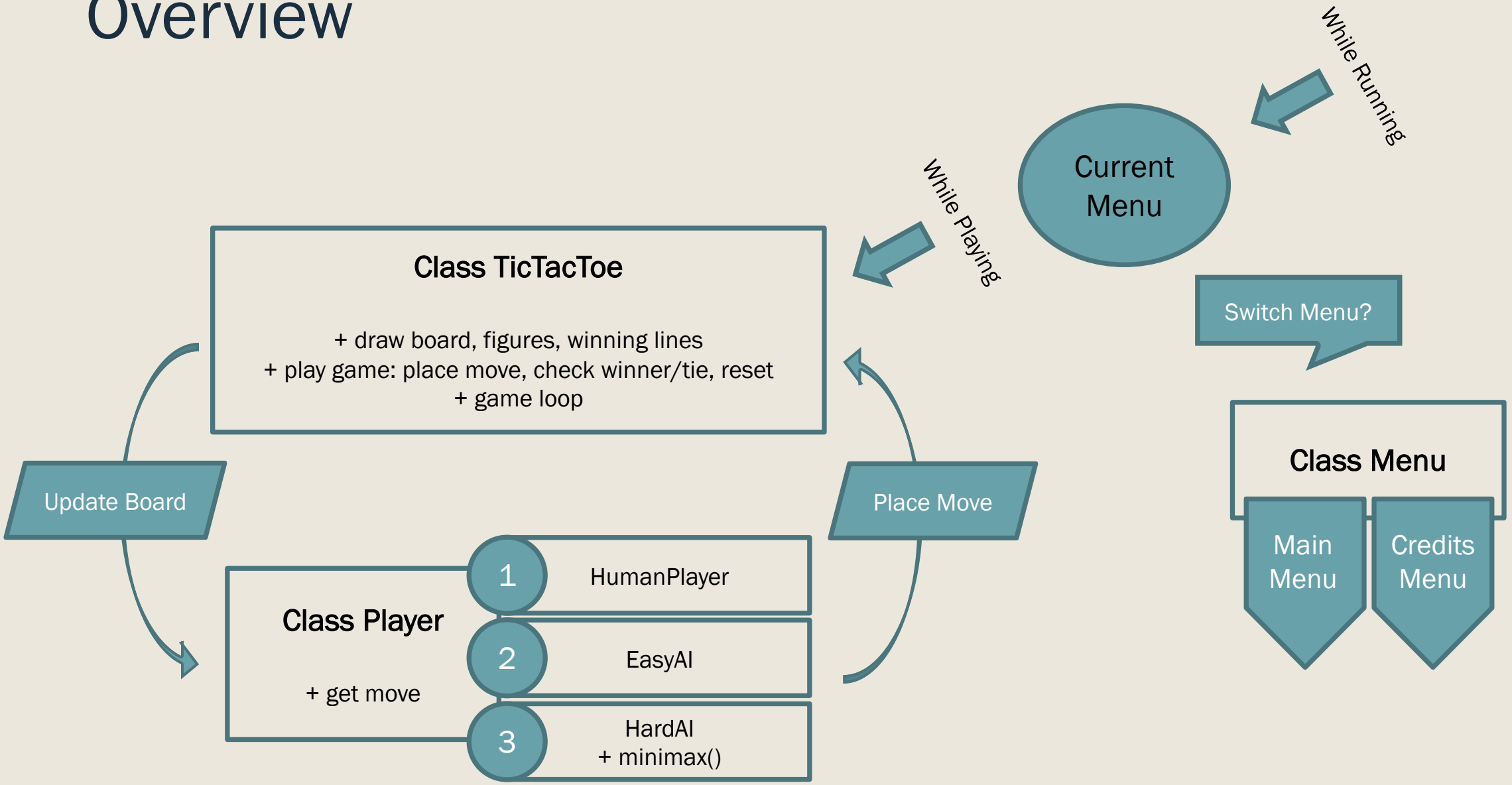
TIC TAC TOE

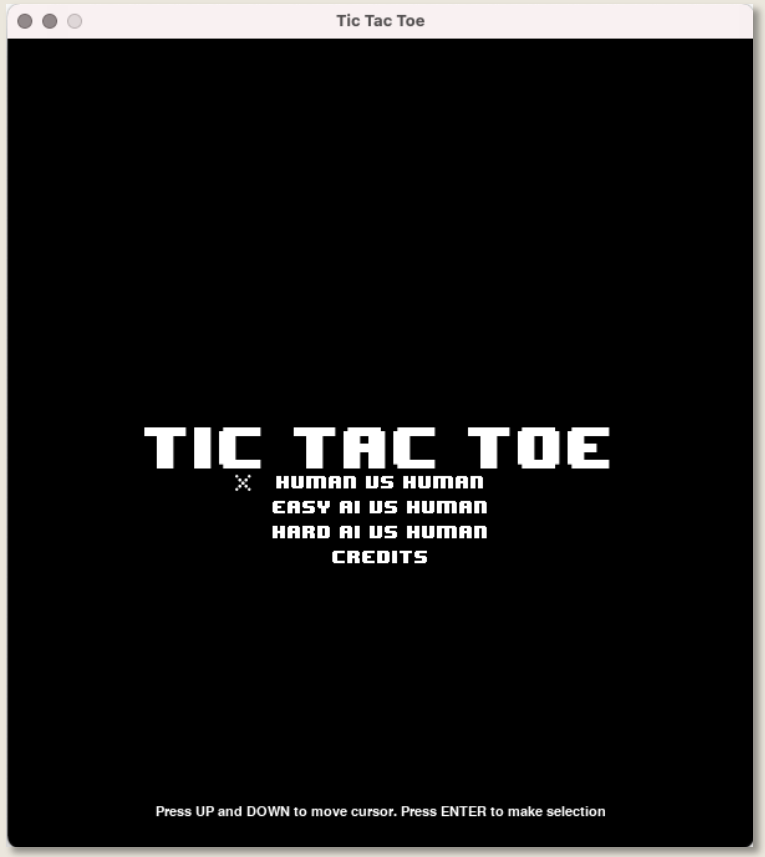
Beijing Wu

Overview

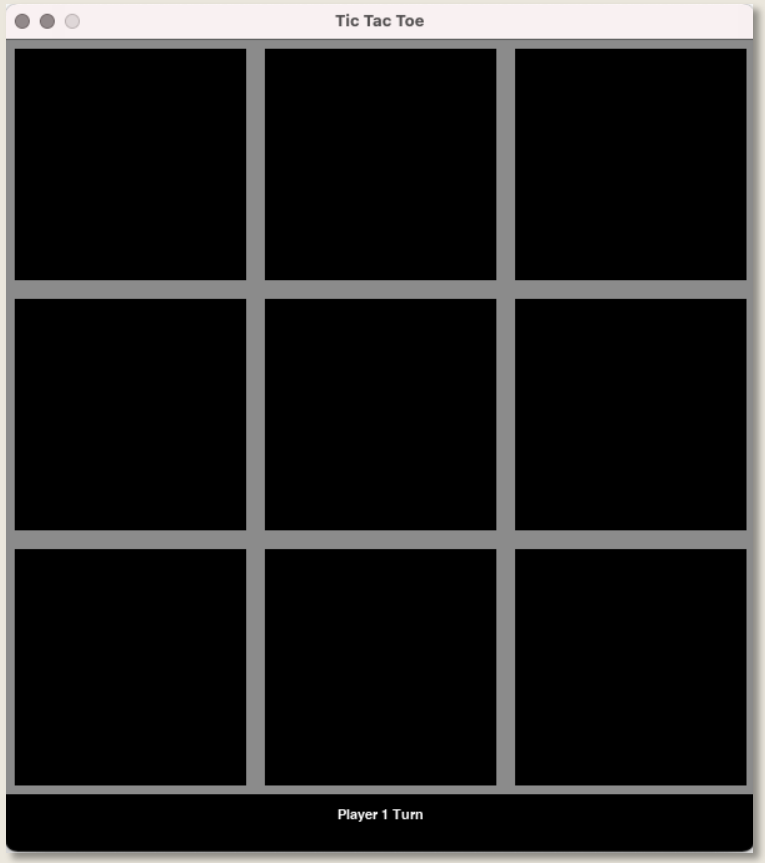

While Running
Current Menu
While Playing
Switch Menu?
Class TicTacToe
+ draw board, figures, winning lines
+ play game: place move, check winner/tie, reset
+ game loop
Update Board
Place Move
Class Menu
Main Menu
Credits Menu
Class Player
+ get move
1
HumanPlayer
2
EasyAI
3
HardAI
+ minimax()

Game Interface: pygame

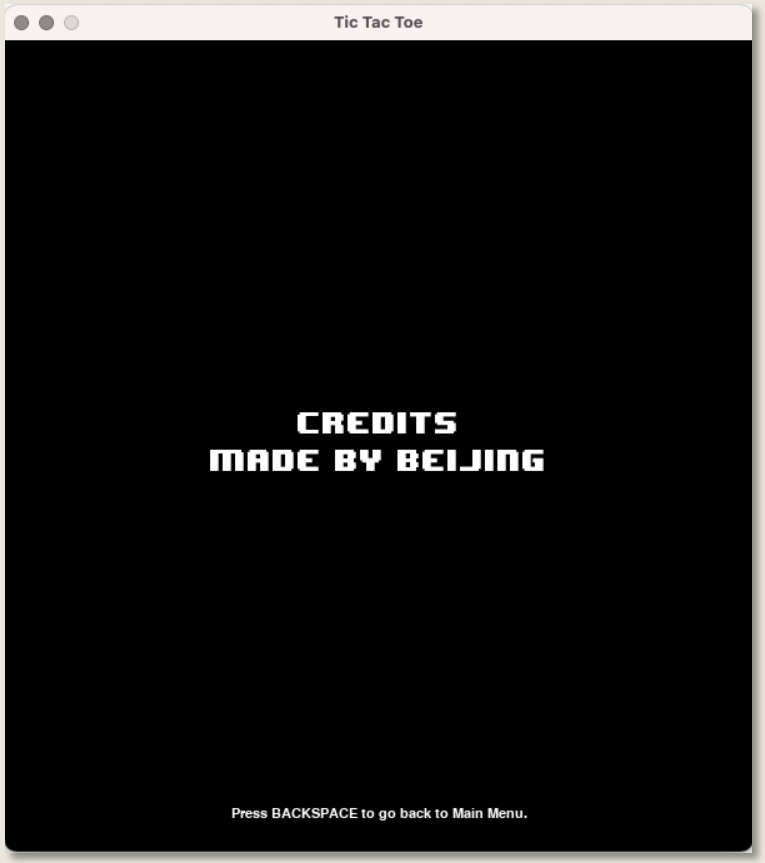
Main Menu



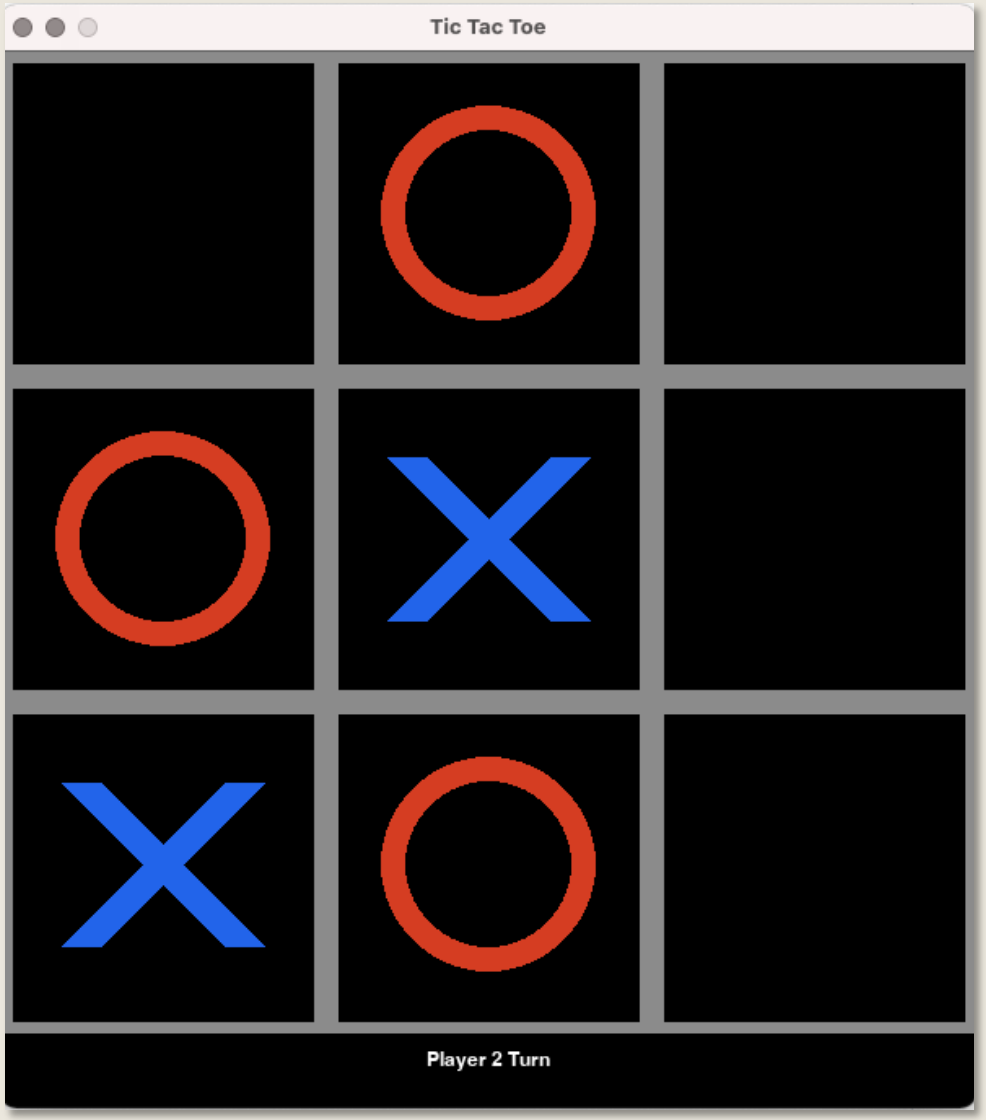
Game Board



Credits



Game Board



- User click at empty spot on the board to make move

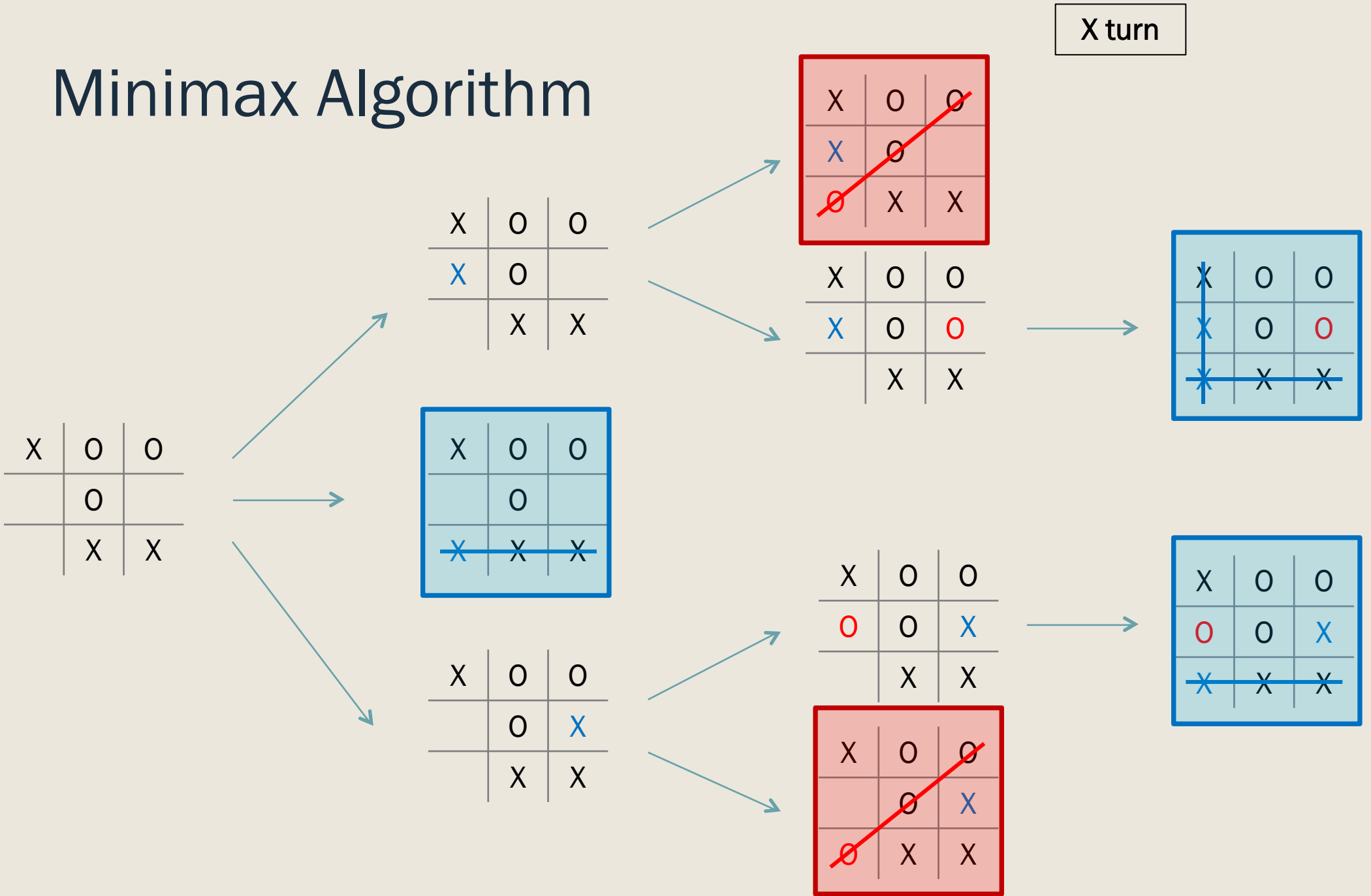
Game Mode

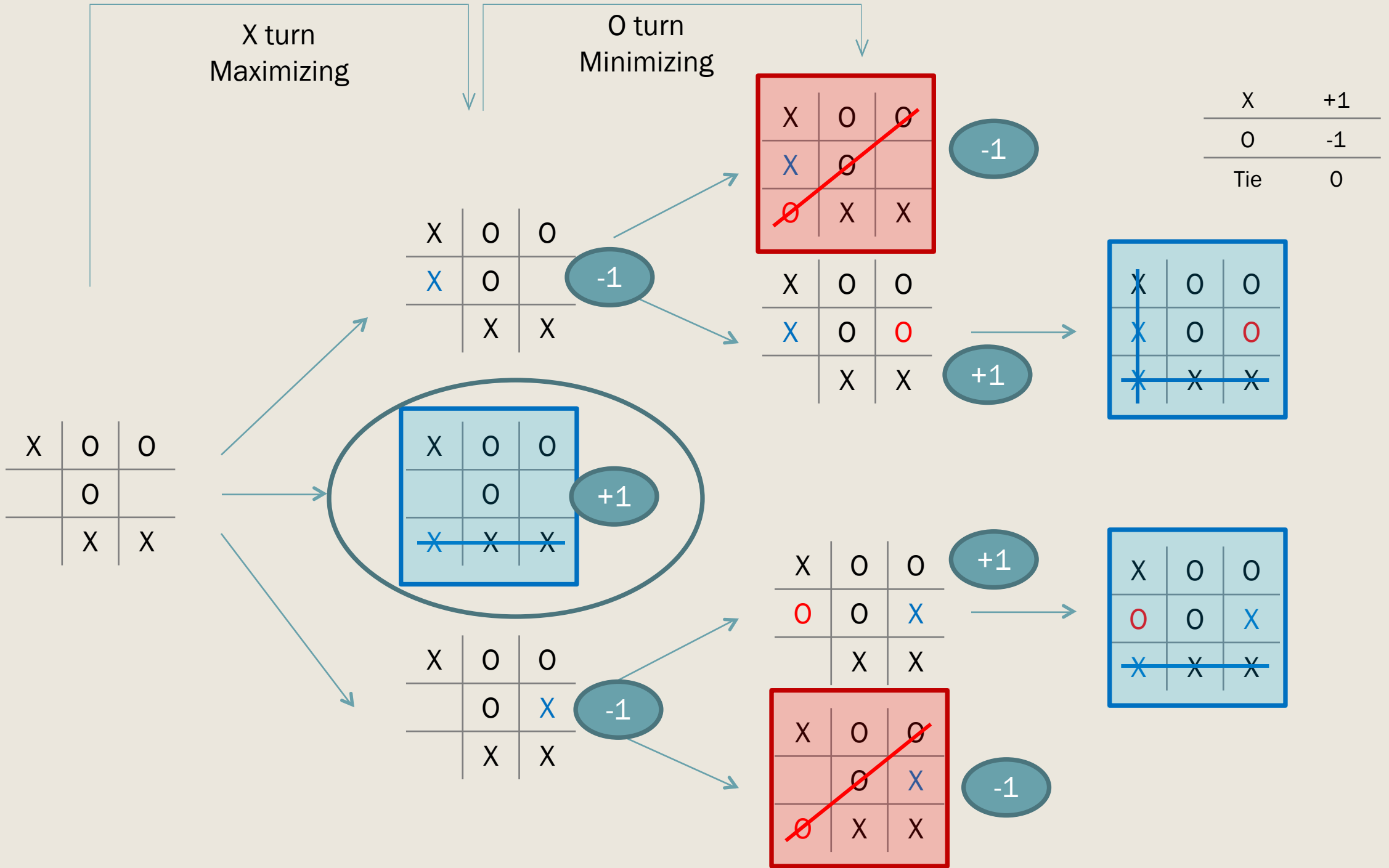
- **Human vs. Human**

- **Easy AI vs. Human**
  - Computer will randomly choose an empty spot on the board to make the move

- **Hard AI vs. Human**
  - Computer will use minimax algorithm to choose the move – maximize chance to win and minimize opponent's chance to win

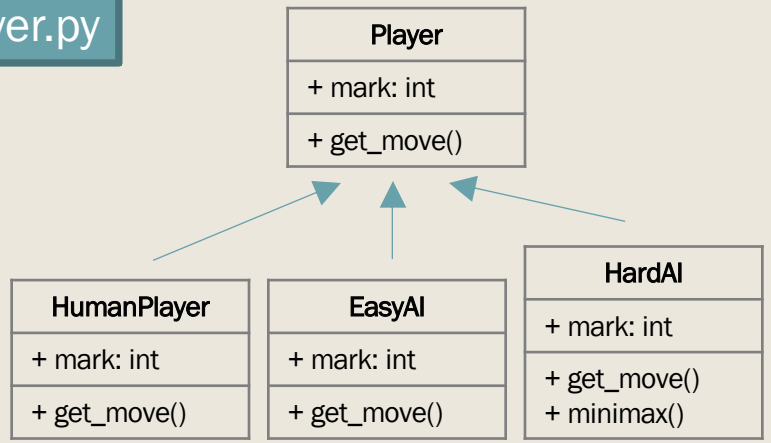
Minimax Algorithm



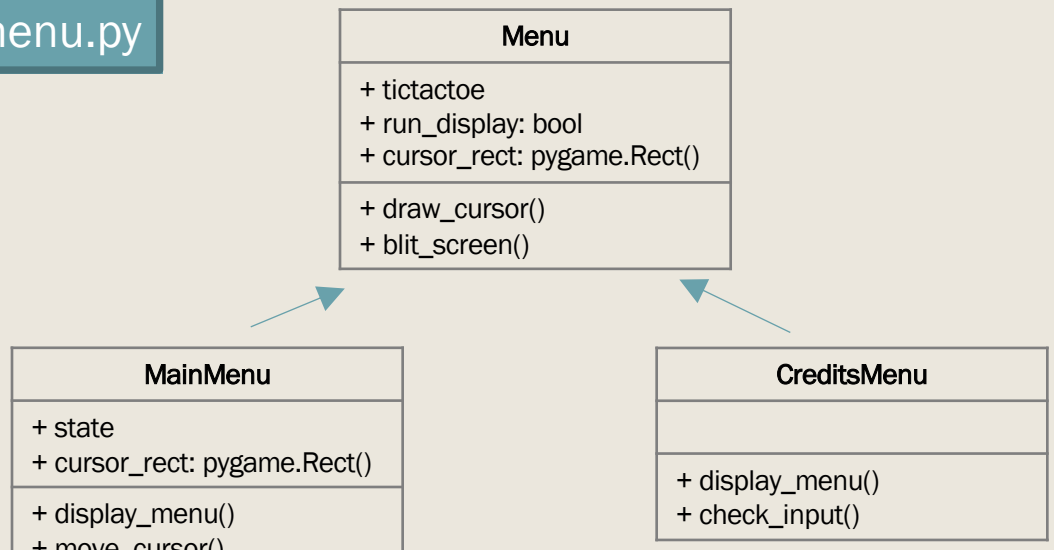

X turn
Maximizing
O turn
Minimizing
X O O
O
X X
X O O
X O
X X
-1
X O O
X O O
X X
+1
X O O
X O
X X
+1
X O O
X O
X X
-1
X O O
X O X
X X
-1
X O O
O O X
X X
+1
X O O
O O X
X X X
X O O
O O X
X X X
-1
X +1
O -1
Tie 0

Full Program Design

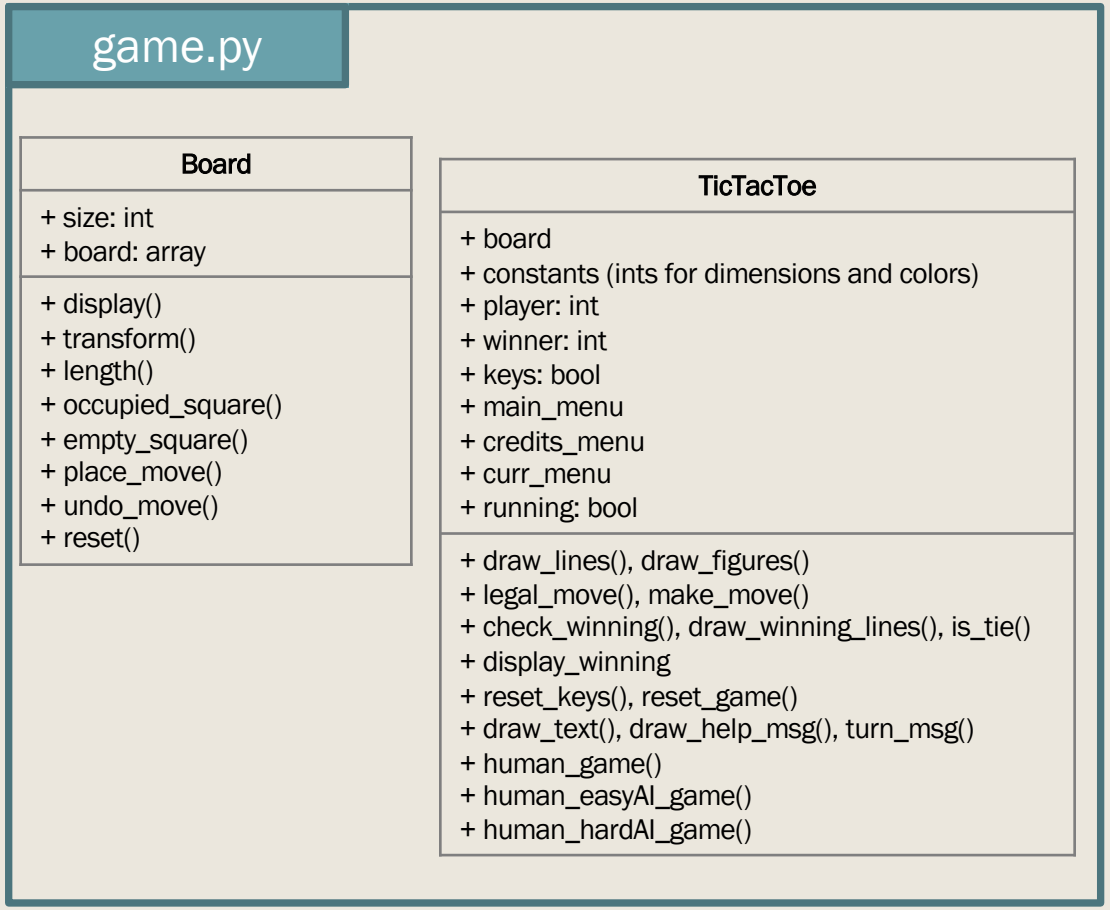
player.py



menu.py



game.py



main.py

```
# Start game
game = tictactoe()
while game.running:
  game.curr_menu.display_menu()
  game.game_loop()
```

Challenges and Future Direction

- Player1 and Player2 are hardcoded (i.e., can’t do human vs. machine yet).

- Future Direction
  - Menu options can be improved.
    - Add “Options” to let user choose player 1 and player 2.
    - Add “Options” to change screen display size, colors.
  - Add score board.
  - The code base can be easily adapted to create more complicated game, such as Connect4.