Bilkent University

Department of Computer Engineering

Object Oriented Software Engineering Project

CS319 Project Group 1G: Super Mario Bros.

Final Report

Ahmet Çandıroğlu, Unas Sikandar Butt, Umut Balkan, Mert Sezer

Course Instructor: Uğur Doğrusöz
Course Assistant: Hasan Balcı

Progress/ Iteration 2

Dec 16 2017

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfilment of the requirements of the Object Oriented Programming course CS319/1

Contents

1 Introduction

Super Mario is a classic game known to almost every person from the ‘90s. The main purpose of the system is to provide a similar game, with minor changes. The game can be classified as an adventure and level based game. Comparing to current standards of gaming, our system may be lacking in the graphics and complexity departments, nonetheless the game will provide a challenging gameplay which is addictive and really enjoyable. The difficulty level of the game is intentionally going to be kept hard so that the game can become a platform for gamers to compete on.

2 System Requirements and Installation

This game will require no installation other than Java Virtual Machine which is installed to billions of computers. As game has been developed in a Java environment, it will be distributed as a Java archive (.jar) file. Thus the user will not have the hassle of going through installation wizards.

As for the system requirements, being a .jar file the system must have a compatible java environment. Also as for the hardware requirements, the game will run on any average machine since there is not much CPU or GPU needed game to run smoothly.

3 Changes to the Design

There is one addition to our design to make the project more suitable for the design patterns that we planned to use.

MapManager class is added during implementation. This class is a controller class to manage map works like collision checks, location updates etc. We did not planned to add this during design phase. However it turned out that we needed a separate controller for map management. Originally map works were handled

in *GameEngine* and *Map* class which damages architecture (MVC) and design patterns. However, this change does not change any functionality since we only moved functions from *GameEngine* to *MapManager*.

MapSelection and *MapSelectionItem* classes are added. These classes are created during implementation for new added feature during second iteration. We did not include these classes to our design however, it is better for us (developers) to have separate classes to represent map selection feature rather than doing it in *UIManager* class.

4 User's Manual

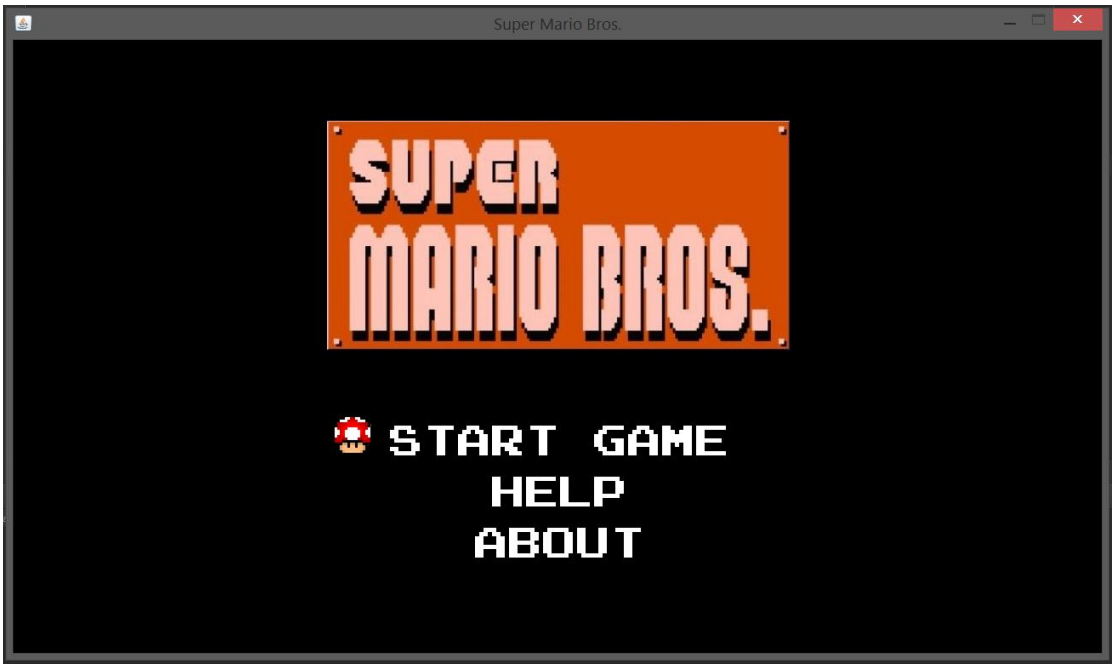

Figure 1 Start screen

Opening the game:

1. Navigate to the directory where the game is kept.
2. Run the Super Mario.jar file.

User will see this screen (figure 1) when he/she first opens the application by double clicking on the icon.

User may select one of “Start Game”, “Help” or “About” by navigating between selection items by using “UP” and “DOWN” keys and “ENTER” to select.

Figure 2 Help Screen

Help screen shows which keys should be used to do certain things. This screen can be opened by selecting “Help” in “Start Screen”.

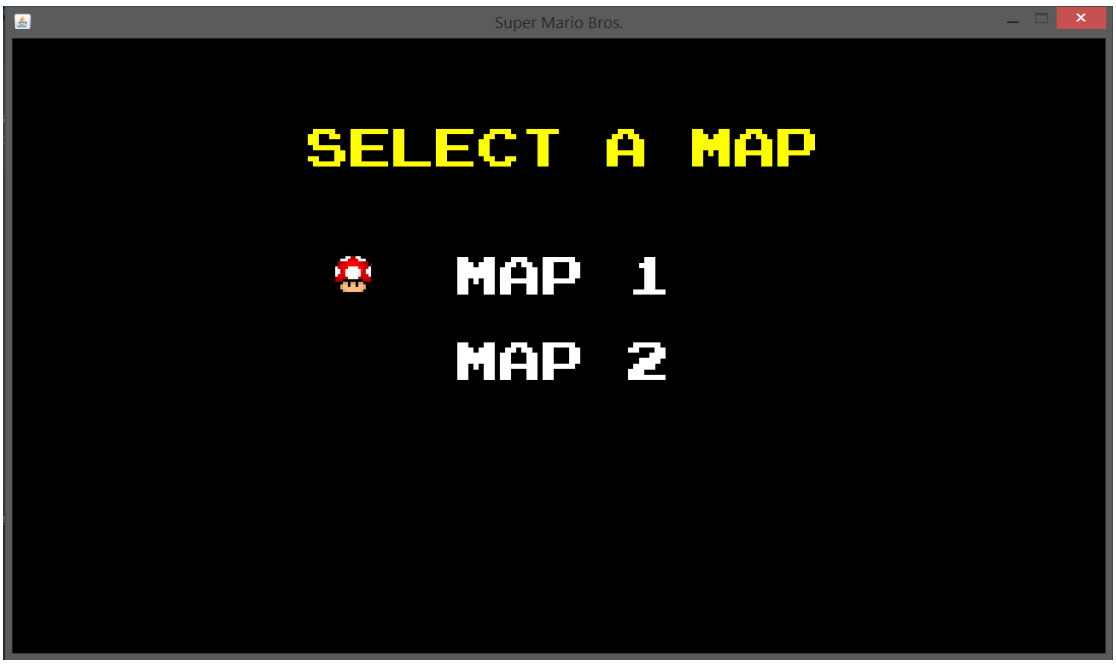

Figure 3 Map Selection Screen

In order to play the game user must select the “Start Game” selection. Then, user will see this screen (figure 3) which is the “Map Selection Screen”. User may select

any map he/she wishes by clicking on the option or using keys as in “Start Screen”.

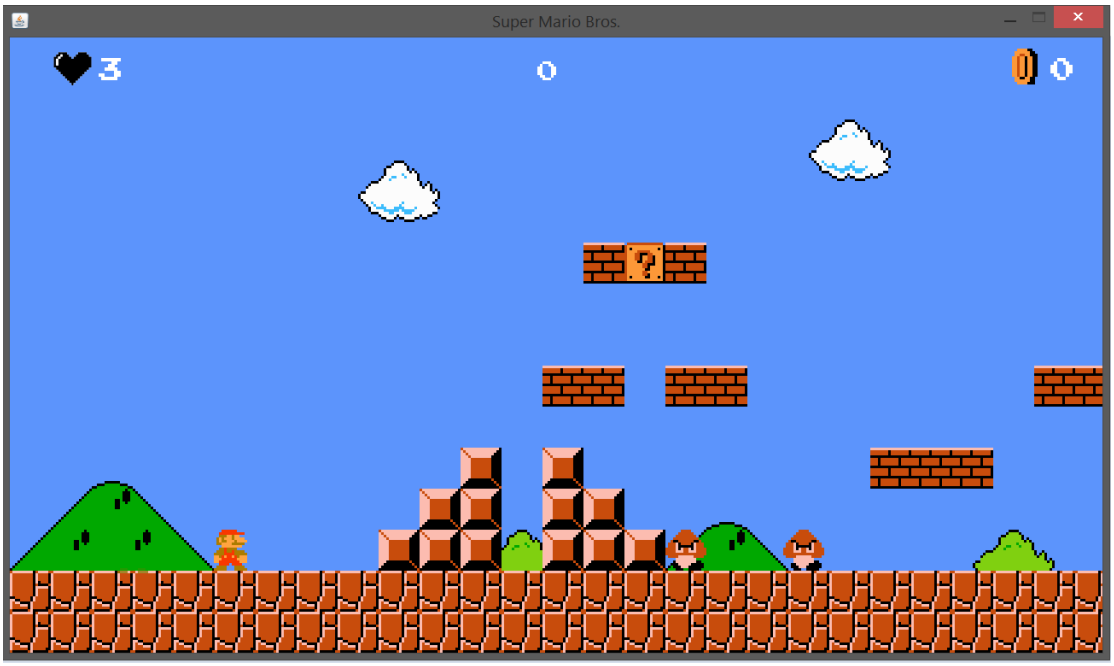

Figure 4 In-game screen

Finally, user can play the game. Player can pause the game in any given time and resume it again. If Mario dies after touching an enemy, map will be reset to beginning which means there is no checkpoint system in the game.

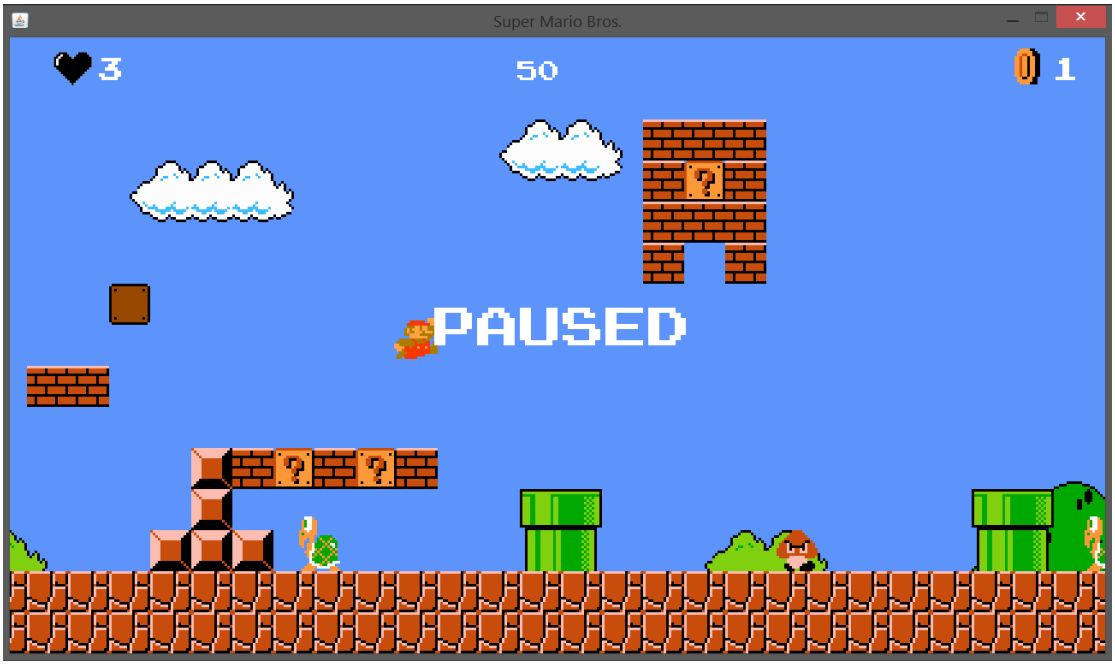

Figure 5 Pause Screen

Player can pause/resume by ESC key.

Keys corresponding

- UP Arrow key – Jump or navigate through options.
- DOWN Arrow key – Navigate through option.
- RIGHT Arrow key – Move Mario to right
- LEFT Arrow key – Move Mario to left.
- SPACE key – Send fireballs
- ESC key – Pause/Resume game or go to start screen when game is over

5 Unfinished Requirements

Most of the requirements are implemented and working fine. However, there are minor things remain to be implemented.

- Quest path: We wanted to make a quest path which includes many maps and unlock map feature. All maps can be played in this implementation and they are all independent from each other.