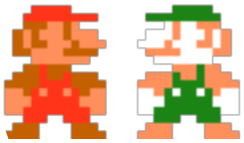
System Requirements Specification



Super Mario Bros.

Authors:
Cody Brown
Nathan McKain
Timothy Rolling

February 20, 2014
Version 1.1

1. Introduction

1.1. Purpose

The purpose of this document is to define what the “Super Mario Bros.” system should do. The audience that this document is intended for includes the client, the project managers, the testing engineers and the system designers. This document’s intention is to: 1) completely and accurately state what functions the program will perform; 2) show the basic layout of the system using “mock ups” of what the user interface will do; 3) give an overview of the entire program; 4) provide definitions of the terminology that this document and the project will use; 5) define the scope of the project.

1.2. Scope

The Super Mario Brothers system is a graphical user interface program that recreates the Super Mario Brothers game for Nintendo. The program allows user to start a game, make the valid moves to proceed through levels, and win the levels. The main goal of the game is to pass through multiple worlds of levels in an attempt to win the game.

1.3. Definitions, Acronyms, and Abbreviations

Beetle A beetle-like enemy

Bowser A dragon-like enemy that appears in castles

Bros. Brothers

Goomba A mushroom-like enemy

Koopa Short for Koopa Troopa (see Koopa Troopa).

Koopa Troopa A turtle-like enemy

Luigi One of two main characters in the game that tries to save Princess Peach; Mario’s brother.

Mario One of two main characters in the game that tries to save Princess Peach; Luigi’s brother.

Peach The princess that the character is trying to save

Piranha Plant Leafy plant enemy that usually comes out of a pipe and chomps

Power-Up A special item that helps the character save Peach

SMB Super Mario Brothers

1.4. Overview

The rest of this document includes four major sections: 1) overall description; 2) specific functional requirements; 3) appendix. The overall description is an overview of the system. The requirements section outlines the specific requirements that the system must satisfy.

2. Overall Description

2.1. Product Perspective

2.1.1. System Interfaces

The SMB program is a stand-alone system that runs within the operating system environment. The system does not communicate with other devices, so no data or functions are accessed outside of the program.

2.1.2. User Interfaces

The program operates using user input devices, including the mouse and keyboard. The system also uses a monitor to show the user the status of the game.

2.1.3. Hardware Interfaces

This system requires a keyboard and mouse, as well as a display.

2.1.4. Software Interfaces

The program will work on computers running Windows XP or newer. The computer must have a Java Virtual Machine of at least version 1.6 (source: http://java.com/).

2.1.5. Communication Interfaces

This system does not communicate with other systems and does not need any communication interfaces.

2.1.6. Operations

Start up : Program begins, Start screen shown
Character Selection : Select between Mario and Luigi
Gameplay : User proceeds through levels
Loss/Victory : User can end or begin again
Off : Program is off

2.2. Product Functions

The main function of SMB is to serve as an entertainment system during which the user utilizes the character's abilities to complete the levels and the game.

2.3. User Characteristics

The program is designed for many audiences. The recommended age for using the system is at least five years old. The user should be able to operate a computer using the mouse and keyboard, and should be able to read basic English on the display.

2.4. Constraints

This program shall require less than 64 Megabytes of memory in addition to the JVM and Operating System requirements.

2.5. Assumptions and Dependencies

For expected behavior, the system should be running a Windows operating system. The system also expects input from a keyboard and mouse.

3. Specific Requirements

3.1. External Interfaces

3.1.1. The system does not require any external interfaces.

3.2. Functions

3.2.1. Phase 1 (Essential Requirements)

3.2.1.1. Start (see Appendix)

3.2.1.1.1. The system shall load into a start screen the option to play the game (which leads to the character selection screen) or enter the level creator.

3.2.1.1.2. The system shall allow the user to exit the game any time.

3.2.1.2. Level Creator

3.2.1.2.1. The level creator shall allow the user to create levels to add to the game.

3.2.1.2.2. The level creator shall allow users to delete created levels (but not default levels).

3.2.1.2.3. The level creator shall allow the player to play the selected level, which will lead to the character selection screen first.

3.2.1.2.4. The level creator shall allow the player to select the music that will be played during the level from a default list, as well as create the map using blocks used in the default levels.

3.2.1.3. Character Selection

3.2.1.3.1. The system should allow the user to pick a character (Mario or Luigi) by clicking on his picture.

3.2.1.3.2. The system shall allow the user to go back to the start screen without selecting a character.

3.2.1.3.3. The system shall show which character is currently selected.

3.2.1.3.4. The system shall begin the game with the selected character when the "Start" button is pressed

3.2.1.4. Game Play (see Appendix)

3.2.1.4.1. The character shall be controlled by using the keyboard buttons and should allow the character to jump, run left and right, duck, sprint, and shoot fireballs.

3.2.1.4.2. The system shall default to the first level at the start.

3.2.1.4.3. The game shall place the selected character at the beginning of the current level.

3.2.1.4.4. The background of the levels should include decorations that the character cannot interact with.

3.2.1.4.5. The levels should include objects, such as bricks, which the character can stand on or that can impede the characters progress.

3.2.1.4.6. The levels shall include enemies that can injure or kill the character.

3.2.1.4.6.1. Goomba shall be included in the game.

3.2.1.4.6.1.1. Goomba shall kill the character if he runs into the side of Goomba.

3.2.1.4.6.1.2. Goomba shall move in one direction unless he collides with another object.

3.2.1.4.6.1.3. Goomba shall fall if he goes off the edge of an object.

3.2.1.4.6.1.4. Goomba shall be smashed and disappear if the character lands on top of him.

3.2.1.4.6.1.5. Goomba will disappear if hit by a fireball.

3.2.1.4.6.2. Koopa Troopa and Beetle shall be included in the game.

3.2.1.4.6.2.1. Koopa and Beetle shall kill the player if he runs into their side.

3.2.1.4.6.2.2. Koopa Troopa shall come in both red and green colors.

3.2.1.4.6.2.2.1. Red Koopa and Beetle will turn around if they reaches an edge

3.2.1.4.6.2.2.2. Green Koopa will fall down if he walks over an edge.

3.2.1.4.6.2.3. Koopa and Beetle will move in one direction unless he collides with another object. They would then turn around.

3.2.1.4.6.2.4. Koopa and Beetle shall be smashed into their shells if the character lands on top of them.

3.2.1.4.6.2.4.1. If the character lands hits the shell, the shell will move quickly in the direction opposite where the character hit.

3.2.1.4.6.2.4.1.1. The shell will kill/injure any enemies/characters that it hits, with the exception of Bowser.

3.2.1.4.6.2.4.1.2. The shell will fall off any edge it goes over, and will change direction if it hits an obstacle.

3.2.1.4.6.3. The Piranha Plant shall be included in the game.

3.2.1.4.6.3.1. The Piranha Plant shall kill the character if he touches the Piranha Plant.

3.2.1.4.6.3.2. The Piranha Plant can only be killed by shells or fireballs.

3.2.1.4.6.4. Bowser shall be included in the game.

3.2.1.4.6.4.1. Bowser can kill the character if they make contact or if the character is hit by Bowser's fire.

3.2.1.4.6.4.2. Bowser shall jump at times and breathe fire that will aim toward the character's position when the fire is released.

3.2.1.4.6.4.3. Bowser shall appear at the end of all castle levels and is defeated by releasing the bridge or being shot by four fireballs.

3.2.1.4.7. The levels shall include various "power-ups".

3.2.1.4.7.1. Power-ups include growth, fire, invincibility, and 1-UP.

3.2.1.4.7.1.1. The grown character power-up comes in the form of a red-spotted mushroom and will increase the character's size. Hitting an enemy will cause the character to return to the small size. Getting this power-up while already grown will have no effect. This mushroom will move in one direction, bounce off obstacles, and fall off edges.

3.2.1.4.7.1.2. The 1-UP power-up gives the character an extra life and comes in the form of a green-spotted mushroom. This mushroom will move in one direction, bounce off obstacles, and fall off edges.

3.2.1.4.7.1.3. The invincible power-up comes in the form of a star and allows the character to kill any enemy by touching them. This power-up is temporary and wears off after a certain amount of time, generally around 10 seconds. This star will bounce around, changing directions when it collides with an obstacle, and can fall off edges.

3.2.1.4.7.1.4. The fireball power-up transforms the character into a fully-grown fire-mode character. This power-up comes in the form of a flower. The character can shoot fireballs capable of killing enemies and also has the benefits of the grown character. This flower is stationary.

3.2.1.4.7.2. Power-ups will appear out of blocks when the character hits them.

3.2.1.4.7.3. Each power-up shall cause a sound effect to play.

3.2.1.4.8. The levels shall include coins.

3.2.1.4.8.1. The character can collect coins by touching them.

3.2.1.4.8.2. Every time the character collects 100 coins, the character shall be awarded an extra life.

3.2.1.4.9. The levels shall include gravity that will pull down on any characters, enemies, or power-ups that are not located on a solid block.

3.2.1.4.10. The levels shall include pipes, which may allow the character to go down to a hidden part of the level.

3.2.1.4.11. The levels may include holes, which will kill the character if he falls down.

3.2.1.4.12. The system shall not allow the player to return to part of the level that has already exited the screen to the left. This edge will act as a wall to the character but not to other objects.

3.2.1.4.13. When the character reaches the end of a level, sound effects will play, then the screen will show the number of lives the character has left, followed by the beginning of the next level (with the exception of the final level, after which the winning screen shall be shown).

3.2.1.4.14. The winning screen shall show the character finding the Princess and display a winning message to the player.

3.2.1.4.15. The game shall play background music during all parts of the game except for loading screens.

3.2.2. Phase 2 (Essential Requirements)

3.2.2.1. Start

3.2.2.1.1. The system shall show the currently selected character on screen running through a demo.

3.2.2.2. Game Play

3.2.2.2.1. The character shall be controlled using the keyboard.

3.2.2.2.1.1. Pressing the up key will allow the character to climb certain objects.

3.2.2.2.1.2. Pressing up will allow the character to swim in water.

3.2.2.2.2. The system shall have up to one checkpoint in each level.

3.2.2.2.3. If the player dies during a level and has lives remaining, the system shall place the player at the most recent checkpoint reached, if any.

3.2.2.2.4. Points will be awarded for activities such as killing enemies, collecting coins, or obtaining power-ups.

3.2.2.2.4.1. The local high score will be tracked and displayed on the start screen.

3.2.2.2.5. The game shall include a store that can be accessed at the end of every level.

3.2.2.2.5.1. Various items can be purchased from the store using coins, up to two of which can be used during levels by pressing the corresponding keyboard shortcut.

3.2.2.2.6. The will be saved at the end of each level and can be loaded later. Each save shall overwrite the previous save of the same game, if any.

3.2.3. Future Phases (Optional Requirements)

3.2.3.1. Gameplay

3.2.3.1.1. The game will allow for additional characters to be used by the player, each with their own set of moves.

3.2.3.1.2. The game shall allow for more enemies to be implemented.

3.2.3.2. Level Creator

3.2.3.2.1. The level creator shall allow for custom songs to be imported and played as background music.

3.2.3.2.2. The level creator shall allow the player to use the default levels as templates for new levels to create edited versions of the default levels.

3.3. Performance Requirements

3.3.1. The system shall arrive at the Start screen within 10 seconds of being opened.

3.3.2. The game shall operate smoothly, without periods of visually noticeable lag between frames.

3.4. Maintainability

3.4.1. The game shall be built in modules for each main screen, such as the start screen and the game play screen. This will allow each module to be modified without affecting the other parts of the system.

3.5. Security

3.5.1. The system will not have any special security features.

3.6. Response Time

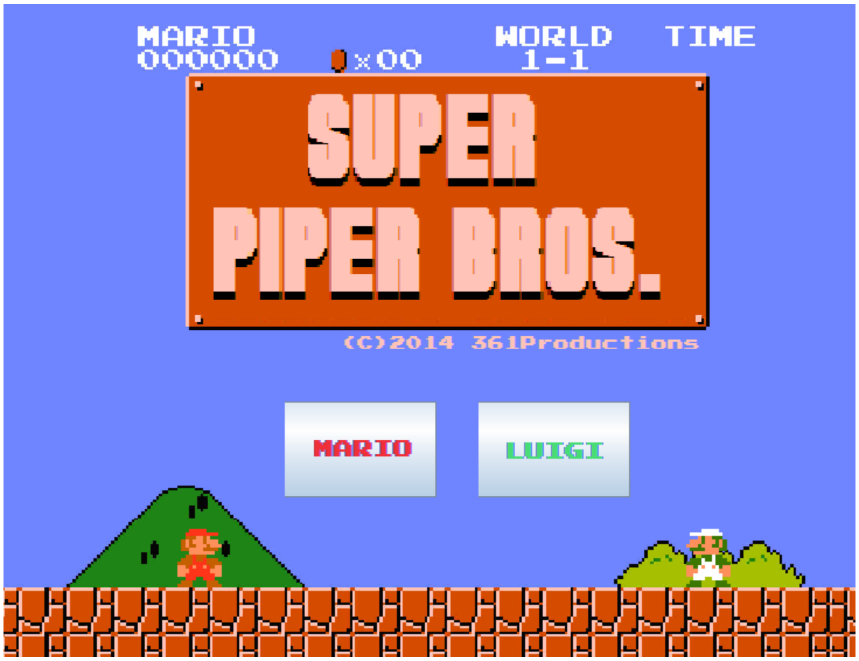
3.6.1. The system shall respond to user input within two frames and within 0.5 seconds, except when loading a new screen.

3.7. Design Constraints

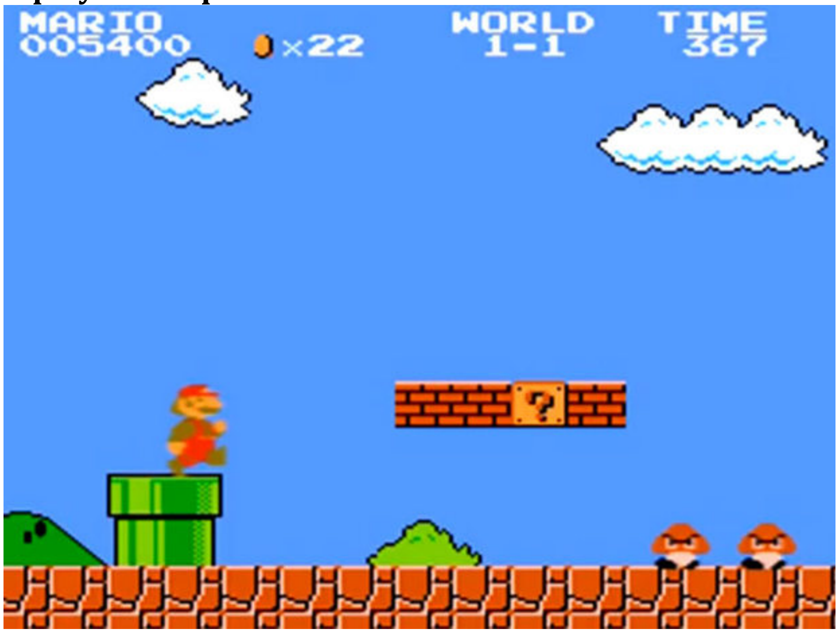
3.7.1. The first version shall implement all functions needed to implement an initial start screen, followed by character selection and game play. All gameplay details including winning, dying, and basic running/jumping movements shall be implemented.

4. Appendix

4.1. Start Screen Mockup



4.2. Gameplay Mockup


MARIO
005400
×22
WORLD
1-1
TIME
367