Test Report
Super Tetris

Group#: 38

Team Name: Binary

Members:

Tongfei Wang : wangt62

Bowen Yuan : yuanb1

Tim Zhang : zhangj14

December 6, 2017

SFWR ENG 3XA3

McMaster University

Contents

List of Tables

List of Figures

1 Functional Requirements Evaluation

1.1 User Input and Navigation Test

Test 1.1.1:	**SP-IN1**
Type:	Functional,Dynamic,Manual
Initial State:	Web browser is open
Input:	User enters the domain of this game
Output:	A new game starts
Pass:	Passed.

Test 1.1.2:	**SP-IN2**
Type:	Functional,Dynamic,Manual
Initial State:	Game is in process
Input:	User presses 'enter'
Output:	Game paused
Pass:	Passed.

Table 1: **Revision History**

Date	**Version**	**Notes**
Dec 6 2017	1.0	document upload

Test 1.1.3:	**SP-IN3**
Type:	Functional,Dynamic,Manual
Initial State:	Game is paused
Input:	User presses 'enter'
Output:	Game resumes
Pass:	Passed.

Test 1.1.4:	**SP-IN4**
Type:	Functional,Dynamic,Manual
Initial State:	Game is paused or in process
Input:	User presses 'r'
Output:	Game is restarted
Pass:	Passed.

Test 1.1.5:	**SP-IN5**
Type:	Functional,Dynamic,Manual
Initial State:	Game is in process
Input:	User presses 'w'
Output:	The current tetromino is rotated clockwise
Pass:	Passed.

Test 1.1.6:	**SP-IN6**
Type:	Functional,Dynamic,Manual
Initial State:	Game is in process
Input:	User presses 'left' or 'right' by one grid
Output:	The current tetromino is moved left/right
Pass:	Passed.

Test 1.1.7:	**SP-IN7**
Type:	Functional,Dynamic,Manual
Initial State:	Game is in process
Input:	User presses 'down
Output:	The current tetromino falls to the bottom(not intersected with other tetrominoes)
Pass:	Passed.

Test 1.1.8:	**SP-IN8**
Type:	Functional,Dynamic,Manual
Initial State:	Game is in process
Input:	User presses '1' or '2' or '3'
Output:	User buys and uses the item they selected('1' to '3' correspanding to four different items)
Pass:	Passed.

1.2 Game Logic Test

Test 1.2.1:	**SP-GL2**
Type:	Structrual,Dynamic,Automated
Initial State:	Game is in Process.
Input:	Coordinate of current tetromino which is a 4*4 matrix
Output:	True
Pass:	Passed

Test 1.2.2:	**SP-GL3**
Type:	Structrual,Dynamic,Automated
Initial State:	Game is in Process.
Input:	row number
Output:	new row
Pass:	Passed.

Test 1.2.3:	**SP-GL4**
Type:	Structrual,Dynamic,Automated
Initial State:	Game is in Process.
Input:	N/A
Output:	A random 4*4 piece(tetromino)
Pass:	Passed.

Test 1.2.4:	**SP-GL7**
Type:	Structrual,Dynamic,Automated
Initial State:	Game is in Process.
Input:	item
Output:	True
Pass:	Passed.

Test 1.2.5:	**SP-GL9**
Type:	Structrual,Dynamic,Automated
Initial State:	Game is in Process.
Input:	Number of rows killedplaying time
Output:	Score increase 100 when one line is cleared
Pass:	Passed.

Test 1.2.6:	**SP-GL10**
Type:	Structrual,Dynamic,Automated
Initial State:	Game is in Process.
Input:	N/A
Output:	The speed of falling decrease for 15s
Pass:	Passed.

Test 1.2.7:	**SP-GL11**
Type:	Structrual,Dynamic,Automated
Initial State:	Game is in Process.
Input:	N/A
Output:	4*4 matrix
Pass:	Passed

Test 1.2.8:	**SP-GL12**
Type:	Structrual,Dynamic,Automated
Initial State:	Game is in Process.
Input:	N/A
Output:	Three rows are cleared
Pass:	Passed.

Test 1.2.9:	**SP-GL13**
Type:	Structrual,Dynamic,Automated
Initial State:	Game is in Process.
Input:	Playfield
Output:	True
Pass:	Passed.

Test 1.2.10:	**SP-GL14**
Type:	Structrual,Dynamic,Automated
Initial State:	State: Game is paused
Input:	N/A
Output:	No move happened
Pass:	Passed.

2 Nonfunctional Requirements Evaluation

2.1 Usability

Test 2.1.1:	**Operating System Support**
Description:	Super Tetris shall run on all major web browsers (Chrome/Firefox/Safari /IE)
Type:	Functional (dynamic, manual)
Tester(s):	Development team
Pass:	We run and play the game through Chrome/Firefox/Safari /IE successfully.

Test 2.1.2:	**Look and Feel**
Description:	The falling of the tetrimino and line clear requires animation
Type:	Functional (dynamic, manual)
Tester(s):	Development team
Pass:	When tetriminos are falling or when a row is eliminated, the animation display correctly. After we run the game hundreds of times we did not capture any incorrect animation display.

Test 2.1.3:	**Difficulty**
Description:	Super Tetris shall follow the same rule in the Classic Tetris.
Type:	Functional (dynamic, manual)
Tester(s):	Development team
Pass:	Average survey score of less than 4 in 'Difficulty' section. The average survey score in Difficulty section amony the surveys we received are less than 4.

2.2 Performance

Test 2.2.1:	**Precision**
Description:	The event shall happen at the correct time.
Type:	Functional (dynamic, manual)
Tester(s):	Development team
Pass:	The boom shall explode when it stops. The ice shall slow the falling time when it is triggered. The line shall be cleared when a complete line is made. They are all triggered at the right timing.

Test 2.2.2:	**Capacity**
Description:	Super Tetris needs less than 10MB in memory.
Type:	Functional (dynamic, manual)
Tester(s):	Development team
Pass:	Test team calculates the size of whole program file before uploading. The whole game file takes less than 2MB in memory

2.3 Robustness

If an unexpected error occurs, the system will stop running at the first time. Additionally, The system is published on a server. The server we bought is robust and it will check if the system is cracked or a dead loop is running. If it is, the server will shut the system down and notice us.

3 Comparison to Existing Implementation

By comparing to the original implementation, this system makes several improvements. First, this system has a module structure which results in several modules while the original one has only one module. Secondly, we make the system connect to the server by coding some parts in HTML.

Then, numerous functions and systems are added. Finally, our codes are well-commented while the existing code does not have any comment.

4 Unit Testing

As the system is implemented by Javascript, Mocha was used for unit testing. The test coverts most of the functions and it called 'supertetristest.js' and can be found in 'src/mochatest/test'.

5 Changes Due to Testing

While testing, we found that when a user uses the item clear random three lines the money was increased as three normal lines were eliminated. However, we planed the item costs $20 and the user gain 10$ if a line is cleared. As a result, every time the user used this item, the users money will not decreased. In the contrary, the user gain $10 every times / he used the item. For solving this problem, we increased th

6 Automated Testing

For Automated Testing, we used mocha unit test. For almost every function, it generates numerous inputs and check if the answers match the outputs and report them as pass or fail. Then it will classify where the issue comes from and what the supposed output is to make is traceable.

7 Trace to Requirements

refer to the table in the end of the document.

8 Trace to Modules

refer to the table in the end of the document.

9 Code Coverage Metrics

Mocha testing shall test 80% of the functions including edge cases. The rest of the codes are mostly in HTML and the purpose of HTML is to connect the system to the website and create the layout. As a result, it is not that meaningful to test and HTML is hard to be tested. Thus, basically, we manually test the layout part. In conclusion, the test coverage shall be above 75%.

	FQ1	FQ2	FQ3	FQ4	FQ5	FQ6	FQ7	FQ8	FQ9	FQ10	FQ11	FQ12	FQ13	FQ14
SP-IN1	1													
SP-IN2							1							
SP-IN3				1										
SP-IN4						1								
SP-IN5		1	1											
SP-IN6		1			1									
SP-IN7		1			1									
SP-IN8												1	1	1
SP-GL2										1				
SP-GL3									1			1	1	1
SP-GL4								1						
SP-GL7												1	1	1
SP-GL9										1		1	1	1
SP-GL10											1	1	1	1
SP-GL11											1	1	1	1
SP-GL12									1	1		1	1	1
SP-GL13										1				
SP-GL14					1									
SP-POC2											1	1	1	1

Figure 1: Traceability Testing Matrix respect to requirements

	M1	M2	M3	M4	M5	M6	M7	M8	M9
SP-IN1	1								
SP-IN2				1		1			
SP-IN3				1		1			
SP-IN4				1		1			
SP-IN5				1		1			
SP-IN6				1		1			
SP-IN7				1		1			
SP-IN8				1		1	1	1	1
SP-GL2					1				
SP-GL3			1	1					
SP-GL4		1	1						
SP-GL7			1		1		1		
SP-GL9			1						
SP-GL10			1	1					1
SP-GL11				1					
SP-GL12			1						
SP-GL13		1	1		1				
SP-GL14			1	1					
SP-POC2				1		1	1		

Figure 2: Traceability Testing Matrix respect to modules