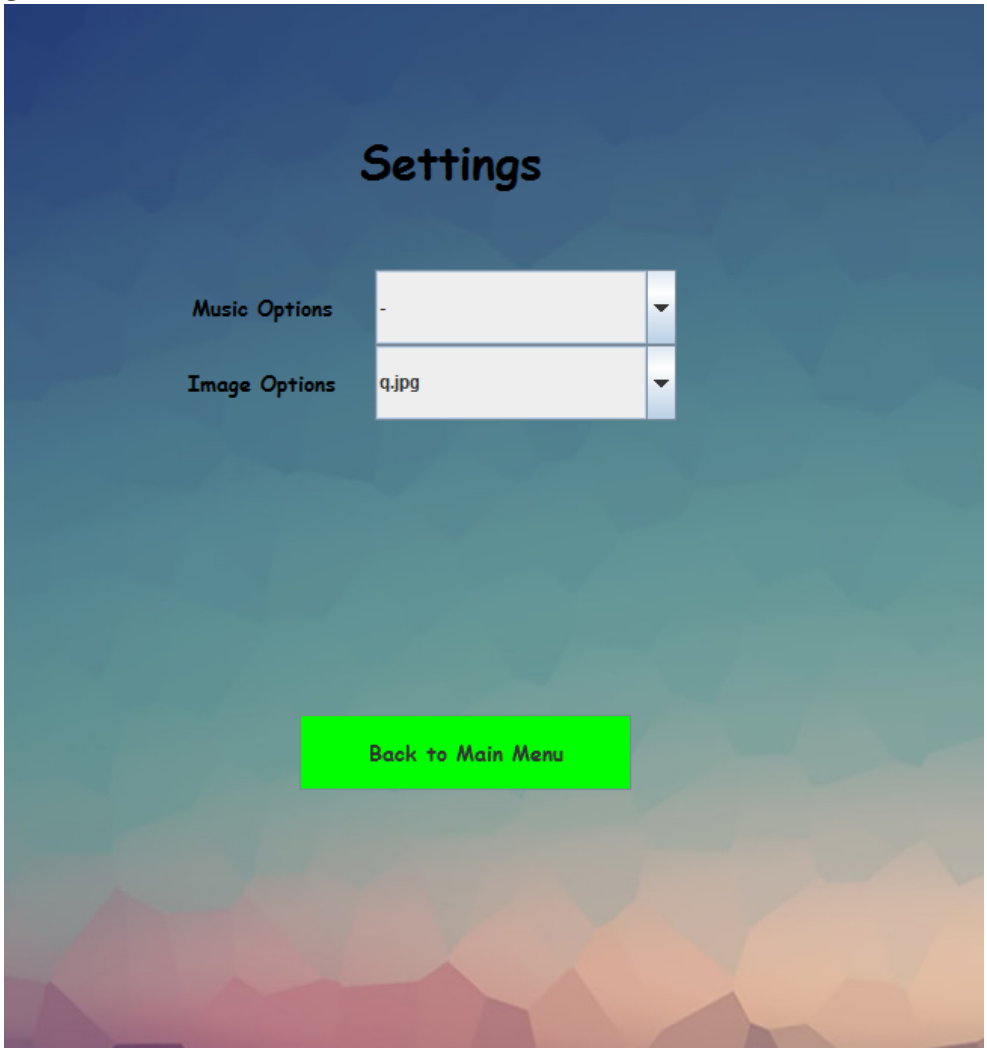
4.2.2 How to Play

1. Player chooses “How to Play” button from the main menu.
2. Player reads the description of the game and familiarize himself/herself with the game.
3. Player chooses “Back to Main Menu” and returns to main menu ready to play the game.



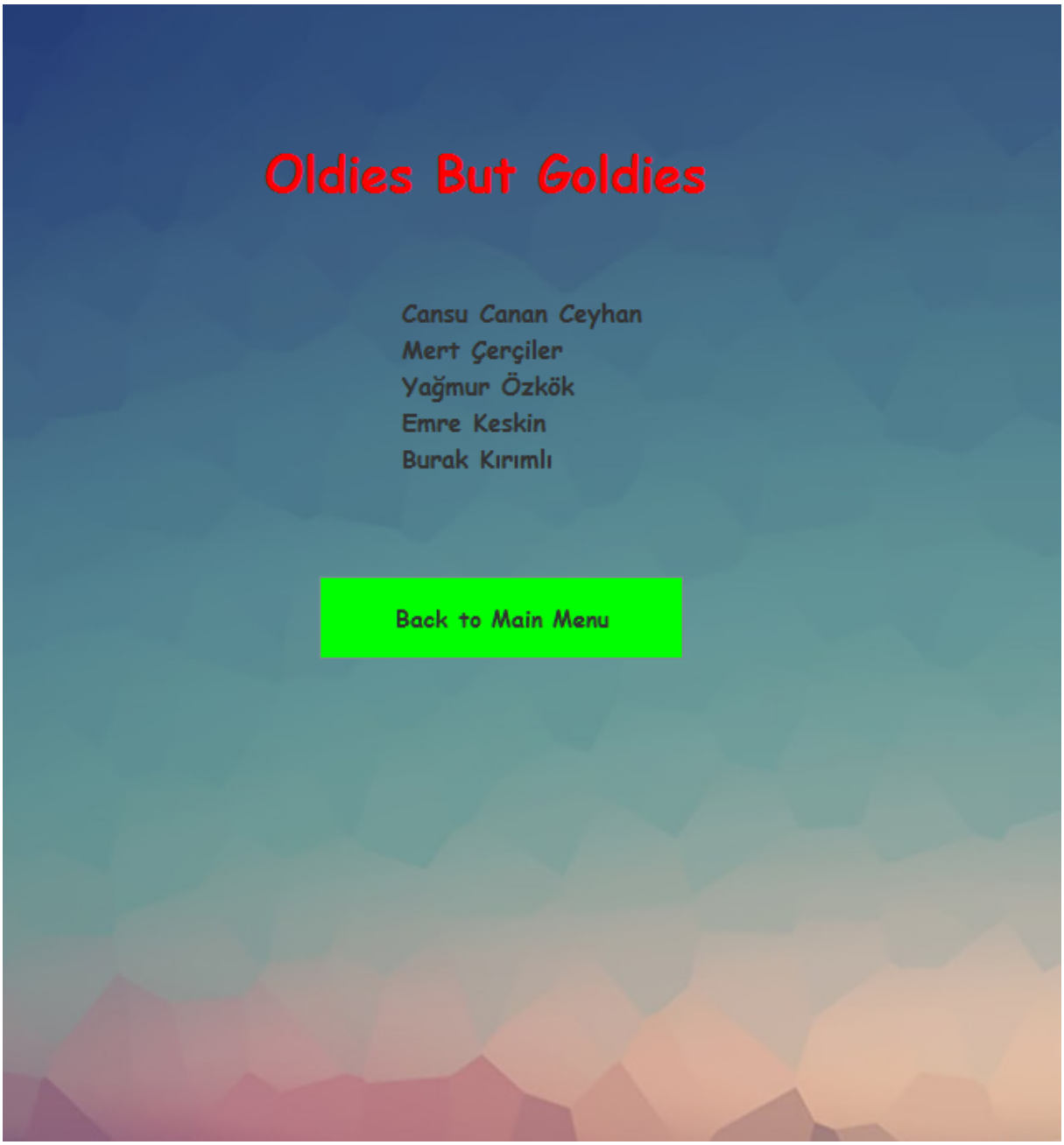
4.2.3 Settings

1. Player chooses “Settings” button from the main menu.
2. Player changes background color of the game.
3. Player changes music playing in the background of the game.
4. Player chooses “Back to Main Menu” and returns to main menu ready to play the game.



4.2.4 Credits

1. Player chooses “Credits” button from the main menu to see who have implemented this game.



5. User's Guide

- Every report rather than final iteration 2 is done with cooperation of everyone in the team.

1. Yağmur Özkök
    - GUI Panel transitions
    - Map class
    - Mode class
    - Card class
    - Button Performances
    - Helping design errors
    - HowToPlayHelper class
2. Cansu Canan Ceyhan
    - Time class
    - Final Report (iteration 2)
    - Demo video
3. Emre Keskin
    - Settings class
    - Credits class
    - HowToPlay class
    - Sound effects
    - Pause Panel
    - Final Report (iteration 2)
    - Background Image
4. Mert Çerçiler
    - Map class
    - Mode class
    - Card class
    - ModeManager class
    - Rolling Stones/Memory Remains class
    - Cube class
    - Keyboard Control
    - Checker class
    - CardButton class
5. Burak Kırımlı
    - MainMenu class
    - ModeSelect class
    - DifficultySelect Screen
    - HighScore algorithm
    - Score Bonus(combo) algorithm
    - HowToPlay class
    - Sound effects