DU-Paths and Test Cases

**Note:** These are final DU-Paths. Else there were multiple DU-paths according to c-use and p-use. But, these are the complete finalized paths. That did coverage of other paths as well. And (Test Pass) means that if a loop shall be discontinue then it did and if it should leave the loop due to condition then it left.

# blastUISpecialGem()

* Variables
  1. Index
  2. k
  3. X\_coordinate
  4. Y\_coordinate
  5. This.listOfGems
  6. This.blast\_coordinates[index][0]
  7. This.blast\_coordinates[index][1]
* DU-Paths
  1. 3->3(1)->3(2)->6->7->8->9->10->11->**3(1)**->14->15
  2. 2->3->4->5->6->7->8->9->10->11->**3(1)->3(2)->**3(1)->14->15
* Test Cases
  1. PATH-1:
     + K<64 (Test Pass)-loop enter
     + K>64 (Test Pass)-loop halt
  2. PATH-2:
     + [K<64&&this.listOfGems.get(k).getValue()==6](Test Pass)
     + [K>64&&this.listOfGems.get(k).getValue()==6](Test Pass)
     + [K<64&&this.listOfGems.get(k).getValue()!=6](Test Pass)
     + [K>64&&this.listOfGems.get(k).getValue()!=6](Test Pass)
     + [K==64&&this.listOfGems.get(k).getValue()==6](Test Pass)
     + [K==64&&this.listOfGems.get(k).getValue()!=6](Test Pass)

# buttonHandler()

* Variables
  1. Source
  2. i
  3. j
* DU-Paths
  1. 5->5(1)->6(1)->7(1)->9->10->11
* Test Cases
  1. PATH-1:
     + [i<8 && j<8 && source==tiles[i][j] ] (Test Pass)
     + [~(i<8) && j<8 && source==tiles[i][j] ] (Test Pass)
     + [i<8 && ~(j<8) && source==tiles[i][j] ] (Test Pass)
     + [~(i<8) && ~(j<8) && source==tiles[i][j] ] (Test Pass)
     + [i<8 && j<8 && source!=tiles[i][j] ] (Test Pass)
     + [~(i<8) && j<8 && source!=tiles[i][j] ] (Test Pass)
     + [i<8 && ~(j<8) && source!=tiles[i][j] ] (Test Pass)
     + [~(i<8) && ~(j<8) && source!=tiles[i][j] ] (Test Pass)

# isValidMove()

* Variables
  1. this.coordinates[0][0]
  2. this.coordinates[0][1]
  3. this.coordinates[1][0]
  4. this.coordinates[1][1]
* DU-Paths
  1. 2->3->4
  2. 7->8->9
  3. 12->13->14
  4. 17->18->19
  5. 2->3->7->8->12->13->17->18->22
* Test Cases
  1. PATH-1:
     + ((this.coordinates[0][0] - 1) >= 0) && ((this.coordinates[0][0] - 1) == this.coordinates[1][0] && this.coordinates[0][1] == this.coordinates[1][1]) (Test Pass)
     + ~((this.coordinates[0][0] - 1) >= 0) && ((this.coordinates[0][0] - 1) == this.coordinates[1][0] && this.coordinates[0][1] == this.coordinates[1][1]) (Test Pass)
     + ~((this.coordinates[0][0] - 1) >= 0) && ~((this.coordinates[0][0] - 1) == this.coordinates[1][0] && this.coordinates[0][1] == this.coordinates[1][1]) (Test Pass)
     + ((this.coordinates[0][0] - 1) >= 0) && ~((this.coordinates[0][0] - 1) == this.coordinates[1][0] && this.coordinates[0][1] == this.coordinates[1][1]) (Test Pass)
  2. PATH-2:
     + ((this.coordinates[0][1] - 1) >= 0) && (this.coordinates[0][0] == this.coordinates[1][0] && (this.coordinates[0][1] - 1) == this.coordinates[1][1]) (Test Pass)
     + ~((this.coordinates[0][1] - 1) >= 0) && (this.coordinates[0][0] == this.coordinates[1][0] && (this.coordinates[0][1] - 1) == this.coordinates[1][1]) (Test Pass)
     + ~((this.coordinates[0][1] - 1) >= 0) && ~(this.coordinates[0][0] == this.coordinates[1][0] && (this.coordinates[0][1] - 1) == this.coordinates[1][1]) (Test Pass)
     + ((this.coordinates[0][1] - 1) >= 0) && ~(this.coordinates[0][0] == this.coordinates[1][0] && (this.coordinates[0][1] - 1) == this.coordinates[1][1]) (Test Pass)
  3. PATH-3:
     + ((this.coordinates[0][1] + 1) < 8) && (this.coordinates[0][0] == this.coordinates[1][0] && (this.coordinates[0][1] + 1) == this.coordinates[1][1]) (Test Pass)
     + ~((this.coordinates[0][1] + 1) < 8) && (this.coordinates[0][0] == this.coordinates[1][0] && (this.coordinates[0][1] + 1) == this.coordinates[1][1]) (Test Pass)
     + ~((this.coordinates[0][1] + 1) < 8) && ~(this.coordinates[0][0] == this.coordinates[1][0] && (this.coordinates[0][1] + 1) == this.coordinates[1][1]) (Test Pass)
     + ((this.coordinates[0][1] + 1) < 8) && ~(this.coordinates[0][0] == this.coordinates[1][0] && (this.coordinates[0][1] + 1) == this.coordinates[1][1]) (Test Pass)
  4. PATH-4:
     + ((this.coordinates[0][0] + 1) < 8) && ((this.coordinates[0][0] + 1) == this.coordinates[1][0] && this.coordinates[0][1] == this.coordinates[1][1]) (Test Pass)
     + ~((this.coordinates[0][0] + 1) < 8) && (((this.coordinates[0][0] + 1) == this.coordinates[1][0] && this.coordinates[0][1] == this.coordinates[1][1]) (Test Pass)
     + ~((this.coordinates[0][0] + 1) < 8) && ~((this.coordinates[0][0] + 1) == this.coordinates[1][0] && this.coordinates[0][1] == this.coordinates[1][1]) (Test Pass)
     + ((this.coordinates[0][0] + 1) < 8) && ~((this.coordinates[0][0] + 1) == this.coordinates[1][0] && this.coordinates[0][1] == this.coordinates[1][1]) (Test Pass)
  5. PATH-5:
     + ~[((this.coordinates[0][0] - 1) >= 0) &&((this.coordinates[0][0] - 1) == this.coordinates[1][0] && this.coordinates[0][1] == this.coordinates[1][1])&& ((this.coordinates[0][1] - 1) >= 0)&& (this.coordinates[0][0] == this.coordinates[1][0] && (this.coordinates[0][1] - 1) == this.coordinates[1][1])&& ((this.coordinates[0][1] + 1) < 8)&& (this.coordinates[0][0] == this.coordinates[1][0] && (this.coordinates[0][1] + 1) == this.coordinates[1][1])&& ((this.coordinates[0][0] + 1) < 8)&& ((this.coordinates[0][0] + 1) == this.coordinates[1][0] && this.coordinates[0][1] == this.coordinates[1][1])] (Test Pass)

# processClick(i,j)

* Variables
  1. This.counter
  2. This.coordinates[0][0]
  3. This.coordinates[0][1]
  4. This.coordinates[1][0]
  5. This.coordinates[1][1]
  6. Temp
  7. this.tiles
  8. delay
  9. timer
  10. nestedTimer
  11. this.gemBoard
  12. temp2
  13. LoginFrame
  14. Delay(Outer)
  15. Timer(Outer)
* DU-Paths
  1. 1->2->3->4->5->6->7->8->9
  2. 11->12->13->14
  3. 16->17->18->19->20->21->22
  4. 50->51
  5. 51->52->53->54->55->56
  6. 17->18->19->20->21->22->23->24->25->26->27->28…48
* Test Cases
  1. PATH-1:
     + this.counter == 1 (Test Pass) Enter block
     + this.counter == 2 (Test Pass) Enter block
     + ~this.counter == 1 (Test Pass) !Enter block
     + ~this.counter == 2 (Test Pass) !Enter block
  2. PATH-2:
     + nil
  3. PATH-3:
     + this.gemBoard.findSpecialGemOccurences()-(Test Pass) Enter block
     + this.gemBoard.findSpecialGemOccurences()-(Test Pass) !Enter block
     + this.gemBoard.findOccurences()-(Test Pass) Enter loop
     + this.gemBoard.findOccurences()-(Test Pass) !Enter loop
  4. PATH-4:
     + nil
  5. PATH-5:
     + nil
  6. PATH-6:
     + this.gemBoard.findSpecialGemOccurences()-(Test Pass) Enter block
     + this.gemBoard.findSpecialGemOccurences()-(Test Pass) !Enter block
     + this.gemBoard.findOccurences()-(Test Pass) Enter loop
     + this.gemBoard.findOccurences()-(Test Pass) !Enter loop