```
┌─────────────────────────────────────┐
│            piece.Piece              │
├─────────────────────────────────────┤
│ availableLocs : list                │
│ locs : list                         │
│ lolim : int                         │
│ moveNum : int                       │
│ name                                │
│ pctext : dict                       │
│ player                              │
│ position                            │
│ seeKing : bool                      │
│ uplim : int                         │
│ x                                   │
│ y                                   │
├─────────────────────────────────────┤
│ availableMoves()                    │
│ checkBounds(intervals)              │
│ checkKing(board)                    │
│ checkpiece(board, intervals)        │
└─────────────────────────────────────┘
                  △
                  │
┌─────────────────────────────────────┐
│            piece.Bishop             │
├─────────────────────────────────────┤
│ availableLocs : list                │
├─────────────────────────────────────┤
│ availableMoves(board)               │
└─────────────────────────────────────┘
```

chess.Game

board : list
cmate : bool
lolim : int
player1 : NoneType
player2 : NoneType
uplim : int
userPiece : NoneType
userx : int
usery : int

fillBoard()
initplacePieces()
printBoard()
updateMove(newloc, pc)

player1 player2

player.Player

cemetary : dict
check : bool
cmate : bool
lolim : int
pieces : dict
playerNumber
poi : NoneType
uplim : int

findKing(moves, plr)
playerPieces(sll)
removePiece(key)

piece.Piece

availableLocs : list
locs : list
lolim : int
moveNum : int
name
pctext : dict
player
position
seeKing : bool
uplim : int
x
y

---

availableMoves()
checkBounds(intervals)
checkKing(board)
checkpiece(board, intervals)

piece.King

availableLocs : list

---

availableMoves(board)
checkpiece(board, intervals)

piece.Piece

availableLocs : list
locs : list
lolim : int
moveNum : int
name
pctext : dict
player
position
seeKing : bool
uplim : int
x
y

---

availableMoves()
checkBounds(intervals)
checkKing(board)
checkpiece(board, intervals)

piece.Knight

availableLocs : list
check : bool
cmate : bool

---

availableMoves(board)
checkpiece(board, intervals)
isCheck(board)

piece.Piece

availableLocs : list
locs : list
lolim : int
moveNum : int
name
pctext : dict
player
position
seeKing : bool
uplim : int
x
y

---

availableMoves()
checkBounds(intervals)
checkKing(board)
checkpiece(board, intervals)

piece.Pawn

availableLocs : list

---

availableMoves(board)
checkpiece(board, intervals)

piece.Piece
availableLocs : list locs : list lolim : int moveNum : int name pctext : dict player position seeKing : bool uplim : int x y
availableMoves() checkBounds(intervals) checkKing(board) checkpiece(board, intervals)

player.Player
cemetary : dict
check : bool
cmate : bool
lolim : int
pieces : dict
playerNumber
poi : NoneType
uplim : int
findKing(moves, plr)
playerPieces(sll)
removePiece(key)

piece.Piece

availableLocs : list
locs : list
lolim : int
moveNum : int
name
pctext : dict
player
position
seeKing : bool
uplim : int
x
y

---

availableMoves()
checkBounds(intervals)
checkKing(board)
checkpiece(board, intervals)

piece.Queen

availableLocs : list

---

availableMoves(board)

piece.Piece

availableLocs : list
locs : list
lolim : int
moveNum : int
name
pctext : dict
player
position
seeKing : bool
uplim : int
x
y

---

availableMoves()
checkBounds(intervals)
checkKing(board)
checkpiece(board, intervals)

piece.Rook

availableLocs : list

---

availableMoves(board)