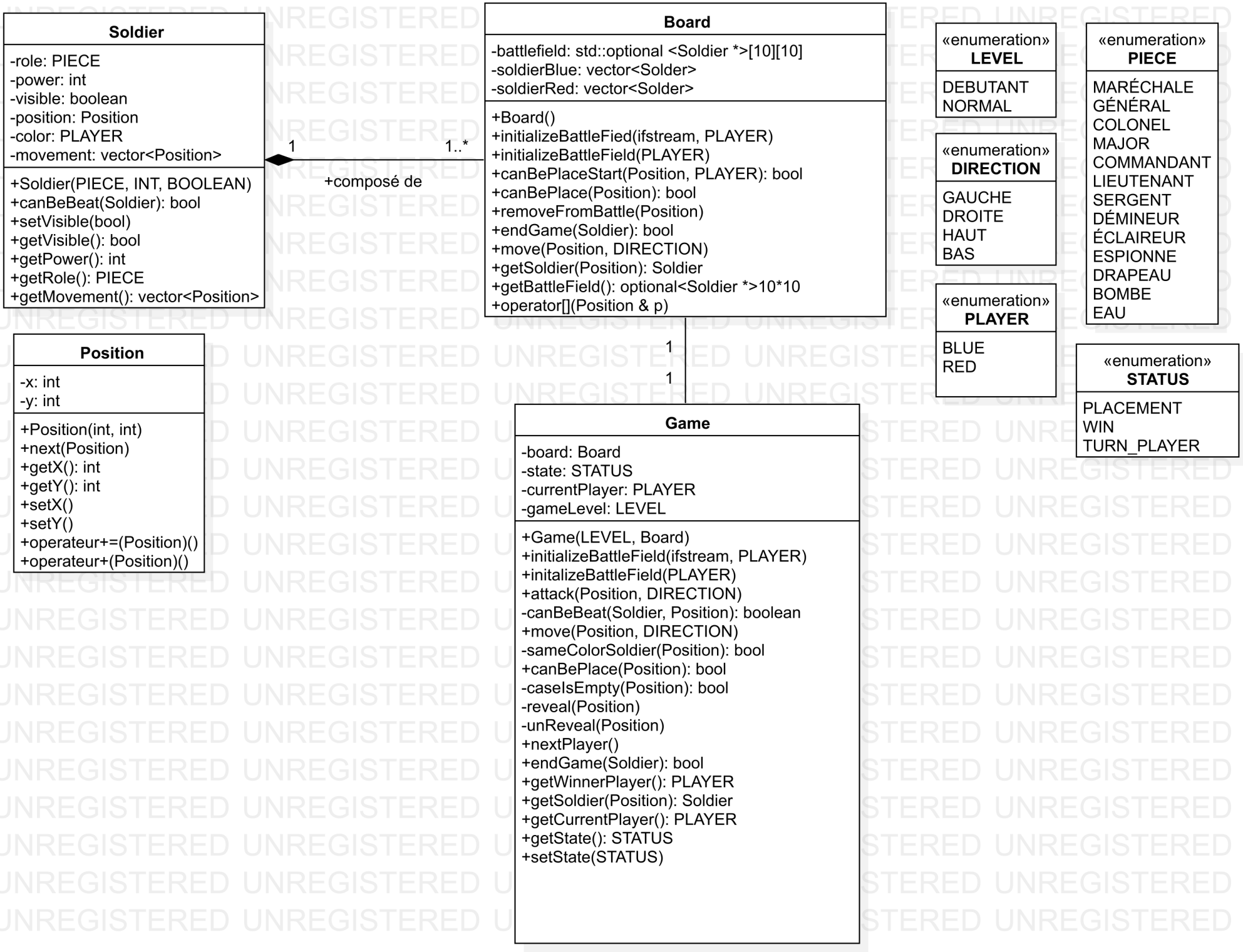
Soldier
-role: PIECE
-power: int
-visible: boolean
-position: Position
-color: PLAYER
-movement: vector<Position>
+Soldier(PIECE, INT, BOOLEAN)
+canBeBeat(Soldier): bool
+setVisible(bool)
+getVisible(): bool
+getPower(): int
+getRole(): PIECE
+getMovement(): vector<Position>
1
1..*
+composé de
Board
-battlefield: std::optional <Soldier *>[10][10]
-soldierBlue: vector<Solder>
-soldierRed: vector<Solder>
+Board()
+initializeBattleFied(ifstream, PLAYER)
+initializeBattleField(PLAYER)
+canBePlaceStart(Position, PLAYER): bool
+canBePlace(Position): bool
+removeFromBattle(Position)
+endGame(Soldier): bool
+move(Position, DIRECTION)
+getSoldier(Position): Soldier
+getBattleField(): optional<Soldier *>10*10
+operator[](Position & p)
«enumeration»
LEVEL
DEBUTANT
NORMAL
«enumeration»
DIRECTION
GAUCHE
DROITE
HAUT
BAS
«enumeration»
PLAYER
BLUE
RED
«enumeration»
PIECE
MARÉCHALE
GÉNÉRAL
COLONEL
MAJOR
COMMANDANT
LIEUTENANT
SERGENT
DÉMINEUR
ÉCLAIREUR
ESPIONNE
DRAPEAU
BOMBE
EAU
«enumeration»
STATUS
PLACEMENT
WIN
TURN_PLAYER
Position
-x: int
-y: int
+Position(int, int)
+next(Position)
+getX(): int
+getY(): int
+setX()
+setY()
+operateur+=(Position)()
+operateur+(Position)()
1
1
Game
-board: Board
-state: STATUS
-currentPlayer: PLAYER
-gameLevel: LEVEL
+Game(LEVEL, Board)
+initializeBattleField(ifstream, PLAYER)
+initalizeBattleField(PLAYER)
+attack(Position, DIRECTION)
-canBeBeat(Soldier, Position): boolean
+move(Position, DIRECTION)
-sameColorSoldier(Position): bool
+canBePlace(Position): bool
-caseIsEmpty(Position): bool
-reveal(Position)
-unReveal(Position)
+nextPlayer()
+endGame(Soldier): bool
+getWinnerPlayer(): PLAYER
+getSoldier(Position): Soldier
+getCurrentPlayer(): PLAYER
+getState(): STATUS
+setState(STATUS)