

<div>DamModel</div> <div> Tile[][]: board ArrayList<Piece>: pieces int: scorePlayer0, scorePlayer1 </div> <div> -void setup() -void setupBoard() -void setupPieces() void placePieces(Piece piece) void removePiece(Piece piece) void removeAllPieces() </div>	<div>ModelData implements Serializable</div> <div> -int: scorePlayer0, scorePlayer1 -int: tileAmount -ArrayList<PieceData>: piecesData </div> <div> +void setSetupMFK(boolean mfk) + boolean getSetupCP() +int getScorePlayer0() +void setScorePlayer0(int score) .. + void savePiece(Piece piece) +ArrayList<Piece> loadPieces() </div>	<div>ModelSettings</div> <div> +boolean: mandatoryFirstKill +boolean: allCrownedPieces +boolean: simpleGame +boolean: vsComputer ... </div> <div> loadSettings() </div>
<div>Piece</div> <div> Player: player -boolean: crowned -boolean: canBeKilledCombo -boolean: beingKilledInCombo, canKill int: xBeforeDragged, yBeforeDragged -Color: crownedColor, attackColor </div> <div> void move() boolean getCanBeeKilledCombo() void setCanBeKilledCombo(boolean ..) boolean getCanBeKilledCombo() .. boolean getCrowned() void setCrowned(boolean crowned) </div>	<div>PieceData implements Serializable</div> <div> int: x, y Player: player </div> <div></div>	<div>Tile</div> <div> enum Type: light, dark Paint: originalColor, highlightColor boolean: highlighted,canBeUsedIn-Combo, isUsedInCombo, int: modelX, modelY </div> <div> void setPiece(Piece piece) Piece getPiece() boolean getCanBeUsedInCombo() void setCanBeUsedInCombo(boolean ..) .. boolean getHighlighted() void setHighlighted(boolean getHigh-lighted) </div>