# TEAM 126 CODE REVIEW

# Criteria of Evaluation

Modularity

10 points

Readability

10 points

Functionality

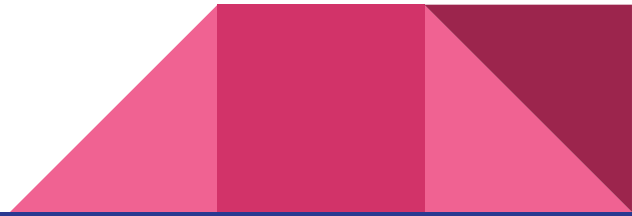
10 points

Comments/Javadocs

10 points

Unit Testing

10 points



# Candidate A: <Dragon Team>

## Modularity

-Data and GUI code separated

-Clear inter-class interactions

-Good implementation of data security

Score: 9/10

## Comments/Javadocs

-No javadocs (some comments seem to be attempted javadocs)

- Fair number of comments

Score: 6/10

## Readability

-Well commented code

-Clear test & variable names

-Fair amount of code duplication

-Class names tend to be confusing

Score: 6/10

## Unit Testing

-Fair number of tests

-Failure on 23/70 tests

-Unclear purpose of tests

Score: 6/10

## Functionality

### Pros

-elegant design

-maintains order

-buggy size button

### Cons

-once the pawn moved to a tile that was not clicked. Could not be reproduced.

Score: 8/10

# Candidate B: <MasterLabyrinth>

## Modularity

-Very efficient code
-Easy to edit

Score:9/10

## Comments/Javadocs

-Well written & sufficient javadocs
-Brief and clean comments

Score: 10/10

## Readability

-Clean code
-Good indentation
-Appropriate method & variable names

Score: 10/10

## Unit Testing

-Low amount of tests
-Pass 20/20 tests
-Visual representation of tests

Score: 8/10

## Functionality

### Pros

-Unique use of KeyListener
-Ingredient List well implemented

### Cons

-Pawns overlap in same space
-Cannot rotate tiles
-No instructions

Score:8/10

# Candidate C: <CSE group>

## Modularity

-Separate ActionListener instance classes
-Rigid categories of classes/packages

Score: 7/10

## Comments/Javadocs

-Good javadocs
-Lack of comments

Score:8/10

## Readability

-Neat and clean code
-Good method & variable naming

Score: 8/10

## Unit Testing

-Pass 33/51 tests
-Revised tests
-Clear Testing logic

Score: 8/10

## Functionality

### Pros

-None

### Cons

-Application freezes on first move

-Cannot initialize game with players

Score: 1/10

# Candidate D: <Master Labyrinth>

## Modularity

-Overconcentrated classes
-GUI and data separated
-Not modular

Score: 6/10

## Readability

-Excessive indentation
-Hard to read

Score: 6/10

## Functionality

Pros
-None

Cons
-Unable to play game

Score: 2/10

## Comments/Javadocs

-Sufficient javadocs
-Lack of comments

Score: 8/10

## Unit Testing

-Passed 8/54 tests
-Too many “NullPointerExceptions”

Score: 2/10

# Candidate E: <Stage_2>

## Modularity

-Unnamed ActionListeners
-Lack of crucial getter methods
-Hard to edit

Score: 5/10

## Readability

-Fairly readable
-Presence of excessive code/code duplication

Score: 7/10

## Functionality

Pros
-Works perfectly

Cons
-None

Score: 10/10

## Comments/Javadocs

-Sufficient & good javadocs
-Fair amount of comments

Score: 9/10

## Unit Testing

-6000+ tests!
-All pass

Score: 10/10

# Final Scores

Candidate A

Score: 35/50

Candidate B

Score: 45/50

Candidate C

Score: 32/50

Candidate D

Score: 24/50

Candidate E

Score: 41/50

# DISCLAIMER

The views and opinions expressed in this review are those of the authors only and do not necessarily reflect that of any external audience including the developers of the work being critiqued. The authors are therefore, not liable for any errors, omissions, or discrepancies in this information or any conflict arising from its display or use.

# Contributors

- Matthew Greene
- Hanming Liu
- Aniruddha Nandi [M]