# **Defect Logs**

Design Inspection Defect Log

|  |  |
| --- | --- |
| **Module** | **Developer** |
| Main Menu | Trevor, Sean |
| Rules | Trevor |
| Settings | Trevor |
| Game | Sean, Dan, Scott, Trevor |
| AI | Dan, Sean |
| Player | Dan, Scott |
| Lookup Tables | Dan |
| Card | Scott |

|  |  |
| --- | --- |
| Product | BlackJack Game (Group 1) |
| Date | 2/12/2016 - 2/13/2016 |
| Authors | Scott McClellan, Sean Kelley, Trevor Edris, Dan Sokoler |
| Moderator | Sean Kelley |
| Inspectors | Scott McClellan, Sean Kelley, Trevor Edris, Dan Sokoler |
| Recorders | Scott McClellan, Sean Kelley, Trevor Edris, Dan Sokoler |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Description | Severity | Fix | Module |
| 1 | No differentiation between 10, J, Q, K | 1 | Increased the range of values for cards from 2-10 to 1-13 where 1 is Ace, and 11-13 is J-K | Deck |
| 2 | No way to distinguish between an Ace being 1 or 11 | 1 | Bust method assumes Ace is 11 until the player goes over 21, then assume it is a 1 | AI |
| 3 | CPU “saw” the wrong card, was viewing what was technically the facedown card | 2 | Changed the card viewed from the first one to the second one | AI |
| 4 | CPU accessed the wrong index in its lookup table, off by 1 | 2 | Fixed the off by one error | AI |
| 5 | CPU only checked the first card when choosing which lookup table to use | 2 | Included an OR to look at the second card in its initial hand | AI |
| 6 | Row/Column numbering bug in CPU soft lookup table access | 1 | Swap row/column accessor indexes | AI |
| 7 | Row/Column numbering bug in CPU hard lookup table access | 1 | Swap row/column accessor indexes | AI |
| 8 | Soft lookup table used the wrong card for Ace checking | 1 | Created if/else to check which card should be used | AI |
| 9 | ArrayIndexOutOfBounds upon 2nd hit from player | 1 | Redo array index calculation | AI |

Testing Defect Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Description | Severity | Fix | Module |
| 1 | Upon starting the program, entering ‘2’ to start game, the cards are printed and an extra newline is observed in the representation of the card | 3 | changed println() to print() in the printing function | Engine |
| 2 | Start the program, enter ‘5’ in the main menu, Null pointer exception occurs | 1 | Added a default case in the switch statement for out of bounds as well as an extra check for non integer input | Engine |
| 3 | After playing the game a bit, an extra space is observed when printing a card with a 10 as the value | 1 | Removed a space from the print statement when printing ‘10’s | Engine |
| 4 | Begin playing the game, enter ‘3’ continuously for several rounds, the game will not end if option 3 is used | 1 | option 3 is a placeholder with no functionality at the moment, future development will rectify | Engine |
| 5 | After playing the game for a while, it was clear that certain card values were extremely more common than others | 3 | Seed for Deck Shuffling was static; Used System.nanoTime to create random seed for deck order shuffling | Deck |
| 6 | Start program, enter ‘2’ to start game, first two cards of player hand were “face down” when printed | 2 | alter printHand() method to display since it was copy pasted from printTable() | Engine |
| 7 | Settings page will randomly take in a newline before taking input | 2 | create a new scanner specifically for settings method | Engine |
| 8 | Human is null when starting a second (or more) game | 1 | Instead of setting human to null the human player’s hand is cleared | Engine/Player |
| 9 | Human player’s hand is not cleared between games | 2 | Wrote a method to clear a single player’s hand | Engine/Player |
| 10 | List of CPUs is not reset between games resulting in CPUs with uncleared hands | 2 | Reset CPU list to hold blank CPUs after each game | Engine |
| 11 | Winners list is set to null in between games, resulting in a NullPointerException | 1 | Winners list is reset to blank arraylist after each game instead of null | Engine |