Project Name: Mario

Test Case - 1

Test Case ID	: TC_UI_1	**Test Designed by**	:	Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	:	9th December 2014
Module Name	: Character naming screen	**Test Executed by**	:	Vernon Chien
Test Title	: Input character name	**Test Execution date**	:	11th December 2014
Description	: Test the character naming page.			

Pre-conditions : Game loaded successfully

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to character naming screen.		Player should be able to name his or her character.	Character of the player is named and player navigated to character selection screen.	Pass	
2	Provide valid name.	Player = Vernon				
3	Click on the "OK" button or press "Enter".					

Post-conditions : Player's name is input into the game application and successfully displayed on the character selection screen.

Project Name: Mario

Test Case – 2

Test Case ID	: TC_UI_2	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	: 9th December 2014
Module Name	: Character selection screen	**Test Executed by**	: Vernon Chien
Test Title	: Verify character selection	**Test Execution date**	: 11th December 2014
Description	: Test the character selection page.		

Dependencies : TC_UI_1

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to character selection screen.		Player should be able to select his or her favorite character	Player's favorite character is selected and player navigated to scene of level 1.	Pass	
2	Click on favorite character.	Player = Mario / Luigi				

Post-conditions : Player's favorite character is selected and the character is displayed on the screen throughout the level game.

Project Name: Mario

Test Case - 3

Test Case ID : TC_ACT_1		**Test Designed by** : Vernon Chien	
Test Priority (Low/Medium/High) : Medium		**Test Designed day** : 9th December 2014	
Module Name : Level game		**Test Executed by** : Vernon Chien	
Test Title : Verify forward movement of the character		**Test Execution date** : 11th December 2014	
Description : Test the forward movement of the character throughout the level game.			

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to level game. (Level 1, 2, 3)		Player should be able to move his or her character forward.	Chosen character is moved forward.	Pass	
2	Press right arrow key					

Post-conditions : Player's chosen character is moved forward and the character is passed through the place where he or she previously stay.

Project Name: Mario

Test Case - 4

Test Case ID	: TC_ACT_2	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Medium	**Test Designed day**	: 9th December 2014
Module Name	: Level game	**Test Executed by**	: Vernon Chien
Test Title	: Verify backward movement of the character	**Test Execution date**	: 11th December 2014
Description	: Test the backward movement of the character throughout the level game.		

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to level game. (Level 1, 2, 3)		Player should be able to move his or her character backward.	Chosen character is moved backward.	Pass	
2	Press left arrow key					

Post-conditions : Player's chosen character is moved backward and the character is passed through the place where he or she previously stay.

Project Name: Mario

Test Case - 5

Test Case ID	: TC_ACT_3	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Medium	**Test Designed day**	: 9th December 2014
Module Name	: Level game	**Test Executed by**	: Vernon Chien
Test Title	: Verify upward movement of the character	**Test Execution date**	: 11th December 2014
Description	: Test the upward movement of the character throughout the level game.		

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to level game. (Level 1, 2, 3)		Player should be able to move his or her character upward.	Chosen character is moved upward.	Pass	
2	Press up arrow key					

Post-conditions : Player's chosen character is moved upward and back to the original position.

Project Name: Mario

Test Case - 6

Test Case ID	:	TC_ACT_4	Test Designed by	:	Vernon Chien
Test Priority (Low/Medium/High)	:	Medium	Test Designed day	:	9th December 2014
Module Name	:	Level game	Test Executed by	:	Vernon Chien
Test Title	:	Verify shooting ability of the character	Test Execution date	:	11th December 2014
Description	:	Test the shooting ability of the character throughout the level game			

Pre-Conditions : Fireballs are collected

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to level game. (Level 1, 2, 3)		Player should be able to shoot the enemies.	Fireball is released, the enemy has been shot and die.	Pass	
2	Press space bar key					

Post-conditions : Fireball is released by the player's chosen character. Enemy is shot by the fireball and die. The character is stayed safe from enemy.

Project Name: Mario

Test Case - 7

Test Case ID	: TC_UI_3	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	: 9th December 2014
Module Name	: Mini game screen	**Test Executed by**	: Vernon Chien
Test Title	: Input of mini game's answers	**Test Execution date**	: 11th December 2014
Description	: Test the mini game page		

Pre-Conditions : Pass level game (Level 1/ 2)

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to mini game.		Player should be able to give answer for each question.	Answer for each question is filled in and submitted.	Pass	
2	Press answer for each question	Player = answer for question				
3	Click on "submit" button					

Post-conditions : Answer for each question is filled in and submitted successfully. 5 extra fireballs are given to the player who answered at least 8 questions correctly. 20 coins are deducted from the player's accumulated coins if the player answered less than 8 questions correctly.

Project Name: Mario

Test Case - 8

Test Case ID	: TC_DISP_1	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	: 9th December 2014
Module Name	: Level game	**Test Executed by**	: Vernon Chien
Test Title	: Display Level	**Test Execution date**	: 11th December 2014
Description	: Test the ability to display level information		

Dependencies   : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to level game. (Level 1, 2 ,3)		Level information should be corresponded to current level.	Level information is corresponding to current level	Pass	
2	Check whether the level information is corresponding to the current level					

Post-conditions   : Level information is corresponding to current level. When player is in level 1, the phrase "Level 1" is displayed on the top of the window. When player is in level 2, the phrase "Level 2" is displayed on the top of the window. When player is in level 3, the phrase "Level 3" is displayed on the top of the window.

Project Name: Mario

Test Case - 9

Test Case ID	: TC_DISP_2	**Test Designed by**	:	Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	:	9th December 2014
Module Name	: Level game	**Test Executed by**	:	Vernon Chien
Test Title	: Display Life	**Test Execution date**	:	11th December 2014
Description	: Test the ability to display life information			

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to level game. (Level 1, 2 ,3)		Life information should be displayed correctly according to the performance of player.	Life is deducted when the character touched the enemy. Current number of life is displayed on top of the window.	Pass	
2	Check the initial number of life					
3	Move the character towards the enemy					
4	Check the current number of life					
5.	Repeat step 3 and 4 until the game is over					

Post-conditions : Life information is displayed correctly. Number of life displayed on top of the window decreased from two to one after the player touched the enemy.

Project Name: Mario

Test Case - 10

Test Case ID	: TC_DISP_3	Test Designed by	: Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	: 9th December 2014
Module Name	: Level game	**Test Executed by**	: Vernon Chien
Test Title	: Display Coin(s)	**Test Execution date**	: 11th December 2014
Description	: Test the ability to display coin(s) information		

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to level game. (Level 1, 2 ,3)		Coin(s) information should be displayed correctly according to the performance of player.	Coin(s) is added to the accumulated coin(s) when the character get the coin(s). Current number of coin(s) is displayed on top of the window.	Pass	
2	Check the initial number of coin(s)					
3	Get coin					
4	Check the current number of coin(s)					
5	Repeat step 3 and 4					

Post-conditions : Coin(s) information is displayed correctly. Number of coin(s) displayed on top of the window increased from zero to one after the player get a coin.

Project Name: Mario

Test Case - 11

Test Case ID	: TC_DISP_4	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	: 9th December 2014
Module Name	: Level game	**Test Executed by**	: Vernon Chien
Test Title	: Display Fireball(s)	**Test Execution date**	: 11th December 2014
Description	: Test the ability to display Fireball(s) information		

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to level game. (Level 1, 2 ,3)		Fireball(s) information should be displayed correctly according to the performance of player.	Fireball(s) is added to the accumulated fireball(s) when the character get the star(s). Current number of fireball(s) is displayed on top of the window.	Pass	
2	Check the initial number of fireball(s)					
3	Get fireball					
4	Check the current number of fireball(s)					
5	Repeat step 3 and 4					

Post-conditions : Fireball(s) information is displayed correctly. Number of fireball(s) displayed on top of the window increased from zero to five after the player get a star.

Project Name: Mario

Test Case - 12

Test Case ID	: TC_DISP_5	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	: 9th December 2014
Module Name	: Level game	**Test Executed by**	: Vernon Chien
Test Title	: Display Fireball(s)	**Test Execution date**	: 11th December 2014
Description	: Test the ability to display Fireball(s) information		

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to level game. (Level 1, 2 ,3)		Fireball(s) information should be displayed correctly according to the performance of player.	Fireball(s) is deducted from the accumulated fireball(s) when the character use it to shoot enemy. Current number of fireball(s) is displayed on top of the window.	Pass	
2	Get fireball					
3	Check the current number of fireball(s)					
4	Shoot enemy					
5	Check the current number of fireball(s)					

Post-conditions : Fireball(s) information is displayed correctly. Number of fireball(s) displayed on top of the window decreased from five to four after the player use it to shoot enemy.

Project Name: Mario

Test Case - 13

Test Case ID	: TC_BUTTON_1	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	: 9th December 2014
Module Name	: Mini game	**Test Executed by**	: Vernon Chien
Test Title	: Agree on deduct coins	**Test Execution date**	: 11th December 2014
Description	: Test the navigation of the game according to the button clicked ("Yes")		

Pre-Conditions : Passed Level Game and Mini Game, at least 20 coins are collected

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Click on "Yes" button		20 coins should be deducted from player's accumulated coin(s). Player should be in the scene of next level	Player is navigated to scene of next level. Number of accumulated coin(s) decreased by 20.	Pass	
2	Check on the current page					

Post-conditions : Player is now in the scene of level 2. Number of coin(s) displayed on top of the window decreased by 20.

Project Name: Mario

Test Case - 14

Test Case ID	: TC_BUTTON_2	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	: 9th December 2014
Module Name	: Mini game	**Test Executed by**	: Vernon Chien
Test Title	: Disagree on deduct coins	**Test Execution date**	: 11th December 2014
Description	: Test the navigation of the game according to the button clicked ("No")		

Pre-Conditions : Passed Level Game and Mini Game, at least 20 coins are collected

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Click on "No" button		Game Over screen should be displayed	Game over screen is displayed.	Pass	
2	Check on the current page					

Post-conditions : Player is now in the game over page and he or she is required to decide whether want to replay the game or not.

Project Name: Mario

Test Case - 15

Test Case ID	: TC_BUTTON_3	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	: 9th December 2014
Module Name	: Game over screen	**Test Executed by**	: Vernon Chien
Test Title	: Replay	**Test Execution date**	: 11th December 2014
Description	: Test the navigation of the game according to the button clicked ("Yes")		

Dependencies　　　: TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to game over screen.		Player should be in scene of level 1.	Scene of level 1 is displayed and the player is in the scene of level 1.	Pass	
2	Click on "Yes" button					
3	Check the current page					

Post-conditions　　: Player is now in the scene of level 1. He or she can replay the game.

Project Name: Mario

Test Case - 16

Test Case ID	: TC_BUTTON_4	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	: 9th December 2014
Module Name	: Game over screen	**Test Executed by**	: Vernon Chien
Test Title	: End game	**Test Execution date**	: 11th December 2014
Description	: Test the navigation of the game according to the button clicked ("No")		

Dependencies : TC_UI_1, TC_UI_2

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to game over screen.		Player should be able to exit the game.	Player exit the game successfully.	Pass	
2	Click on "No" button					
3	Check the current page					

Post-conditions : Player is not in any page of the game. Player exit the game successfully.

Project Name: Mario

Test Case - 17

Test Case ID	: TC_BUTTON_5	**Test Designed by**	: Vernon Chien
Test Priority (Low/Medium/High)	: Low	**Test Designed day**	: 9th December 2014
Module Name	: Character selection screen	**Test Executed by**	: Vernon Chien
Test Title	: Verify instruction loaded	**Test Execution date**	: 11th December 2014
Description	: Test the navigation to instruction page.		

Dependencies   : TC_UI_1

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)	Notes
1	Navigate to character selection screen.		Instruction page should be displayed.	Instruction page is displayed. Player is able to read the instruction of the game.	Pass	
2	Click on "instruction" button.					
3	Check the current page.					

Post-conditions   :   Instruction page is displayed. Instruction of the game is shown to player.

Test case ID: Unique ID for each test case. Follow some convention to indicate types of test. E.g. 'TC_UI_1' indicating 'user interface test case #1'.

Test priority (Low/Medium/High): This is useful while test execution. Test priority for business rules and functional test cases can be medium or higher whereas minor user interface cases can be low priority. Test priority should be set by reviewer.

Module Name – Mention name of main module or sub module.

Test Designed By: Name of tester

Test Designed Date: Date when wrote

Test Executed By: Name of tester who executed this test. To be filled after test execution.

Test Execution Date: Date when test executed.

Test Title/Name: Test case title. E.g. verify login page with valid username and password.

Test Summary/Description: Describe test objective in brief.

Pre-condition: Any prerequisite that must be fulfilled before execution of this test case. List all pre-conditions in order to successfully execute this test case.

Dependencies: Mention any dependencies on other test cases or test requirement.

Test Steps: List all test execution steps in detail. Write test steps in the order in which these should be executed. Make sure to provide as much details as you can. Tip – to efficiently manage test case with lesser number of fields use this field to describe test conditions, test data and user roles for running test.

Test Data: Use of test data as an input for this test case. You can provide different data sets with exact values to be used as an input.

Expected Result: What should be the system output after test execution? Describe the expected result in detail including message/error that should be displayed on screen.

Post-condition: What should be the state of the system after executing this test case?

Actual result: Actual test result should be filled after test execution. Describe system behavior after test execution.

Status (Pass/Fail): If actual result is not as per the expected result mark this test as failed. Otherwise update as passed.

Notes/Comments/Questions: To support above fields if there are some special conditions which can't be described in any of the above fields or there are questions related to expected or actual results mention those here.

Add following fields if necessary:

Defect ID/Link: If test status is fail, then include the link to defect log or mention the defect number.

Test Type/Keywords: This field can be used to classify tests based on test types. E.g. functional, usability, business rules etc.

Requirements: Requirements for which this test case is being written. Preferably the exact section number of the requirement doc.

Attachments/References: This field is useful for complex test scenarios. To explain test steps or expected result using a visio diagram as a reference. Provide the link or location to the actual path of the diagram or document.

Automation? (Yes/No): Whether this test case is automated or not. Useful to track automation status when test cases are automated.

Taken from http://www.softwaretestinghelp.com/test-case-template-examples/