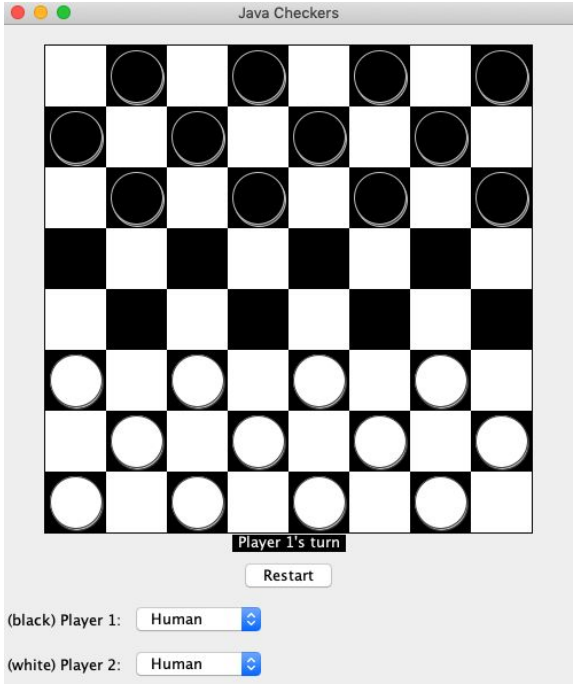
Software Testing Final Project:

A Game of Checkers

Debanik Purkayastha, Shaunak Shah,
Vinicius Lepca

Software Under Test

- Java-based Checkers game
  - Desktop App, both Windows and macOS
  - Small-scale app, few active users
  - Object Oriented Encapsulation of essential checkers components
    - Board
    - Game
    - Move
    - Player
  - 3 different play-styles
    - Computer player: generates moves using built-in move generator
    - Human player: PvP on the same computer
    - Network player: Play against a friend on a network connection



Testing Overview

- Black-box integration testing of Game object
  - Uses an amalgamation of Board, Move, and Player Objects
  - Utilize techniques such as Equivalence Partitioning, Boundary Analysis, and Error Guessing
- White-box unit testing of various components
  - Aim to achieve 100% Statement Coverage
  - Utilize techniques such as Equivalence Partitioning, Boundary Analysis, and Error Guessing
- Mock testing Network functionality
  - Engage in behavior-based mock tests to ensure that network communication works as expected
    - Ensure sockets send/receive messages
    - Actions trigger message send and message receive

Demo

- Whitebox unit tests
- Blackbox integration tests
- Network Mock testing

Results

Checkers.model

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
ComputerPlayer		0%		0%	49	51	119	121	5	7	0	1
Board		95%		82%	14	59	4	73	0	19	0	1
NetworkPlayer		83%		n/a	1	3	1	3	1	3	0	1
HumanPlayer		83%		n/a	1	3	1	3	1	3	0	1
Game		100%		98%	1	42	0	92	0	14	0	1
Move		100%		n/a	0	14	0	26	0	14	0	1
Player		100%		n/a	0	2	0	2	0	2	0	1
Total	588 of 1,457	59%	103 of 224	54%	66	174	125	320	7	62	0	7

- JaCoCo Coverage
- Faults
- Soundness of overall SUT

Thank you

Any Questions?