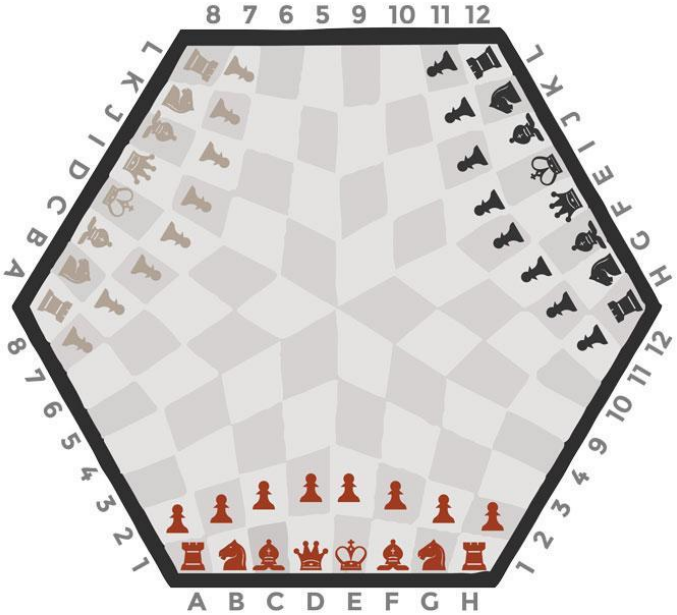
Trio-Chess

SSW690 Software Engineering Studio

Problem Statement

To validate the moves played by all the three players on a hexagonal chessboard in order to lead a legitimate game.



Development Plan

Roles and Responsibilities

	OGADINMA	RUCHA
Developer	✓	✓
Tester	✓	✓
Documentation	✓	✓
Designer	✓	✓
Customer Representative	✓	✓

Development Plan

Environment

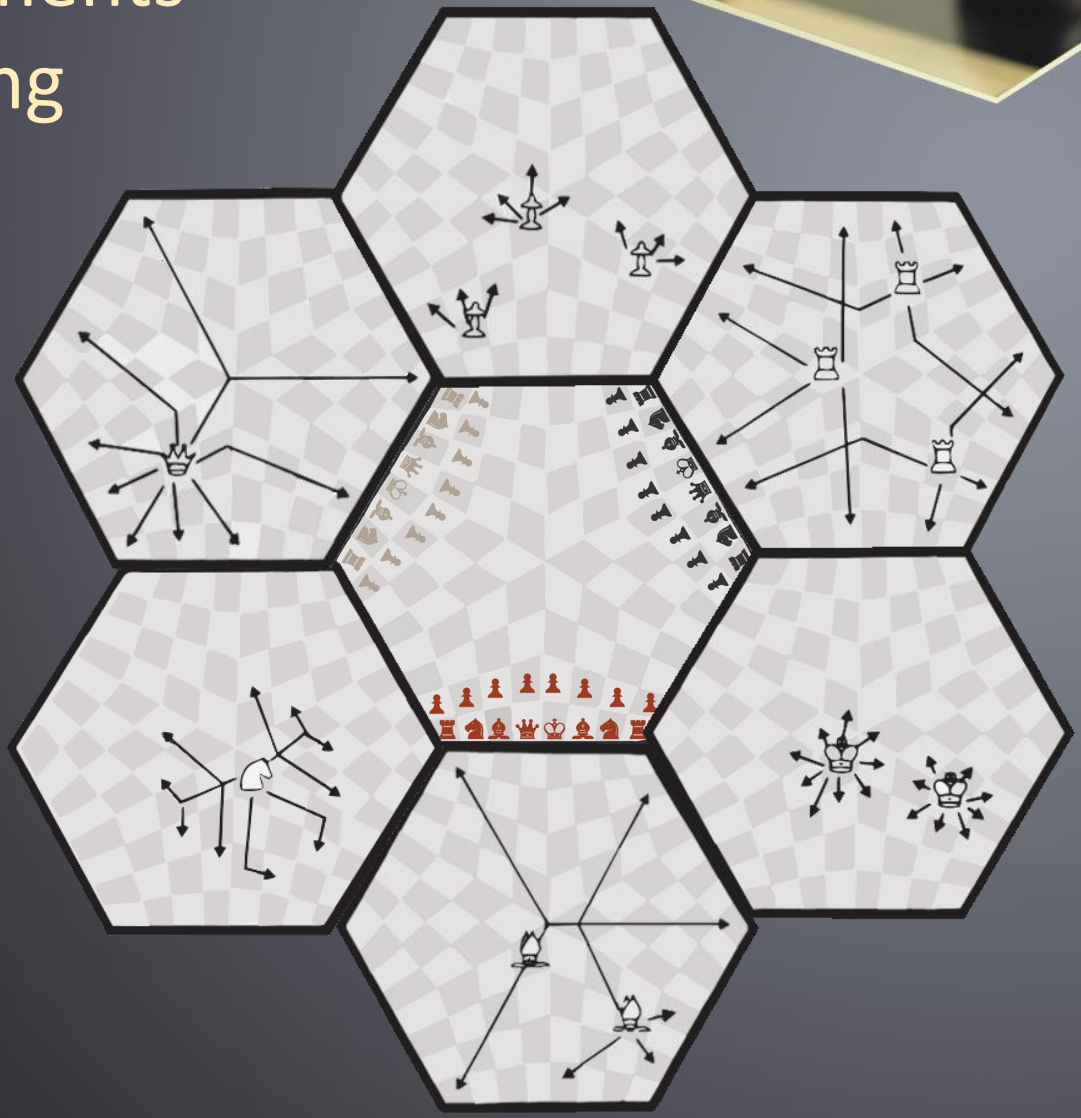
Process

- Agile Scrum

Tools, Libraries

- Python 3.7
  - turtle
  - unittest
  - math
  - pylint
  - Coverage
  - radon
- Visual Studio Code
- GitHub
  - https://github.com/RuchaCB/Trio_Chess

Requirements Gathering



Functional Requirements

- **UC 01: The system shall display hexagonal chess board.**
- **UC 02: The system shall display pieces on the chess board**
- **UC 03: The system shall take chess standard level user inputs**
- **UC 04: The system shall define player turns**
- **UC 05: The system shall identify pieces and their location on the board**
- **UC 06: The system shall validate pieces moves**
- **UC 07: The system shall move pieces on the board**

Non-Functional Requirements

- **The system shall be:**
  - **UC N01: Usable**
  - **UC N02: Testable**
  - **UC N03: Reliable**
  - **UC N04: Maintainable**

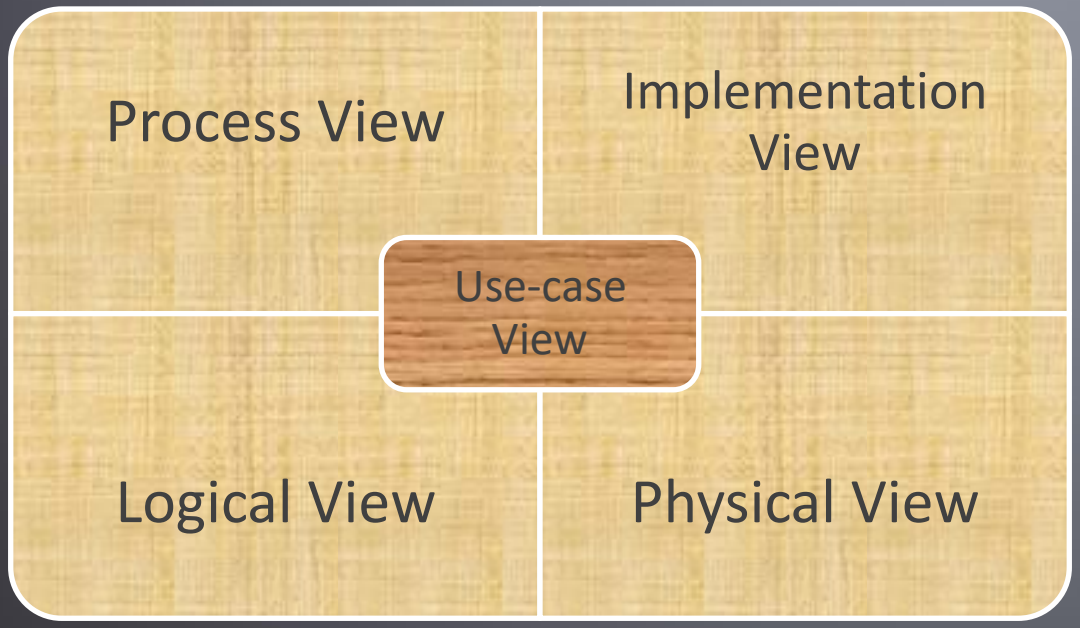
Estimations

1. Unadjusted Actor's Weight = UAW = 3
2. Unadjusted Use Cases Weight = UUCW = 50
3. Unadjusted Use Case point(UCP) = UAW+UUCW = 53
4. Total Technical Factor(TTF) = 33
5. Environmental Total Factor (ETF) = 15
4. Technical Complexity Factor (TCF) = 0.6 + .01 * TTF =0.93
5. Environmental Complexity Factor (ECF) = 1.4 - (0.03 *ETF) = 0.95
6. Calculate the Adjusted UCP

   i.e. AUCP = UUCP * TCF * EF = 46.8
7. Productivity Factor = 14
8. Hours = UCP*PF = 656
9. Effort in weeks = 16

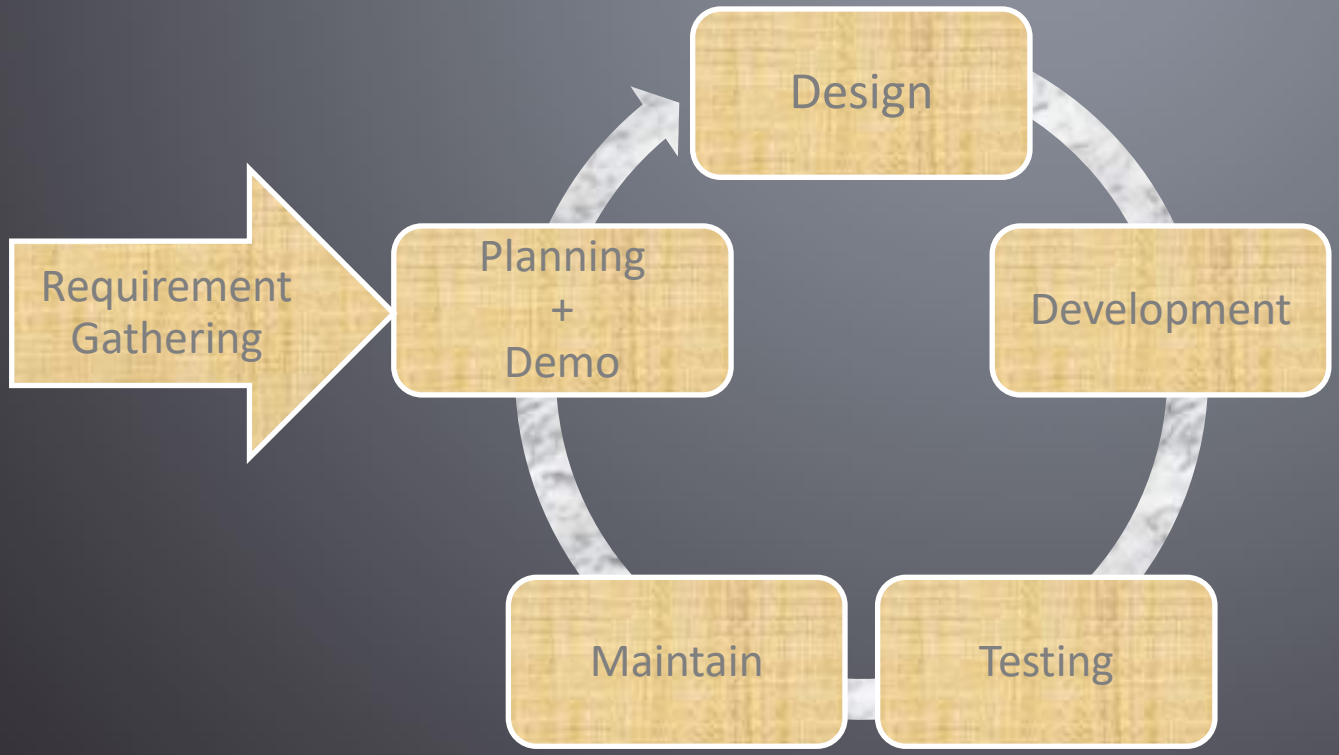
Architecture

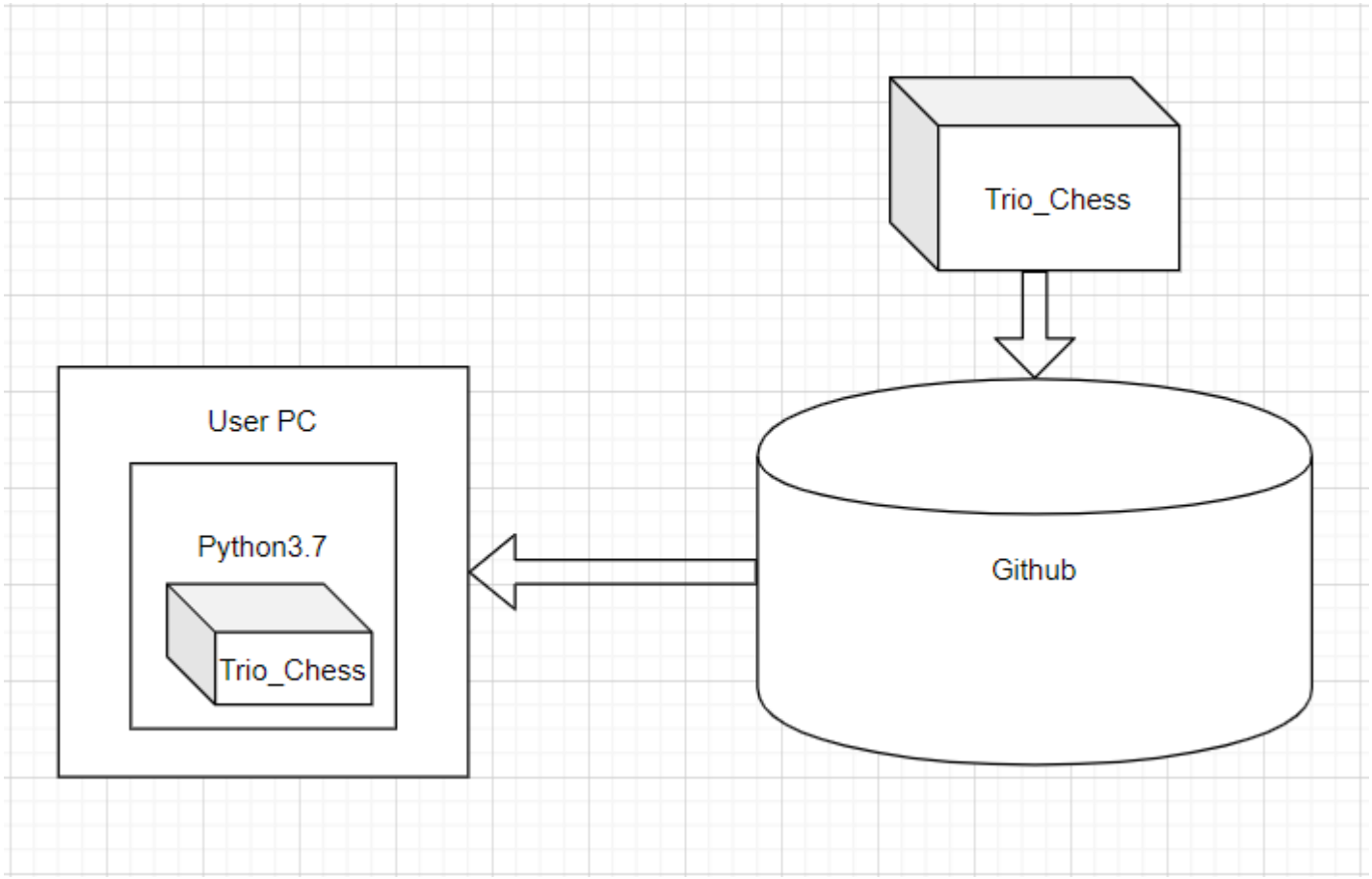
4+1 Model



Architecture

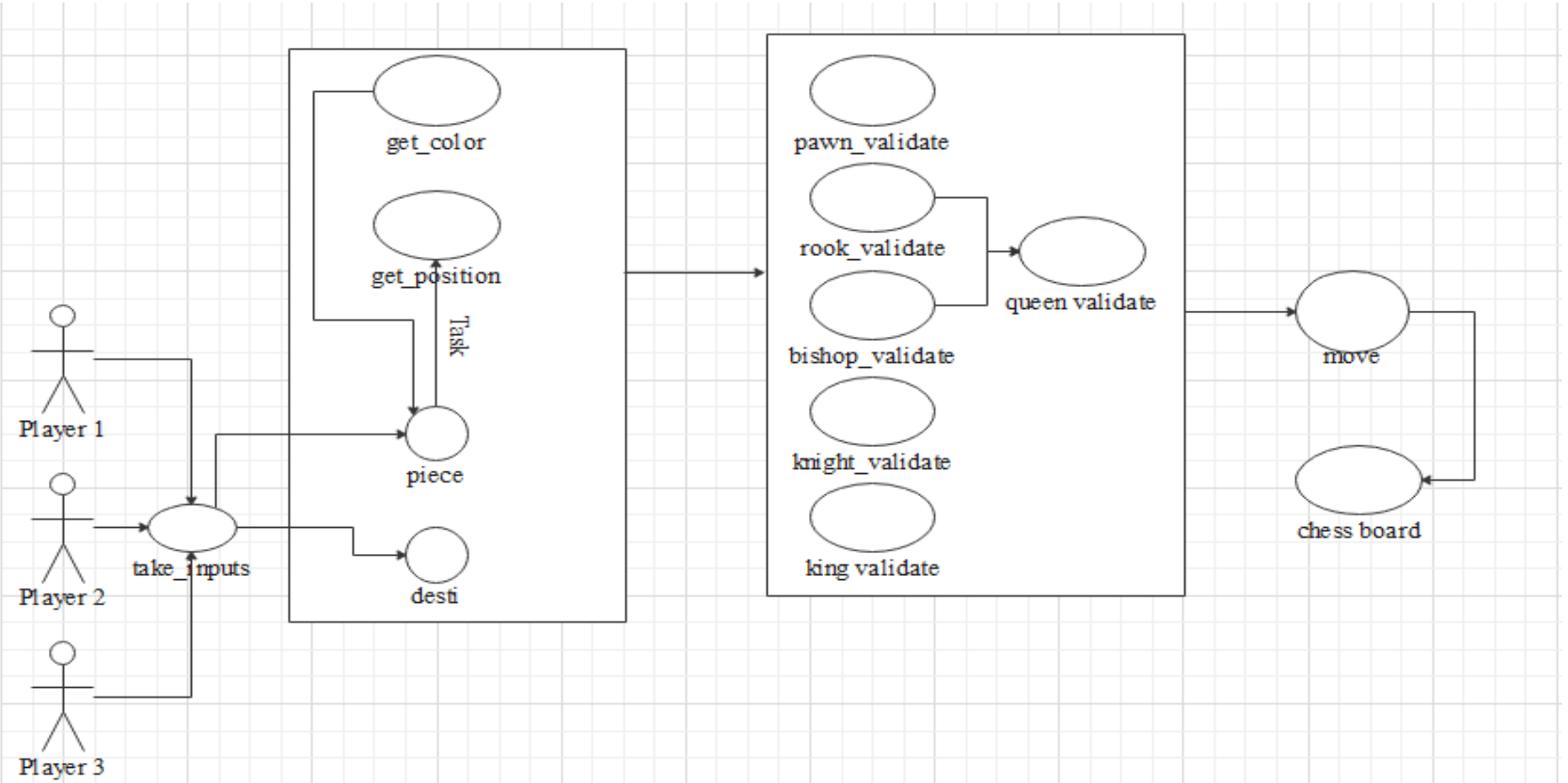
Process View





Architecture

Physical View

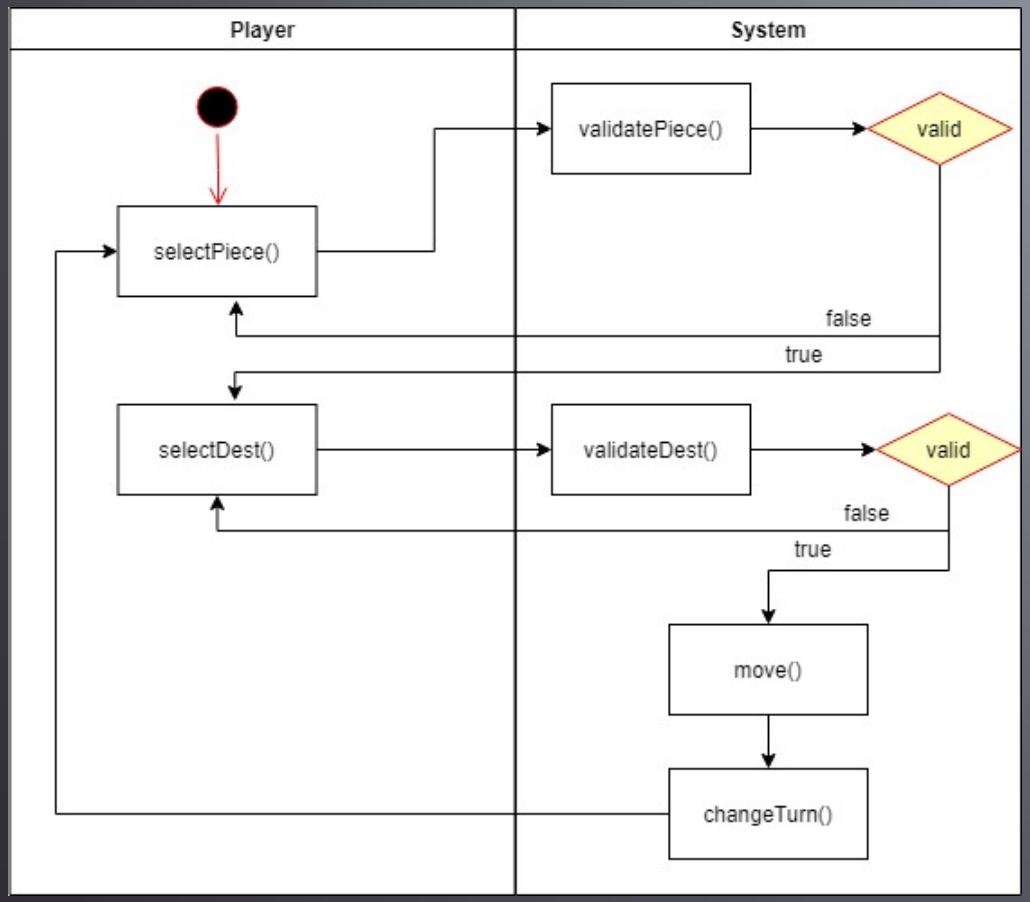


Architecture

Use Case View

Architecture

Logical View



Architecture

Implementation View

Use Case Scenario

REQ 1	UC01	UC02	UC03	UC04	UC05	UC06	UC07	UC08	
REQ 2	X								
REQ 3				X					
ROQ 4			X						
REQ 5							X		
REQ 6					X				
REQ 7						X			
REQ 8		X						X	

Cyclomatic Complexity

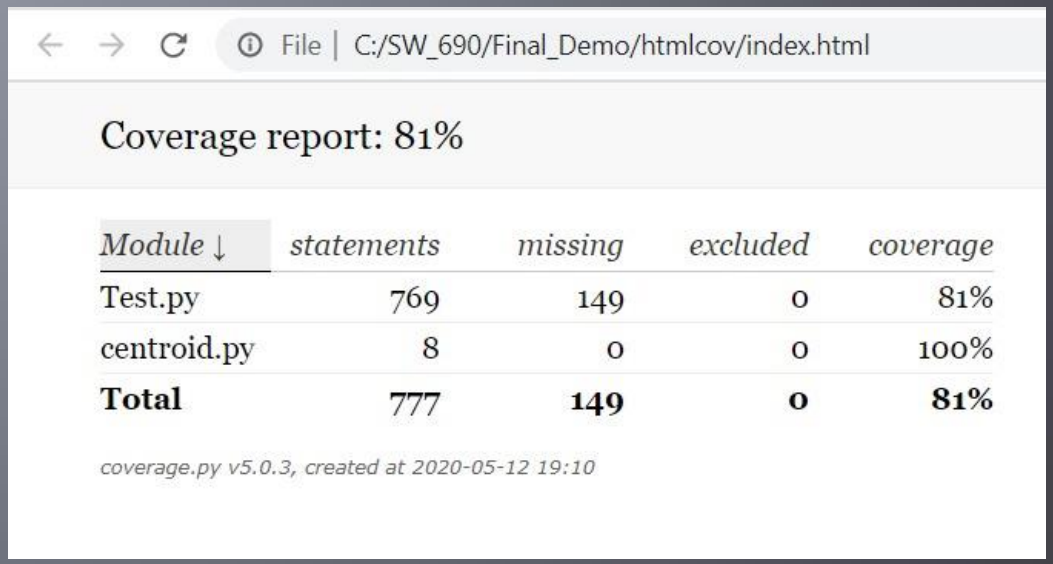
```
                          > radon cc -s Final.py
Final.py
    F 678:0 pawn_validate - F (54)
    F 608:0 king_validate - D (23)
    F 790:0 move - C (19)
    F 627:0 knight_validate - C (17)
    F 580:0 bishop_validate - C (13)
    F 523:0 rook_validate - C (11)
    F 888:0 take_inputs - B (7)
    F 778:0 update - B (6)
    F 871:0 ask - A (5)
    F 551:0 triangle - A (4)
    F 516:0 get_pos - A (3)
    F 600:0 queen_validate - A (3)
    F 857:0 validate_desti - A (3)
    F 864:0 capture - A (3)
    F 157:0 alternate_color_1 - A (2)
    F 163:0 alternate_color_2 - A (2)
    F 505:0 start_pos - A (2)
    F 850:0 validate_piece - A (2)
    F 108:0 square - A (1)
    F 127:0 square_2 - A (1)
    F 142:0 square_3 - A (1)
PS C:\SW_690\Final_Demo>
```

LOC Stats

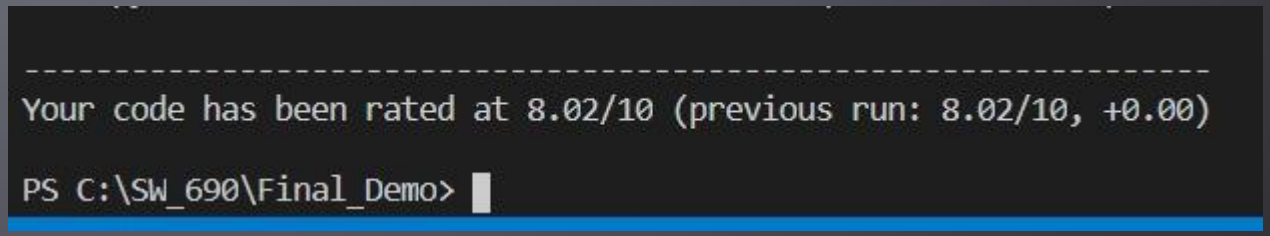
```
PS C:\SW_690\Final_Demo> radon raw Final.py
Final.py
    LOC: 905
    LLOC: 730
    SLOC: 750
    Comments: 13
    Single comments: 20
    Multi: 30
    Blank: 105
    - Comment Stats
        (C % L): 1%
        (C % S): 2%
        (C + M % L): 5%
PS C:\SW_690\Final_Demo>
```

Static Code Analysis

Code Coverage Analysis



Code Standard Analysis



Unit Testing

```
PS C:\SW_690\Final_Demo> & "C:/Program Files
Running Trio Chess Game unit tests
..........
----------------------------------------------
Ran 10 tests in 0.006s

OK
PS C:\SW_690\Final_Demo>
```

Trio-Chess Game -Metrics

Method used: Goal Question Metrics:

Goal: To design a Trio-Chess board for players that meets standard Chess game functionality

Question: Does the project meets Chess standard functionality?

Metric: Design and developed software met the standard required.

Post Morten

What would you do different?

- Piece recognition using game-tracking software tools
- Cross Entropy of the Trio-Chess game. It is intended as a more descriptive form of classification accuracy, because it takes into account the margin of probability scores (results) of the game.

Post Morten

What would you do same?

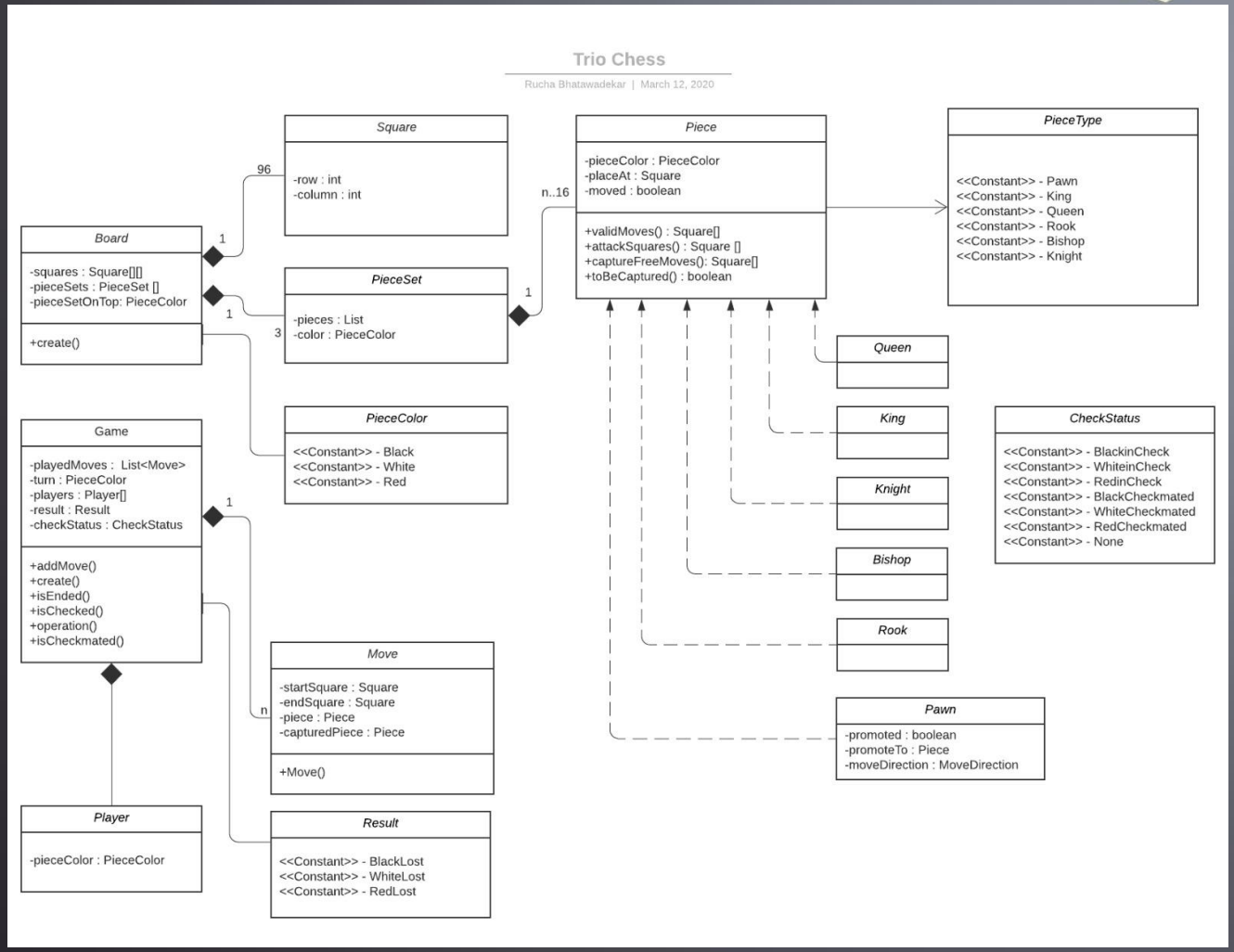
- More on team collaboration and communication on sub-projects.
- Feature extraction using scale-invariant feature transform (SIFT) and histogram of oriented gradients (HOG)

Future Work

Tracking movements of pieces within a specified area with the application Machine Learning and Artificial intelligent

Improvement on Corner-based using corner detection and board recognition software.

Future Work

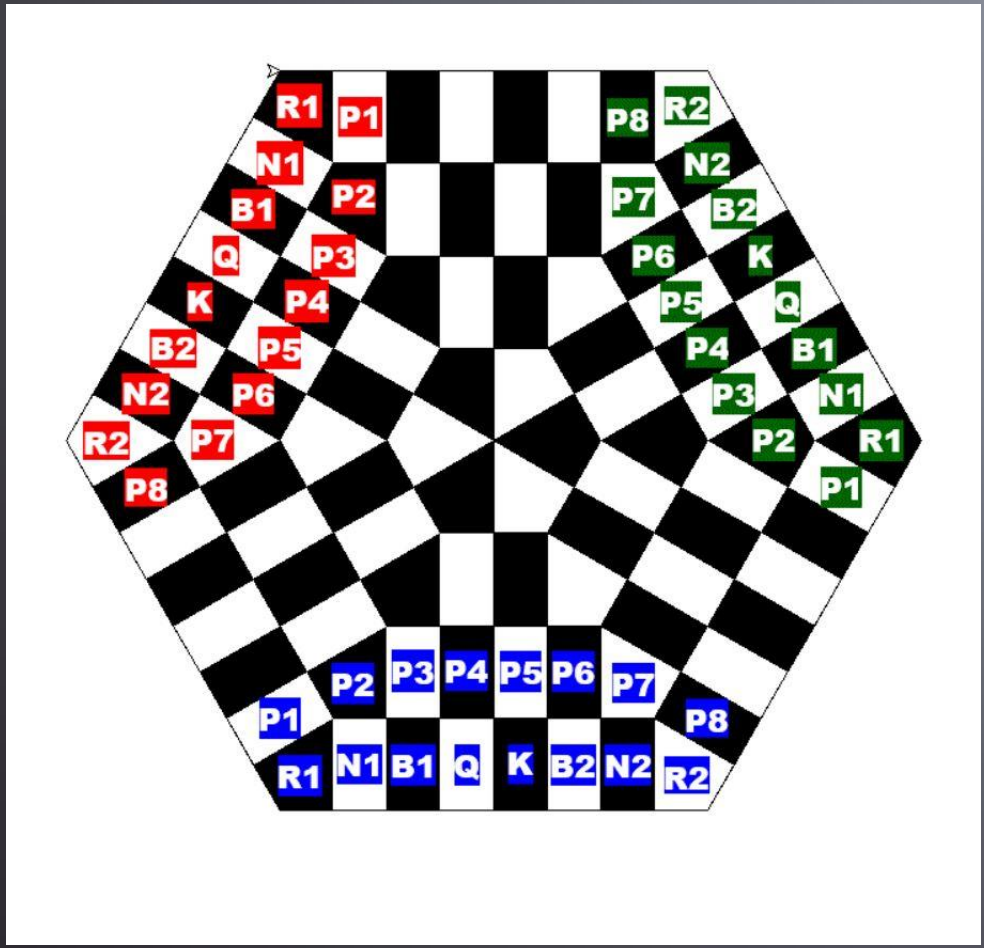


Conclusions

This project presented a proof of concepts of Trio-Chess game for corner detection which is more robust to real life game scenarios.

The major problem encountered was the lack of comprehensive data set of pieces images on the board.

Conclusions



References

- https://web.stanford.edu/class/cs231a/prev_projects_2016/CS_231A_Final_Report.pdf
- N. Dalal and B. Triggs, ”Histograms of Oriented Gradientsfor Human Detection”, inProc. IEEE Conf. Computer Visionand Pattern Recognition, 2005
- N. Loewke. (2015).Depth-based image segmentation[Online]. Available:http://stanford.edu/class/ee367/Winter2015/reportloewke.pdf
- https://tallyfy.com/uml-diagram/#activity-diagram