Project Deliverable 2

Felix Leong
Dr. Baron
SER216 T Th 13:30-14:45
February 21, 2023

Table of Contents

Personal Process Log

Date	Start	Stop	Interruption Time	Delta Time	Phase	Gradescope Score	Comments
2/7/2023	19:00	19:41	20 min	21 min	Planning	N/A	Setting up documents
2/8/2023	14:32	15:47	15 min	60 min	Design	N/A	class diagram, clarification, understanding interfaces
2/9/2023	15:59	16:47	5 min	43 min	Design	N/A	laying out piece taking calculations, determining final methods
2/9/2023	16:53	18:11	15 min	63 min	Coding	N/A	begin setting up CheckersGameLogic class
2/9/2023	19:19	21:07	10 min	98 min	Coding	N/A	further implementation of CheckersGameLogic
2/9/2023	21:10	21:24	3 min	11 min	Design	N/A	tracing Checker movement algorithm
2/9/2023	21:26	22:14	5 min	43 min	Coding	N/A	implementing Checker valid move algorithm
2/11/2023	11:26	13:15	25 min	84 min	Coding	N/A	further implementation of CheckersGameLogic
2/14/2023	11:48	13:15	15 min	72 min	Coding	N/A	wrapping up implementation and adding loggers
2/16/2023	15:56	17:08	0 min	72 min	Coding	1.8/20	refactoring code, adding documentation, changing structure, removing imports
2/16/2023	17:12	19:41	20 min	129 min	Testing	4/20	debugging program based on Gradescope results
2/16/2023	20:10	20:39	0 min	29 min	Testing	4/20	running sample test for error exploration
2/16/2023	20:50	21:08	0 min	18 min	Post	4/20	generating documentation for project, documenting sample runs
2/17/2023	13:50	15:16	15 min	71 min	Testing	20/20	refractor getSquare and rework program references accordingly

2/17/2023	15:17	16:39	10 min	72 min	Post	20/20	generate documentation for project, complete postmortem documents

Defect Log

Sl. No.	Date	Defect Type	Defect Inject Phase	Defect Removal Phase	Fix Time	Fix Ref	Description
001	2/9/2023	Interface	Coding	Coding	5 min	X	Scanner cannot read console input, added run { standardInput=System.in } to app/build.gradle
002	2/9/2023	Data	Coding	Coding	1 min	X	int value of char reads ASCII code instead of number value, use substring() instead of charAt()
003	2/9/2023	Structure	Design	Planning	N/A	X	doTurn occurring before nextTurn means no turn to perform on initial step
004	2/9/2023	Syntax	Coding	Coding	20 min	X	substring not parsing String correctly, redesigned code format to not rely on error causing method
005	2/11/2023	Interface	Coding	Coding	3 min	X	TextConsole not showing changes on board, resolved by correcting order of parameters in function call
006	2/16/2023	Package	Design	Coding	10 min	X	Disallowed packages, refactor code to avoid use of disallowed packages
007	2/16/2023	Function	Coding	Testing	15 min	X	getWinningPlayer() not returning correct values, error traced back to generateValidMoves() not returning null when supposed to
008	2/16/2023	Function	Coding	Testing	7 min	X	generateValidMoves() not returning null when it should
009	2/17/2023	Data	Coding	Testing	58 min	X	getSquare() not returning the correct position
010	2/17/2023	Function	Design	Testing	15 min	X	getWinningMove() not detecting player positions on white tiles, fixed by removing black tile check from generateValidMoves()
011	2/17/2023	Build	Post	Post	20 min	X	Gradle task documentation not functioning correctly

Design Form

File Structure

```
-edu
        -ser216
                -checkers
                        -core
                                CheckersDriver.java
                                CheckersGame.java
                                CheckersGameLogic.java (needs implementation)
                        -ui
                                CheckersTextConsole.java (needs implementation)
                                CheckersViewer.java
-docs
        -<files generated from JavaDoc>
```

Questions

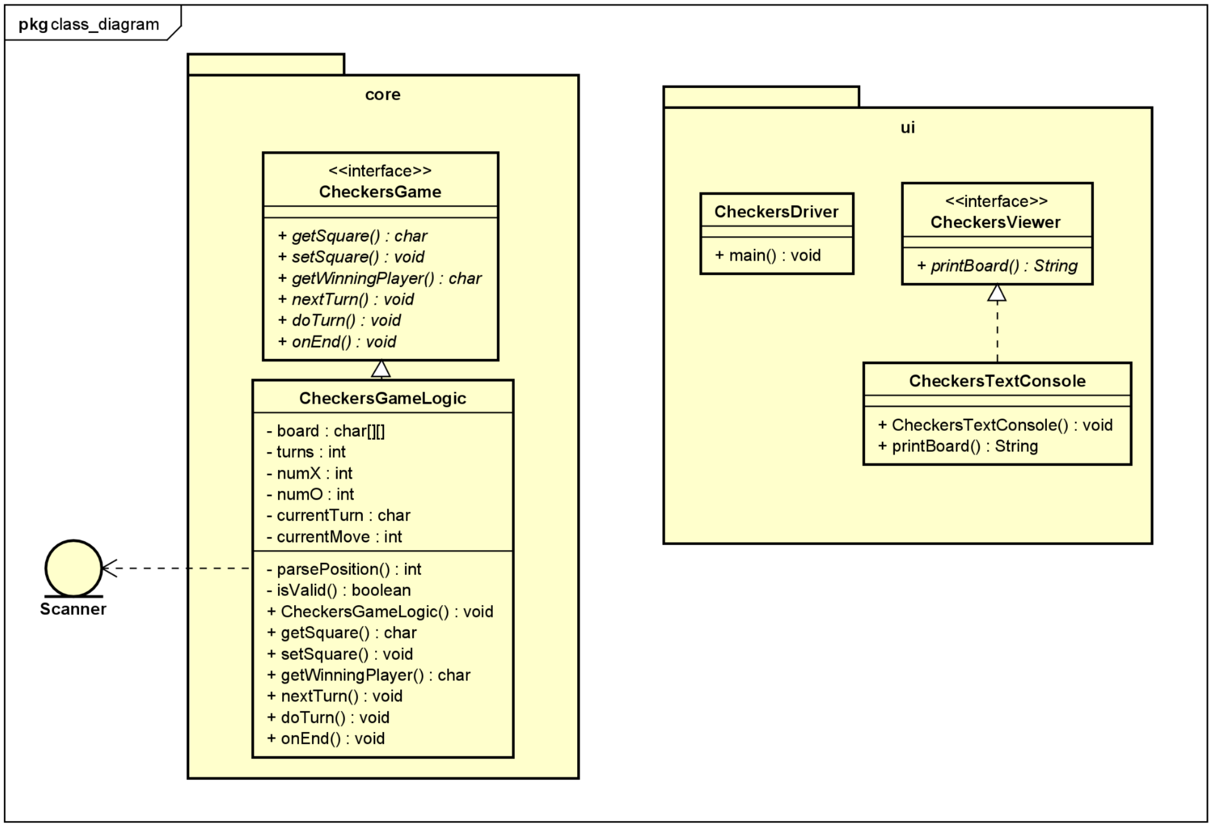
What is the best way to represent the board?

- checkers will only be on alternating squares
- minimum 4x4 array needed
- board range: a-h, 1-8
- internal range 0-7, 0-7
- row if x % 2 = 1, offset piece by remainder

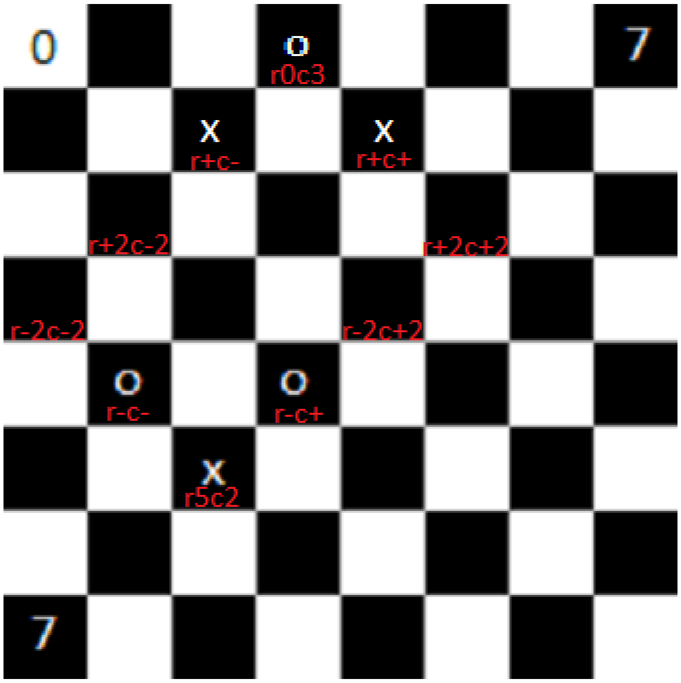
Java Packages

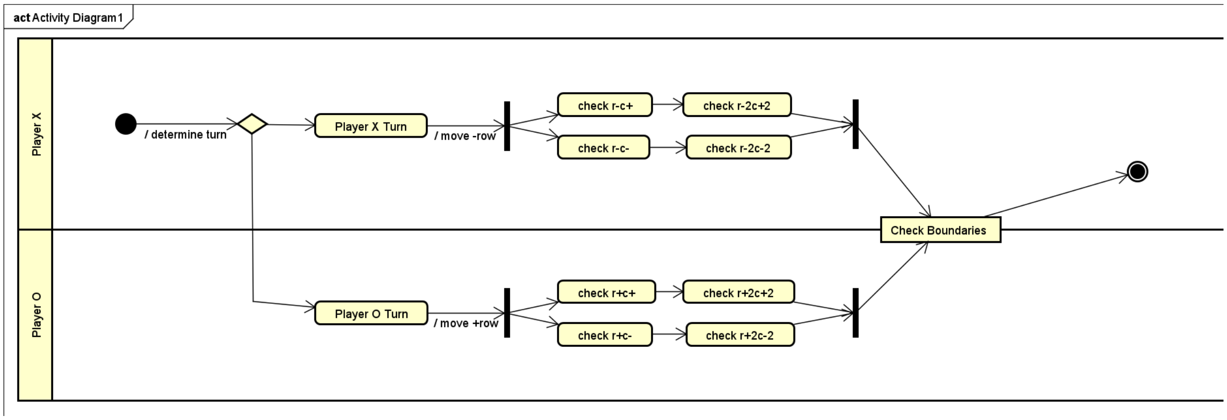
java.util.Scanner
java.util.regex.*

Class Diagram



Checkerboard Model





IsValid

1. receive player turn
    a. throw exception if turn not valid
2. check r-c+ and r-c- positions
3. Is the original position and new position the same?
    a. yes - not valid, throw exception
    b. no - continue
4. Is the move diagonal? (both r and c changed)
    a. yes - continue
    b. no - not valid, throw exception
5. Are there pieces in those positions?
    a. yes - are the pieces the players own?
        i. yes - player cannot move
        ii. no - can the pieces be jumped
            1. yes - take pieces
            2. no, continue
    b. no, make a move
6. Is the piece on the border?
    a. yes - piece movement may not be valid depending on direction
    b. no - continue
7. End of Function

CheckersPiecePositon - class

- track row and col position

Function: generateValidMoves(position, piece)

1. Is the starting position valid? (ie. is it on the board)
2. Is the specified position possible? (ie. is it on a black tile?)
3. Is the piece at the boundary (for example, is black at row=0)
    a. yes, cannot move any further (return moves)
4. check diagonal left and right

```
        a. check if left diag points to a wall
            i. yes, then ignore all left moves
        b. check if right diag points to a wall
            i. yes, then ignore all right moves
    5. check if diag contains a piece
        a. yes, then which piece is it
            i. own piece - cannot move, not valid
            ii. opponent piece - check if jump can be valid
                1. diag beyond opponent must be empty
        b. no, diag is empty
            i. move is valid
```

Project Summary

Time Distribution

Time in Phase (minutes)	Actual Time	% of Total
Planning	21 mins	2.37%
Design	1 hr 54 mins	12.87%
Code	7 hr 12 mins	48.76%
Test	3 hr 49 mins	25.85%
Postmortem	1 hr 30 mins	10.16%
TOTAL	14 hr 46 mins	100%

Defect Summary

Defects Injected	Actual Number of Defects	% of Total
Planning	0	0%
Design	3	27%
Code	7	64%
Test	0	0%
Postmortem	1	9%
TOTAL	11	100%

Performance Summary

	Actual
Program Size (LOC)	643 LOC
Productivity (LOC/Hr)	44 LOC/hr
Defect Rate (Defects/KLOC)	17 Defects/KLOC

Project Reflection

In which Phase did you spend most of your effort? (Look at the time spent in the different phases of this assignment to answer this question)

Most of my effort was spent in the coding phase as indicated by the highest percentage of time spent there. I tested while coding, meaning that a lot of effort was also diverted toward testing, however in my opinion, I spent the most critical thinking during coding on structure and logic.

In which Phase did you introduce the greatest number of defects?

The coding phase introduced the greatest number of defects.

Did you find it useful to follow a systematic process and track your effort and defects?

It was difficult to keep track of PSP toward the end of the project as I began to really focus down on certain aspects of the product. But, the process was useful for determining what I had already done in the state of the project and what errors were already fixed. It might have been more useful to me had I spent more time documenting errors.