**Team 13 Test Plan**

Peter Xeros

Adam McDonald

Avi Agarwal

Maxwell Jones

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 001 -** Checking User Interface  **System:** Cards Against Humanity Bot  **Module:** User Interface  **Severity:** 1 |
| **Instructions** | 1. Run the jar file 2. Select options in the menu 3. Click the back to menu 4. Select play |
| **Expected result** | * Verify that a options menu did pop up * Verify that the a window with the bot playing the game did open |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 002 -** Database of Cards  **System:** Cards Against Humanity Bot  **Module:** Database  **Severity:** 1 |
| **Instructions** | 1. ssh into the system with the database 2. Move to the correct directory 3. Open mysql 4. Use MySQL *show tables* command |
| **Expected result** | * Verify that there is a white card table * Verify that there is a black card table * Verify that there is a card combinations table |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 003 -** Identifying elements In webpage  **System:** Cards Against Humanity Bot  **Module:** Web interface  **Severity:** 1 |
| **Instructions** | 1. Run the jar file 2. Select play in the menu 3. Look at the the console |
| **Expected result** | * Verify that the console does contain “identifying elements...” * Verify that the console does contain “elements found” |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 004 -** Distinguish game components from other webpage elements  **System:** Cards Against Humanity Bot  **Module:** Web Interface  **Severity:** 1 |
| **Instructions** | 1. Run the jar file 2. Select play in the menu 3. Look at the the console |
| **Expected result** | * Verify that the console does contain “game state discerned from elements” * Verify from the console that the program is correctly reading the webpage |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 005 -** Run AI in neutral mode where the AI does not modify the database  **System:** Cards Against Humanity Bot  **Module:** Web Interface  **Severity:** 3 |
| **Instructions** | 1. Run the jar file 2. Select options from the menu 3. Uncheck the box that says update database 4. Select play in the menu 5. Wait for a round to finish 6. Look at the the console |
| **Expected result** | * Verify that the console says “round ended” * Verify that the console does not say “calculating new weightages” * Verify that the console did not say “tables updated” |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 006 -** AI can read and write from Database  **System:** Cards Against Humanity Bot  **Module:** Database  **Severity:** 1 |
| **Instructions** | 1. Run Jar File. 2. Select Offline mode 3. Give fake inputs through Unit testing 4. Look at database to see if updated 5. Ask for output for the best combination via Unit Testing |
| **Expected result** | * Verify that the tables are formatted correctly. * Verify the output from the AI is the right card |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 007 -** AI interacts with online interface  **System:** Cards Against Humanity Bot  **Module:** Web Interface  **Severity:** 1 |
| **Instructions** | 1. Run the jar file 2. Select play in the menu 3. Look at the the console |
| **Expected result** | * Verify that the console does contain “control passed to AI” |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 008 -** AI functions in an offline mode  **System:** Cards Against Humanity Bot  **Module:** UI  **Severity:** 2 |
| **Instructions** | 1. Run the jar file 2. Select play offline in the menu 3. Input your hand and the black card 4. Click go |
| **Expected result** | * Verify that the window now displays the name of the recommended card |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 009 -** AI can handle bugs in SQL database  **System:** Cards Against Humanity Bot  **Module:** Database  **Severity:** 2 |
| **Instructions** | 1. Insert a bug (SQL injection, etc.) into the database you have made for the AI 2. Run the AI and verify that the right card is picked via unit testing |
| **Expected result** | * Verify that, regardless of the bug, the output is the same |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 010 -** AI can determine the best possible white card for the current round by weighting card combinations  **System:** Cards Against Humanity Bot  **Module:** Web Interface  **Severity:** 1 |
| **Instructions** | 1. Run the jar file 2. Select play in the menu 3. Wait a round 4. Look at the the console |
| **Expected result** | * Verify that the console does contain the list of all cards found by the program. * Verify that the list contains the saved values of the cards. * Verify that the AI chose the one with the highest number |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 011 -** AI can take advantage of exchanging its hand for a black card rule.  **System:** Cards Against Humanity Bot  **Module:** AI  **Severity:** 3 |
| **Instructions** | 1. Set the black card rule 2. Give the AI a hand that would require the use of the black card rule via Unit Testing 3. Run AI |
| **Expected result** | * The Black card count should go down by 1 and a request for a new deck should be sent |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 012 -** The AI can decisively pick between card combinations of equal weight  **System:** Cards Against Humanity Bot  **Module:** AI  **Severity:** 3 |
| **Instructions** | 1. Give the AI a hand where the best pick is a tie via unit testing 2. Have the AI pick via unit testing |
| **Expected result** | * Verify that a response is given |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 013 -** The AI can adapt to different senses of humour  **System:** Cards Against Humanity Bot  **Module:** AI  **Severity:** 3 |
| **Instructions** | 1. Start a new game with the database and the hands given by mock input 2. Run a unit test of several games where the card is chosen is always a card with a specific category. |
| **Expected result** | * Verify that the AI will start to choose cards by this category even if they have a lower weight |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 014 -** Database keeps track of in-game statistics  **System:** Cards Against Humanity Bot  **Module:** Database  **Severity:** 2 |
| **Instructions** | 1. Run the jar file 2. Select play in the menu 3. Wait a few rounds |
| **Expected result** | Verify that statistics about the game are listed in the window |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 015 -** AI knows when to play a throw away round.  **System:** Cards Against Humanity Bot  **Module:** AI  **Severity:** 3 |
| **Instructions** | 1. Provide the AI with a black card with low weight and a deck of cards with one card that has a notably low average weight via Unit Testing 2. Run the round |
| **Expected result** | * Card |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 016 -** Options menu that users can dictate AI functionality  **System:** Cards Against Humanity Bot  **Module:** UI  **Severity:** 3 |
| **Instructions** | 1. Run the jar file 2. Select options in the menu |
| **Expected result** | * Verify that there is an options menu displayed |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 017 -** An AI that can understand the overarching state of the game.  **System:** Cards Against Humanity Bot  **Module:** AI  **Severity:** 2 |
| **Instructions** | 1. Run the jar file 2. Select play in the menu 3. Wait a round 4. Look at the the console |
| **Expected result** | * Verify that cards are identified in the console * Verify that the current state of the game is displayed in the console * Verify that the AI is playing the game on the website |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 018 -** AI can handle black cards that require more than 1 white card inputs  **System:** Cards Against Humanity Bot  **Module:** AI  **Severity:** 2 |
| **Instructions** | 1. From UI menu, select play 2. Spectate the AI until you see it handle a multiple input black card in a humorous way |
| **Expected result** | * Verify that the AI makes a logical choice of white cards for the given black card |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 019 -** Options menu that allows the user to tell the AI to switch games  **System:** Cards Against Humanity Bot  **Module:** UI  **Severity:** 3 |
| **Instructions** | 1. Run the Jar file. 2. Have the User choose the option for the AI to switch games automatically 3. Have the AI play a game to completion |
| **Expected result** | * After the game is completed, the user should come back to see the AI in a different lobby. |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 020 -** AI keeps track of wins and losses using the database  **System:** Cards Against Humanity Bot  **Module:** Database  **Severity:** 3 |
| **Instructions** | 1. Run the jar file 2. Select options from the menu 3. Uncheck “find a new game when done” 4. Select play in the menu 5. Let the AI play the game 6. ssh into the system with the database 7. Move to the correct directory 8. Open mysql 9. View gamesWonTable |
| **Expected result** | * Verify that total games won is listed in this table |

|  |  |
| --- | --- |
| **Identification and**  **classification** | **Test Case 021 -** AI knows how to manage seeing an unrecognized white or black card.  **System:** Cards Against Humanity Bot  **Module:** Database  **Severity:** 3 |
| **Instructions** | 1. Run the jar file 2. Select options from the menu 3. Uncheck “find a new game when done” 4. Select play in the menu 5. Let the AI play the game 6. ssh into the system with the database 7. Move to the correct directory 8. Open mysql 9. View gamesWonTable |
| **Expected result** | * Verify that total games won is listed in this table |