

# Software Architecture Document

## Purpose:

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

## Deployment :

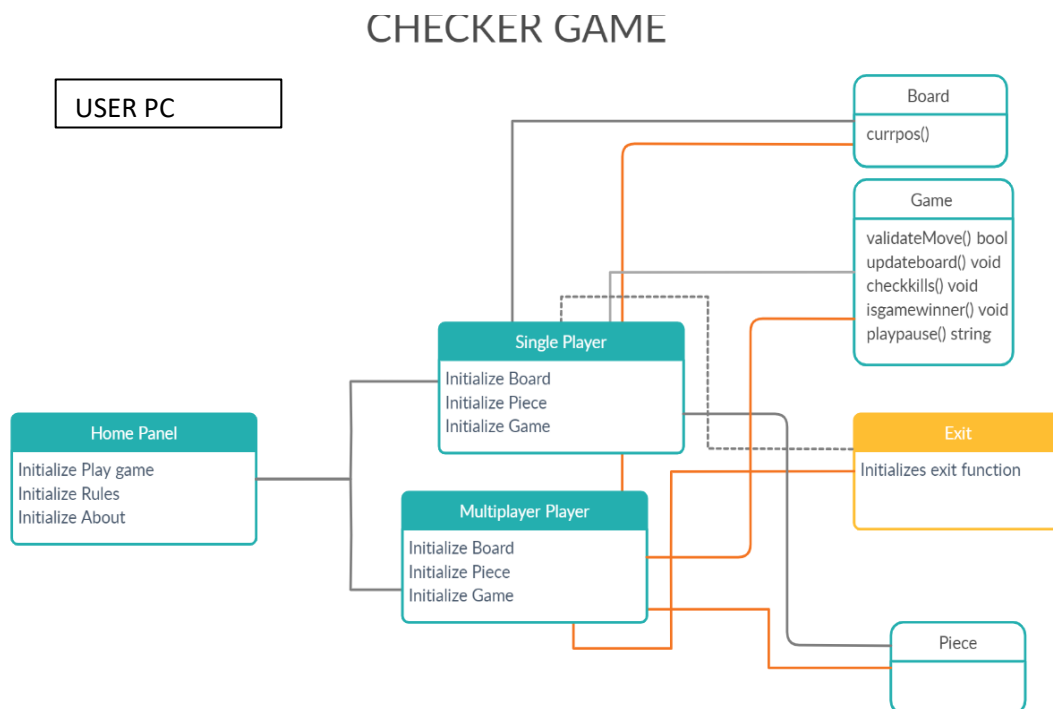
In the deployment it's working on User's PC and then, when user will load the game it load the

Home Panel, in home panel it loads the Play game, rules and about.

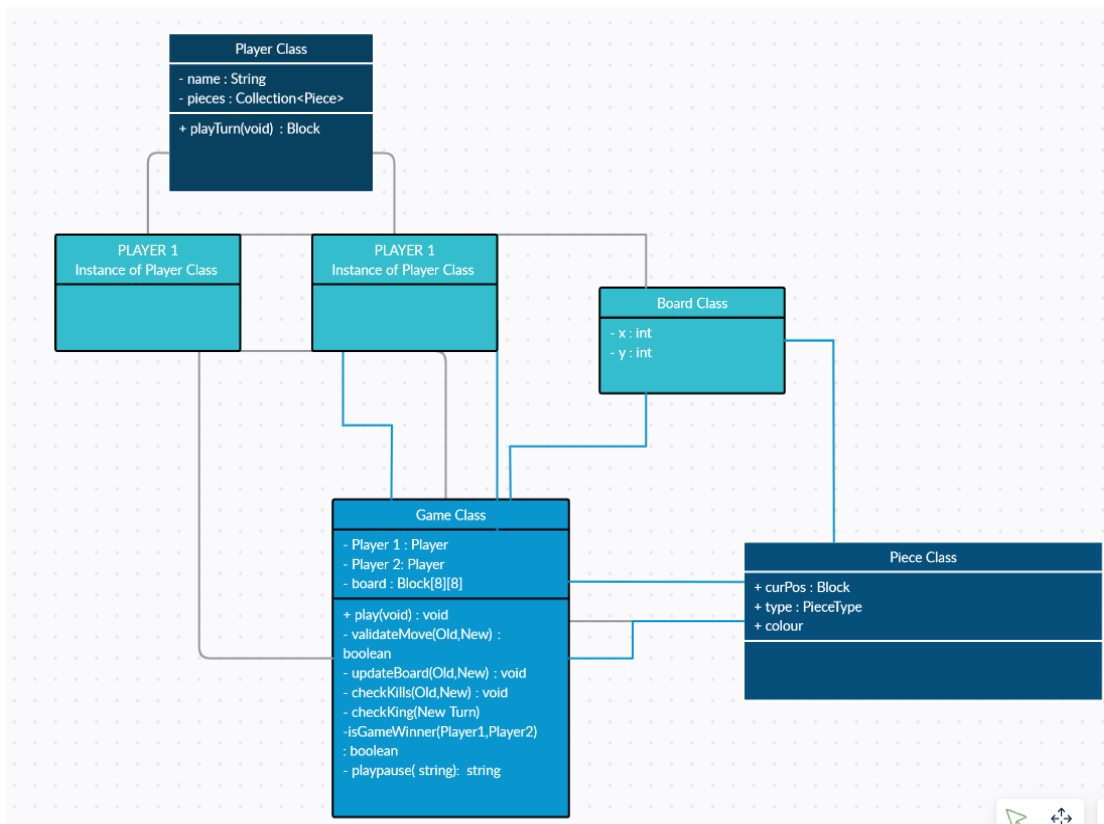
Single/Multiplayer: Call Board, Game, Exit, Piece.

Exit, Helps user to exit from the game.

Deployment View:



## Class Diagram:



## Classes:

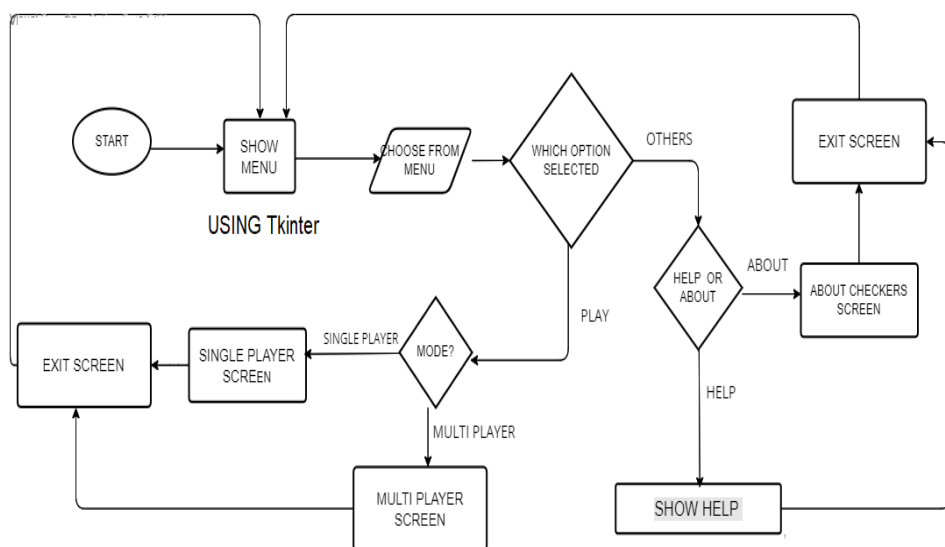
- **Player Class:** contains details of the player as well as their tokens: it has a function `playTurn` where after every player's turn we will change the count variable (global) which will decide whose turn it is.
- **Game Class:** This class has all the functions regarding game play which are `validateMove`, `updateBoard`, `checkKills`, `checkKing`, `isGameWinner`, `playpause`.
- **Piece Class:** It has `curpost`(function) which gives current position of checker, `checkerkills` takes the record of how many checker removed, `isGameWinner` find the Winner of game.
- **Baord,** It helps to get the distance of x and y position of checker from board.

## Scenarios:

## Flow chart of Game:

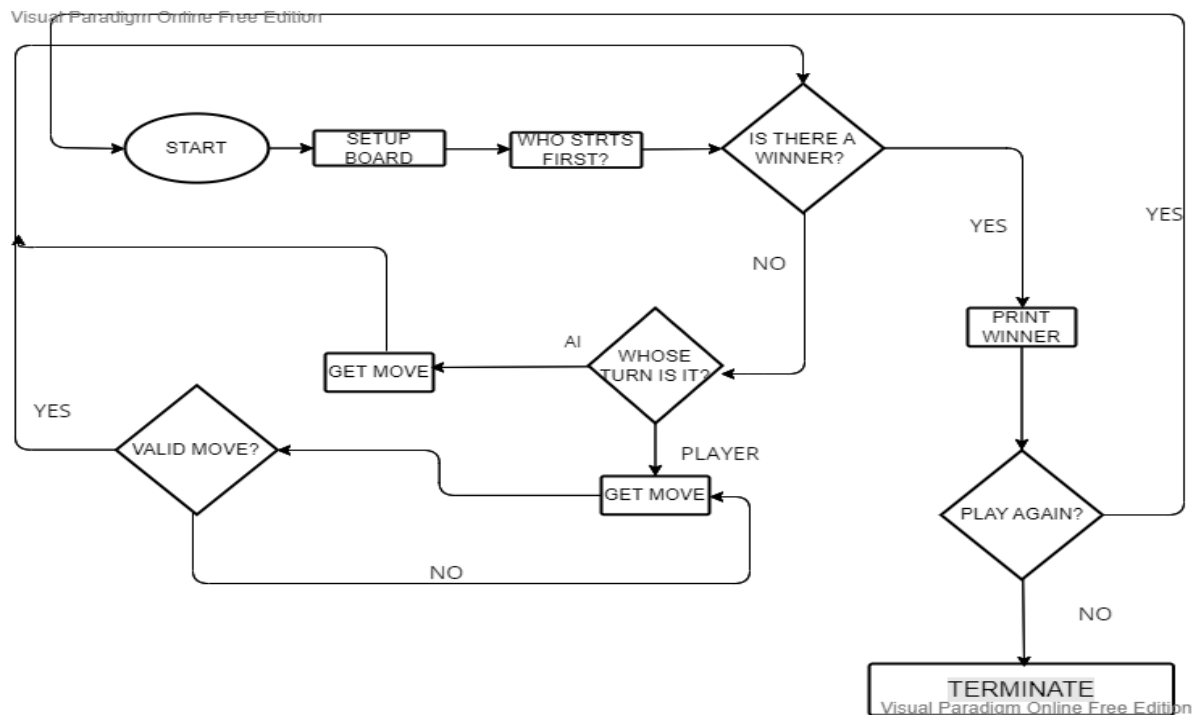
### MAIN SCREEN:

This Flow-chart show, what will happen on the Main Screen.



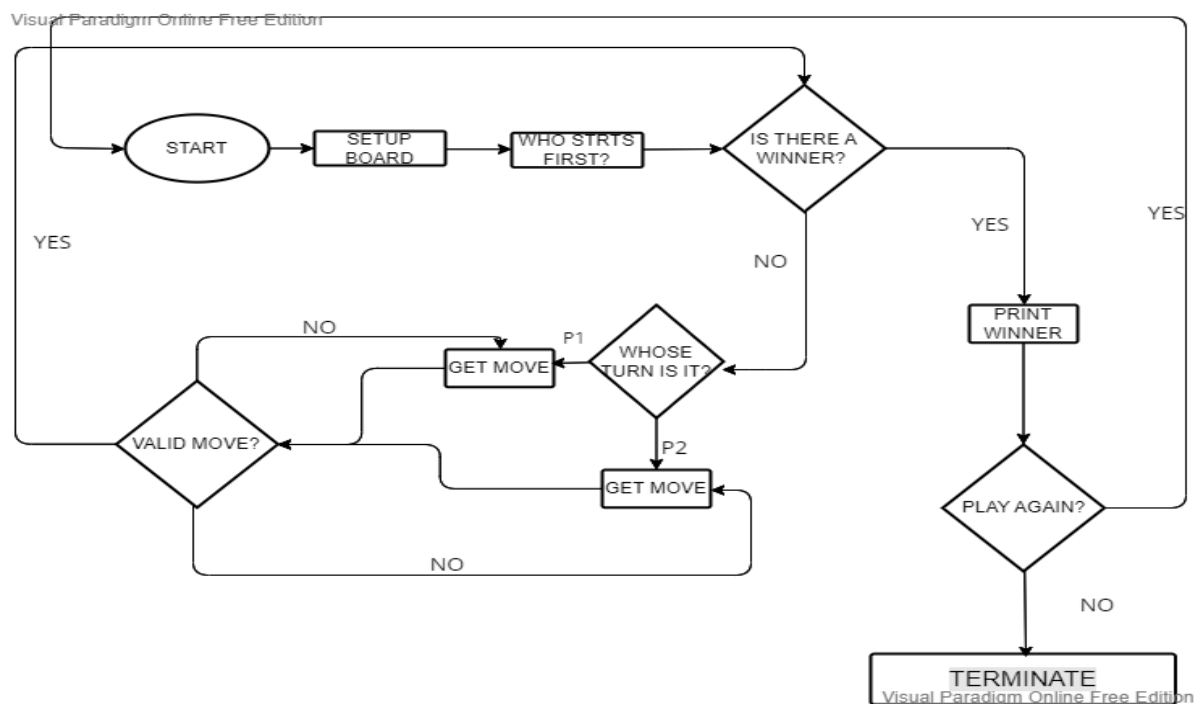
Tkinter use to for menu program, then flow shows, choose menu where we need to choose options to play or go for about option, if user select play option then get a mode to choose single player or Multiplayer.

## Single Player:



Single player mode, to choose setup a board, to find who'll go first, then check for moves validate move and goes in loop until get move has no input, after that checks for print winner and then ask for play again and end.

## MULTI-PLAYER:



## Use Case scenarios:

Use case name: Move piece

Actors: Player (Primary)

Pre-conditions: \*\*\*\*\*

Post-conditions: \*\*\*\*\*

Basic Path:

1. Player selects one piece and the destination square, and submit a move request.
2. System validates destination square.
3. System moves the piece and calculates the moving.
4. System displays the moving result.

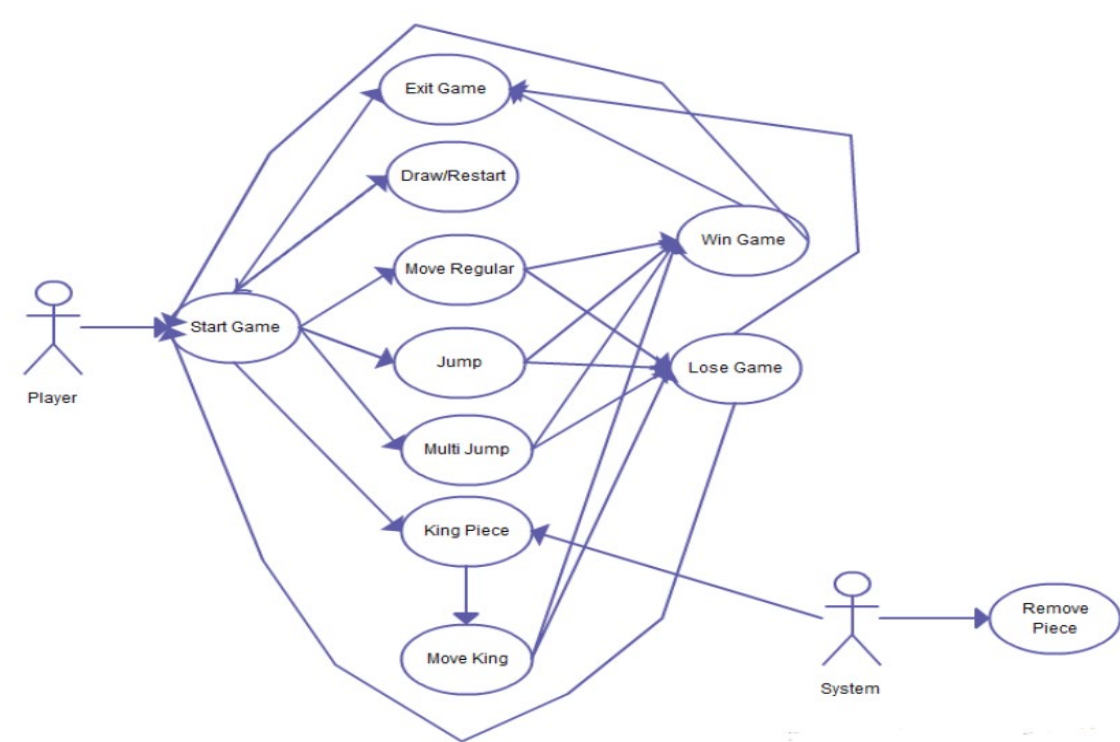
Exception Path:

2a. destination square is not valid:

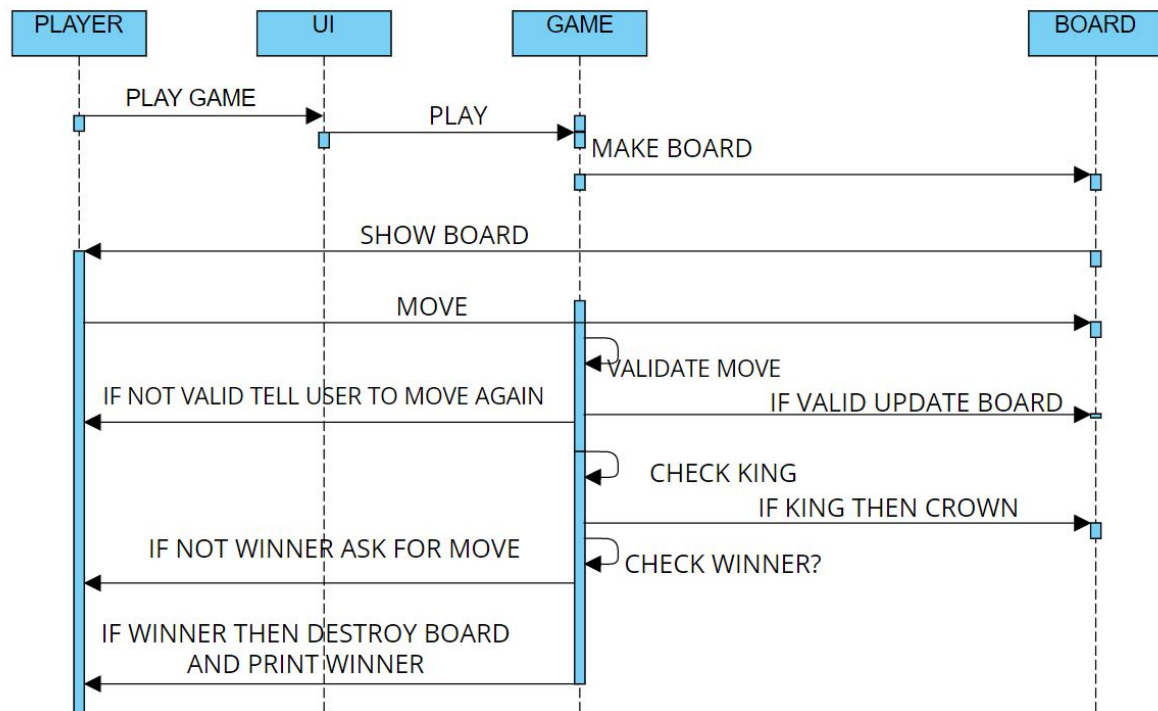
2a1. System \*\*\*\*

Business Rule:

5. valid destination square: .....
6. calculating rule, such as king piece, win game.....



## SEQUENCE DIAGRAM:



The user will click on the play game button on the GUI interface then UI will send the play command to game ,game will send the create board command to the board class.

The board class will show the board to the user

User will move and the move will be validated by the GAME class if the move is valid then the board will be updated else user will be told to move again.

The game will check if any king is made then the piece will be crowned.

The Game will check if there is any winner then declare the winner else ask user for another move.

## Libraries used:

Tkinter

Pygame

## Resources

:

RAM to load the application

GUI cursor

Heap Memory