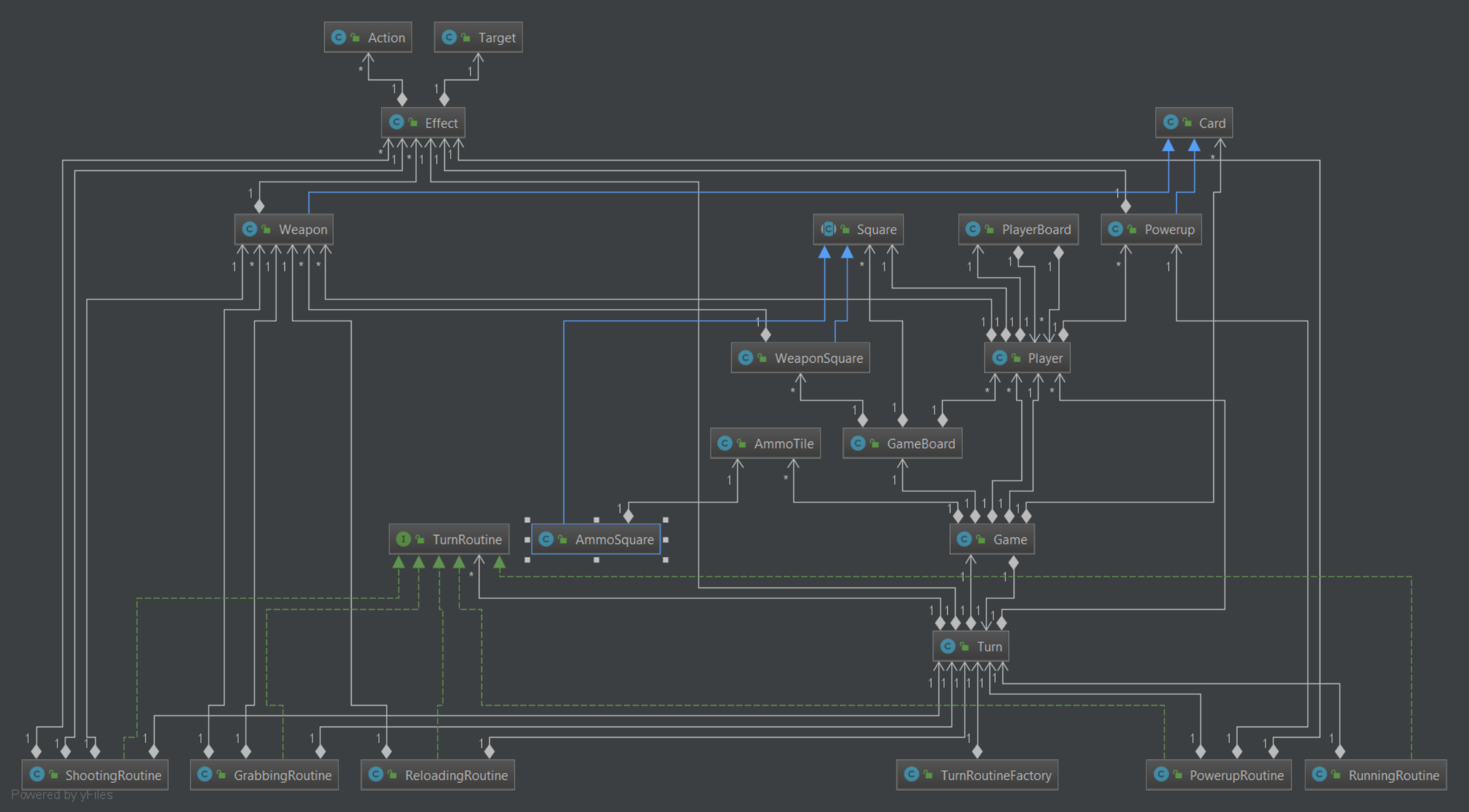
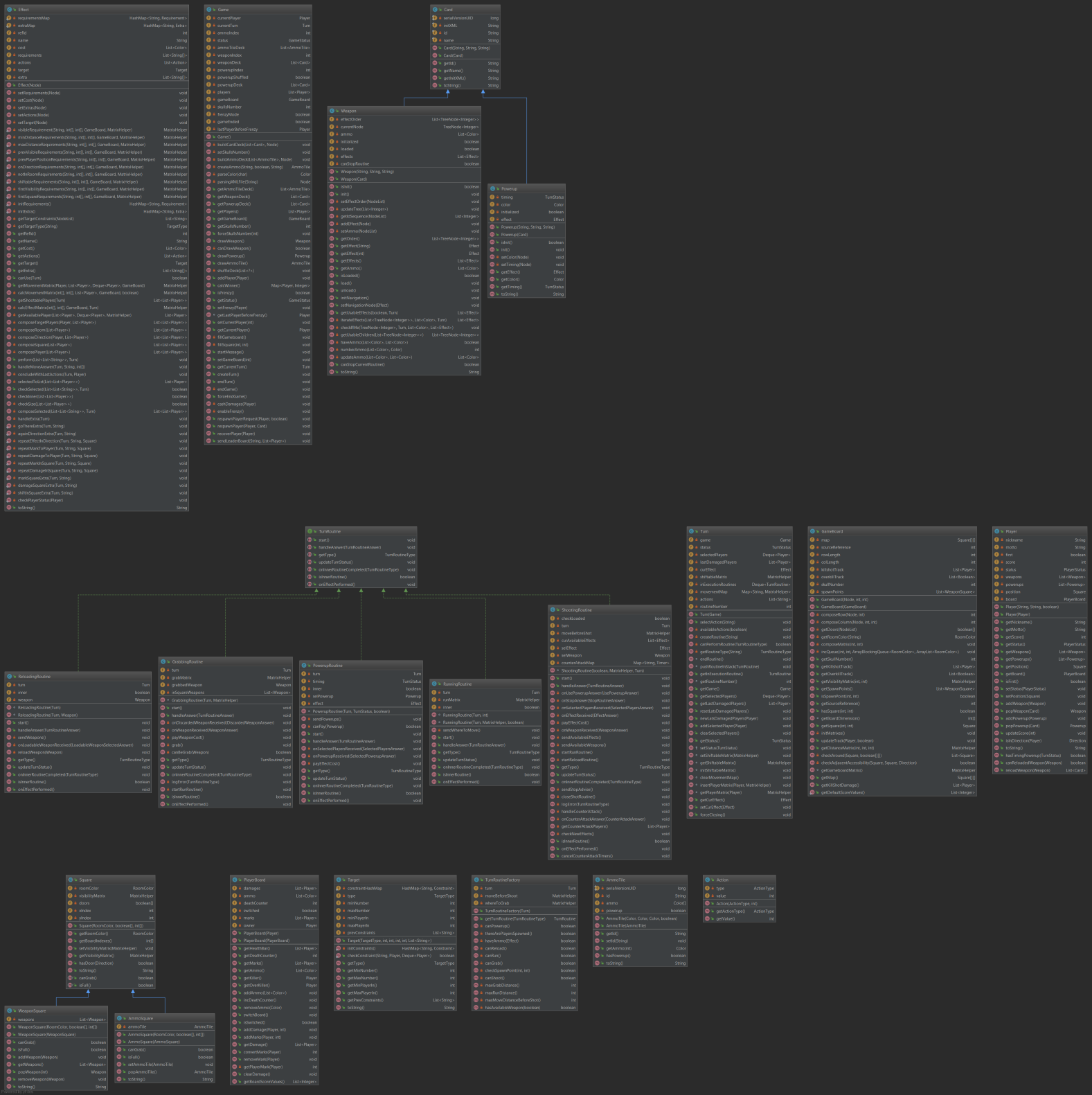

Action
Target
Effect
Card
Weapon
Square
PlayerBoard
Powerup
WeaponSquare
Player
AmmoTile
GameBoard
TurnRoutine
AmmoSquare
Game
Turn
ShootingRoutine
GrabbingRoutine
ReloadingRoutine
TurnRoutineFactory
PowerupRoutine
RunningRoutine
Powered by yFiles


Effect
requirementMap HashMap<String, Requirement>
extraMap HashMap<String, Extra>
refId int
name String
cost List<Color>
requirements List<String>[]
actions List<Action>
target Target
extra List<String>[]
Effect(Node)
setRequirements(Node) void
setCost(Node) void
setExtras(Node) void
setActions(Node) void
setTarget(Node) void
visibleRequirement(String, int[], int[], GameBoard, MatrixHelper) MatrixHelper
minDistanceRequirements(String, int[], int[], GameBoard, MatrixHelper) MatrixHelper
maxDistanceRequirements(String, int[], int[], GameBoard, MatrixHelper) MatrixHelper
prevVisibleRequirements(String, int[], int[], GameBoard, MatrixHelper) MatrixHelper
prevPlayerPositionRequirements(String, int[], int[], GameBoard, MatrixHelper) MatrixHelper
onDirectionRequirements(String, int[], int[], GameBoard, MatrixHelper) MatrixHelper
notInRoomRequirements(String, int[], int[], GameBoard, MatrixHelper) MatrixHelper
shiftpieceRequirements(String, int[], int[], GameBoard, MatrixHelper) MatrixHelper
firstVisibleRequirements(String, int[], int[], GameBoard, MatrixHelper) MatrixHelper
firstSquareRequirements(String, int[], int[], GameBoard, MatrixHelper) MatrixHelper
initRequirements() HashMap<String, Requirement>
initExtra() HashMap<String, Extra>
getPropertyConstraints(NodeList) List<String>
getTargetType(String) TargetType
getRefId() int
getName() String
getCost() List<Color>
getActions() List<Action>
getTarget() Target
getExtra() List<String>[]
canUseTurn() boolean
getMovementMatrix(Player, List<Player>, Deque<Player>, GameBoard) MatrixHelper
calcMovementMatrix(int[], int[], List<Player>, GameBoard, boolean) MatrixHelper
getShootablePlayers(Turn) List<List<Player>>
calcEffectMatrix(int[], int[], GameBoard, Turn) MatrixHelper
getAvailablePlayer(List<Player>, Deque<Player>, MatrixHelper) List<Player>
composeTargetPlayers(Player, List<Player>) List<List<Player>>
composeRoom(List<Player>) List<List<Player>>
composeDirection(Player, List<Player>) List<List<Player>>
composeSquare(List<Player>) List<List<Player>>
composePlayer(List<Player>) List<List<Player>>
perform(List<List<String>>, Turn) void
handleMoveAnswer(Turn, String, int[]) void
conclude(WhiteActions(Turn, Player)) void
selectedList(List<List<Player>>) List<Player>
checkSelected(List<List<String>>, Turn) boolean
checkInner(List<List<Player>>) boolean
checkSize(List<List<Player>>) boolean
composeSelected(List<List<String>>, Turn) List<List<Player>>
handleExtra(Turn) void
goThereExtra(Turn, String) void
againDirectionExtra(Turn, String) void
repeatEffectInDirection(Turn, String, Square) void
repeatMarkToPlayer(Turn, String, Square) void
repeatDamageToPlayer(Turn, String, Square) void
repeatMarkInSquare(Turn, String, Square) void
repeatDamageInSquare(Turn, String, Square) void
markSquareExtra(Turn, String) void
damageSquareExtra(Turn, String) void
shiftInSquareExtra(Turn, String) void
checkPlayerStatus(Player) void
toString() String
Game
currentPlayer Player
currentTurn Turn
ammoIndex int
status GameStatus
ammoTileDeck List<AmmoTile>
weaponIndex int
weaponDeck List<Card>
powerupIndex int
powerupShuffled boolean
powerupDeck List<Card>
players List<Player>
gameBoard GameBoard
skullsNumber int
frenzyMode boolean
gameBoard boolean
lastPlayerBeforeFrenzy Player
Game()
buildCardDeck(List<Card>, Node) void
getSkullsNumber() int
buildAmmoDeck(List<AmmoTile>, Node) void
createAmmo(String, boolean, String) AmmoTile
parseColor(char) Color
parsingXMLFile(String) Node
getAmmoTileDeck() List<AmmoTile>
getWeaponDeck() List<Card>
getPowerupDeck() List<Card>
getPlayers() List<Player>
getGameBoard() GameBoard
getSkullsNumber() int
forceSkullsNumber(int) void
drawWeapon() Weapon
canDrawWeapon() boolean
drawPowerup() Powerup
drawAmmoTile() AmmoTile
shuffleDeck(List<?>) void
addPlayer(Player) void
calcWinner() Map<Player, Integer>
isFrenzy() boolean
getStatus() GameStatus
setFrenzyPlayer() void
getLastPlayerBeforeFrenzy() Player
setCurrentPlayer(int) void
getCurrentPlayer() Player
fillGameboard() void
fillSquare(int, int) void
startMessage() void
setGameBoard(int) void
getCurrentTurn() Turn
startTurn() void
endTurn() void
endGame() void
forceEndGame() void
calcDamages(Player) void
enableFrenzy() void
respawnPlayerRequest(Player, boolean) void
respawnPlayer(Player, Card) void
recoverPlayer(Player) void
sendLeaderBoard(String, List<Player>) void
Card
serialVersionUID long
assXML String
id String
name String
Card(String, String, String)
Card(Card)
getId() String
getName() String
getIntXML() String
toString() String
Weapon
effectOrder List<TreeNode<Integer>>
currentNode TreeNode<Integer>
ammo List<Color>
initialized boolean
loaded boolean
effects List<Effect>
canStopRoutine boolean
Weapon(String, String, String)
Weapon(Card)
isInit() boolean
init() void
setEffectOrder(NodeList) void
updateTree(List<Integer>) void
getSkipSequence(NodeList) List<Integer>
addEffect(Node) void
setAmmo(NodeList) void
getOrder() List<TreeNode<Integer>>
getEffect(String) Effect
getEffect(int) Effect
getEffects() List<Effect>
getAmmo() List<Color>
isLoaded() boolean
load() void
unload() void
reRepartition() void
setNavigationNodeEffect() void
getUsableEffects(boolean, Turn) List<Effect>
iterateEffects(List<TreeNode<Integer>>, List<Color>, Turn) List<Effect>
checkMax(TreeNode<Integer>, Turn, List<Color>, List<Effect>) void
getUsableChildren(List<TreeNode<Integer>>) List<TreeNode<Integer>>
haveAmmo(List<Color>, List<Color>) boolean
numberAmmo(List<Color>, Color) int
updateAmmo(List<Color>, List<Color>) List<Color>
canStopCurrentRoutine() boolean
toString() String
Powerup
timing TurnStatus
color Color
initialized boolean
effect Effect
Powerup(String, String, String)
Powerup(Card)
isInit() boolean
init() void
setColor(Node) void
setTiming(Node) void
getEffect() Effect
getColor() Color
getTiming() TurnStatus
toString() String
TurnRoutine
start() void
handleAnswer(TurnRoutineAnswer) void
getType() TurnRoutineType
updateTurnStatus() void
onInnerRoutineCompleted(TurnRoutineType) void
isInnerRoutine() boolean
onEffectPerformed() void
ReloadingRoutine
turn Turn
inner boolean
weapon Weapon
ReloadingRoutine(Turn)
ReloadingRoutine(Turn, Weapon)
start() void
handleAnswer(TurnRoutineAnswer) void
sendWeapons() void
onLoadableWeaponReceived(LoadableWeaponSelectedAnswer) void
reloadWeapon(Weapon) void
getType() TurnRoutineType
updateTurnStatus() void
onInnerRoutineCompleted(TurnRoutineType) void
isInnerRoutine() boolean
onEffectPerformed() void
GrabbingRoutine
turn Turn
grabMatrix MatrixHelper
grabbedWeapon Weapon
inSquareWeapons List<Weapon>
GrabbingRoutine(Turn, MatrixHelper)
start() void
handleAnswer(TurnRoutineAnswer) void
onDiscardedWeaponReceived(DiscardedWeaponAnswer) void
onWeaponRecieved(WeaponAnswer) void
payWeaponCost() void
grab() void
canBeGrabWeapon() boolean
getType() TurnRoutineType
updateTurnStatus() void
onInnerRoutineCompleted(TurnRoutineType) void
logError(TurnRoutineType) void
startRunRoutine() void
isInnerRoutine() boolean
onEffectPerformed() void
PowerupRoutine
turn Turn
timing TurnStatus
inner boolean
setPowerup Powerup
effect Effect
PowerupRoutine(Turn, TurnStatus, boolean)
sendPowerups() void
canPayPowerup() boolean
start() void
handleAnswer(TurnRoutineAnswer) void
onSelectedPlayersReceived(SelectedPlayersAnswer) void
onPowerupsRecieved(SelectedPowerupsAnswer) void
payEffectCost() void
getType() TurnRoutineType
updateTurnStatus() void
onInnerRoutineCompleted(TurnRoutineType) void
logError(TurnRoutineType) void
isInnerRoutine() boolean
onEffectPerformed() void
RunningRoutine
turn Turn
runMatrix MatrixHelper
inner boolean
RunningRoutine(Turn, int)
RunningRoutine(Turn, MatrixHelper, boolean)
sendWhereToMove() void
start() void
handleAnswer(TurnRoutineAnswer) void
getType() TurnRoutineType
updateTurnStatus() void
onInnerRoutineCompleted(TurnRoutineType) void
isInnerRoutine() boolean
onEffectPerformed() void
ShootingRoutine
checkLoaded boolean
turn Turn
moveBeforeShot MatrixHelper
curAvailableEffects List<Effect>
selEffect Effect
selWeapon Weapon
counterAttackMap Map<String, Timer>
ShootingRoutine(boolean, MatrixHelper, Turn)
start() void
handleAnswer(TurnRoutineAnswer) void
onUsePowerupAnswer(UsePowerupAnswer) void
onStopAnswer(StopRoutineAnswer) void
onSelectedPlayersReceived(SelectedPlayersAnswer) void
onEffectRecieved(EffectAnswer) void
payEffectCost() void
onWeaponRecieved(WeaponAnswer) void
sendAvailableEffects() void
sendAvailableWeapons() void
startRunRoutine() void
startReloadRoutine() void
getType() TurnRoutineType
updateTurnStatus() void
onInnerRoutineCompleted(TurnRoutineType) void
sendStopAdvise() void
closeShotRoutine() void
logError(TurnRoutineType) void
handleCounterAttack() void
onCounterAttackAnswer(CounterAttackAnswer) void
getCounterAttackPlayers() List<Player>
checkNewEffects() void
isInnerRoutine() boolean
onEffectPerformed() void
cancelCounterAttackTimers() void
Turn
game Game
status TurnStatus
selectedPlayers Deque<Player>
lastDamagedPlayers List<Player>
curEffect Effect
shiftableMatrix MatrixHelper
inExecutionRoutines Deque<TurnRoutine>
movementMap Map<String, MatrixHelper>
actions List<String>
routineNumber int
Turn(Game)
selectActions(String) void
availableActions(boolean) void
crossRoutine(String) void
canPerformRoutine(TurnRoutineType) boolean
getRoutineTypeString() TurnRoutineType
endRoutine() void
pushRoutineInStack(TurnRoutine) void
getExecutionRoutine() TurnRoutine
getRoutineNumber() int
getGame() Game
getSelectedPlayers() Deque<Player>
getLastDamagedPlayers() List<Player>
resetLastDamagedPlayers() void
newLastDamagedPlayers(Player) void
addSelectedPlayer(Player) void
clearSelectedPlayers() void
getStatus() TurnStatus
setStatus(TurnStatus) void
setShiftableMatrix(MatrixHelper) void
getShiftableMatrix() MatrixHelper
initShiftableMatrix() void
clearMovementMaps() void
insertPlayerMatrix(Player, MatrixHelper) void
getPlayerMatrix(Player) MatrixHelper
getCurEffect() Effect
setCurEffect(Effect) void
forceClosing() void
GameBoard
map Square[][]
sourceReference int
rowLength int
colLength int
killshotTrack List<Player>
eventoTrack List<Boolean>
skullNumber int
spawnPoints List<WeaponSquare>
GameBoard(Node, int, int)
GameBoard(GameBoard)
composeRow(Node, int) int
composeColumn(Node, int, int) int
getDoors(NodeList) boolean[]
getRoomColor(String) RoomColor
composeMatrix(int, int) void
incQueue(int, int, ArrayBlockingQueue<RoomColor>, ArrayList<RoomColor>) void
getSkullNumber() int
getKillshotTrack() List<Player>
getEventoTrack() List<Boolean>
getVisibilityMatrix(int, int) MatrixHelper
getSpawnPoints() List<WeaponSquare>
isSpawnPoint(int, int) boolean
getSourceReference() int
hasSquare(int, int) boolean
getBoardDimension() int[]
getSquare(int, int) Square
initMatrices() void
updateTrack(Player, boolean) void
getDistanceMatrix(int, int, int) MatrixHelper
checkAround(Square, boolean[][]) List<Square>
checkAdjacentAccessibility(Square, Square, Direction) boolean
getGameboardMatrix() MatrixHelper
getMap() Square[][]
getKillShotDamages() List<Player>
getDefaultScoreValues() List<Integer>
Player
nickname String
motto String
first boolean
score int
status PlayerStatus
weapons List<Weapon>
powerups List<Powerup>
position Square
board PlayerBoard
Player(String, String, boolean)
Player(Player)
getNickname() String
getMotto() String
getScore() int
getStatus() PlayerStatus
getWeapons() List<Weapon>
getPowerups() List<Powerup>
getPosition() Square
getBoard() PlayerBoard
isFirst() boolean
setStatus(PlayerStatus) void
setPosition(Square) void
addWeapon(Weapon) void
popWeapon(Card) Weapon
addPowerup(Powerup) void
popPowerup(Card) Powerup
updateScore(int) void
isInDirection(Player) Direction
toString() String
hasTimingPowerup(TurnStatus) boolean
addReloadedWeapon(Weapon) boolean
reloadWeapon(Weapon) List<Card>
Square
roomColor RoomColor
visibilityMatrix MatrixHelper
doors boolean[]
xIndex int
yIndex int
Square(RoomColor, boolean[], int[])
getRoomColor() RoomColor
getBoardIndexes() int[]
setVisibilityMatrix(MatrixHelper) void
getVisibilityMatrix() MatrixHelper
hasDoor(Direction) boolean
toString() String
canGrab() boolean
isFull() boolean
PlayerBoard
damages List<Player>
ammo List<Color>
deathCounter int
switched boolean
marks List<Player>
owner Player
PlayerBoard(Player)
PlayerBoard(PlayerBoard)
getHeartNbr() int
getDeathCounter() int
getMarks() List<Player>
getAmmo() List<Color>
getKiller() Player
getOverKiller() Player
addAmmo(List<Color>) void
incDeathCounter() void
removeAmmo(Color) void
switchBoard() void
isSwitched() boolean
addDamage(Player, int) void
addMarks(Player, int) void
getDamage() List<Player>
convertMarks(Player) int
removeMark(Player) void
getPlayerMark(Player) int
clearDamage() void
getBoardScoreValues() List<Integer>
Target
constraintHashMap HashMap<String, Constraint>
type TargetType
minNumber int
maxNumber int
minPlayerIn int
maxPlayerIn int
prevConstraints List<String>
Target(TargetType, int, int, int, int, List<String>)
initConstraints() HashMap<String, Constraint>
checkConstraint(String, Player, Deque<Player>) boolean
getType() TargetType
getMinNumber() int
getMaxNumber() int
getMinPlayerIn() int
getMaxPlayerIn() int
getPrevConstraints() List<String>
toString() String
TurnRoutineFactory
turn Turn
moveBeforeShoot MatrixHelper
whereToGrab MatrixHelper
TurnRoutineFactory(Turn)
getTurnRoutine(TurnRoutineType) TurnRoutine
canPowerup() boolean
thereArePlayersSpawned() boolean
haveAmmo(Effect) boolean
canReload() boolean
canRun() boolean
canGrab() boolean
checkSpawnPoint(int, int) boolean
canShoot() boolean
maxGrabDistance() int
maxRunDistance() int
maxMoveDistanceBeforeShot() int
hasAvailableWeapon(boolean) boolean
AmmoTile
serialVersionUID long
id String
ammo Color[]
powerup boolean
AmmoTile(Color, Color, Color, boolean)
AmmoTile(AmmoTile)
getId() String
setId(String) void
getAmmo(int) Color
hasPowerup() boolean
toString() String
Action
type ActionType
value int
Action(ActionType, int)
getActionType() ActionType
getValue() int
WeaponSquare
weapons List<Weapon>
WeaponSquare(RoomColor, boolean[], int[])
WeaponSquare(WeaponSquare)
canGrab() boolean
isFull() boolean
addWeapon(Weapon) void
getWeapons() List<Weapon>
popWeapon(int) Weapon
removeWeapon(Weapon) void
toString() String
AmmoSquare
ammoTile AmmoTile
AmmoSquare(RoomColor, boolean[], int[])
AmmoSquare(AmmoSquare)
canGrab() boolean
isFull() boolean
setAmmoTile(AmmoTile) void
popAmmoTile() AmmoTile
toString() String