**Software Test Plan**

**for**

**ZetaFish**

Version 1.5

December 12, 2010

Prepared by:

Patrick Michalina Chad Albrecht  
Malick Noor



**Team Zeta Dragons**

CSC 478B - Software Engineering Capstone

Fall 2010

University of Illinois at Springfield



Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Malick Noor | 10/20/2010 | Initial Version | 1.0 |
| Malick Noor | 11/2/2010 | Rework section 4 and add section 5 | 1.1 |
| Malick Noor | 11/13/2010 | Remove section 4 and 5 and rework on section 3 | 1.2 |
| Malick Noor | 12/8/2010 | Rework Section 2 and 3 | 1.3 |
| Malick Noor | 12/10/2010 | Rework Section 3.3,3.9.1 and 3.9.2 | 1.4 |
| Chad Albrecht | 12/12/2010 | Fixed spelling and grammar errors. | 1.5 |

Table of Contents

[1 A Brief Introduction of the Software 1](#_Toc279934799)

[1.1 About Software 1](#_Toc279934800)

[1.2 Functional Overview 1](#_Toc279934801)

[2 A Brief Review of the Software Requirements 1](#_Toc279934802)

[3 Test Plan 2](#_Toc279934803)

[3.1 Test 1: 2](#_Toc279934804)

[3.1.1 Input 2](#_Toc279934805)

[3.1.2 Test 2](#_Toc279934806)

[3.1.3 Result 3](#_Toc279934807)

[3.2 Test 2: 3](#_Toc279934808)

[3.2.1 Test 3](#_Toc279934809)

[3.2.2 Test Cases 3](#_Toc279934810)

[3.2.3 Output 3](#_Toc279934811)

[3.3 Test 3: 3](#_Toc279934812)

[3.3.1 Input 3](#_Toc279934813)

[3.3.2 Test Cases 3](#_Toc279934814)

[3.3.3 Output 4](#_Toc279934815)

[3.4 Test 4: 4](#_Toc279934816)

[3.4.1 Test 4](#_Toc279934817)

[3.4.2 Output 4](#_Toc279934818)

[3.5 Test 5: 4](#_Toc279934819)

[3.5.1 Input 4](#_Toc279934820)

[3.5.2 Output 4](#_Toc279934821)

[3.6 Test 6: 4](#_Toc279934822)

[3.6.1 Input 4](#_Toc279934823)

[3.6.2 Output 5](#_Toc279934824)

[3.7 Test 7: 5](#_Toc279934825)

[3.7.1 Test 5](#_Toc279934826)

[3.8 Test 8: 5](#_Toc279934827)

[3.8.1 Test 5](#_Toc279934828)

[3.9 Test 9: 5](#_Toc279934829)

[3.9.1 Test 5](#_Toc279934830)

[3.9.2 Result 5](#_Toc279934831)

[3.10 Test 10: 5](#_Toc279934832)

[3.10.1 Test 5](#_Toc279934833)

[3.11 Test 11: 6](#_Toc279934834)

[3.11.1 Test 6](#_Toc279934835)

[3.12 Test 12: 6](#_Toc279934836)

[3.12.1 Test 6](#_Toc279934837)

[3.13 Test 13: 6](#_Toc279934838)

[3.13.1 Test 6](#_Toc279934839)

[3.14 Test 14: 6](#_Toc279934840)

[3.14.1 Test 6](#_Toc279934841)

[3.14.2 Result 7](#_Toc279934842)

[3.15 Test 15: 7](#_Toc279934843)

[3.15.1 Test 7](#_Toc279934844)

[3.16 Test 16: 7](#_Toc279934845)

[3.16.1 Test 7](#_Toc279934846)

# A Brief Introduction of the Software

## About Software

ZetaFish is an implementation of the classic card game of Go Fish. The software will allow 2-6 players to enjoy a fun game without the hassle of cleaning up the cards. The software is being built as part of group project in the CSC 478 Software Engineering Capstone course at UIS.

## Functional Overview

The game of ZetaFish will provide all the functions you would find in a real game of Go Fish. With the computer as the dealer, players will be able to request cards from opponents, receive cards from opponents if they have the requested card and “go fish” from the ocean of cards if they do not. Mechanisms for starting and stopping a game as well as re-dealing will be provided.

# A Brief Review of the Software Requirements

ZetaFish has many specifications and requirements. The test plan has been designed to validate this software and check that all the requirements specified in Requirement Document are fulfilled.

The following table shows all requirements for which the test plan is designed.

|  |  |
| --- | --- |
| **Requirement # as documented in SRS Document** | **Requirements to be tested:** |
| 3.1.1 | The game will be graphical in nature. |
| 3.1.2 | The game should allow 2-6 players |
| 3.1.3 | Game will provide a means to establish and terminate a connection to other players. |
| 3.1.4 | Once the game has started, no new players will be allowed to join. |
| 3.1.5 | The game provide for a restart. |
| 3.1.6 | The game will provide for a re-deal. |
| 3.1.7 | The game will keep score. |
| 3.1.8 | Each book will be worth 1 point |
| 3.1.9 | If a player abnormally exits, the game should continue and the player’s cards will be returned to the ocean. |
| 3.1.10 | The game shall provide a means of visually showing each player the number of cards held in other players hands. |
| 3.1.11 | The game shall provide a means of allow a player to request a card from an opponent. |
| 3.1.12 | The game shall provide a means of automatically transferring a card (if held) to a player from an opponent. |
| 3.1.13 | The game shall provide a means of automatically sending a player to the ocean in the case where a card is not held by an opponent. (Go Fish) |
| 3.1.14 | The game shall provide a means of detecting the end of a player’s turn and pass it to the next player. |
| 3.1.15 | A standard deck of 52 playing cards will be used. |
| 3.1.16 | Cards should be easily viewable on a 1024 x 768 resolution color display. |

# Test Plan

## Test 1:

Testing Requirement # 3.1.1 of SRS, i.e., game will be graphical in nature.

### Input

Start the software and see how it look and try to use it.

### Test

Test to check that the whole software is graphical in nature. Software is considered to be graphical if it does not require its user to learn any kind of command language. There should be graphics in the software and mouse and key board is used as an input device

### Result

Test passes if the software fulfills the graphical interface requirement, that is, it does not require user to use any command language, there should be graphics instead of only text and mouse and keyboard is used as an input device

## Test 2:

Testing Requirement # 3.1.2 of SRS, i.e., the game should allow 2-6 players

### Test

There is a limit for minimum and maximum number of players. This test is designed to check that game should allow 2 to 6 players and should not allow more than 6 and less than 2 players

### Test Cases

1. Test for one player
2. Test for 2 players
3. Test for 3 players
4. Test for 4 players
5. Test for 5 players
6. Test for 6 players
7. Test for more than 6 players i.e 7th player

### Output

The game is designed for 2 to 6 players, so if the number of players is within the permitted range, games precede.

In case of one player, the game would not start and remain disabled until the second player joins the game session.

Game should not allow the 7th player to join the game session.

## Test 3:

Testing Requirement # 3.1.3 of SRS, i.e., Game will provide a means to establish and terminate a connection to other player.

### Input

1. Click on the player’s hand with which we want to connect.
2. Send a message through chat pane.

### Test Cases

1. Test to establish a connection with a player who is not connected.
2. Test to terminate a connection with a player who is previously connected.
3. Send a message and see if all the other players get it or not.

This test is to be performed with 2, 3, 4, 5 and 6 players setting.

### Output

This test passes if the player can terminate and establish connections with other players. When a player connects to a new game, barring any validation issues, he/she is allowed into the game.

For an active game, players would remain connected to each other throughout the game.

A second type connection is through chatting, any player can send a message to all the other players throughout the duration of the game.

## Test 4:

Testing the requirement # 3.1.4 of SRS, i.e., once the game has started, no new players will be allowed to join.

### Test

Once the game is started, try to add a new player.

This test is performed in 2, 3, 4, 5, 6 players setting.

### Output

This test passes if the player is not allowed (rejected) to join the game once the game started.

## Test 5:

Testing the requirement # 3.1.5 of SRS, i.e., the game provide for a restart.

### Input

Click the “Start New Game” button.

This test is performed in 2, 3, 4, 5, 6 players setting.

### Output

This test passes if the game is restarted when the button is clicked. If the game doesn’t restart, the test fails.

## Test 6:

Testing the requirement # 3.1.6 of SRS, i.e., the game will provide for a re-deal.

### Input

Click the “Start New Game” button.

This test is performed in 2, 3, 4, 5, 6 players setting.

### Output

By clicking the button, the game should re-deal. The test doesn’t pass, if the game fails to re-deal.

## Test 7:

Testing the requirement # 3.1.7 of SRS, i.e., the game will keep score.

### Test

To check this requirement, the tester should have to play the game and check whether the software keeps track of the score or not.

## Test 8:

Testing the requirement # 3.1.8 of SRS, i.e., each book will be worth 1 point.

### Test

To check this requirement, tester should have to play the game and have to check what each book worth.

## Test 9:

Testing the requirement # 3.1.9 of SRS, i.e., if a player abnormally exits, the game should continue and the player’s cards will be returned to the ocean.

### Test

1. Play two player game and let one player abnormally exit the game.
2. Play 3 or more player game and let one player exit the game abnormally.
3. Play 3 or more player game and let the host (player who started the game) exit the game abnormally.

### Result

1. The game should end as the minimum requirement for the game is 2 players.
2. The game continues, and the player’s card will return to the Ocean.
3. The game should come to an end.

## Test 10:

Testing the requirement # 3.1.10 of SRS, i.e., the game shall provide a means of visually showing each player the number of cards held in other players hands.

### Test

To check this feature, tester has to play the game in a 2 players, 3 players, 4 players, 5 players and 6 players setting and has to check whether he is able to see the number of cards held in other players hand or not

## Test 11:

Testing the requirement # 3.1.11 of SRS, i.e., the game shall provide a means to allow a player to request a card from an opponent.

### Test

To check this feature, tester has to play the game and see if he can make a request or not. Again, the player has to test this feature in 2 players, 3 players, 4 players, 5 players and 6 players setting.

## Test 12:

Testing the requirement # 3.1.12 of SRS, i.e., the game shall provide a means of automatically transferring a card (if held) to a player from an opponent.

### Test

During the game, when a player request for card from opponent, the game should automatically transfer the card, if held, to the requesting player.

To check this feature, tester has to play the game with 2, 3, 4, 5 and 6 players setting and see if the software is doing what it is supposed to do, i.e. transfer the card automatically.

## Test 13:

Testing the requirement # 3.1.13 of SRS, i.e., the game shall provide a means of automatically sending a player to the ocean in the case where a card is not held by an opponent. (Go Fish)

### Test

During the game, when a player request for card from opponent and the opponent don’t have that card, game should automatically send the player to the ocean.

To check this feature, tester has to play the game with 2, 3, 4, 5 and 6 players setting and see if the software is doing what it is supposed to do, i.e. send the player automatically to the Ocean.

## Test 14:

Testing the requirement # 3.1.14 of SRS, i.e., the game shall provide a means of detecting the end of a player’s turn and pass it to the next player.

### Test

Tester has to check this feature while playing the game in all 5 settings, i.e. 2, 3, 4, 5 and 6 Players setting. Tester has to check that

1. If the player doesn’t request card, instead, he passes his turn, game should detect end of player’s turn.
2. The game should detect end of player’s turn if the player get a card from his opponent or take a card from ocean.

### Result

To pass this test, the game should detect ends of player’s turn and pass the turn to the next player.

## Test 15:

Testing the requirement # 3.1.15 of SRS, i.e., a standard deck of 52 playing cards will be used.

### Test

To test this requirement, the tester has to play the game till the end and see all the check the deck by playing and seeing that none of the card is duplicated or missing

## Test 16:

Testing the requirement # 3.1.16 of SRS, i.e., cards should be easily viewable on a 1024 x 768 resolution color display.

### Test

This is a subjective test and the tester has to play game and give his reviews