

KRONOS

Acceptance Test Procedures

Brandon Smith, Nieka Gutenberg, Joseph Coppin, Ryan Frazier, Trevor Jewkes

December 2, 2016

1 Server–OLD

1.1 Server

Validation:

1. Client can connect to the server when opening the client.

Verification:

Unit Test for connecting to server.

2 Client GUI

2.1 Login

2.1.1 Create user

Validation:

1. Create a new user to 'log in' to access server.
2. Create username to identify player.
3. Create password.
4. Ability to change password if needed
5. Successfully login or logout each game.

Verification:

Unit Tests for creating user functions.

1. unit test for function createAccount()
2. unit test for function userSettings()

2.2 Main Menu

2.2.1 Main menu

Validation:

1. Menu to edit account/profile
2. Choose specific game
 - (a) Choose Hearts
 - (b) Choose Spades
3. Option to join game or create game
 - (a) In joining game, able to select specific game to join
 - (b) In creating game, choose options for gameplay
 - Ex: Choose name of private game

- Ex: Choose number of human/computer players
 - i. Different levels of AI
 - ii. Choose Private or Public

Verification:

Unit Test for choosing game, ensuring only specific options work when creating/joining games.

1. unit test for function chooseGame()
2. unit test for function gameOptions()

2.3 Game Table/GUI

2.3.1 Cards

Validation:

1. View cards, ensure cards are delt properly
2. Cards match values listed on card
3. See how many cards other users have (face down)

Verification:

Unit Tests for how cards will look, how cards are shuffled, how many cards are visible to player.

1. unit test for function createTable()
2. unit test for function showHand()
3. unit test for function drawCard()

2.3.2 Scoreboard

Validation:

1. See scores/bid
2. Reflects appropriate score at the end of each turn
3. Players info and cards sent each time a player selects a card.

Verification:

Unit Tests for ensuring correct information is passed to scoreboard.

1. unit test for function displayCenter()

2.3.3 Moves

Validation:

1. See when other players move
2. Each turn is shown with appropriate amount of time
3. Can see past hand from option menu

Verification:

Unit Test for ensuring correct information is sent to display.

1. unit test for function redrawTable()
2. unit test for function getHint()
3. unit test for function viewHistory()

2.4 Game End Event

Validation:

1. If connection removed prematurely, AI will fill in so gameplay will continue without problems
2. End at proper time
3. Score for round displays on screen
4. Option to play again
5. Option to return to main menu

Verification:

Unit Test for making sure all premature exits are taken care of by creating and using AI.
Unit Test for making sure each function dealing with new game works correctly.

1. unit test for function endOfGameDisplay()

2.5 Other

2.5.1 Multiplayer Options

Validation:

1. AI fills in as needed
2. 1-3 other humans can join game
3. Disconnecting user is replaced with AI that takes place
4. After determined amount of time of inactivity, player is removed from game
5. When disconnected accidentally, can rejoin same game if game is still in play

Verification:

Unit Test for ensuring that multiplayer functions work as expected.