SE 3XA3: Test Report
Mari0

Team 9, Ninetendo
David Hobson - hobsondd
Jose Miguel Ballesteros - ballesjm
Jeff Pineda - pinedaj

December 8, 2016

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	**Version**	**Notes**
2016-12-07	1.0	Section 1,2,5,6,9
2016-12-08	1.0	Section 4,7

1 Functional Requirements Evaluation

1.1 Input Testing

Test Case	*Initial State*	*Input*	*Expected Result*	*Result*
Run Right	Idle	RIGHT_ARROW_KEY/D_KEY	Player moves right	PASS
	Moving Left	RIGHT_ARROW_KEY/D_KEY	Player moves right	PASS
Run Left	Idle	LEFT_ARROW_KEY/A_KEY	Player moves left	PASS
	Moving Right	LEFT_ARROW_KEY/A_KEY	Player moves left	PASS
Stop	Moving Right	No input	Player stops	PASS
	Moving Left	No input	Player stops	PASS
Jump	Idle	SPACE_BAR	Player jumps	PASS
	Moving	SPACE_BAR	Player jumps	PASS
	Airborne	SPACE_BAR	Nothing	PASS
Fire Blue Portal	Any In-Game State	LEFT_CLICK	Blue Portal	PASS
Fire Orange Portal	Any In-Game State	RIGHT_CLICK	Orange Portal	PASS
Pause	Any In-Game State	P_KEY	Game is Paused	PASS
Quit	Paused	LEFT_CLICK on QUIT	Main Menu shown	PASS
Play	Main Menu	LEFT_CLICK on START	Game Begins	PASS
Help	Main Menu	LEFT_CLICK on HELP	Help Menu shown	PASS

1.2 Collision Testing

Test Case Collision Object	*Initial State*	*Condition*	*Expected Result*	*Result*
Wall	Player is moving	Player hits wall	Wall prevents movement	PASS
Platform	Player is idle/falling	Player hits Platform	Wall prevents movement	PASS
Castle	Player is Moving	Player runs into Castle	Level is won	PASS
Score Box	Player is airborne	Player hits score Box	Score increases by 10	PASS
Blue Portal	Player is moving	Player hits portal	Character teleports to orange portal	PASS
Orange Portal	Player is moving	Player hits portal	Character teleports to blue portal	PASS
Goomba	Player is falling	Foot first Contact	Goomba dies	PASS
	Player is idle/moving	Front first contact	Game restarts	PASS

2 Nonfunctional Requirements Evaluation

Description: The following tests were executed by each member of our development team and a couple colleagues from our faculty. Engineering students were better suited for testing the current state of our project since it is an early development view of the open source game that is being recreated. Each participant was asked to give their honest feedback and suggestions.

2.1 Look and Feel Requirements

1. Game Environment

   Results: All testers were able to explore all the game environments without issues.

2. Game Hude/Interface

   Results: All testers stated that the location of the counter was not not obstructive in their opinion.

2.2 Usability and Humanity Requirements

1. Ease of Learning

   Results: All testers stated the game was easy to play and the controls were easy to learn.

2. Entertainment

   Results: All testers stated that the game can be more enjoyable if it were stretched out to a longer period of time and had more gameplay, but is currently too short to provide a good amount of entertainment.

2.3 Performance Requirements

1. Controls/Commands

   Results: All testers noticed no delays or malfunction from the controls.

2.4 Operational and Environment Requirements

1. Operating System Support

   Results: Each tester was able to run the game on their own operating system. These include Windows and OSX 10.

2.5 Security Requirements

1. Altering Information

   Results: No tester indicated any type of alterations done to their current process or files.

2.6 Cultural Requirements

1. Spelling and Grammar

   Results: All testers indicated that the game had no spelling or grammar mistakes.

2. Offensive Content

   Results: The testers indicated that they found no source of offensive content that would be directed at them or people from another culture.

2.7 Legal Requirements

1. License Adherence

   Results: The testers stated that they believe that the game is not breaching its current license.

2.8 Health and Safety Requirements

1. Epileptic Prevention

   Results: The testers indicated that they believe that the game would not trigger any epileptic seizures, although it is worth noting that none of the testers has ever had a history of epileptic seizures.

3 Comparison to Existing Implementation

Since the functional and nonfunctional requirements that Mari0 must meet are derived from the original implementation, all system tests that test whether or not these requirements are met will be the metric used to determine the likeness of our reimplementation to the original product. We made sure that we tested and compared our product to the current product that is available. Our tests ensure that our game meets the same requirements as the original. Our tests and surveys included a lot of the nonfunctional requirements such as Look and Feel, and Performance Requirements. Many of the different functional requirements were tested alongside with the original product to ensure accuracy. These requirements will be movement, portal colllision, wall and ground collision, enemy collision, and player physics.

4 Unit Testing

Test Case Name	TPG: Play Game Button
Initial State	User is viewing the main menu screen.
Input	User left clicks the Play Game button
Expected Results	User is taken to the game level and given control of Mario
Actual Results	User is taken to the game level and given control of Mario
Test Result	Pass

Test Case Name	THP: Help Button
Initial State	User is viewing the main menu screen
Input	User left clicks the Help button
Expected Results	User is taken to a screen that lists the controls of the game
Actual Results	User is taken to a screen that lists the controls of the game
Test Result	Pass

Test Case Name	TGP: Game Pause Functionality
Initial State	User is in a level playing the game
Input	The user presses down either the ESC key or the 'P' key on their keyboard
Expected Results	The game is fully paused (game events no longer occur, user can no longer provide inputs, and audio stops).
Actual Results	The game is fully paused.
Test Result	Pass

Test Case Name	TST: In Game User Interface Score Tracker
Initial State	User is in a level playing the game
Input	User increases their score by collecting coins or killing enemies
Expected Results	Score increases by the value associated with coins and enemies
Actual Results	Score increases by the value associated with coins and enemies
Test Result	Conditional Pass (See next two unit test cases)

Test Case Name	TCCIS: Colleting Coins to Increase Score
Initial State	User is in a level playing the game
Input	User collects a coin by jumping from underneath and hitting a question mark block
Expected Results	Score increases by a value of 10
Actual Results	Score increases by a value of 10
Test Result	Pass

Test Case Name	TKEIS: Killing Enemies to Increase Score
Initial State	User is in a level playing the game
Input	User lands on an enemy, killing the enemy
Expected Results	Score increases by a value of 50
Actual Results	Score does not increase
Test Result	Fail

Test Case Name	TKE: Killing Enemies
Initial State	User is in a level playing the game
Input	User lands on an enemy
Expected Results	Enemy is killed (removed from game)
Actual Results	Enemy is killed
Test Result	Pass

Test Case Name	TDE: Dying to Enemies
Initial State	User is in a level playing the game
Input	User runs into an enemy (Player's current yvalue is less than or equal to Enemy's yvalue)
Expected Results	Player is killed (removed from game) and game restarts the level
Actual Results	Player is killed and player is placed at the beginning of the level
Test Result	Pass

Test Case Name	TDF: Dying to Pit Fall
Initial State	User is in a level playing the game
Input	User falls into a pit fall (Falls off the screen)
Expected Results	Player is killed (removed from game) and game restarts the level
Actual Results	Player is killed and player is placed at the beginning of the level
Test Result	Pass

Test Case Name	TFBVH: Firing a Blue Portal on a valid horizontal surface
Initial State	User is in a level playing the game
Input	Player left clicks on a horizontal surface that has enough space to fit the length of the portal
Expected Results	A Blue Portal is placed horizontally on the surface
Actual Results	A Blue Portal is placed horizontally on the surface
Test Result	Pass

Test Case Name	TFBVV: Firing a Blue Portal on a valid vertical surface
Initial State	User is in a level playing the game
Input	Player left clicks on a vertical surface that has enough space to fit the length of the portal
Expected Results	A Blue Portal is placed vertically on the surface
Actual Results	A Blue Portal is placed vertically on the surface
Test Result	Pass

Test Case Name	TFBNH: Firing a Blue Portal on a non-valid horizontal surface
Initial State	User is in a level playing the game
Input	Player left clicks on a horizontal surface that does not have enough space to fit the length of the portal
Expected Results	A Blue Portal is not placed horizontally on the surface
Actual Results	A Blue Portal is not placed horizontally on the surface
Test Result	Pass

Test Case Name	TFBNV: Firing a Blue Portal on a non-valid vertical surface
Initial State	User is in a level playing the game
Input	Player left clicks on a vertical surface that does not have enough space to fit the length of the portal
Expected Results	A Blue Portal is not placed vertically on the surface
Actual Results	A Blue Portal is not placed vertically on the surface
Test Result	Pass

Test Case Name	TFOVH: Firing an Orange Portal on a valid horizontal surface
Initial State	User is in a level playing the game
Input	Player right clicks on a horizontal surface that has enough space to fit the length of the portal
Expected Results	An Orange Portal is placed horizontally on the surface
Actual Results	An Orange Portal is placed horizontally on the surface
Test Result	Pass

Test Case Name	TFOVV: Firing an Orange Portal on a valid vertical surface
Initial State	User is in a level playing the game
Input	Player right clicks on a horizontal surface that has enough space to fit the length of the portal
Expected Results	An Orange Portal is placed vertically on the surface
Actual Results	An Orange Portal is placed vertically on the
Test Result	Pass

Test Case Name	TFONH: Firing an Orange Portal on a non-valid horizontal surface
Initial State	User is in a level playing the game
Input	Player right clicks on a horizontal surface that does not have enough space to fit the length of the portal
Expected Results	An Orange Portal is not placed horizontally on the surface
Actual Results	An Orange Portal is not placed horizontally on the surface
Test Result	Pass

Test Case Name	TFONV: Firing an Orange Portal on a non-valid vertical surface
Initial State	User is in a level playing the game
Input	Player right clicks on a vertical surface that does not have enough space to fit the length of the portal
Expected Results	An Orange Portal is not placed vertically on the surface
Actual Results	An Orange Portal is not placed vertically on the surface
Test Result	Pass

Test Case Name	TMLS: Move Left when there is open space to the left of the character
Initial State	User is in a level playing the game, player is grounded or in the air
Input	Player presses the 'Left Arrow' key or the 'A' key
Expected Results	Character's x velocity becomes negative and character is displaced to the left
Actual Results	Character's x velocity becomes negative and character is displaced to the left
Test Result	Pass

Test Case Name	TMRS: Move Right when there is open space to the right of the character
Initial State	User is in a level playing the game, player is grounded or in the air
Input	Player presses the 'Right Arrow' key or the 'D' key
Expected Results	Character's x velocity becomes positive and character is displaced to the right
Actual Results	Character's x velocity becomes positive and character is displaced to the right
Test Result	Pass

Test Case Name	TMLNS: Move Left when there is no open space to the left of the character
Initial State	User is in a level playing the game, player is grounded or in the air
Input	Player presses the 'Left Arrow' key or the 'A' key
Expected Results	Character's x velocity becomes zero and character is not displaced
Actual Results	Character's x velocity becomes zero and character is not displaced
Test Result	Pass

Test Case Name	TMRNS: Move Right when there is no open space to the right of the character
Initial State	User is in a level playing the game, player is grounded or in the air
Input	Player presses the 'Right Arrow' key or the 'D' key
Expected Results	Character's x velocity becomes zero and character is not displaced
Actual Results	Character's x velocity becomes zero and character is displaced
Test Result	Pass

Test Case Name	TJ: Jump
Initial State	User is in a level playing the game and the character is grounded
Input	Player presses the 'Up Arrow' key or the W Key
Expected Results	Character's y velocity becomes positive and character is displaced upwards
Actual Results	Character's y velocity becomes positive and character is displaced upwards
Test Result	Pass

5 Changes Due to Testing

After the testing was complete no urgent fixes that interfered with the requirements were needed to be made.

6 Automated Testing

No automated testing methods were used for the testing of this product.

7 Trace to Requirements

Tests	Requirements
TPG	R1
THB	R1, R2
TGP	R3
TST	R4, R13
TCCIS	R9, R13
TKEIS	R11, R13
TKE	R11
TDE	R10
TDF	R12
TFBVH	R5
TFBVV	R5
TFBNH	R5
TFBNV	R5
TFOVH	R5
TFOVV	R5
TFONH	R5
TFONV	R5
TMLS	R6
TMRS	R6
TMLNS	R6
TMRNS	R6
TH	R7

Table 2: Trace between Tests to Requirements

8 Trace to Modules

Tests	Modules
TPG	M1, M2, M4, M10
THB	M1, M2, M4, M10
TGP	M3, M10
TST	M4, M7, M10
TCCIS	M4, M7, M8, M9, M10, M11
TKEIS	M4, M7, M9, M10, M11
TKE	M7, M8, M9, M11
TDE	M7, M8, M11
TDF	M8, M11
TFBVH	M2, M5, M6, M11
TFBVV	M2, M5, M6, M11
TFBNH	M2, M5, M6, M11
TFBNV	M2, M5, M6, M11
TFOVH	M2, M5, M6, M11
TFOVV	M2, M5, M6, M11
TFONH	M2, M5, M6, M11
TFONV	M2, M5, M6, M11
TMLS	M3, M8, M9, M11
TMRS	M3, M8, M9, M11
TMLNS	M3, M8, M9, M11
TMRNS	M3, M8, M9, M11
TH	M3, M8, M9, M11

Table 3: Trace between Tests to Modules

9 Code Coverage Metrics

No accurate code coverage metrics were achieved with our current test suit since it was all a result of manual and survey testing from other people.