

Object Oriented Software Engineering

CS319

Term Project Final Report

Kubitz

Group No: 3F

Group Name: Wasted Potentials

Mertkan Akkus 21602951

Yasin Alptekin Ay 21601849

Ahmet Furkan Bıyık 21501084

Ramazan Mert Cinar 21601985

Yaman Yagiz Tasbag 21601639

Introduction	3
Implemented Functionalities	3
Design Changes	4
3.1 Low Level Changes	4
3.2 Gameplay Changes	4
Lessons Learnt	5
User's Guide	5
5.1 System Requirements & Installation	5
5.2 How to Use	6
5.2.1 Selecting Multiplayer Game Mode	6
5.2.1.1 Joining Lobby	7
5.2.1.2 Creating Lobby	7
5.2.1.3 Lobby Screen	7
5.2.2 Selecting Singleplayer (Survival / Daily Challenge) Game	8
5.2.3 Playing The Game	8
5.2.4 Guide for Settings	9

1. Introduction

We have started the implementation of Kubitz when we were done with the design phase. All the group members used the IntelliJ IDEA IDE to implement the different parts of the project. We split the work in basically three sections which are server-side, main logic of the game and graphic user interface. While these three main parts are being implemented by different group members, github has been used by us to track the work done. The implemented functionalities, design changes, what we have learnt and a simple user guide will be provided in the rest of this report.

2. Implemented Functionalities

We have aimed to implement the high priority functions of the game and a User Interface that will support these basic functions in the first iteration. Right now, our system provides the functionalities below:

- Configuring graphic and sound settings. User specified settings are saved to a file and read from the file.
- Navigation through menus, each menu interaction is implemented.
- Tutorials for each of the game modes.
- Countdown time controller to be used in Survival game mode.
- Lobby system with all functionalities except chat and invite feature.
- Implementation of all the game modes.
- Server controllers and database usage.

3. Design Changes

At first we planned how to design the project and then we moved on to write the design and analysis reports according to these reports. However, when we started to implement the project we saw that the design is not completely accurate. Therefore, we decided to redesign and improve the system. The changes which occurred after the design phase are stated in the sections below.

3.1 Low Level Changes

Some of the classes were not sufficient to implement the project so we have changed some of the classes and added new classes. For instance we did not have a lobby class at our structure design stage. However we needed to implement such a class so that implementing the UI and would be easier. In addition, we changed some classes' constructors since we need some properties in objects for example we added Config parameter to SettingsScreen's to get or set configuration. We also added new methods in some objects in order to improve the code in terms of readability and reliability such as initializeResources method to screens to make constructor be shorter.

3.2 Gameplay Changes

Some of the game modes were not solid at the design state. Such as, how many game cards will be given to the users as objective in the classical game mode, how many seconds are needed to switch the boards in the switch game mode and how many extra seconds will be given as a bonus to the players after a card has been solved in survival mode. Some minor changes have also been applied such as the theme song of the game. However, since this is a low priority task it can still be implemented later.

4. Lessons Learnt

During both implementation and design phase, we have encountered some problems about being synchronized and learned how to overcome those by experiencing it. First of all, it was a challenge for us to be available at the same time to arrange a meeting and use that time efficiently. Therefore we have decided about the issues to tackle prior to meeting and also about the dedicated time for the meeting. It was another task for to divide the project into subtasks and distribute them amongst the team members. We have realized that there were some overlapping subtasks but still, we have managed to handle them and also learned to do better.

To be synchronized, we are using GitHub as a git service and using many functionalities of it. For instance, we are using the issue tracker and pull request system of the git service. We are adding the issues of any type, e.g. bug, error, to the system and mark them as resolved when it is fixed. We are creating a branch for major features and merge them with the master branch. We believe that those functionalities of git service increases the productivity.

Hence, we think that teamwork is much faster and professional when it can be managed. The team members should be communicating frequently for a better product.

5. User's Guide

5.1 System Requirements & Installation

The only need to play Kubitz game is Java runtime environment independently from the operating system. Since our game occupies so little space, players shouldn't have concerns about memory. For the users who do not have Java installed to their system can

follow this link <https://www.java.com/tr/download/> to download Java. After the installation, the only thing that users need to do is executing the “Kubitz.jar” file by double clicking it.

5.2 How to Use

5.2.1 Selecting Multiplayer Game Mode

To play multiplayer game modes, a user needs to navigate to play menu by clicking the “Play” button in the main menu. This will take the user to play menu in which player will choose multiplayer or singleplayer. After clicking “Multiplayer” button, the user will be taken to Lobbies screen. After this point, the user can either choose the multiplayer game mode by choosing the lobby with the game mode from the lobby list or create a lobby. In lobbies screen, player can also filter lobbies. By pressing “Filter” button, the player will be taken to lobby filter page in which he/she can specify desired lobby features. Lobbies screen can be seen in the figure 1.1.

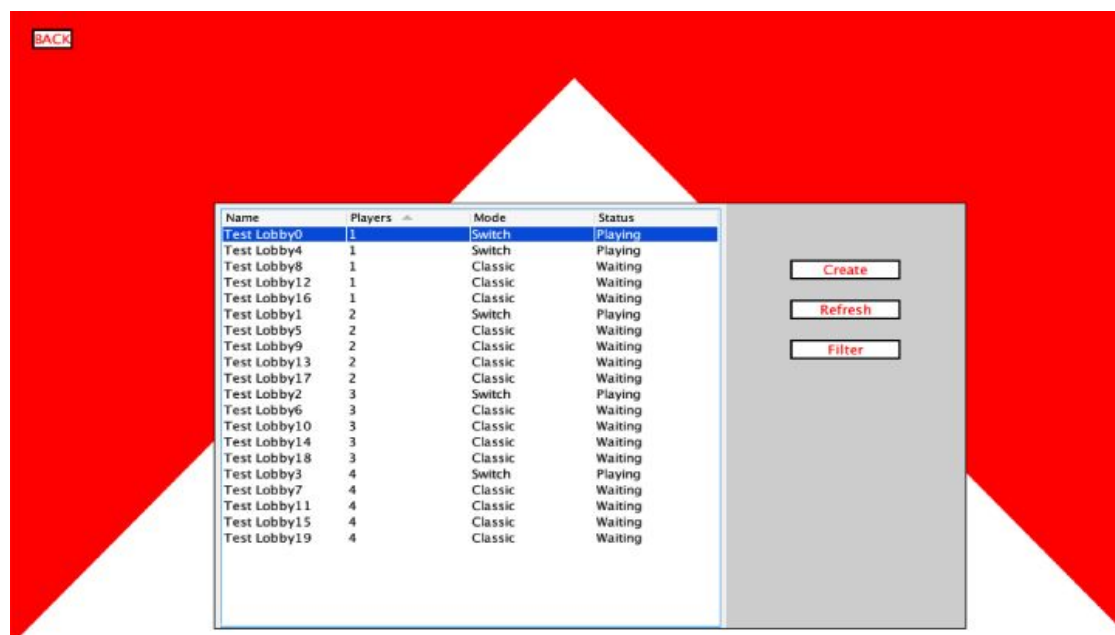


Figure 1.1: Lobbies Screen

5.2.1.1 Joining Lobby

In order to create a lobby the precondition is that the player must be in lobbies screen for which the instructions are given in section 5.2.1. For joining a lobby, the player should only double click on a lobby which he/she wants to join. This will take the player to the lobby screen.

5.2.1.2 Creating Lobby

In order to create a lobby the precondition is that the player must be in lobbies screen for which the instructions are given in section 5.2.1. To create a lobby the player should click on “Create” button in the lobbies screen which will take her/him to lobby creation screen. In this screen, player must enter a name for the lobby, choose the game mode (from switch and classic game modes), set the lobby size (for switch mode maximum is 2, for classic it is 4). If the player checks the “Private Game” checkbox, other players will not be able to join the player’s lobby from the lobbies screen thus, players will be able to join the lobby only via invites that are sent by any player in that lobby. After clicking “Create Lobby” button, the user will be taken to the lobby screen

5.2.1.3 Lobby Screen

This screen is displayed after the players joins a lobby either by creating or selecting from the list. There are 3 functions that can be used by the lobby admin (creator of the lobby) and one function that every player can use. The lobby admin can start the game by clicking “Start” button and kick a player by choosing the player by clicking on his/her name and then clicking “Kick” button. The lobby admin can also change the lobby settings by clicking “Settings” button. This action will take the player to lobby settings screen which is similar to lobby creation screen (see 5.2.1.2 for lobby creation instructions). Each player in

the lobby can invite another player by typing his/her name to the textfield. After clicking the “Invite button”, an invitation to the given player will be sent. There is also a simple chat system implemented in the lobby screen. Players can enter a message and send by pressing enter button on their keyboard.

5.2.2 Selecting Singleplayer (Survival / Daily Challenge) Game

To play singleplayer game modes, a user needs to navigate to play menu by clicking the “Play” button in the main menu. This will take the user to play menu in which player will choose multiplayer or singleplayer. A player should press the Survival or Daily Challenge button to play a singleplayer game. The game screens appears immediately after pressing the related buttons. See section 5.2.3 for how to play the game.

5.2.3 Playing The Game

Players will be on the play screen after the game starts. At the top right corner of this screen there is a game card as the objective, a cube sample at the middle left and the grid at the middle right. Players should arrange the cubes in such an order that will match the game card above to complete the objective.

The arrangement of the cubes can be done by the Q, W, E, A, S, D keys from the keyboard. By hitting the W, A, S, D keys user can change the faces of the cube up, left, down, right respectively and by hitting the Q and E keys player can rotate the cube left and right respectively.

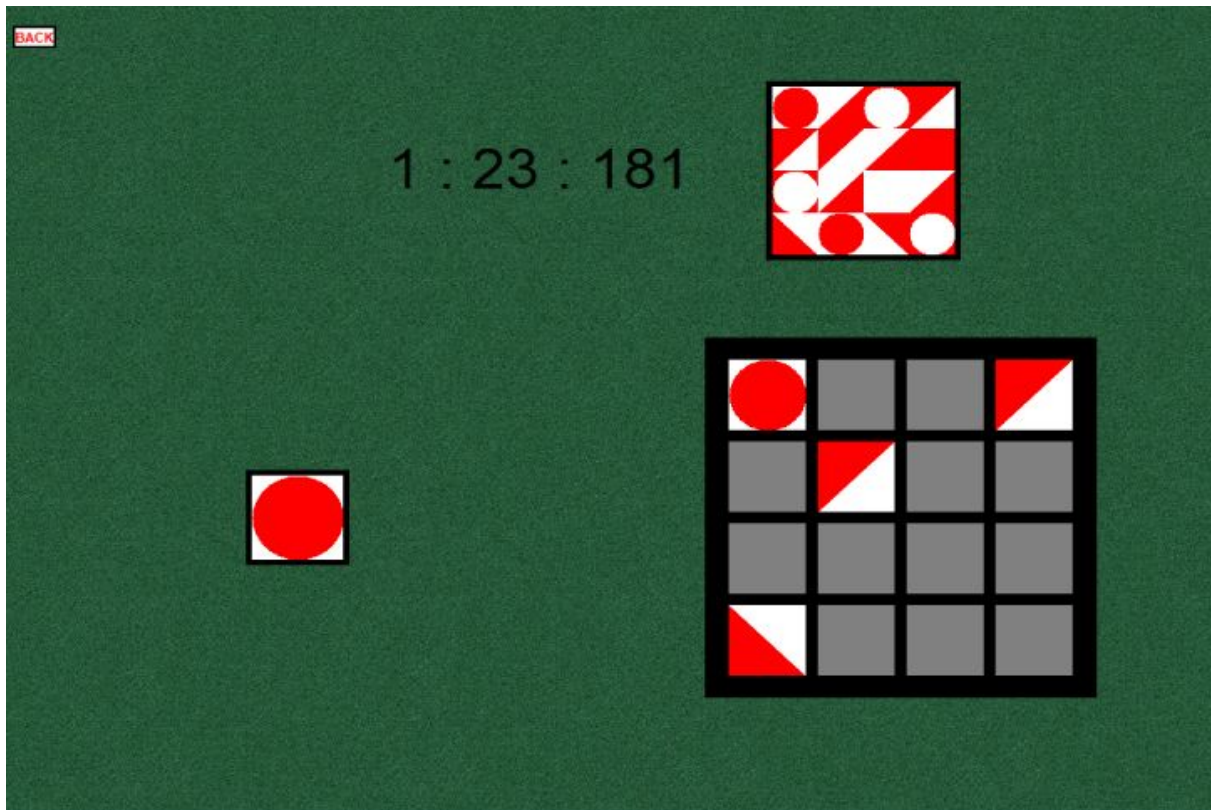


Figure 1.2:sample Gameplay

5.2.4 Guide for Settings

Players can access settings screen by clicking settings button in the main menu. In the settings screen, players can change the randomly generated nicknames by entering a nickname and clicking submit button after writing the desired nickname.

Another setting that players can arrange is the resolution of the game. Below the Graphics section player should select the suitable resolution from the drop down button, they can also select the Full Screen option.

The last part of the settings screen is the sound options, which includes the master volume, effects volume, music volume. Player who wants to configure the sound should scroll the desired volume bar - scroll right to volume up, left to volume down.

After completing configuring the settings, player must save the settings or click cancel to keep the settings as it was before.