**City University of Hong Kong**

CS3343 Software Engineering Practice

2023/24 Semester A

Bug Report

Project Title: Connect Four Game

Group 27

|  |  |
| --- | --- |
| **Name** | **Student Number** |
| Fong Tsz Wai | 57156110 |
| JI Xin | 40147182 |
| Ka Lam Mark Alexander LEE | 56759365 |
| LIN Xiaoyang | 57126676 |
| SONG Rui | 57126559 |

Bug-2701

**Title**: Crashes Caused by System Compatibility Issues

**Status**: Closed

**Description**:

To read the data in SysConfiguration.txt, we need to open this file, but the path expression used to open the file is unsuitable for all systems.

**Reproduce Step:**

1. 1-Start new game

2. 1-Human vs Human

3. Enter Name

4. Enter Symbol

5. Enter Name

6. Enter Symbol

**Type**: Compatibility

**Component**: Main

**OS**: macOS Sonoma 14.1.2

**Severity**: Critical

**Reported**: 2023-10-11 14:30:06 HKT by LIN Xiaoyang

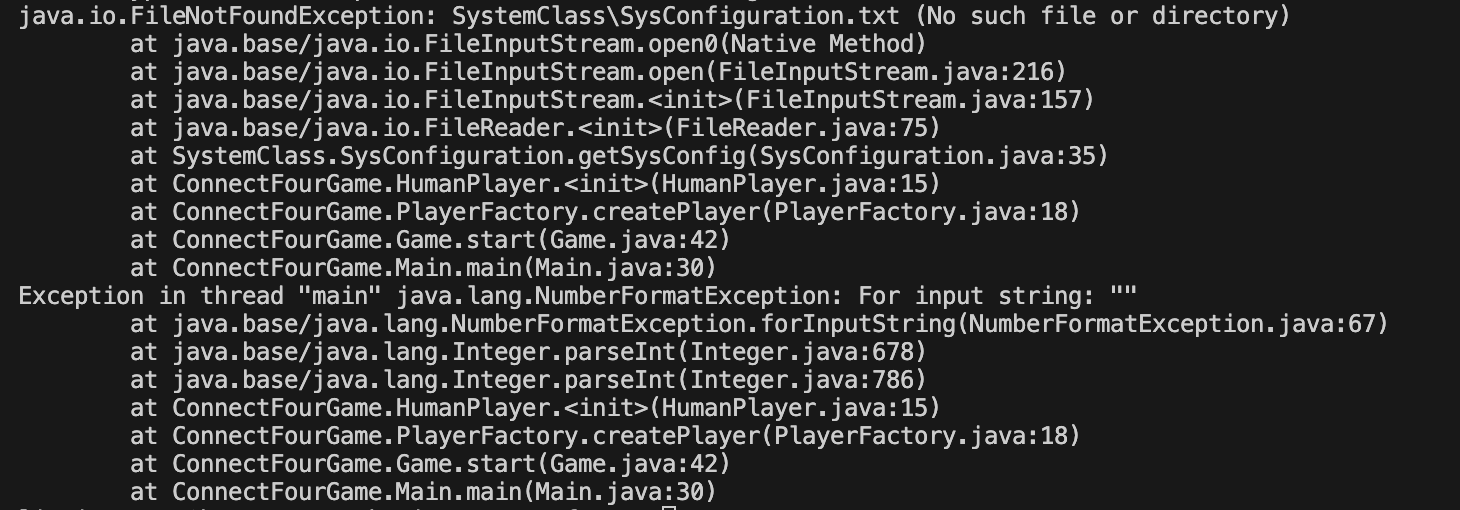
**Last Modified**: 2023-10-11 21:37:13 HKT

**Assignee**: SONG Rui

**Problem**:

When running the program using macOS Sonoma 14.1.2, the following error occurs after entering the information of both players and the symbol.

The error messages are shown in Figure 1.1, which indicates an error related to file opening.

Figure 1.1 Error Display

SONG Rui 2023-10-11 21:37:13 HKT

The debugging results after updating the code are the same as mine. I followed the bug reproducing method provided by LIN Xiaoyang and did not encounter a program crash. Considering that the operating systems used by me, and LIN Xiaoyang are different, I am using Windows 10, while LIN Xiaoyang is using macOS Sonoma 14.1.2. I tracked the error screenshot provided by LIN Xiaoyang and found that the problem may be due to the different path expressions between macOS and Windows.

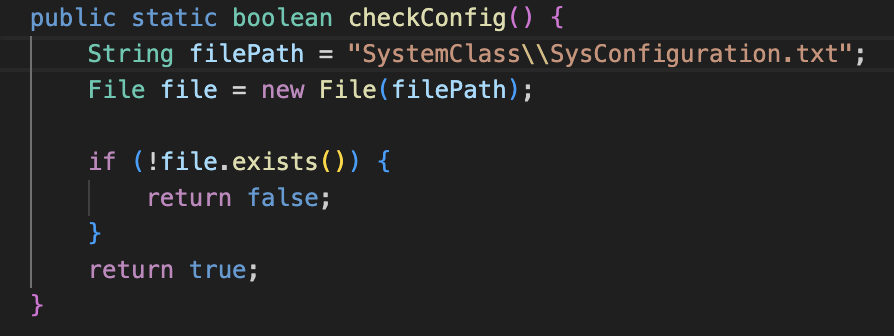


Figure 1.2 Code with Bug

Figure 1.2 shows the code with bug. To ensure compatibility with all systems, I used FILE\_SEPARATOR to make the file path expression valid for all systems, as Figure 1.3 shown.

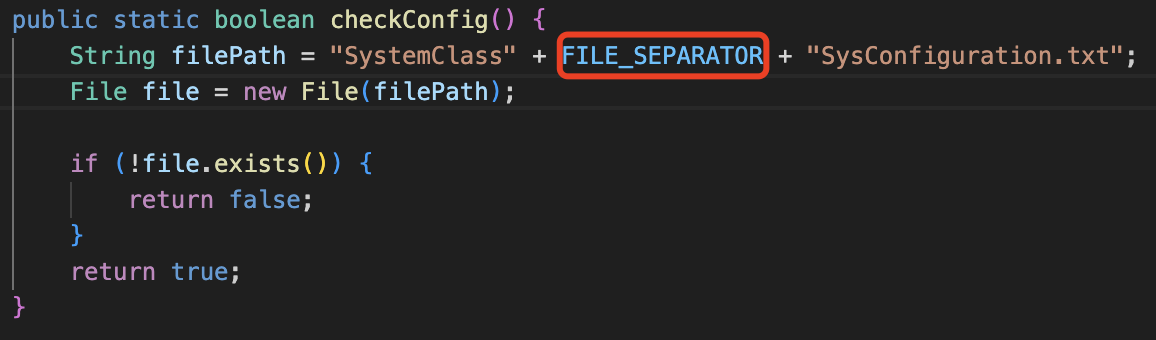


Figure 1.3 Code without Bug

In addition, I have made modifications to all similar places. Although these bugs have not been discovered yet, encountering program crashes due to different systems is inevitable in the subsequent running process.



Figure 1.4 All Paths Are Compatible with Different Systems

Bug-2702

**Title**: Players Can Have the Same Name or Symbol

**Status**: Closed

**Description**:

Two players have the same symbol, causing confusion.

**Reproduce Step:**

1. 1-Start new game

2. 1-Human vs Human

3. Enter Name

4. Enter Symbol

5. Enter Name

6. Enter Symbol

**Type**: Performance

**Component**: Game

**OS**: macOS Sonoma 14.1.2

**Severity**: Major

**Reported**: 2023-10-15 14:34:22 HKT by LIN Xiaoyang

**Last Modified**: 2023-10-16 22:01:20 HKT

**Assignee**: SONG Rui

**Problem**:

If two players use the same character to represent their pieces, it will be visually impossible to distinguish between the pieces of both sides, which will seriously affect the gaming experience.

Figure 2.1 shows the actual outputs. But the expected output is having distinct kinds of symbols for each player so that the players can recognize their own pieces and their opponent’s pieces.

As shown in Figure 2.1 when player Lin enters the first move as 0, its symbol, “o”, falls on the left-bottom corner of the grid. However, when player Song enters 1, its symbol, also “o”, falls on the right of Lin’s first move.

The problem is that both players have the same symbol, “o”, making it impossible to distinguish their pieces.

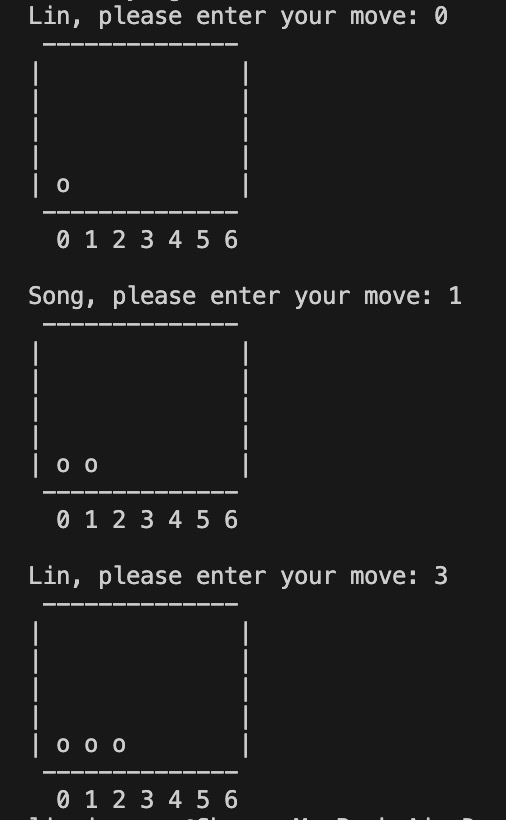


Figure 2.1 Bug Output Display

SONG Rui 2023-10-15 21:38:55 HKT

I added a while loop and an if statement to detect whether the two players have the same symbol to represent their pieces. If they have the same pieces, then the second player is asked to reset their piece.

LIN Xiaoyang 2023-10-16 21:50:52 HKT

In the future, the AIPlayer will automatically fill in its symbol, which may conflict with HumanPlayer’s symbol. Please consider this case.

SONG Rui 2023-10-16 22:01:20 HKT

Okay. The Old version and the new version code are displayed in the following page.

Initial Code:

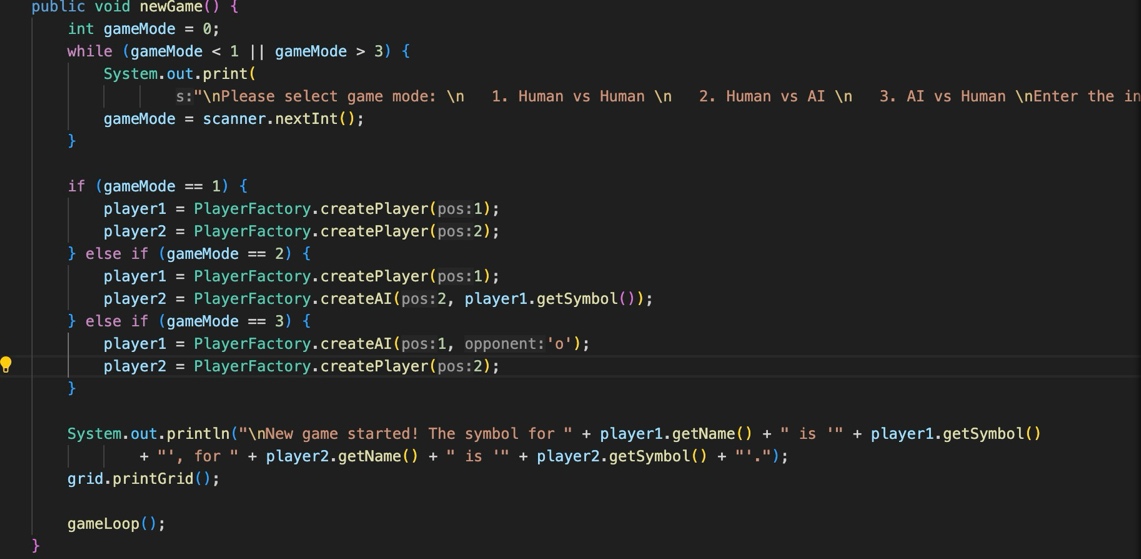


Figure 2.2 Old Version Code with Bug

Updated Code:



Figure 2.3 New Version Code without Bug

Bug-2703

**Title**: Game Not Finished When Connect Five

**Status**: Closed

**Description**:

The game will not be finished if one connects five in a line.

**Reproduce Step:**

1. 1-Start new game

2. 1-Human vs Human

3. Enter Name

4. Enter Symbol

5. Enter Name

6. Enter Symbol

7. Make Step through the following order:

0-5-1-6-3-5-4-6-2

8. The game is not finished

**Type**: Functional

**Component**: Referee

**OS**: Windows10

**Severity**: Critical

**Reported**: 2023-10-17 16:50:13 HKT SONG Rui

**Last Modified**: 2023-10-18 19:07:00 HKT

**Assignee**: Fong Tsz Wai

**Problem**:

When the chess pieces on the board connect to 4, Referee.checkWinner() will return true, indicating that the game is over, and the player has won. However, if you skip connecting four and directly connect five, the referee will not think that you have won and will still return false. The game will continue to let the next player continue playing chess, as shown in figure 3.1.

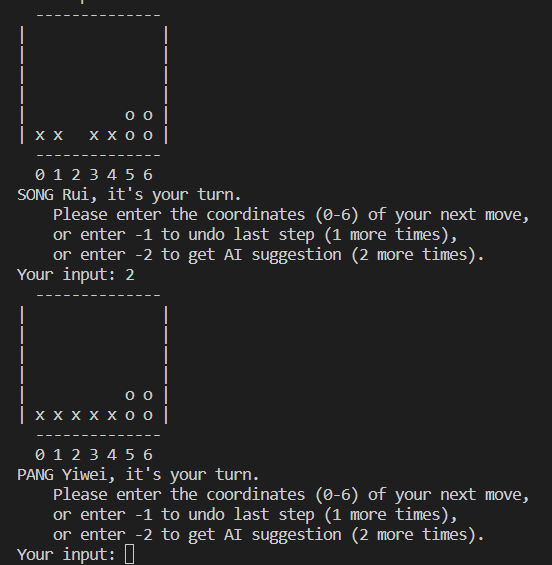


Figure 3.1 Connect Five Not Finish

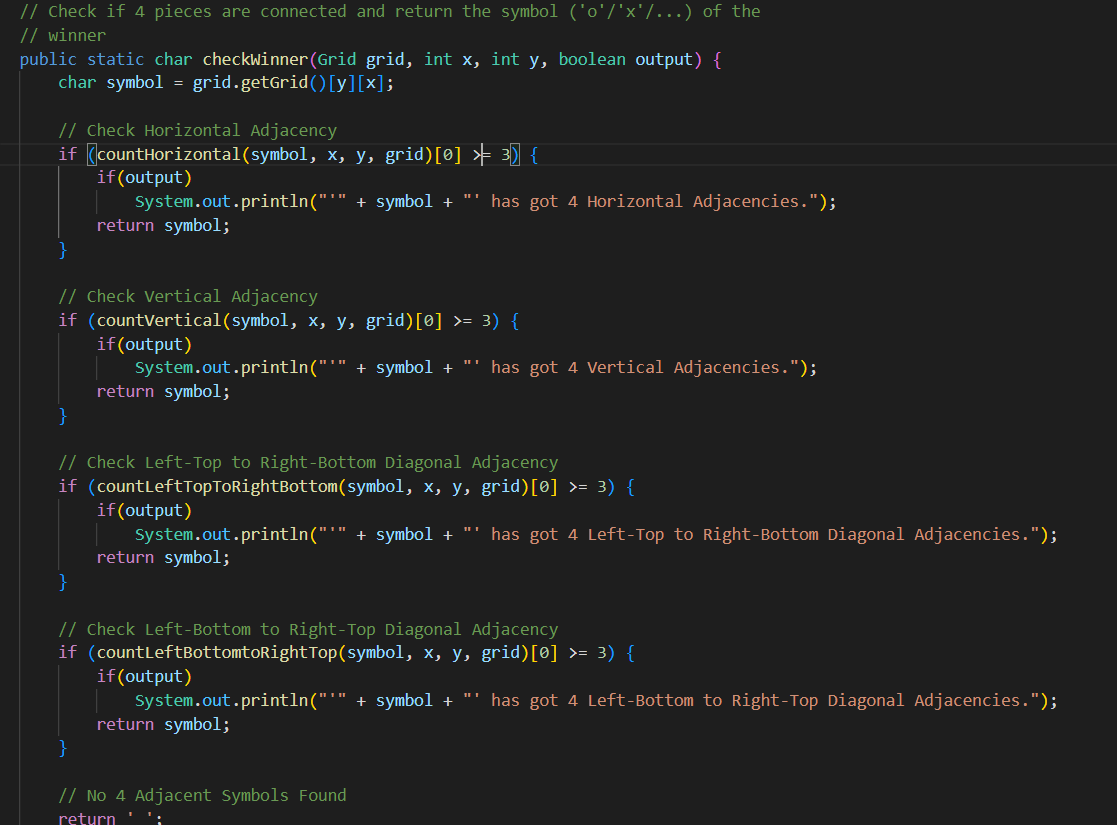


Figure 3.2 Code Without Connect Five Bug

Fong Tsz Wai 2023-10-18 19:07:00 HKT

The problem is fixed. The problem is that after playing a move, it will only check whether there are three adjacent ones in the move just played but ignore the possible four adjacent ones. This has been corrected now as Figure 3.2.

Bug-2704

**Title**: Game initialized error

**Status**: Closed

**Description**:

If we enter some string when selecting game mode, the game cannot move on anymore.

**Reproduce Step:**

1. 1-Start new game
2. Enter some string (for example: “abc”)
3. The game cannot move on

**Type**: Functional

**Component**: Game

**OS**: Windows10

**Severity**: Major

**Reported**: 2023-10-20 19:08:00 HKT SONG Rui

**Last Modified**: 2023-10-21 20:11:00 HKT

**Assignee**: Mark Lee

**Problem**:

When selecting a game mode, you need to enter the serial number of the corresponding game mode. So, the input type should be an integer. But if the user enters a string incorrectly like Figure 4.1 (which may happen when the user does not understand the input rules or accidentally enters the wrong string), the software no longer accepts corrected input, but loops forever to output the request but the user cannot input a new value (Figure 4.2).

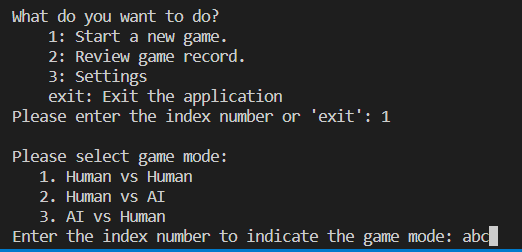


Figure 4.1 User Input a String

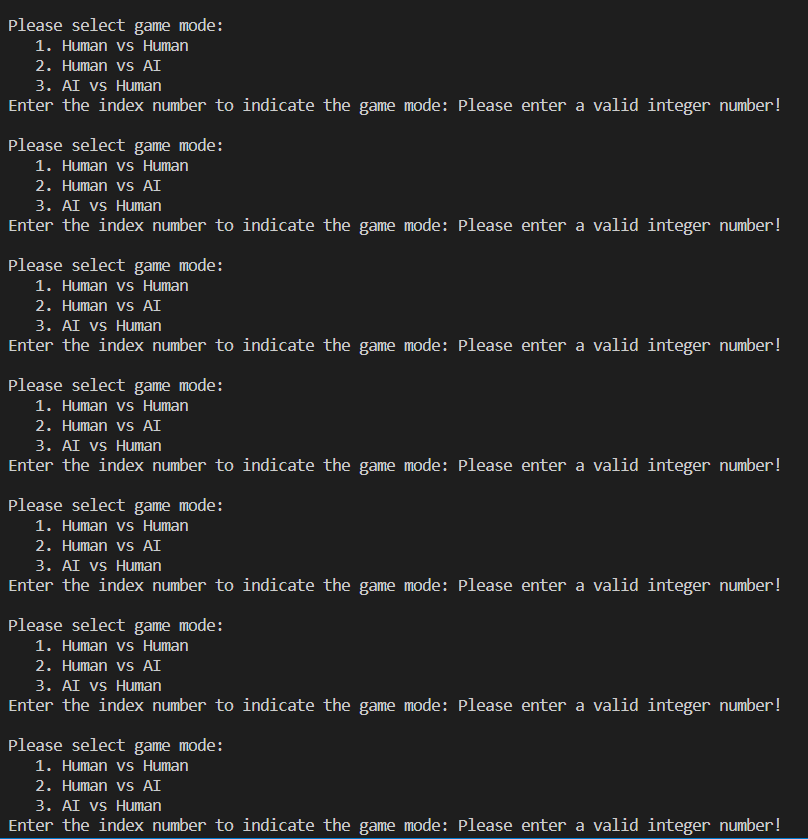


Figure 4.2 The Software print forever

Mark Lee 2023-10-21 20:11:00 HKT

The problem has been solved. The cause of the problem is that scanner.nextInt() will throw new InputMismatchException if it finds that the next input value is not an integer. But we have overlooked a problem. If the next input is not an integer, nextInt() will not accept this input. The input stream will still stay before the non-integer input and will not continue to go after this input. After we caught this exception, we did not clear the input of the buffer, so when it returns to the nextInt through the while loop, it will still face the same input, causing the program to loop in this while loop forever. The solution is not difficult. We only need to clear the incorrect input in the buffer by next() after catching the InputMismatchException, as shown in Figure 4.3 in line 37.

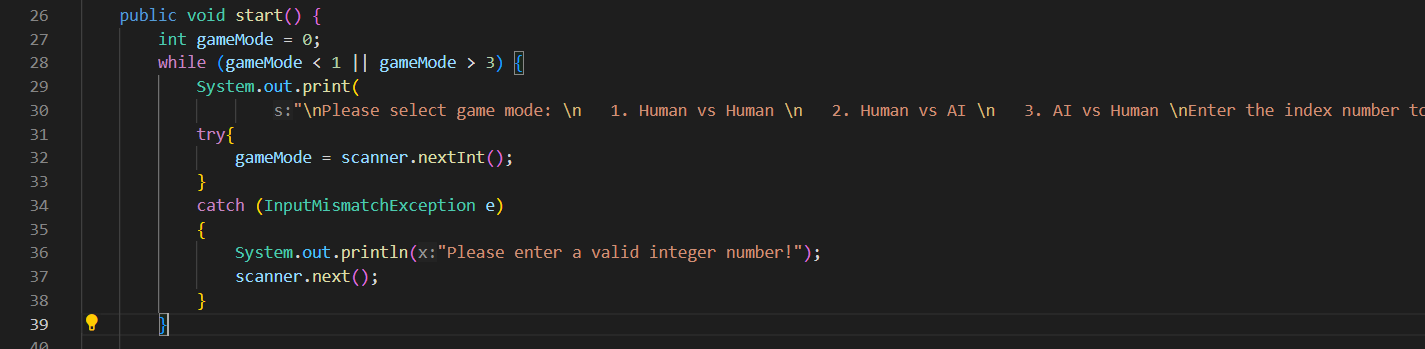


Figure 4.3 Add next() to clear incorrect input

Bug-2705

**Title**: GamaGo Does Not Take Center

**Status**: Closed

**Description**:

GamaGo should choose center first if no other aspects influence.

**Reproduce Step:**

1. 1-Start new Game
2. 3-AI vs Human
3. 1-GamaGo
4. Enter Name
5. Enter Symbol
6. GamaGo goes randomly

**Type**: Functional

**Component**: GamaGo

**OS**: Windows10

**Severity**: Minor

**Reported**: 2023-10-25 10:23:00 HKT Fong Tsz Wai

**Last Modified**: 2023-10-26 14:52:00 HKT

**Assignee**: JI Xin

**Problem**:

When I tried to use GamaGo to play against some online four-bang algorithms, I found that GamaGo's results were not good enough. One of the reasons is that I found that other websites will give priority to the middle column when the profit value of each step is similar, while GamaGo will randomly move to a position in this case, as shown in the Figure 5.1, this will This caused GamaGo to fall into a disadvantage. After reading relevant information, I found that in four-piece chess, more players occupying the middle column will have a greater probability of winning. Therefore, I suggest that GamaGo should add some weights to the middle columns so that it will take the middle columns first. However, this weight should not be too large, causing GamaGo to always go to the middle column. Its weight value should not lead to skipping a level but should only lead to priority among peers.

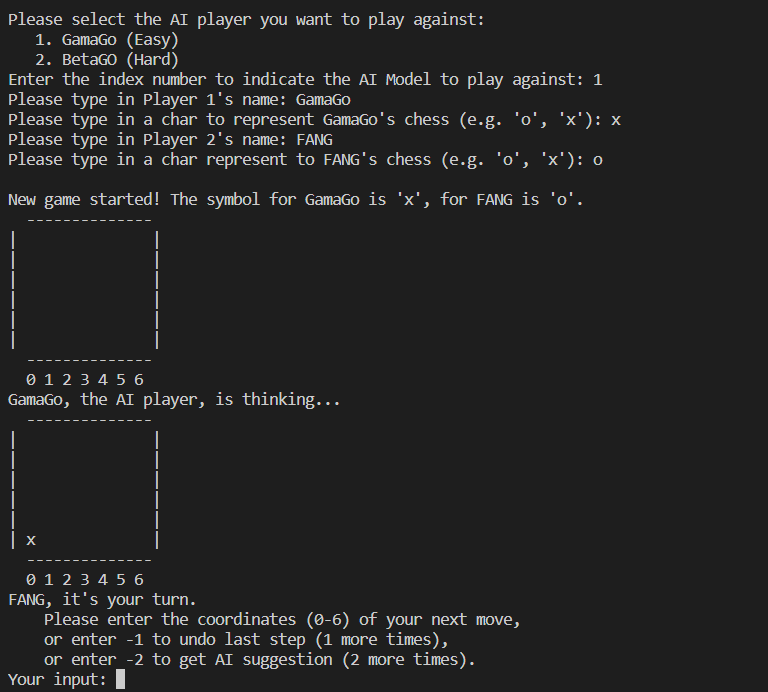


Figure 5.1 GamaGo Move Randomly

JI Xin 2023-10-26 13:12:00 HKT

This issue has been fixed. You are right, we ignored this issue when designing. Now I've given each column a different weight (Figure 2.10), the middle column has a weight of 8 and either side of it has a weight of 6. Because the difference at each level is at least 10, adding a weight within ten will not lead to skipping a level. Now GamaGo will give priority to the middle position when playing at the same level.

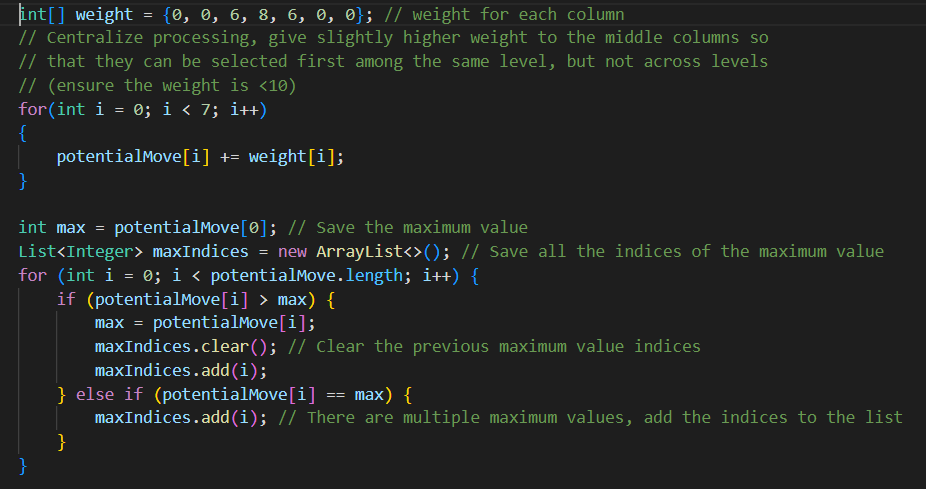


Figure 5.2 Code with Wight for Columns

Fong Tsz Wai 2023-10-26 14：52：00 HKT

Thank you for your work. After testing, the performance of GamaGo against the previous algorithm has been greatly improved.

Bug-2706

**Title**: BetaGo initialized error

**Status**: Closed

**Description**:

BetaGo cannot function successfully no matter in first player or second player.

**Reproduce Step:**

1. 1-Start new Game
2. 2-Human vs AI/3-AI vs Human
3. 2-BetaGo
4. Enter Name
5. Enter Symbol
6. When it is BetaGo’s turn

**Type**: Functional

**Component**: BetaGo

**OS**: Windows10

**Severity**: Major

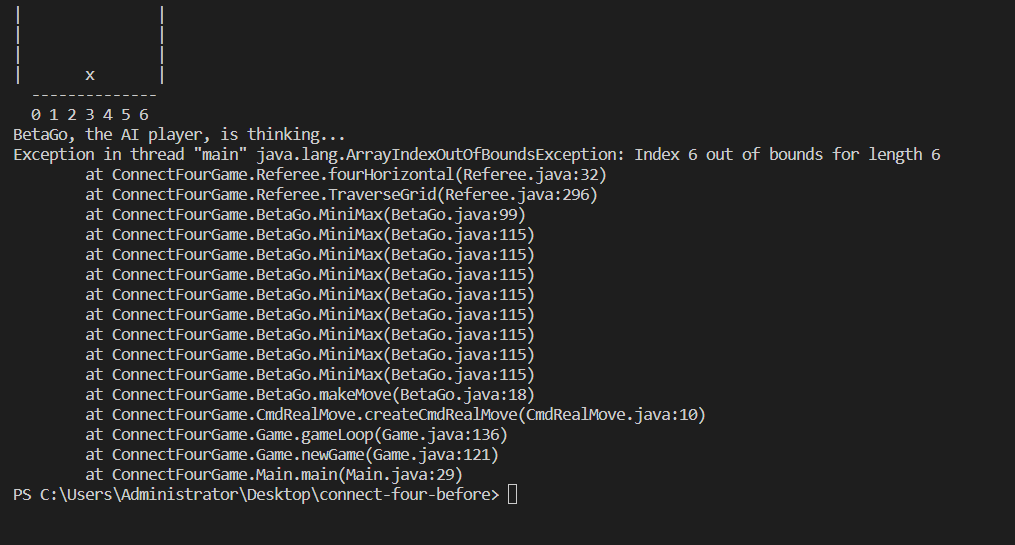
**Reported**: 2023-11-11 13:45:14 HKT by Fong Tsz Wai

**Last Modified**: 2023-11-13 21:38:05 HKT

**Assignee**: Ji Xin

**Problem**:

As mentioned in the description, the BetaGo is not in function and outputs ArrayIndexOutOfBoundsException. The error is caused by Referee.fourHorizontal() and it claims that “Index 6 out of bounds for length 6”. Details are shown in Figure 6.1.

Figure 6.1 Exception occurred when against BetaGo

Ji Xin 2023-11-13 21:38:05 HKT

Now the BetaGo can function in a good situation. The original code and the fixed code are captured in Figure 6.2 and Figure 6.3. The error was found out that due to handling parameter mistakenly. Variable x should imply the column index in the checkerboard while y should imply the row index in the checkerboard.

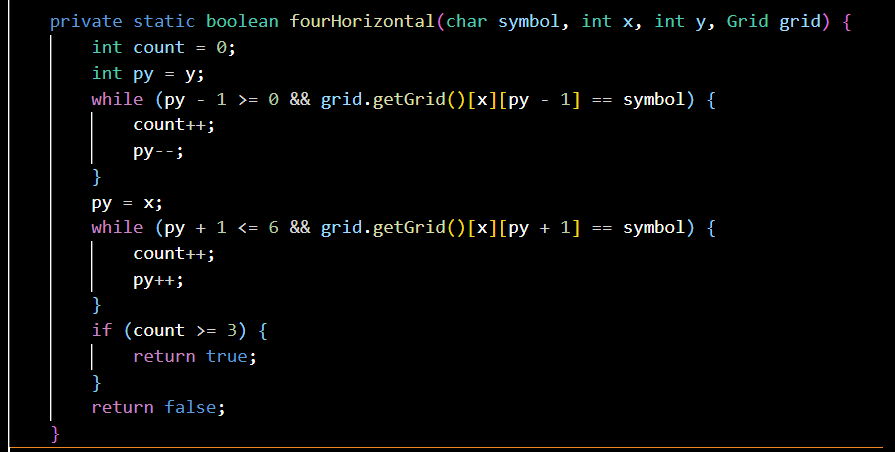


Figure 6.2 Original code: mishandling of x, y parameters

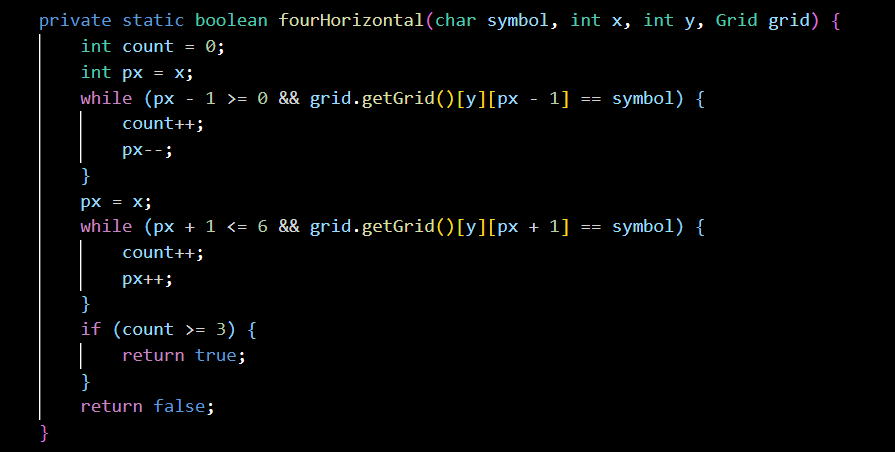


Figure 6.3 Fixed code

Bug-2707

**Title**: Index Out of Bounds Error in Replay: No Moves Left for AI Suggestion

**Status**: Closed

**Description**:

During game replay, after all the moves have been visited, select AI suggestion for next move, Connect4 will crash as there are no moves left and throw an IndexOutOfBoundsException.

**Reproduce Step:**

1. Run Connect4.
2. Start a new Game
3. Select and completely play any game
4. Select review game record
5. Select any game
6. Press 2 until all moves have been traversed
7. Press 0 for AI suggestion
8. Program Crashes

**Type**: Functional

**Component**: CmdList

**OS**: macOS Ventura 13.5.2

**Severity**: Major

**Reported**: 2023-11-25 13:02:18 HKT by Mark Lee

**Last Modified**: 2023-11-25 13:38:15 HKT

**Assignee**: Mark Lee

**Problem**:

No array bounds checking during AI suggestion for next move, causing IndexOutOfBoundsException when there are no moves left.

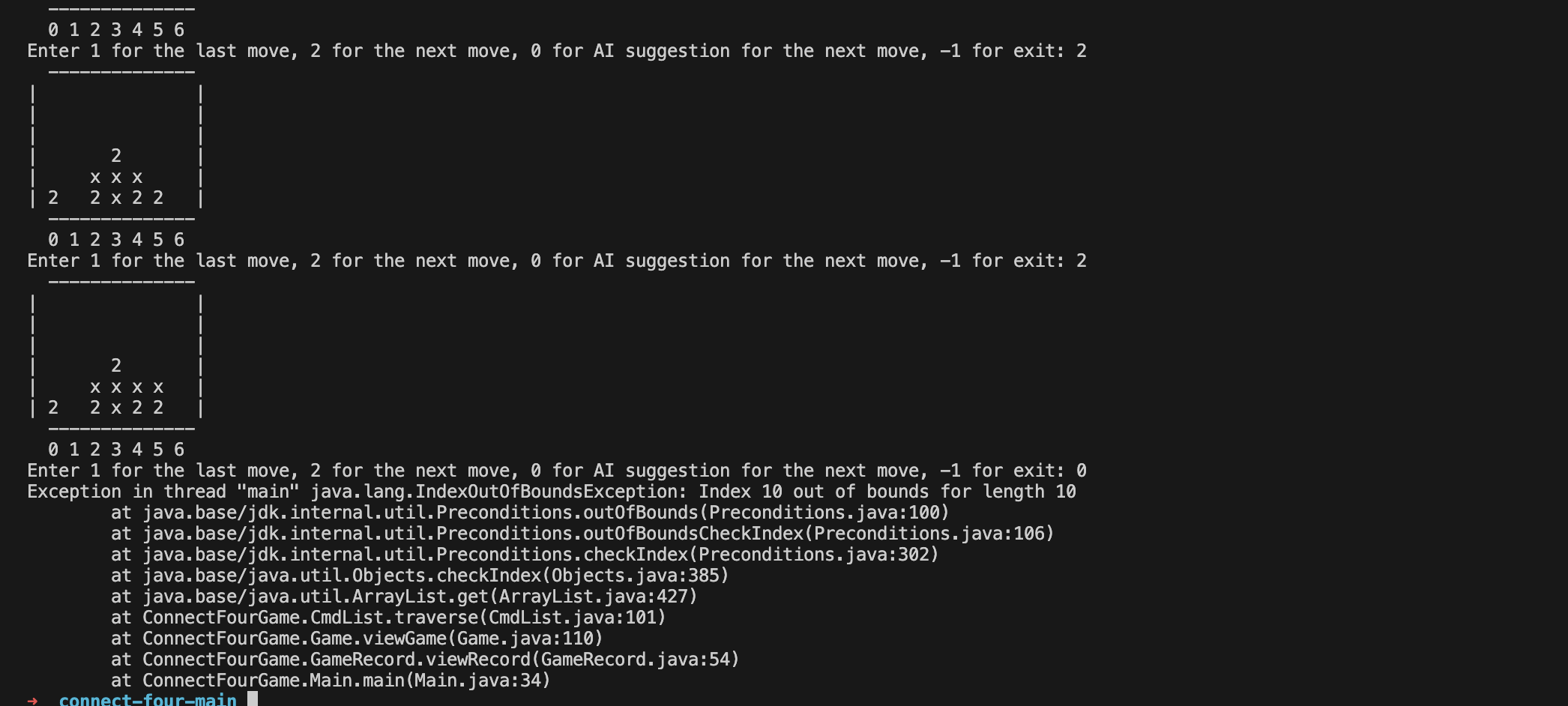


Figure 7.1 Outputs with Bug

Mark 2023-11-25 13:38:15 HKT

I fixed the bug by adding a checking for whether there are any more moves left (checking list.size()). If there are no more moves left, it will print “It's already the last move.”, else it will continue.

Bug-2708

**Title**: BetaGo Does Not Take Center

**Status**: Closed

**Description**:

Same as GamaGo, BetaGo should choose center first if no other aspects influence

**Reproduce Step:**

1. 1-Start new Game

2. 3-AI vs Human

3. 2-BetaGo

4. Enter Name

5. Enter Symbol

6. BetaGo choose to go to 0

**Type**: Functional

**Component**: BetaGo

**OS**: Windows10

**Severity**: Minor

**Reported**: 2023-11-26 15:01:22 HKT JI Xin

**Last Modified**: 2023-11-26 15:03:35 HKT

**Assignee**: Fong Tsz Wai

**Problem**:

Like GamaGo, BetaGo should also add a priority to the middle column.

Fong Tsz Wai 2023-11-26 15:03:35 HKT

I fixed this problem by adding a considerSequence to BetaGo(Figure 8.1). Now BetaGo will consider each column in sequence through this sequence, and only select later columns if they are greater than the current column. This method ensures that when the column values are the same, priority will be given to the numbers in the sequence order, and these numbers are all set to the index of the middle column.

A computer screen with text

Description automatically generated

Figure 8.1 Code with ConsiderSequence

Bug-2709

**Title**: Cannot Input Human’s Name

**Status**: Closed

**Description**:

The software does not allow users to enter name

**Reproduce Step:**

1. 1-start new game
2. 1-Human vs Human
3. Cannot enter name

**Type**: Functional

**Component**: PlayerFactory

**OS**: Windows 10

**Severity**: Critical

**Reported**: 2023-11-02 20:03:25 HKT by Mark Lee

**Last Modified**: 2023-11-03 19:55:02 HKT

**Assignee**: Fong Tsz Wai

**Problem**:

After selecting the battle mode, the player is not asked to enter the player's name. Instead, the player is directly asked to enter the symbol like Figure 9.1.

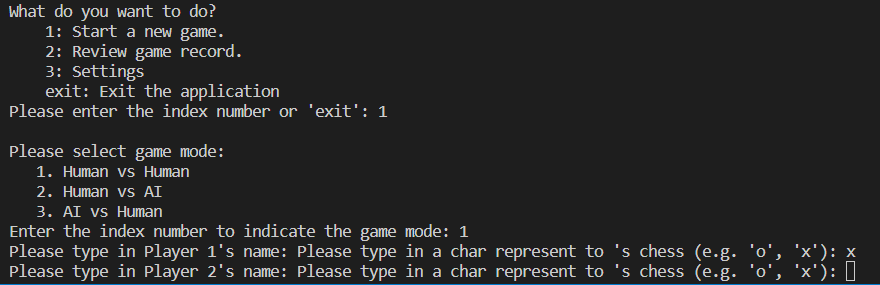


Figure 9.1 Cannot Input Name

Fong Tsz Wai 2023-11-03 19:55:02 HKT

I was trying to make it possible for the player's name to be entered as a name with spaces, so I used scanner.nextLine() instead of scanner.next(), but I overlooked a problem. The previous scanner.next() will stay in front of \n in the input stream, instead of reading \n and then staying behind it. Therefore nextLine() will start reading from \n. If next() finds that it is \n at the beginning, it will skip it, but nextLine() will not do this. It will only read a \n, so it will not receive anything. I've fixed this by clearing the cache of \n with an empty nextLine() as shown in Figure 9.2.

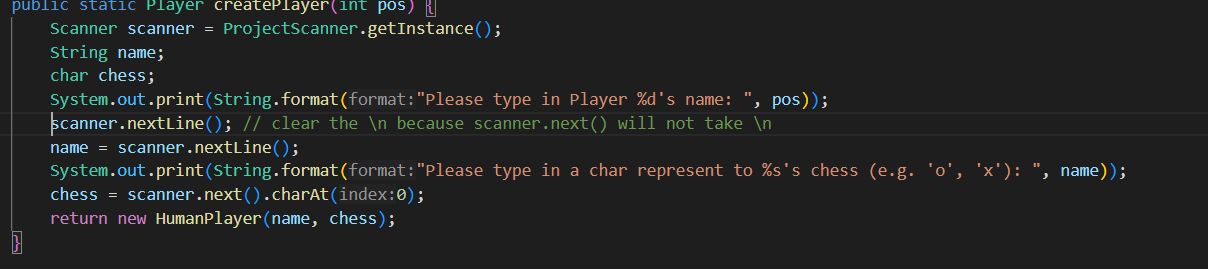


Figure 9.2 NextLine to Clear Buffer

Bug-2710

**Title**: Index Error for Configuration

**Status**: Closed

**Description**:

When selecting which configuration to change, the program throws IndexOutOfBoundsException

**Reproduce Step:**

1. 3-Settings

2. 2-allowedHelp

3. IndexOutOfBoundsException

**Type**: Functional

**Component**: SysConfiguration

**OS**: Windows 10

**Severity**: Major

**Reported**: 2023-12-1 20:35:14 HKT by SONG Rui

**Last Modified**: 2023-12-1 21:07:16 HKT

**Assignee**: SONG Rui

**Problem**:

A problem occurs when entering settings and selecting a certain piece of content to modify. If you choose to modify the last item, an exception will be thrown (Figure 10.1). If you choose to modify the previous items, you cannot modify the correct item. It seems that there is a problem with the index when modifying SysConfiguration.

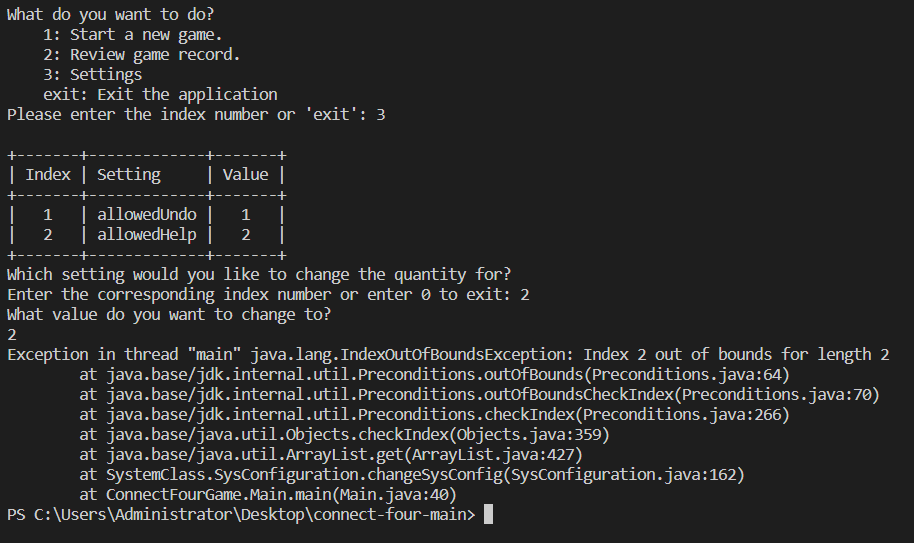


Figure 10.1 Throw Exception

SONG Rui 2023-12-1 21:07:16 HKT

I solved this bug by letting the actual index = the input index - 1. Customers who use our software may not necessarily have a foundation in programming, which means they are not necessarily accustomed to index starting from 0. Therefore, we hope to make a more customer-friendly operation, that is, the index displayed to customers starts from 1. This will cause a problem. The index entered by the customer is not the actual index but needs to be subtracted by 1. We forgot this problem here, resulting in an error.

Bug-2711

**Title**: redoOneCmd Method in CmdMove Does Not Work

**Status**: Closed

**Description**:

The grid should get back to the previous status (before undo) while an undo followed by a redo operation.

**Reproduce Step:**

1. Initialize the Player, Grid, CmdList, CmdMove

2. getMove()

3. get the current move’s chess coordinate

4. get the current move’s chess symbol

5. Determine the symbol with the coordinate in the grid, the symbol should be the same as the symbol in step 4.

6. undoOneCmd()

7. Determine the symbol with the coordinate in the grid, the symbol should be empty

8. redoOneCmd()

9. Determine the symbol with the coordinate in the grid, the symbol should be the same as the symbol in step 4 & 5, but it was shown in different places.

**Type**: Functional

**Component**: CmdMove

**OS**: macOS Sonoma 14.1.1

**Severity**: Major

**Reported**: 2023-12-1 21:35:14 HKT by JI Xin

**Last Modified**: 2023-12-1 22:38:55 HKT

**Assignee**: SONG Rui

**Problem**:

A problem occurred when calling redoOneCmd method, it wrongly requests the parameter of ‘y’ in grid.update(). In this case, the redo operation was not performed as we desired.

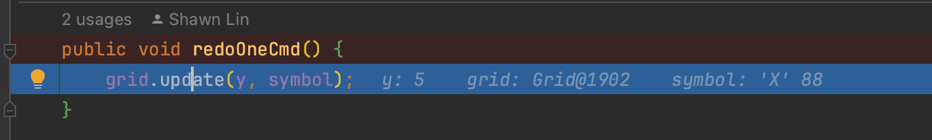


Figure 11.1 Redo Operation Problem

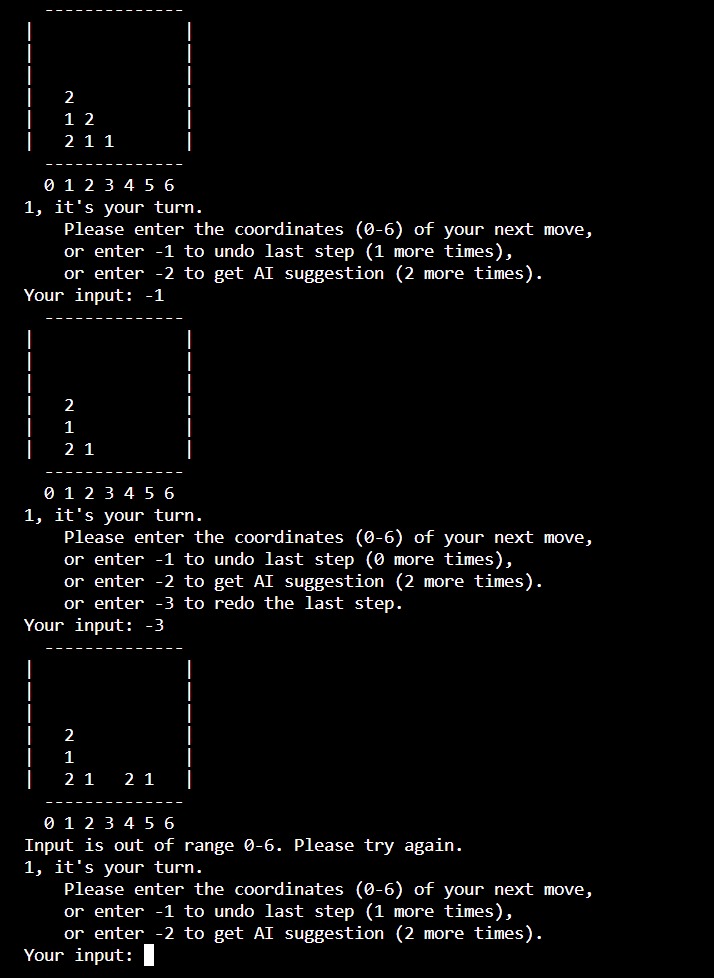


Figure 11.2 Redo Operation Problem in Game

SONG Rui 2023-12-1 22:38:55 HKT

I modified the bug by changing the parameter in the redoOneCmd() to Grid.update(x, symbol).

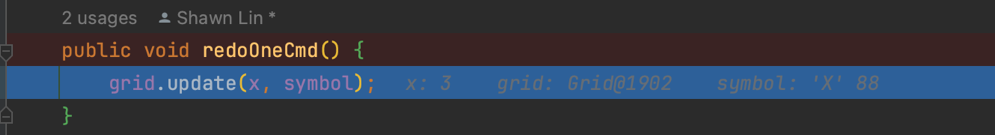


Figure 11.3 Redo Operation Bug Fixed

After modification, once redoOneCmd() is called, it would update the grid with the correct coordinate.

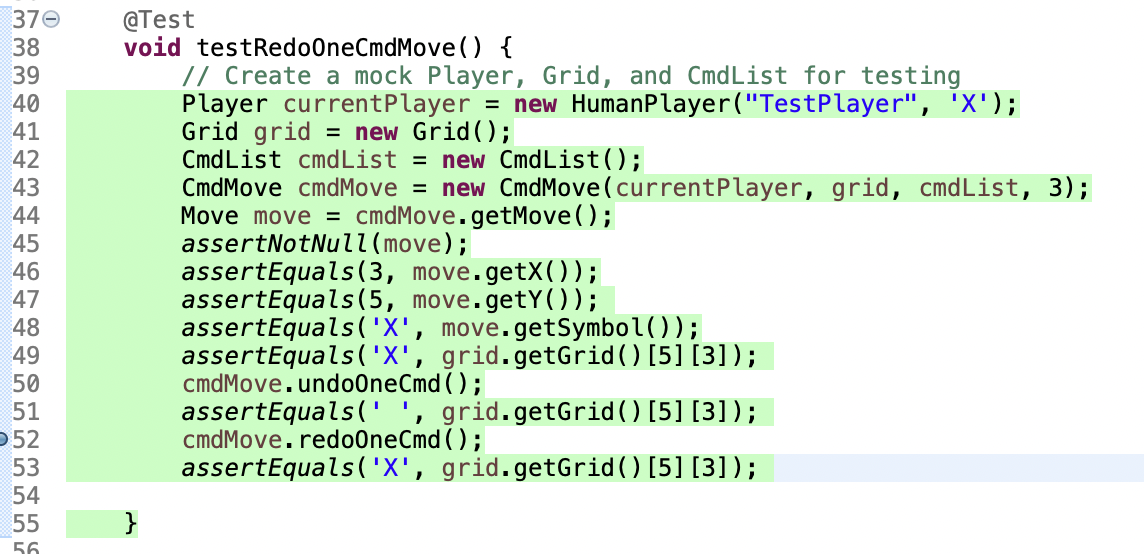


Figure 11.4 Redo Test Coverage

Bug-2712

**Title**: No exception handling for non-integer input

**Status**: Closed

**Description**:

Missing Exception handling in several places where integer inputs are required.

**Reproduce Step:**

1. 1- Start New Game
2. Type in non-integer value (Exception)/ Integer value (keep going)
3. Input name and chess for human player(s)
4. Input non-integer value during game

{Or}

1. 3- Change Settings
2. Type in non-integer value (Exception)/ Integer value referenced to existing setting {1, 2} (keep going)
3. Input non-integer value (Exception)

**Type**: Functional

**Component**: CmdMove

**OS**: macOS Sonoma 14.1.1

**Severity**: Minor

**Reported**: 2023-12-1 21:40:27 HKT by JI Xin

**Last Modified**: 2023-12-2 14:31:10 HKT

**Assignee:** Fong Tsz Wai

**Problem**:

Some parts of the program in which user input are required do not consider the situation that user wrongly input a non-integer value then the program crashes due to InputMismatchException. This unhandled bug will lead to unfriendly user’s experience and should be dealt with after all Major and Critical bugs are closed. Details shown in Figure 12.1.

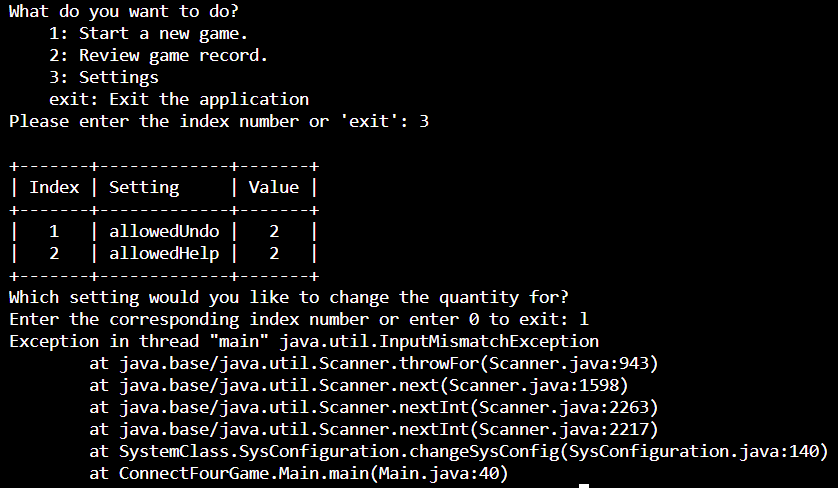


Figure 12.1 InputMismatchException Details



Figure 12.2 Modified SysConfiguration.changeSysConfig() source code

Fong Tsz Wai 2023-12-3 14:31:10 HKT

Yes, this bug really does affect user’s experience and should be considered earlier during the developing process. Thank you, Jesse, for your carefully reviewing and testing of the program. The source code has been modified properly using try-catch syntax as shown in Figure 12.2.