Analysis of Software Artifacts

GameMatrix Project

Integration Test Plan & Report

V1.00

Team 3

Lulu Ding
Shailendra Mishra
Justin Rockwood
Wenjie(Stella) Wang
Randy Winters

Revision History

Version No.	Comments	Change Date	Changed by
V0.90	For one use case	February 7th	Stella, Lulu
V0.91	For functionality baseline	February 11th	Lulu
V0.91	Revise	February 12th	Stella
V0.92	Changes the test cases based on new features	February 15th	Stella, Lulu
V1.00	Fault tolerance	March 12th	Lulu

Test Plan

Test Strategy:

The test plan consists of test cases that fall under two categories:

✧ Basic Functionality
These tests are used to verify the basic functionality of the system.

✧ Dependability:
These tests are used to provide a baseline for fault tolerance.

Test Pass/Fail Criteria:

✧ Basic Functionality
100% passed

✧ Dependability
100% passed

✧ Performance
Get basic data for next round of testing

Test Configuration (for basic functionality):

Client:

```
C:\analysis\GameMatrix> list
C:\analysis\GameMatrix> java game.TicTacToeApplet
```

Server:

```
JBOSS_HOME\bin\run –c default
```

Test Cycle:

2 hours as estimated test cycle

Test Cases:

Test cases are described in the following sections.

Basic Functionality

Test Case #F1	
Use Case	A client, being the first player of the game server, requests to start the application
Precondition	Clean environment
Test Sequence	1. Start the server at the command prompt 2. Start the client application
Expected Result	1. An empty tic-tac-toe board 2. Logon button enabled 3. Logout, New Game, Quit Game button disabled

Test Case #F2	
Use Case	A client requests to start the application, who is the not the first player of the game server
Precondition	At least one player has started the application
Test Sequence	1. Start the client application
Expected Result	1. An empty tic-tac-toe board 2. Logon button enabled 3. Logout, New Game, Quit Game button disabled

Test Case #F3	
Use Case	A client requests to Logon, who will be the very first logged-in player in the game
Precondition	Player has started the application, but has not logged in yet.
Test Sequence	1. Click the Logon button 2. In the new window, enter the player's nickname (for example: Randy). 3. Click on OK button.
Expected Result	1. A new "Logon" window pops up 2. Empty player list 3. A new record with "Randy" being the primary key is created in the Database 4. New Game button Enabled

Test Case #F4	
Use Case	A client requests to Logon, there is already one player in the game
Precondition	Player has started the application, and has not logged in yet. There is at least one player in the game already
Test Sequence	1. Click the Logon Button 2. In the new window, enter the player nickname (for example: Justin). 3. Click on OK button.

Expected Result	1. A new "Logon" window pops up 2. Should see all the available players except the player himself 3. A new record with "Justin" being the primary key is created in the Database.

Test Case #F5	
Use Case	This test case involves the client requesting to log on using the nick name that already used by another player
Precondition	There are at least one player logged in
Test Sequence	1. Click the Logon Button 2. Enter the this player's nickname (which is the same as the name of a player who has already logged in) 3. Click on OK button
Expected Result	1. Should see the message "Nickname already existed." at the console 2. No change to the database

Test Case #F6	
Use Case	Player (A) requests to start a new game with the player (B) whom he has selected; B is idle (i.e., idle means the player is not in any game at this moment) and accepts the request
Precondition	There are at least two players logged into the game server
Test Sequence	1. A selects new game 2. A selects B in player list 3. A sees a "Connecting player B" message in a pop up window 4. B gets a pop up window with "A wants to play with you" message 5. B selects Accept 6. B gets a pop up window with "Connecting player A, please wait…" message 7. A and B get connected, both of them get "Game starts now!" message
Expected Result	1. Test sequence 3 2. Test sequence 4 3. Test sequence 6 4. Test sequence 7 5. Tic-tac-toe board enabled for A and B, A gets X and B gets O

Test Case #F7	
Use Case	Player (A) requests to start a new game with the player (B) whom he has selected; B is idle and declines the request
Precondition	There are at least two players logged into the game server
Test Sequence	1. A selects new game 2. A selects B in player list 3. A sees a "Connecting player B" message in a pop up window 4. B gets a pop up window with "A wants to play with you" message 5. B selects Decline 6. A gets a pop up window with "B declines the request" message

Expected Result	1. Test sequence 3 2. Test sequence 4 3. Test sequence 6 4. Tic-tac-toe board still disabled

Test Case #F8	
Use Case	Player (A) requests to start a new game with the player (B) whom he has selected; B just logs off at this instant (player A does not know this)
Precondition	There are at least one player logged into the game server
Test Sequence	1. A selects new game 2. A selects B in player list 3. At the same time, B logs off 4. A sees a “Connecting player B” message in a pop up window
Expected Result	1. Test sequence 4 2. A gets a pop up window with “Player B logs off message”

Test Case #F9	
Use Case	Player (A) requests to start a new game with the player (B) he has selected, and A cancels the request while waiting for B’s response, but at this time B has not received the invitation message.
Precondition	At least two players are logged in
Test Sequence	1. A selects new game 2. A selects B in player list 3. A sees a “Connecting player B” message in a pop up window 4. A clicks Cancel before B gets invitation
Expected Result	1. Tic-tac-toe board still disabled 2. B gets invitation message 3. B gets “A canceled the request” message

Test Case #F10	
Use Case	Player (A) requests to start a new game with the player (B) he has selected, and A cancels the request while waiting for B’s response
Precondition	At least two players are logged in
Test Sequence	1. A selects new game 2. A selects B in player list 3. A gets a pop up window with “Connecting player B” message 4. A clicks Cancel after B gets invitation
Expected Result	1. Tic-tac-toe board still disabled 2. B gets invitation message 3. B gets “A canceled the request” message

Test Case #F11	
Use Case	Player A and Player B are in a game. A quits during the game
Precondition	Game has started
Test Sequence	1. A selects quit 2. A gets “You are in the game, do you want to quit?”

	3. A selects Yes button
Expected Result	1. Tic-tac-toe board of A cleared and disabled 2. B gets "A quits the game" message 3. B selects OK button 4. Tic-tac-toe board of A is cleared and disabled

Test Case #F12	
Use Case	Player A and Player B are in a game. A logs out during the game
Precondition	Game started
Test Sequence	1. A selects Logout 2. A gets "You are in the game, do you want to log out?" 3. A selects Yes button
Expected Result	1. Tic-tac-toe board of A cleared and disabled 2. B gets "A logged out" message 3. B selects OK button 4. Tic-tac-toe board of A is cleared and disabled

Test Case #F13	
Use Case	The game server logic judges the result of the game
Precondition	N.A.
Test Sequence	White box unit test
Expected Result	The logic can judge the result correctly.

Test Case #F14	
Use Case	Player A and Player B have finished the game. A won
Precondition	Both players in the game The game server logic judges the result of the game correctly (test case #15)
Test Sequence	1. Plays a game to let A win
Expected Result	1. A gets a "You won" message 2. B gets a "You lost" message 3. Both tic-tac-toe boards are cleared and disabled

Test Case #F15	
Use Case	Player A and Player B have finished the game. B won
Precondition	Both players in the game The game server logic judges the result of the game correctly (test case #15)
Test Sequence	1. Plays a game to let B win
Expected Result	4. A gets a "You lost" message 5. B gets a "You won" message 6. Both tic-tac-toe boards are cleared and disabled

Test Case #F16	
Use Case	Player A and Player B have finished the game. The game tied
Precondition	Both players in the game The game server logic judges the result of the game correctly (test case #15)
Test Sequence	1. Plays a game to have a tie at the end
Expected Result	1. A gets a "Game ties" message 2. B gets a "Game ties" message

	3. Both tic-tac-toe boards are cleared and disabled

Test Case #F17	
Use Case	Player B accepts A's invitation and waits for connection. B cancels before the game starts
Precondition	A and B are both logged on. A is not in any game.
Test Sequence	1. A selects new game 2. A selects B in player list 3. A sees a "Connecting player B" message in a pop up window 4. B gets a pop up window with "A wants to play with you, if you want to play with him/her, current game will be ended" message 5. B selects Accept 6. B gets a pop up window with "Connecting player A, please wait…" message 7. B selects Cancel
Expected Result	1. A gets "B canceled" message 2. Both tic-tac-toe boards are still disabled

Test Case #F18	
Use Case	Player wants to refresh the player list
Precondition	Any time after application is started
Test Sequence	1. Player selects Refresh button
Expected Result	1. Player list is updated to the most current version

Test Case #F19	
Use Case	Player list refreshes automatically
Precondition	Any time after the application has started
Test Sequence	1. Add a new player
Expected Result	1. The player list is refreshed automatically at appropriate time intervals and any newly added players will appear in the list

Test Case #F20	
Use Case	The behavior of Nickname text field
Precondition	Logon Window pops up
Expected Result	1. Only accepts strings of no more than 25 characters 2. Only accepts strings consisting of alphanumeric symbols 3. If the input is illegal, a hint will pop up to guide the player

Test Case #F21	
Use Case	The behavior of Logon button
Expected Result	1. The Logon button is enabled after a player has started the application and it is disabled after the player has logged on 2. Logon button and logout button are never enabled/disabled at the same time

Test Case #F22	
Use Case	The behavior of Logout button

Expected Result	1. The Logout button is always enabled after the player has logged in 2. Logout button and Logon button are never enabled/disabled at the same time 3. If the Logout button is selected when player is in the game, a pop up window with message "You are in the game, do you want to logout" will pop up 4. If the Logout button is selected when player is not in the game, the player will be logged off without being hinted

Test Case #F23	
Use Case	The behavior of New game button
Expected Result	1. New game button is always enabled when player logs in successfully and one player in the player list is selected

Test Case #F24	
Use Case	The behavior of Quit button
Expected Result	1. Quit button is enabled when the player is in the game, and is disabled in other situations 2. If the Quit button is selected when player is in the game, a pop up window with message "You are in the game, do you want to quit" will pop up

Test Case #F25	
Use Case	The behavior of Refresh button
Expected Result	1. Refresh button is always enabled after player has logged in

Test Case #F26	
Use Case	The behavior of player list
Expected Result	1. Player list should not support multi-selection 2. Scroll bar is enabled when the viewable size is not enough to show all the players 3. Player list only shows the players that are not in any game (the player himself should not be in the list)

Test Case #F27	
Use Case	The behavior of tic-tac-toe board
Expected Result	1. Empty board is shown but not enabled when player is not in the game 2. The player who initiates the game gets X, and the player who accepts the game invitation gets O 3. If the game is terminated, the board should be disabled and cleared

Test Case #F28	
Use Case	The behavior of Cancel button in any pop up window
Expected Result	1. No impact on the system except in the "waiting for connected window" 2. In "waiting for connect window", pressing Cancel will stop the connection attempt between two players.

Dependability

- **Fault tolerance**

  Environment descriptions:

  - Which servers have you replicated?
    [Put your answer here.]
  - Which is the primary and which is the backup?
    [Put your answer here.]
  - What should we expect to see on a successful fault tolerance run?
    [Put your answer here.]

Test Case #D1	
Use Case	Injecting a process crash fault
Precondition	Both replicas run properlly
Test Sequence	Kill primary replica process during login
Expected Result	1. Client should detect (and report) the crash of the primary replica 2. Client should visibly start communication with the backup replica 3. Backup replica should communicate with the database (or third-tier) 4.

Test Case #D2	
Use Case	Injecting a process crash fault
Precondition	Both replicas run properlly
Test Sequence	Kill primary replica process during setting up connection
Expected Result	1. Client should detect (and report) the crash of the primary replica 2. Client should visibly start communication with the backup replica 3. Backup replica should communicate with the database (or third-tier) 4.

Test Case #D3	
Use Case	Injecting a process crash fault
Precondition	Both replicas run properlly
Test Sequence	Kill primary replica process during the game
Expected Result	1. Client should detect (and report) the crash of the primary replica 2. Client should visibly start communication with the backup replica 3. Backup replica should communicate with the database (or third-tier) 4.

Test Case #D4	
Use Case	Injecting a process crash fault
Precondition	Both replicas run properlly
Test Sequence	Kill primary replica process after the game but before the start of the new game
Expected Result	1. Client should detect (and report) the crash of the primary replica 2. Client should visibly start communication with the backup replica 3. Backup replica should communicate with the database (or third-tier)

	4.

- **Recovery**

Environment descriptions:

- Which machines are "forbidden" for the re-launch of the replica?
  [Put your answer here.]

- What should we expect to see on a successful recovery run?
  [Put your answer here.]

Test Case #D5	
Use Case	Tolerating the sequence
Precondition	The old replica has been killed, the backup replica acts as the current replica
Test Sequence	Re-launch the old primary replica during logging in Kill the current replica immediately after the old primary replica is re-launched
Expected Result	1. The old primary replica has been re-launched and re-registers with Naming server 2. The old primary replica can take over as primary, client can continue to get service 3.

Test Case #D6	
Use Case	Tolerating the sequence
Precondition	The old replica has been killed, the backup replica acts as the current replica
Test Sequence	Re-launch the old primary replica during setting up connection Kill the current replica immediately after the old primary replica is re-launched
Expected Result	1. The old primary replica has been re-launched and re-registers with Naming server 2. The old primary replica can take over as primary, client can continue to get service 3.

Test Case #D7	
Use Case	Tolerating the sequence
Precondition	The old replica has been killed, the backup replica acts as the current replica
Test Sequence	Re-launch the old primary replica during the game Kill the current replica immediately after the old primary replica is re-launched
Expected Result	1. The old primary replica has been re-launched and re-registers with Naming server 2. The old primary replica can take over as primary, client can continue to get service 3.

Test Case #D8	
Use Case	Tolerating the sequence
Precondition	The old replica has been killed, the backup replica acts as the current replica

Test Sequence	Re-launch the old primary replica after the game but before the start of the new game Kill the current replica immediately after the old primary replica is re-launched
Expected Result	1. The old primary replica has been re-launched and re-registers with Naming server 2. The old primary replica can take over as primary, client can continue to get service 3.

Exception handling

Environment descriptions:

- Which exceptions are you handling in your code?
  - SQLException
  - RemoteException
  - FinderException
  - PlayerAlreadyLoggedOnException
  - PlayerNotLoggedOnException
  - PlayerAleradyInGameException

- Which exceptions are you not handling?
  [Put your answer here.]

Test Case #D9	
Use Case	SQLException
Precondition	
Test Sequence	
Expected Result	1. Exception is handled

Test Case #D10	
Use Case	RemoteException
Precondition	
Test Sequence	Re-launch the old primary replica during setting up connection Kill the current replica immediately after the old primary replica is re-launched
Expected Result	1. Exception is handled

Test Case #D11	
Use Case	FinderException
Precondition	
Test Sequence	Re-launch the old primary replica during the game Kill the current replica immediately after the old primary replica is re-launched
Expected Result	1. Exception is handled

Test Case #D12	

Use Case	PlayerAlreadyLoggedOnException
Precondition	
Test Sequence	
Expected Result	1. Exception is handled

Test Case #D13	
Use Case	PlayerNotLoggedOnException
Precondition	
Test Sequence	
Expected Result	1. Exception is handled

Test Case #D14	
Use Case	PlayerAleradyInGameException
Precondition	
Test Sequence	
Expected Result	1. Exception is handled

Test Report

Case No	Result	Description
F1	Pass	
F2	Pass	
F3	Pass	
F4	Pass	
F5	Pass	
F6	Pass	
F7	Fail	If B declines A's request, A cannot see the declination notification. However, A can start another new game without warning
F8	Pass	
F9	Pass	
F10	Pass	
F11	Pass	
F12	Pass	
F13	Pass	
F14	Fail	Sometimes, the "win", "tier", or "lost" message shows before the last step is showed
F15	Pass	
F16	Pass	
F17	Pass	
F18	Not tested	Refresh button not supported yet
F19	Pass	
F20	Fail	Name length restriction not supported yet
F21	Pass	
F22	Pass	
F23	Pass	
F24	Pass	
F25	Fail	Refresh button not supported yet
F26	Fail	The player list cannot refresh automatically (only refreshed when user selects new game). If the user doesn't select "Logoff" before he terminates the game by using the "X" button, the player cannot be removed from database. If the player selects to play with the player that is not in the game, no warning is showed. (A message indicates connecting with <unknown>)
F27	Fail	If the game is terminated, the board is disabled but not cleared
F28	Fail	"Cancel" during connection is not supported yet. (In order to achieve this, a pop up window is needed besides showing it on status bar)

Case No	Result	Steps that lead to error (if fail)
D1		
D2		
D3		
D4		
D5		

D6		
D7		
D8		
D9		
D10		
D11		
D12		
D13		
D14		