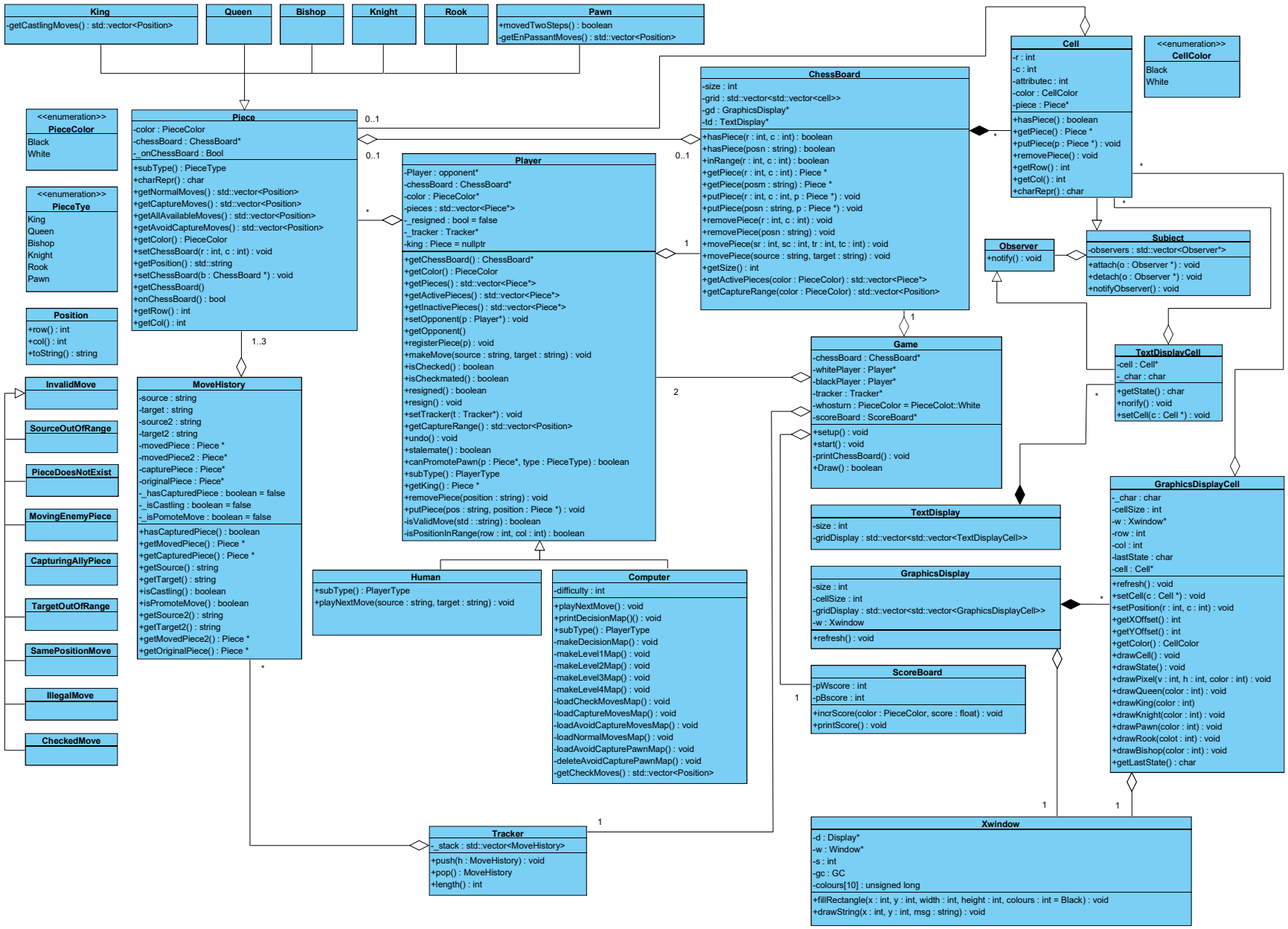
King
-getCastlingMoves() : std::vector<Position>
Queen
Bishop
Knight
Rook
Pawn
+movedTwoSteps() : boolean
-getEnPassantMoves() : std::vector<Position>
Cell
-r : int
-c : int
-attributec : int
-color : CellColor
-piece : Piece*
+hasPiece() : boolean
+getPiece() : Piece *
+putPiece(p : Piece *) : void
+removePiece() : void
+getRow() : int
+getCol() : int
+charRepr() : char
<<enumeration>>
CellColor
Black
White
ChessBoard
-size : int
-grid : std::vector<std::vector<cell>>
-gd : GraphicsDisplay*
-td : TextDisplay*
+hasPiece(r : int, c : int) : boolean
+hasPiece(posn : string) : boolean
+inRange(r : int, c : int) : boolean
+getPiece(r : int, c : int) : Piece *
+getPiece(posm : string) : Piece *
+putPiece(r : int, c : int, p : Piece *) : void
+putPiece(posn : string, p : Piece *) : void
+removePiece(r : int, c : int) : void
+removePiece(posn : string) : void
+movePiece(sr : int, sc : int, tr : int, tc : int) : void
+movePiece(source : string, target : string) : void
+getSize() : int
+getActivePieces(color : PieceColor) : std::vector<Piece*>
+getCaptureRange(color : PieceColor) : std::vector<Position>
<<enumeration>>
PieceColor
Black
White
Piece
-color : PieceColor
-chessBoard : ChessBoard*
-_onChessBoard : Bool
+subType() : PieceType
+charRepr() : char
+getNormalMoves() : std::vector<Position>
+getCaptureMoves() : std::vector<Position>
+getAllAvailableMoves() : std::vector<Position>
+getAvoidCaptureMoves() : std::vector<Position>
+getColor() : PieceColor
+setChessBoard(r : int, c : int) : void
+getPosition() : std::string
+setChessBoard(b : ChessBoard *) : void
+getChessBoard()
+onChessBoard() : bool
+getRow() : int
+getCol() : int
<<enumeration>>
PieceTye
King
Queen
Bishop
Knight
Rook
Pawn
Position
+row() : int
+col() : int
+toString() : string
Player
-Player : opponent*
-chessBoard : ChessBoard*
-color : PieceColor*
-pieces : std::vector<Piece*>
-_resigned : bool = false
-_tracker : Tracker*
-king : Piece = nullptr
+getChessBoard() : ChessBoard*
+getColor() : PieceColor
+getPieces() : std::vector<Piece*>
+getActivePieces() : std::vector<Piece*>
+getInactivePieces() : std::vector<Piece*>
+setOpponent(p : Player*) : void
+getOpponent()
+registerPiece(p) : void
+makeMove(source : string, target : string) : void
+isChecked() : boolean
+isCheckmated() : boolean
+resigned() : boolean
+resign() : void
+setTracker(t : Tracker*) : void
+getCaptureRange() : std::vector<Position>
+undo() : void
+stalemate() : boolean
+canPromotePawn(p : Piece*, type : PieceType) : boolean
+subType() : PlayerType
+getKing() : Piece *
+removePiece(position : string) : void
+putPiece(pos : string, position : Piece *) : void
-isValidMove(std : :string) : boolean
-isPositionInRange(row : int, col : int) : boolean
Observer
+notify() : void
Subject
-observers : std::vector<Observer*>
+attach(o : Observer *) : void
+detach(o : Observer *) : void
+notifyObserver() : void
Game
-chessBoard : ChessBoard*
-whitePlayer : Player*
-blackPlayer : Player*
-tracker : Tracker*
-whosturn : PieceColor = PieceColot::White
-scoreBoard : ScoreBoard*
+setup() : void
+start() : void
-printChessBoard() : void
+Draw() : boolean
TextDisplayCell
-cell : Cell*
-_char : char
+getState() : char
+norify() : void
+setCell(c : Cell *) : void
MoveHistory
-source : string
-target : string
-source2 : string
-target2 : string
-movedPiece : Piece *
-movedPiece2 : Piece*
-capturePiece : Piece*
-originalPiece : Piece*
-_hasCapturedPiece : boolean = false
-_isCastling : boolean = false
-_isPomoteMove : boolean = false
+hasCapturedPiece() : boolean
+getMovedPiece() : Piece *
+getCapturedPiece() : Piece *
+getSource() : string
+getTarget() : string
+isCastling() : boolean
+isPromoteMove() : boolean
+getSource2() : string
+getTarget2() : string
+getMovedPiece2() : Piece *
+getOriginalPiece() : Piece *
InvalidMove
SourceOutOfRange
PieceDoesNotExist
MovingEnemyPiece
CapturingAllyPiece
TargetOutOfRange
SamePositionMove
IllegalMove
CheckedMove
Human
+subType() : PlayerType
+playNextMove(source : string, target : string) : void
Computer
-difficulty : int
+playNextMove() : void
+printDecisionMap()() : void
+subType() : PlayerType
-makeDecisionMap() : void
-makeLevel1Map() : void
-makeLevel2Map() : void
-makeLevel3Map() : void
-makeLevel4Map() : void
-loadCheckMovesMap() : void
-loadCaptureMovesMap() : void
-loadAvoidCaptureMovesMap() : void
-loadNormalMovesMap() : void
-loadAvoidCapturePawnMap() : void
-deleteAvoidCapturePawnMap() : void
-getCheckMoves() : std::vector<Position>
TextDisplay
-size : int
-gridDisplay : std::vector<std::vector<TextDisplayCell>>
GraphicsDisplay
-size : int
-cellSize : int
-gridDisplay : std::vector<std::vector<GraphicsDisplayCell>>
-w : Xwindow
+refresh() : void
ScoreBoard
-pWscore : int
-pBscore : int
+incrScore(color : PieceColor, score : float) : void
+printScore() : void
GraphicsDisplayCell
-_char : char
-cellSize : int
-w : Xwindow*
-row : int
-col : int
-lastState : char
-cell : Cell*
+refresh() : void
+setCell(c : Cell *) : void
+setPosition(r : int, c : int) : void
+getXOffset() : int
+getYOffset() : int
+getColor() : CellColor
+drawCell() : void
+drawState() : void
+drawPixel(v : int, h : int, color : int) : void
+drawQueen(color : int) : void
+drawKing(color : int)
+drawKnight(color : int) : void
+drawPawn(color : int) : void
+drawRook(colot : int) : void
+drawBishop(color : int) : void
+getLastState() : char
Tracker
-_stack : std::vector<MoveHistory>
+push(h : MoveHistory) : void
+pop() : MoveHistory
+length() : int
Xwindow
-d : Display*
-w : Window*
-s : int
-gc : GC
-colours[10] : unsigned long
+fillRectangle(x : int, y : int, width : int, height : int, colours : int = Black) : void
+drawString(x : int, y : int, msg : string) : void
0..1
0..1
0..1
*
1..3
1
2
1
*
*
*
*
*
1
1
1
1
*