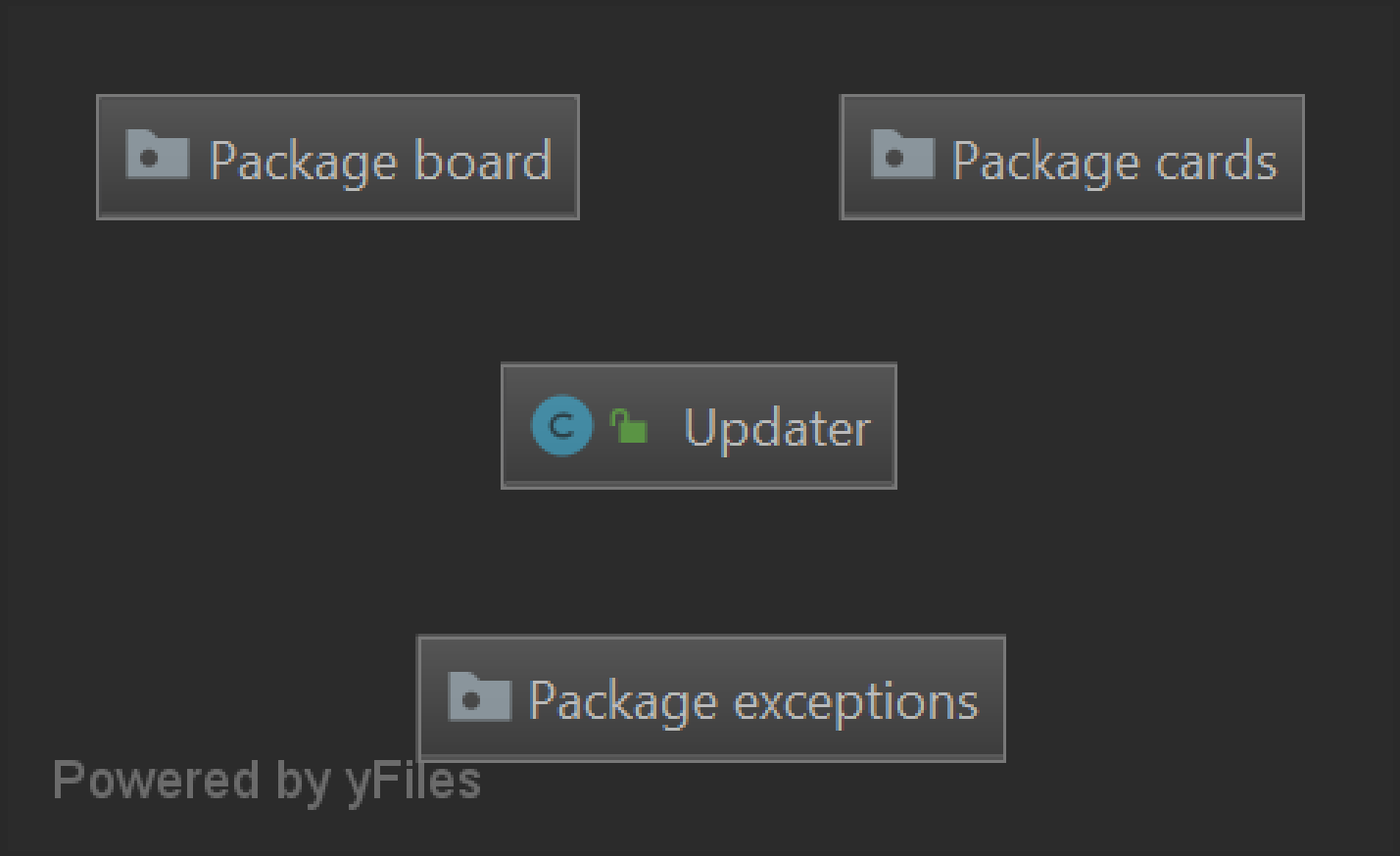
Software Engineering project 2019

Implementation of Adrenaline by Filip Neduk.

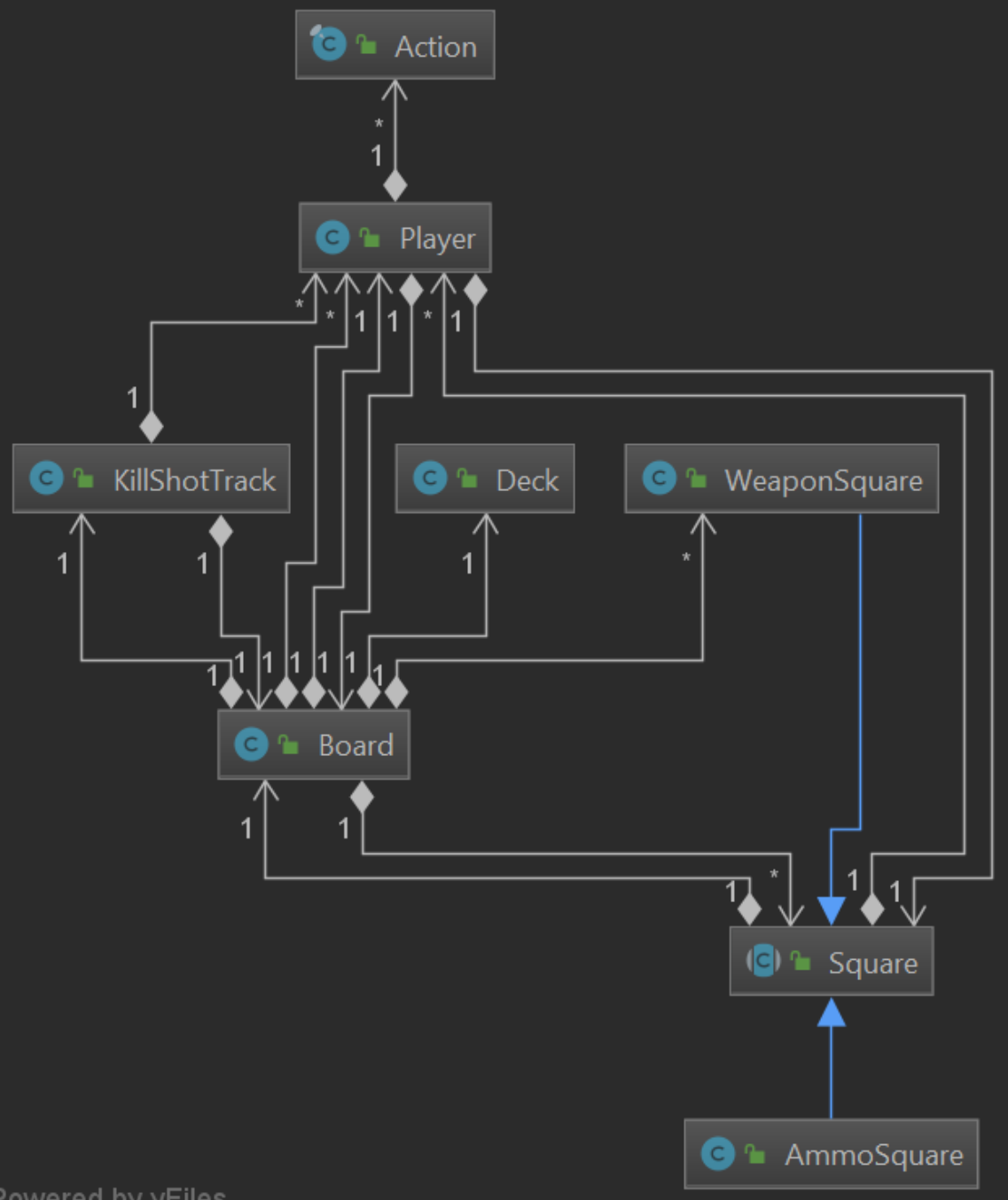
Marco Bagatella, Riccardo Bassani, Davide Aldé.

model


Package board
Package cards
Updater
Package exceptions
Powered by yFiles

model.board

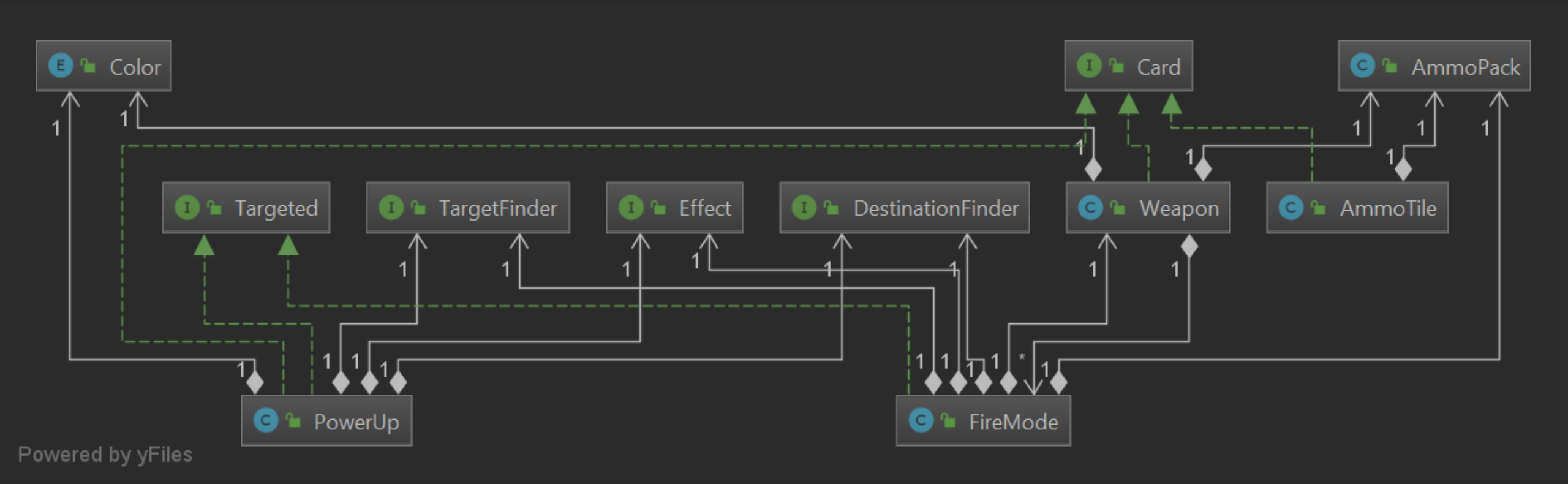
- Contains a class representing the Board and other classes representing the elements of the Board.
- A Board is made of Squares, that can be AmmoSquares or WeaponSquares.
- A Board contains a KillShotTrack, some Decks and the Players of the game.
- Every Player can do some Actions.



model.cards

- Weapon, PowerUp and AmmoTile implement the interface Card.
- An AmmoTile contains an Ammopack. Weapons, PowerUps and FireModes have a cost in AmmoPack.
- FireMode and PowerUp implement the interface Targeted.
- TargetFinder, Effect and DestinationFinder are functional interfaces used by FireMode, PowerUp.
- Cards, as well as other components of the game, can have a Color. Namely, Weapons and PowerUp must have a Color while AmmoTile must not.

model.cards


Color
Card
AmmoPack
Targeted
TargetFinder
Effect
DestinationFinder
Weapon
AmmoTile
PowerUp
FireMode
Powered by yFiles

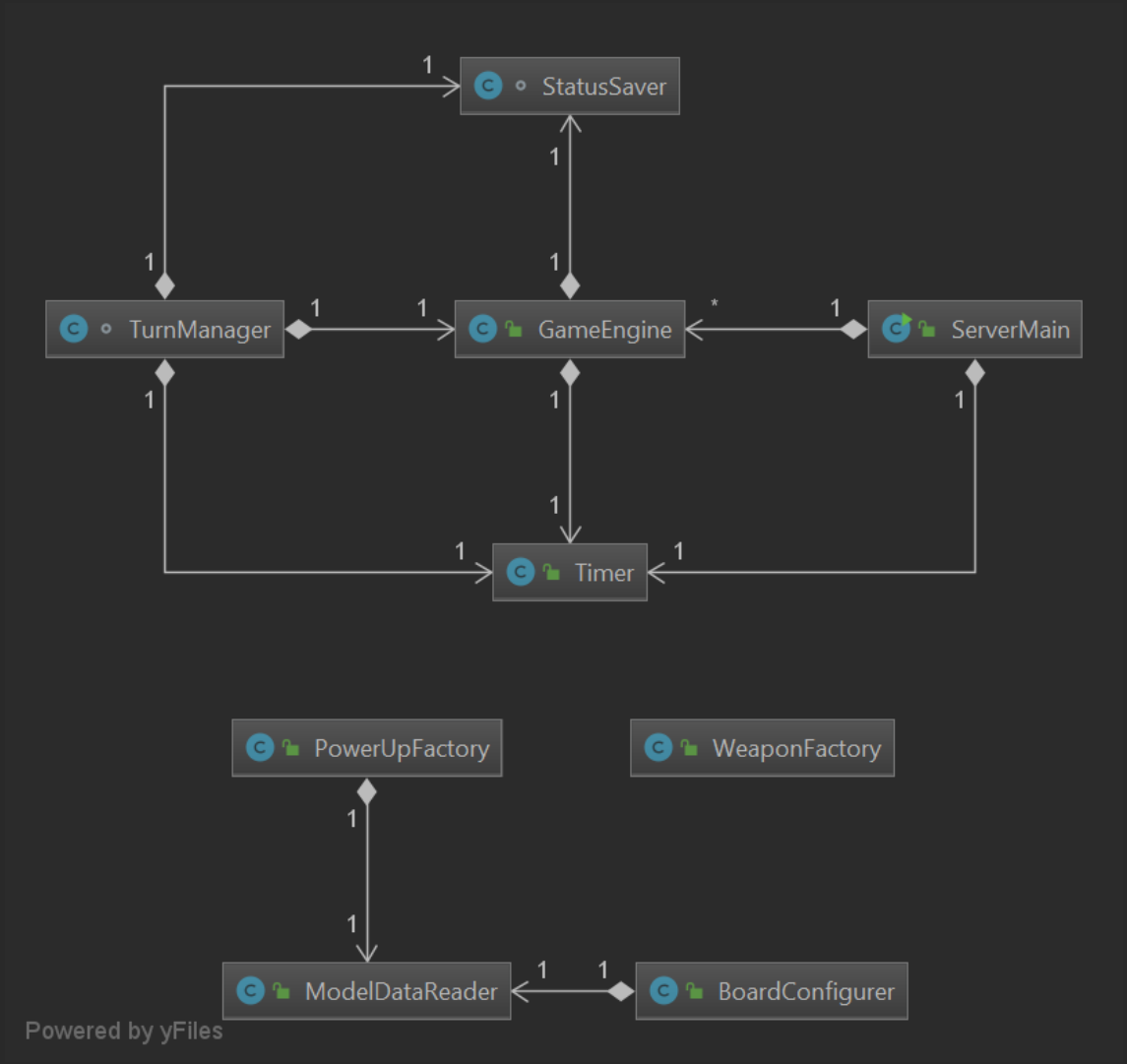
model.Updater

- The class responsible for updating the ClientModel by copying the necessary values from the ServerModel.
- The ClientModel class will be presented later.

Updater	
get(String, Weapon, boolean)	JsonObject
get(String, int)	JsonObject
get(String, int, Player, boolean)	JsonObject
get(String, Player, PowerUp)	JsonObject
get(String, Player, Weapon)	JsonObject
get(String, Player, AmmoPack)	JsonObject
get(String, Player, Square)	JsonObject
get(String, Player)	JsonObject
get(String, Player, int)	JsonObject
get(String, Player, List<Player>)	JsonObject
get(String, Square, Weapon)	JsonObject
get(String, Player, boolean)	JsonObject
get(String, Square)	JsonObject
getModel(Board, Player)	JsonObject
createSimplePlayer(Player, Board)	SimplePlayer
getFreshUpdate(String)	JsonObject
getRenderMessage()	JsonObject
toSimpleWeapon(Weapon)	SimpleWeapon
toSimpleSquare(WeaponSquare)	SimpleSquare
toSimpleSquare(AmmoSquare)	SimpleSquare

Powered by yFiles

controller


StatusSaver
TurnManager
GameEngine
ServerMain
Timer
PowerUpFactory
WeaponFactory
ModelDataReader
BoardConfigurer
Powered by yFiles

controller.ServerMain

- The class containing the main() method of the Server.
- Manages connections and matchmaking.

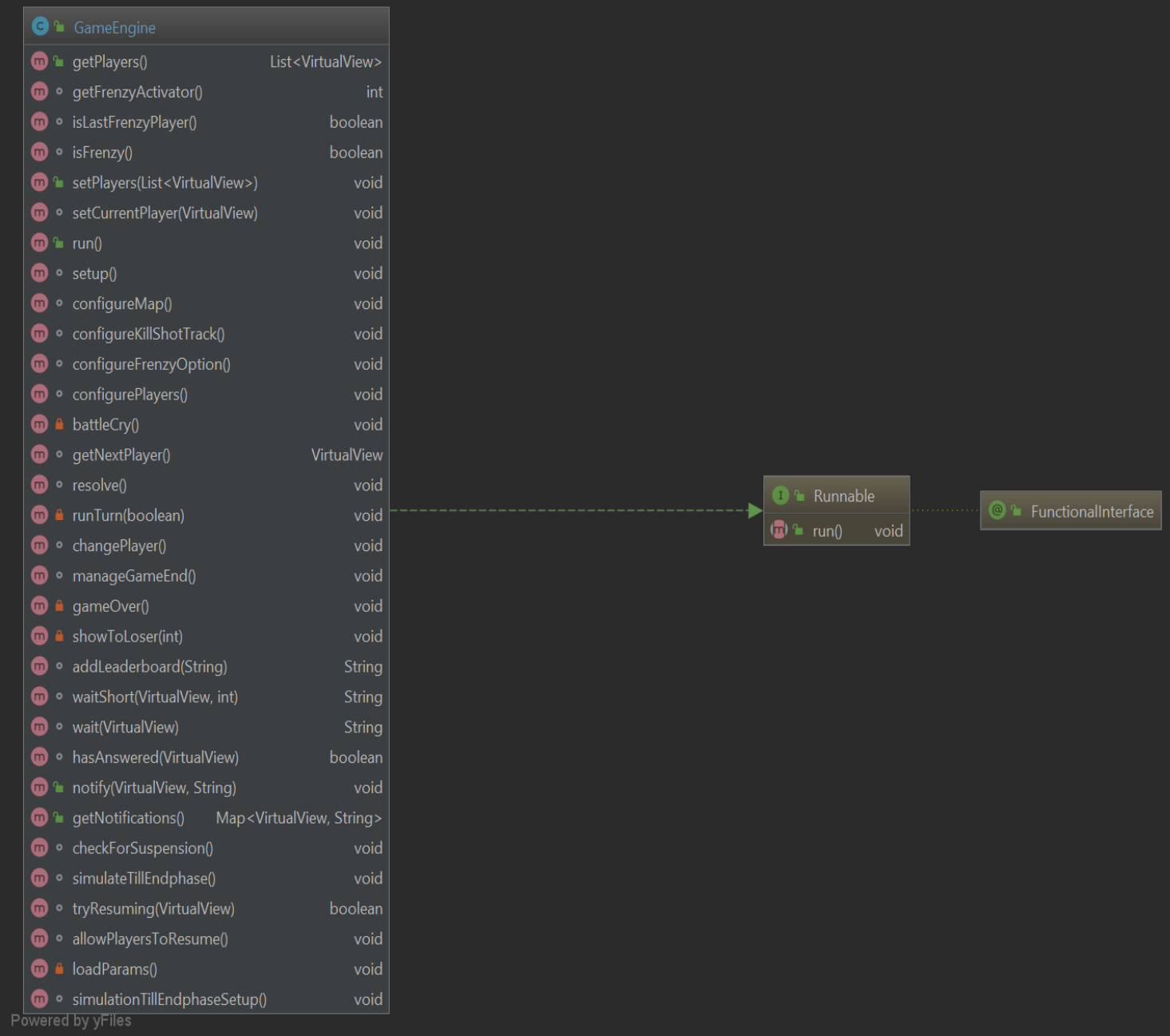
ServerMain

getInstance()	ServerMain
main(String[])	void
setup()	void
untrackGame(GameEngine)	void
addPlayer(VirtualView)	void
login(VirtualView)	boolean
canResume(String)	boolean
resume(VirtualView)	boolean
removeSuspendedPlayers()	void
getPlayers()	List<VirtualView>
initializeLogger()	void
loadConfig()	Properties
manageInput()	void
refreshConnections()	void
matchmaking()	void
getAlreadyConnected()	String

Powered by yFiles

controller.GameEngine

- The class responsible for running a game.
- Is instantiated and run by ServerMain.



controller.TurnManager

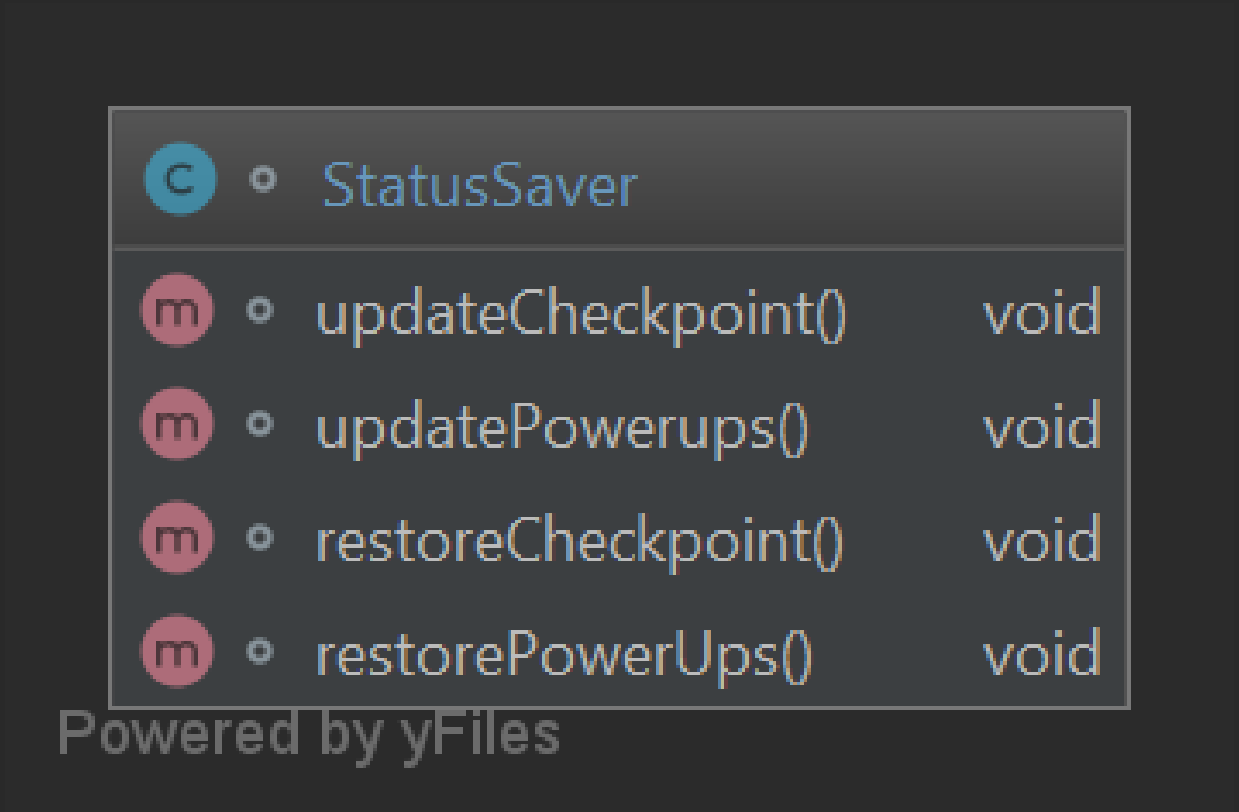
- The class responsible for running a turn.
- Handles the exchange of messages with the user.

TurnManager	
runTurn()	void
joinBoard(Player, int, boolean)	void
executeAction()	boolean
executeActualAction(Action)	void
handleUsingPowerUp()	boolean
usePowerUp()	void
handleMoving(Action)	void
handleCollecting()	void
handleShooting()	void
applyFireMode(FireMode)	void
reload(int)	boolean
reloadMandatory()	void
handleTargetingScope(Player, List<Player>)	boolean
askTargetsForGrenade()	void
handleTagbackGrenade(Player)	boolean
handleDeaths()	void
replaceWeapons()	void
replaceAmmoTiles()	void
getDamagesList()	List<Integer>
updateDead()	void
askConfirmation(String)	boolean
askConfirmation(String, Player)	boolean
resetJoinBoard(Player, boolean)	void
resetPowerUp()	void
resetAction()	void
toStringList(List)	List<String>
toUserStringList(List<List<Player>>)	List<String>
updateAndSendModel()	void
updateAndNotifyAll()	void
restoreAndNotify()	void
getVirtualView(Player)	VirtualView
getDead()	List<Integer>
handlePayment(AmmoPack)	void
mandatoryConversion(Color)	void
setDead(List<Integer>)	void

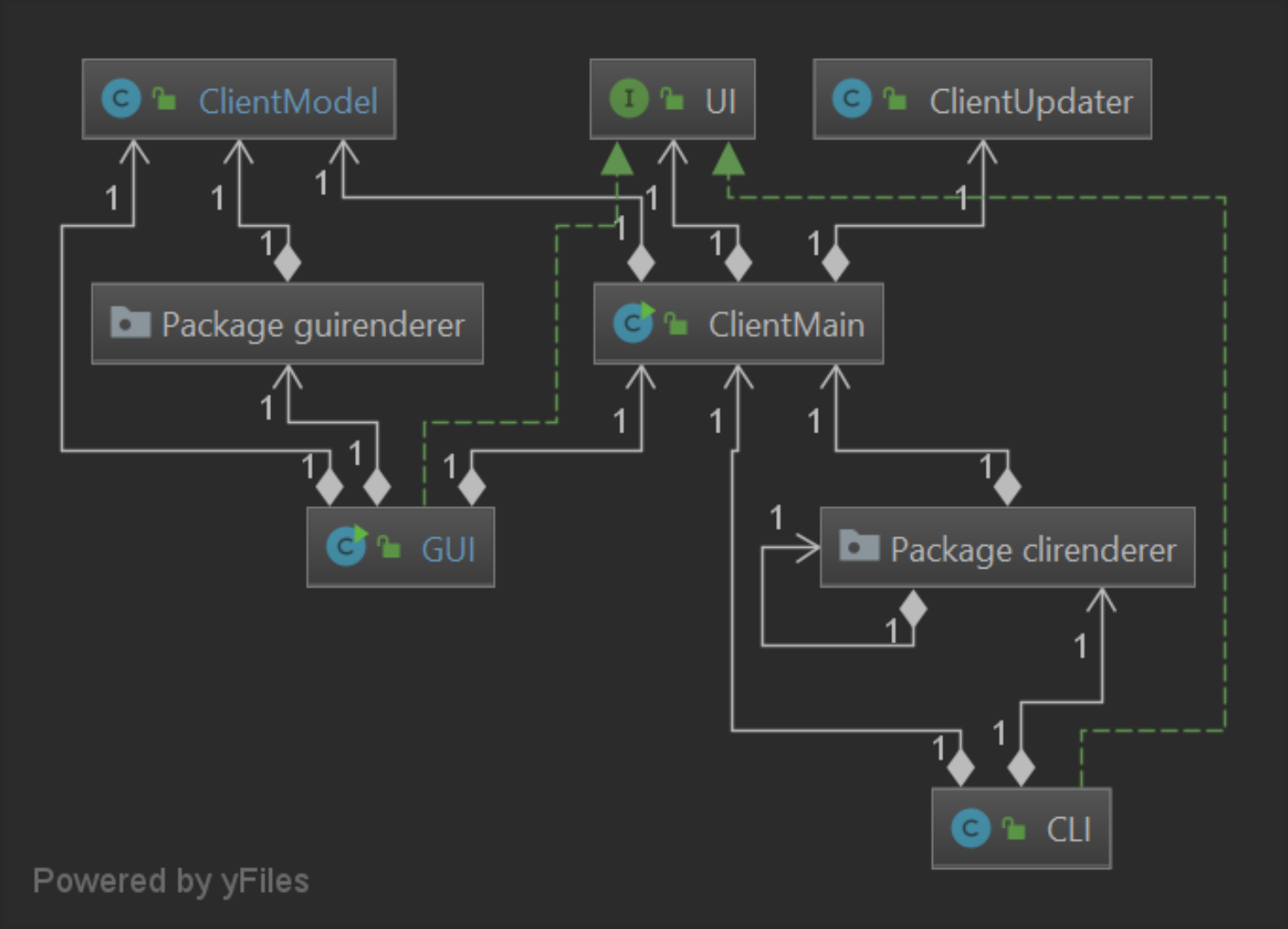
Powered by yFiles

controller.StatusSaver

- Supports the TurnManager allowing it to save checkpoints in order to make possible for the player to reset his actions.



view


ClientModel
UI
ClientUpdater
Package guirenderer
ClientMain
GUI
Package clirenderer
CLI
Powered by yFiles

view.ClientMain

- The class containing the main() method of the Server.

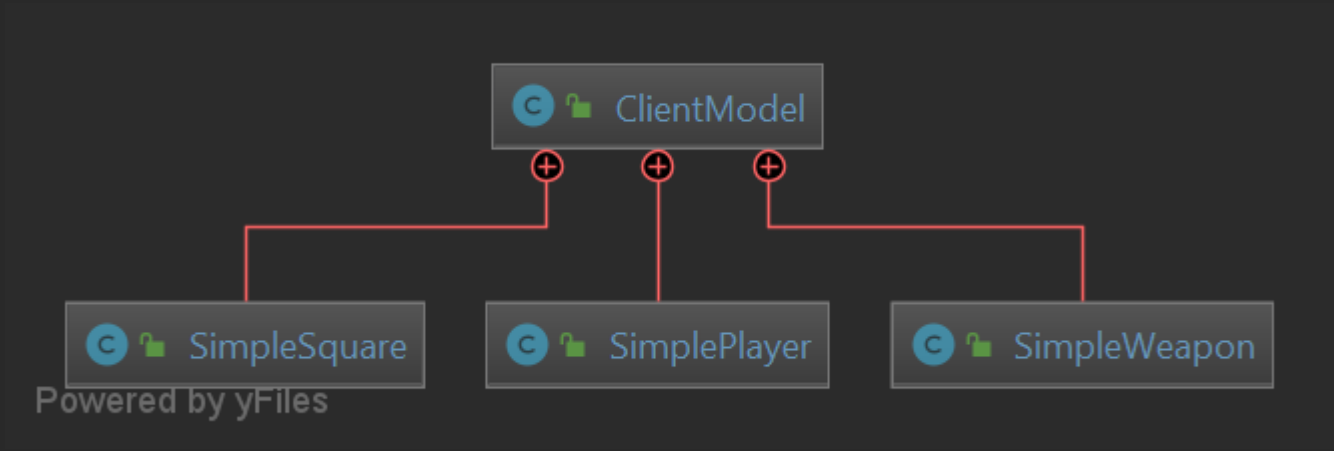
ClientMain

main(String[])	void
initializeLogger()	void
loadConfig(String[])	Properties
setup(String[])	void
choose(String, String, List<String>)	int
display(String)	void
getInput(String, int)	String
update(JsonObject)	void
getClientModel()	ClientModel
setClientModel(ClientModel)	void
showSuspension()	void
showDisconnection()	void
showEnd(String)	void

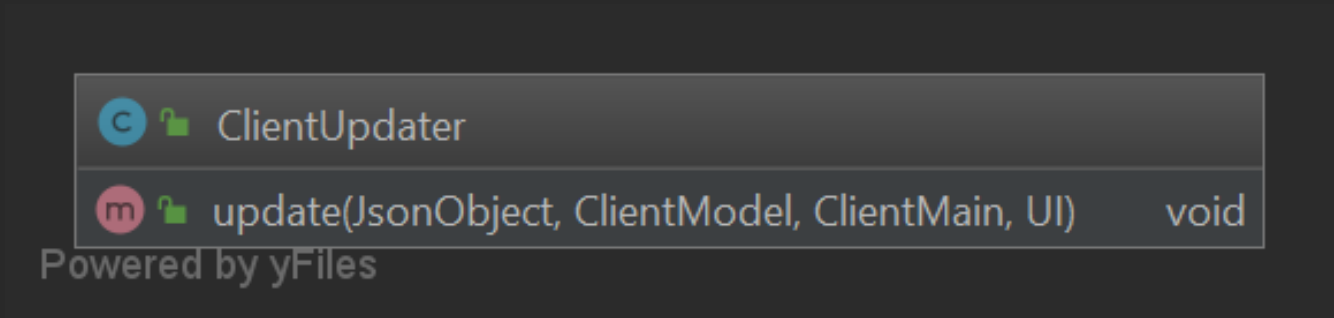
Powered by yFiles

view.ClientModel & view.ClientUpdater

Contains a simplified version of the model.
Contains three inner classes.

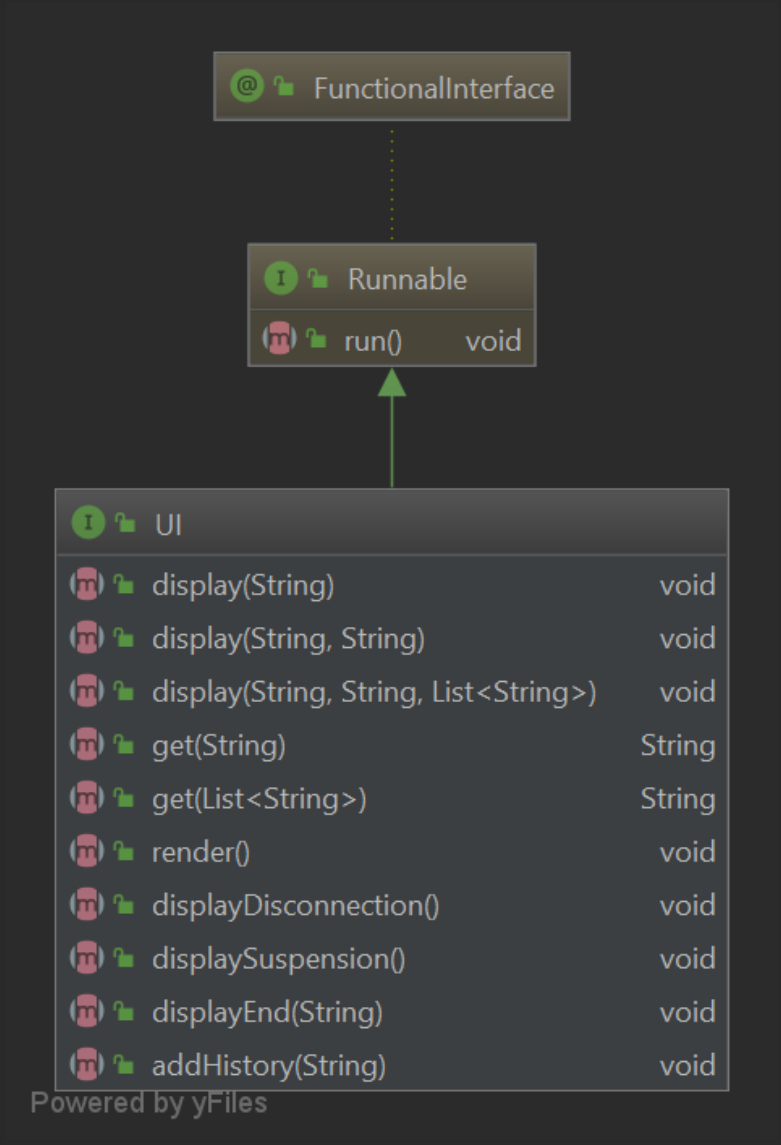


Updates the ClientModel, depending on the type of JsonObject it receives.

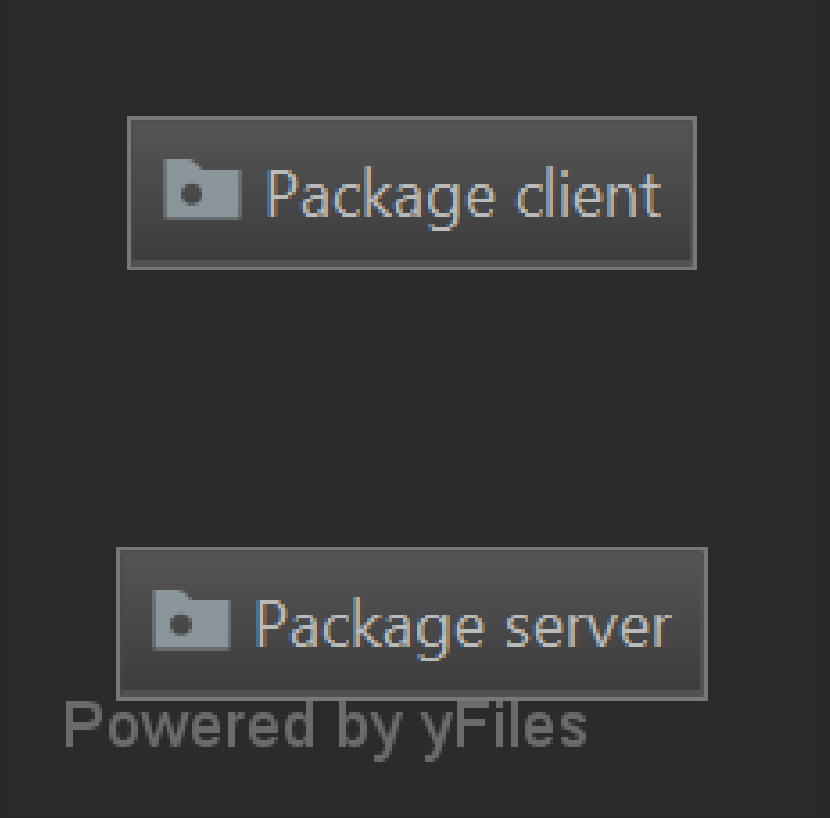


view.UI

- Interface for user interface and for managing client's input/output.
- Is implemented by CLI, GUI.

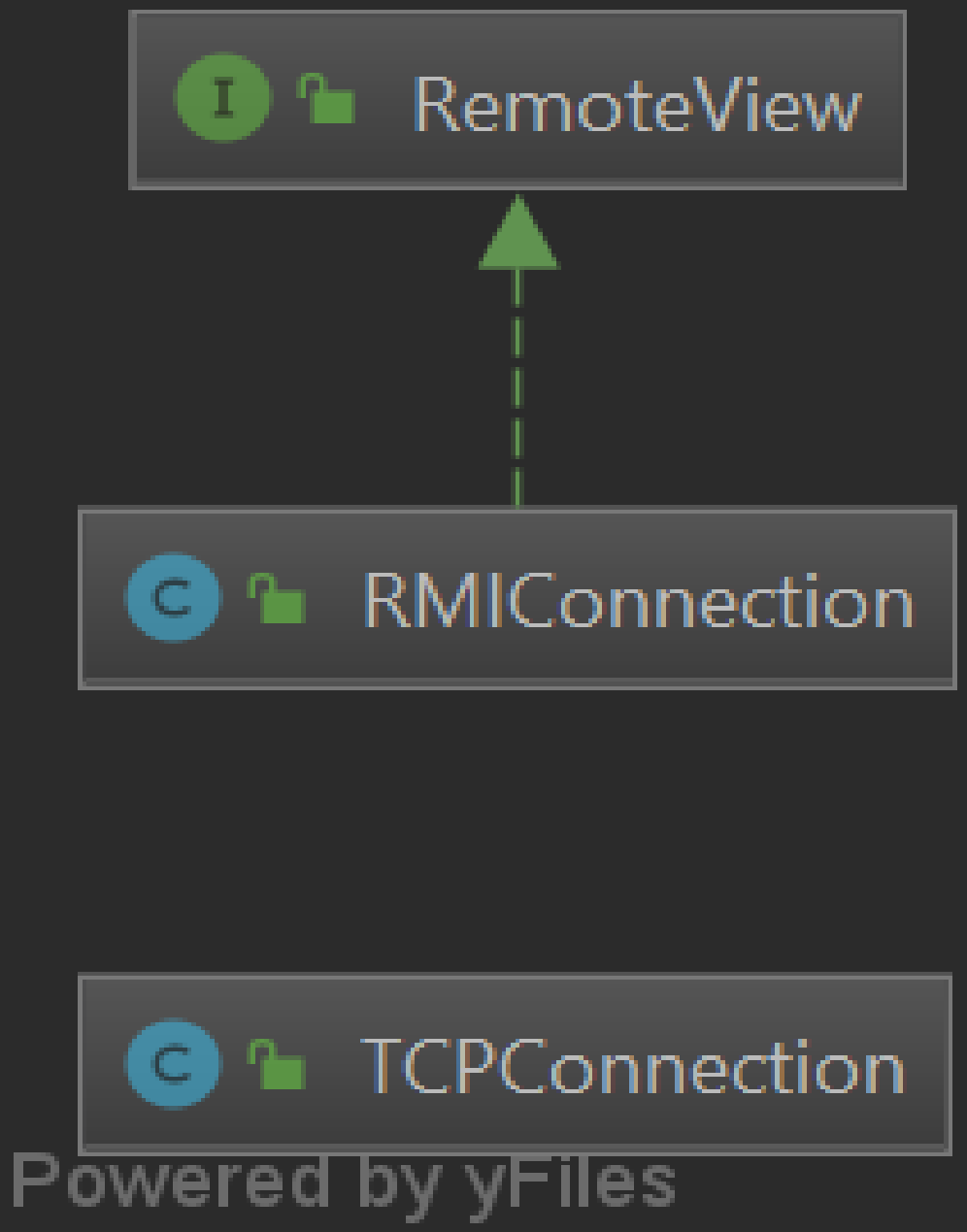


network

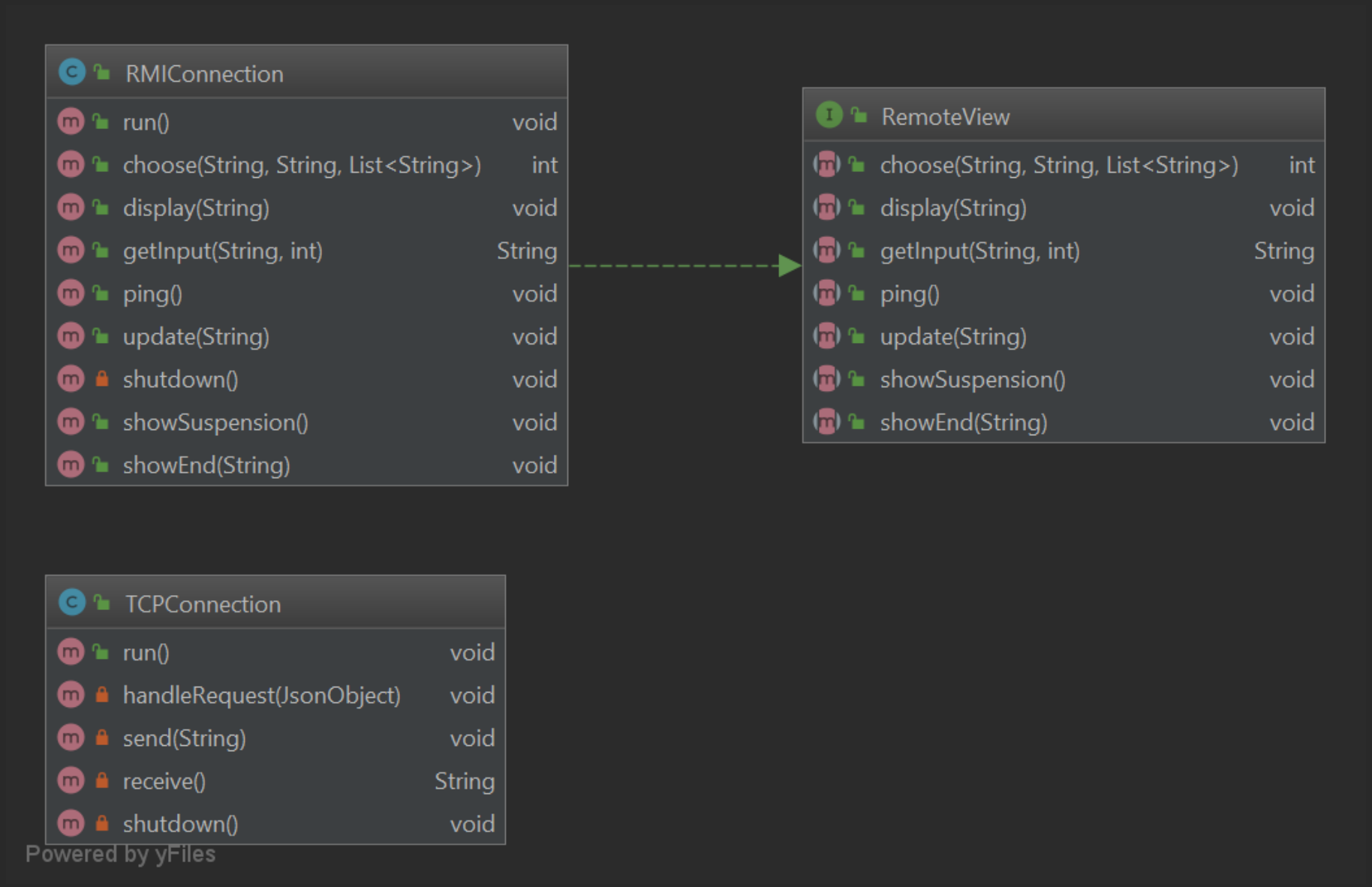


network.client

- Two classes for two different types of connection: RMI and TCP.
- RMIConnection implements the interface RemoteView.

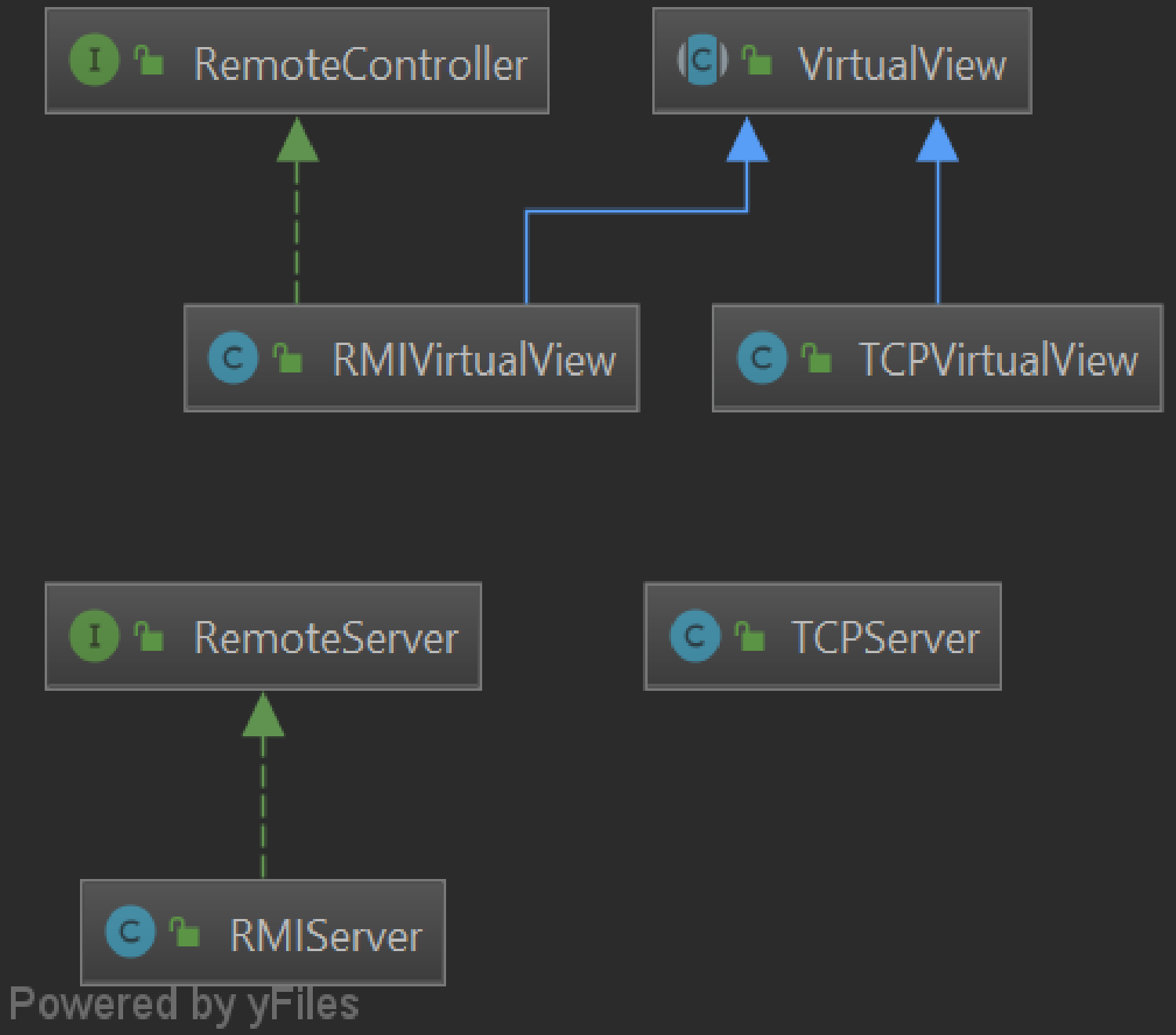


network.client

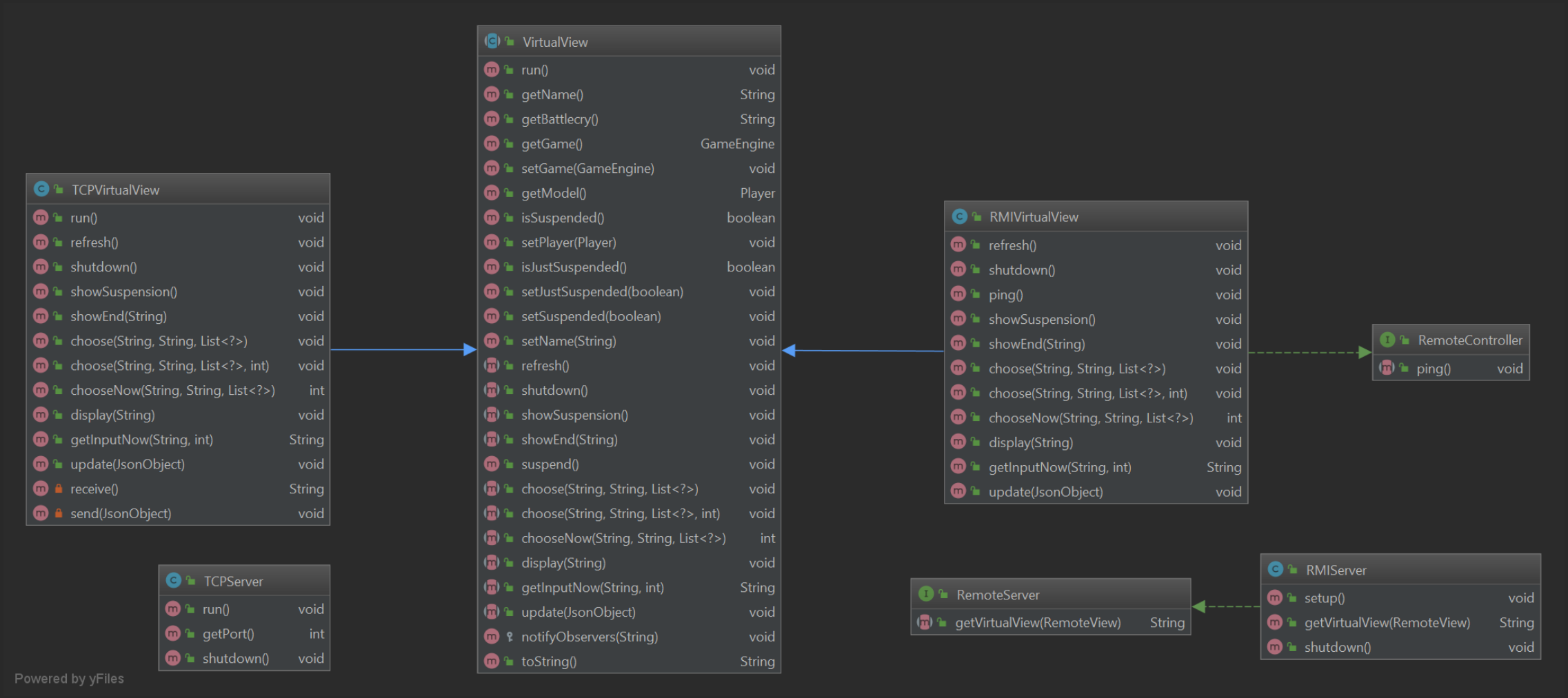


network.server

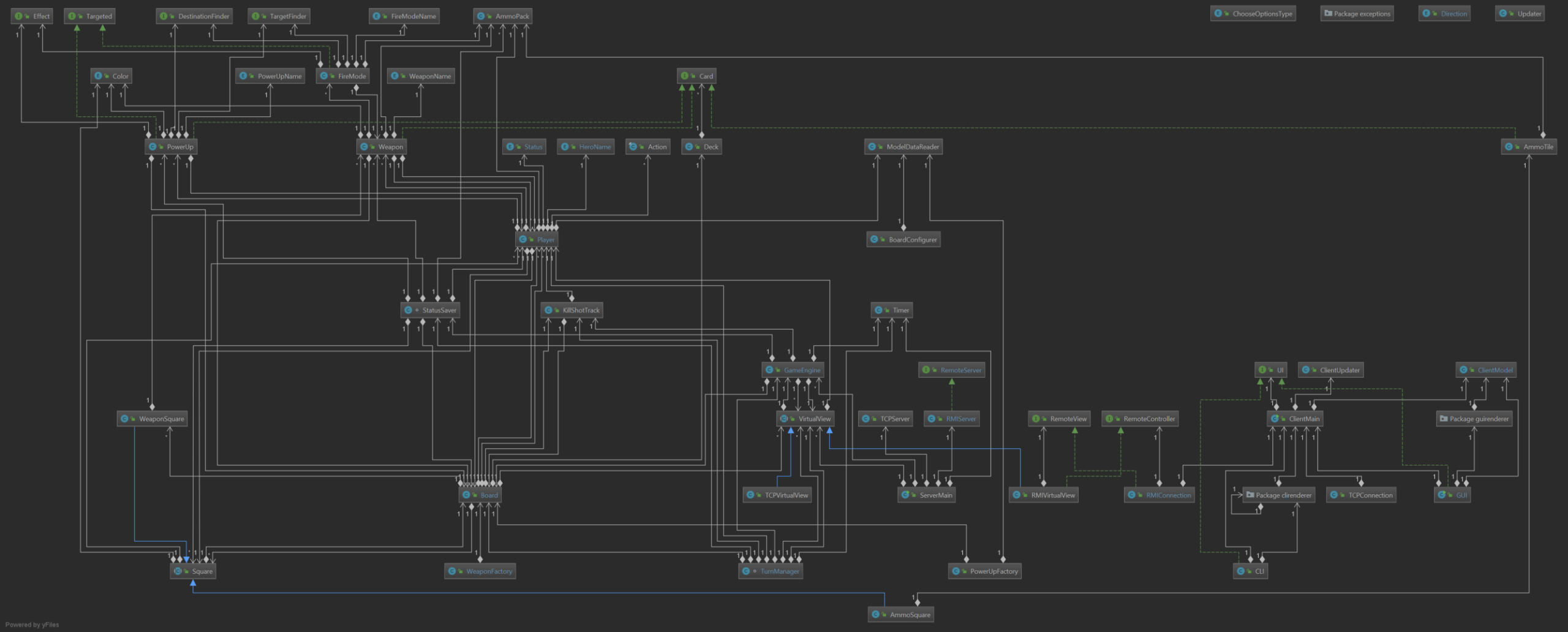
- A VirtualView allows the Server to communicate with the Client.
- Depending on the user choice, for every connection the Server can implement a TCPVirtualView or an RMIVirtualView, taking advantage of a TCPServer class or of an RMIServer class.



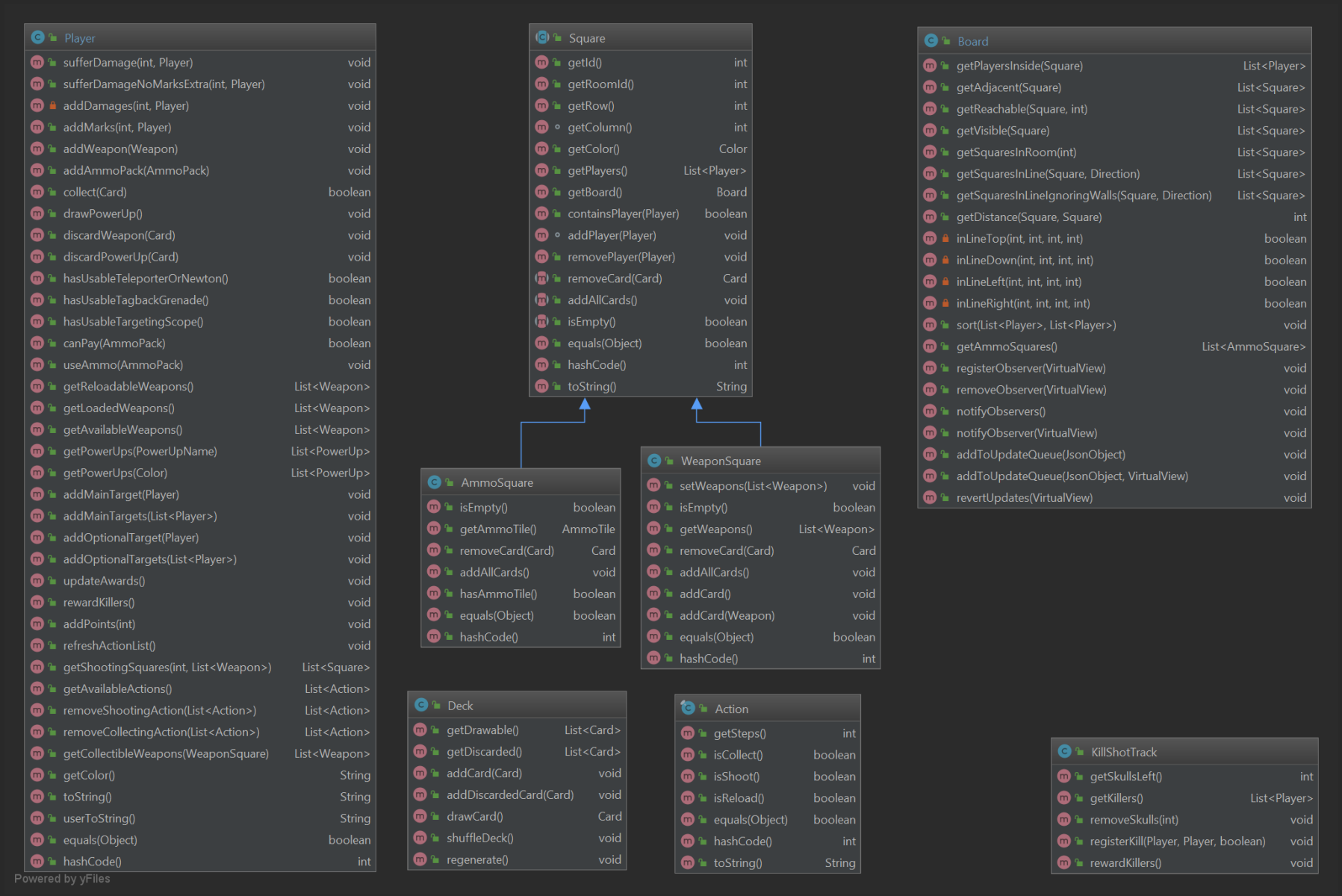
network.server



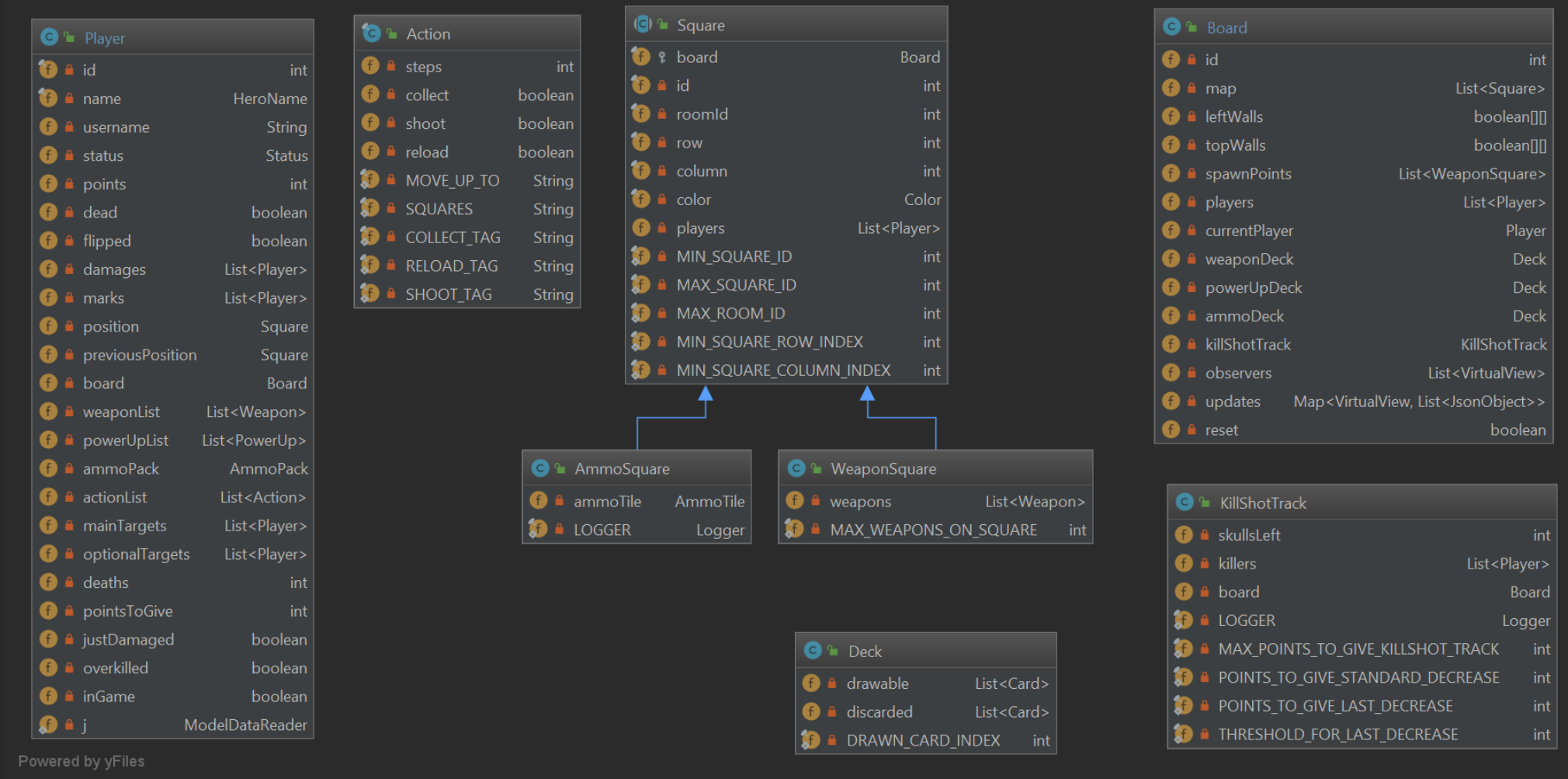
COMPREHENSIVE DIAGRAM



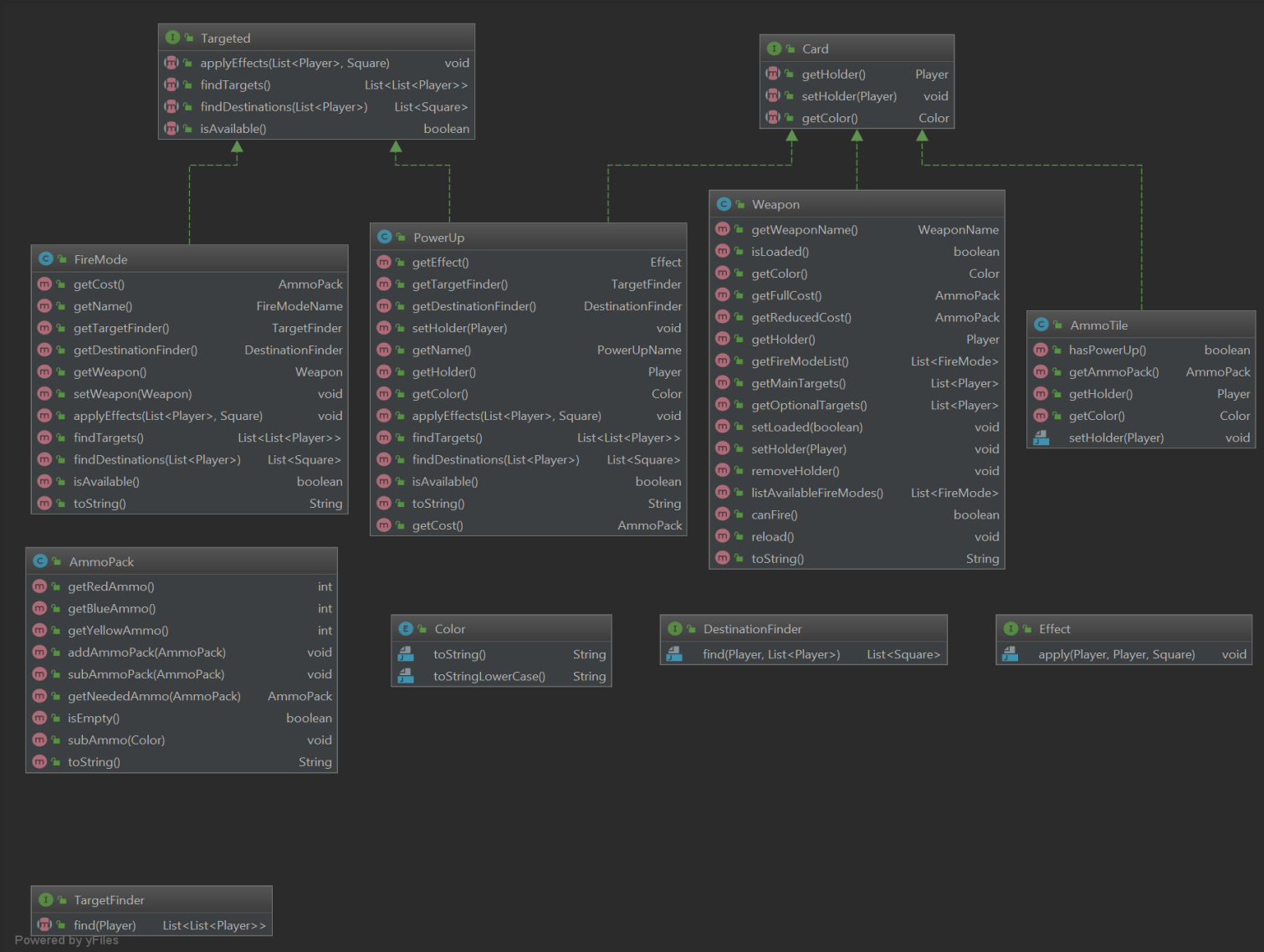
Methods summary: model.board



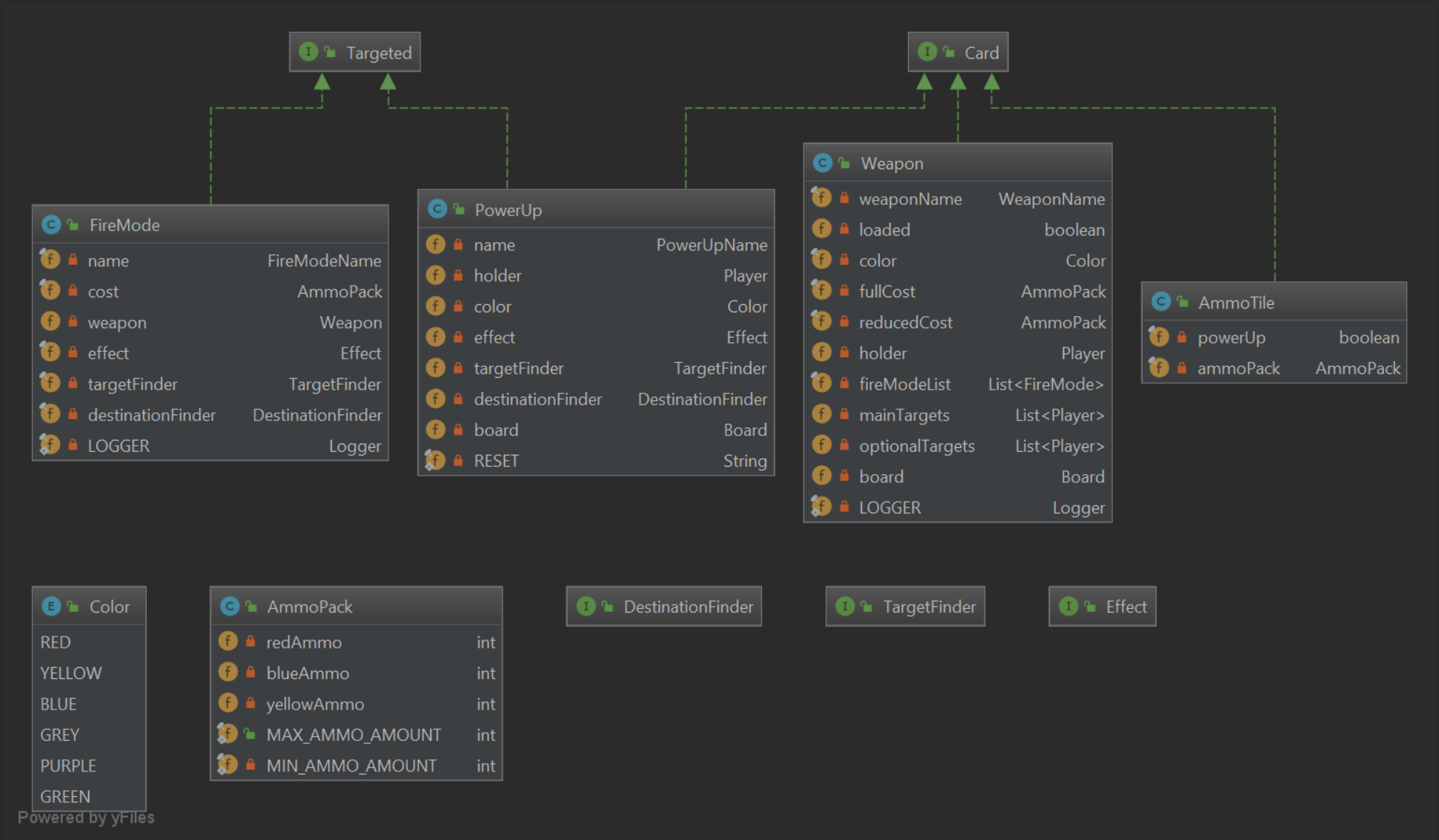
Attributes summary: model.board



Methods summary: model.cards



Attributes summary: model.cards



Methods summary: controller

TurnManager	
runTurn()	void
joinBoard(Player, int, boolean)	void
executeAction()	boolean
executeActualAction(Action)	void
handleUsingPowerUp()	boolean
usePowerUp()	void
handleMoving(Action)	void
handleCollecting()	void
handleShooting()	void
applyFireMode(FireMode)	void
reload(int)	boolean
reloadMandatory()	void
handleTargetingScope(Player, List<Player>)	boolean
askTargetsForGrenade()	void
handleTagbackGrenade(Player)	boolean
handleDeaths()	void
replaceWeapons()	void
replaceAmmoTiles()	void
getDamagesList()	List<Integer>
updateDead()	void
askConfirmation(String)	boolean
askConfirmation(String, Player)	boolean
resetJoinBoard(Player, boolean)	void
resetPowerUp()	void
resetAction()	void
toStringList(List)	List<String>
toUserStringList(List<List<Player>>)	List<String>
updateAndSendModel()	void
updateAndNotifyAll()	void
restoreAndNotify()	void
getVirtualView(Player)	VirtualView
getDead()	List<Integer>
handlePayment(AmmoPack)	void
mandatoryConversion(Color)	void
setDead(List<Integer>)	void

GameEngine	
getPlayers()	List<VirtualView>
getFrenzyActivator()	int
isLastFrenzyPlayer()	boolean
isFrenzy()	boolean
setPlayers(List<VirtualView>)	void
setCurrentPlayer(VirtualView)	void
run()	void
setup()	void
configureMap()	void
configureKillShotTrack()	void
configureFrenzyOption()	void
configurePlayers()	void
battleCry()	void
getNextPlayer()	VirtualView
resolve()	void
runTurn(boolean)	void
changePlayer()	void
manageGameEnd()	void
gameOver()	void
showToLoser(int)	void
addLeaderboard(String)	String
waitShort(VirtualView, int)	String
wait(VirtualView)	String
hasAnswered(VirtualView)	boolean
notify(VirtualView, String)	void
getNotifications()	Map<VirtualView, String>
checkForSuspension()	void
simulateTillEndphase()	void
tryResuming(VirtualView)	boolean
allowPlayersToResume()	void
loadParams()	void
simulationTillEndphaseSetup()	void

PowerUpFactory	
createPowerUp(PowerUpName, Color)	PowerUp

ServerMain	
getInstance()	ServerMain
main(String[])	void
setup()	void
untrackGame(GameEngine)	void
addPlayer(VirtualView)	void
login(VirtualView)	boolean
canResume(String)	boolean
resume(VirtualView)	boolean
removeSuspendedPlayers()	void
getPlayers()	List<VirtualView>
initializeLogger()	void
loadConfig()	Properties
manageInput()	void
refreshConnections()	void
matchmaking()	void
getAlreadyConnected()	String

StatusSaver	
updateCheckpoint()	void
updatePowerups()	void
restoreCheckpoint()	void
restorePowerUps()	void

ModelDataReader	
analyzer(String)	JsonObject
analyzer(String, String, int)	JsonObject
getIntBC(String)	int
getIntBC(String, String, int)	int
getBooleanBC(String, String, int)	boolean
getInt(String)	int
getInt(String, String, int)	int
getBoolean(String, String, int)	boolean
getColorBC(String, String, int)	Color
getColorBC(String)	Color
getInt(JsonObject, String)	int
getBoolean(JsonObject, String)	boolean
getColor(JsonObject, String)	Color

BoardConfigurer	
configureMap(int)	Board
configurePlayerOptions(int, Board)	void
configureDecks(Board)	void
setAmmoTilesAndWeapons(Board)	void
configureKillShotTrack(int, Board)	void
simulateScenario()	Board

WeaponFactory	
createWeapon(WeaponName)	Weapon
getWeaponTree(WeaponName)	JsonObject
getColor(JsonObject)	Color
getFullCost(JsonObject)	AmmoPack
getReducedCost(AmmoPack, Color)	AmmoPack
getFireModeName(String)	FireModeName
getFireModeCost(JsonObject)	AmmoPack
getTargetFinder(JsonObject)	TargetFinder
getDestinationFinder(JsonObject)	DestinationFinder
getEffect(JsonObject)	Effect
createEffect(int, int)	Effect
cartesian(List<List<Player>>, List<List<Player>>)	List<List<Player>>

Timer	
start()	void
stop()	void
reset()	void
update()	void
isRunning()	boolean
isOver()	boolean
pause()	void
resume()	void
getTimeLeft()	long

Attributes summary: controller

GameEngine

players	List<VirtualView>
leaderboard	List<VirtualView>
currentPlayer	VirtualView
frenzyActivator	int
board	Board
killShotTrack	KillShotTrack
timer	Timer
statusSaver	StatusSaver
gameOver	boolean
frenzy	boolean
lastFrenzyPlayer	boolean
exitGame	boolean
notifications	Map<VirtualView, String>
resuming	List<VirtualView>
endphaseSimulation	boolean
turnDuration	int

ServerMain

instance	ServerMain
players	List<VirtualView>
waitingPlayers	List<VirtualView>
currentGames	List<GameEngine>
tcpServer	TCPServer
rmiServer	RMIServer
timer	Timer
executor	ExecutorService
in	BufferedReader
running	boolean
oldMessage	String
LOGGER	Logger
SLEEP_TIMEOUT	int
SERVER_LOG_FILENAME	String

TurnManager

board	Board
statusSaver	StatusSaver
playerConnections	List<VirtualView>
currentPlayerConnection	VirtualView
currentPlayer	Player
dead	List<Integer>
killShotTrack	KillShotTrack
gameEngine	GameEngine
timer	Timer
frenzy	boolean
actionsLeft	int

ModelDataReader

LOGGER	Logger
parser	JsonParser
boardConfFile	String
miscellaneous	String
DATA_NOT_FOUND	String

BoardConfigurer

j	ModelDataReader
BOARDS	String
AMMO_TILES	String
AMMO_SQUARE	String
WEAPON_SQUARE	String
ID	String

StatusSaver

board	Board
playersPositions	List<Square>
playersDamages	List<List<Player>>
playersMarks	List<List<Player>>
playersPowerups	List<List<PowerUp>>
playersAmmoPacks	List<AmmoPack>
currentPlayerWeapons	List<Weapon>
currentPlayerLoadedWeapons	List<Boolean>
squareWeapons	List<List<Weapon>>
LOGGER	Logger

PowerUpFactory

board	Board
j	ModelDataReader
TARGETING_SCOPE_DMG	String
NEWTON_MAX_DISTANCE	String
TAGBACK_GRENADE_MARKS	String

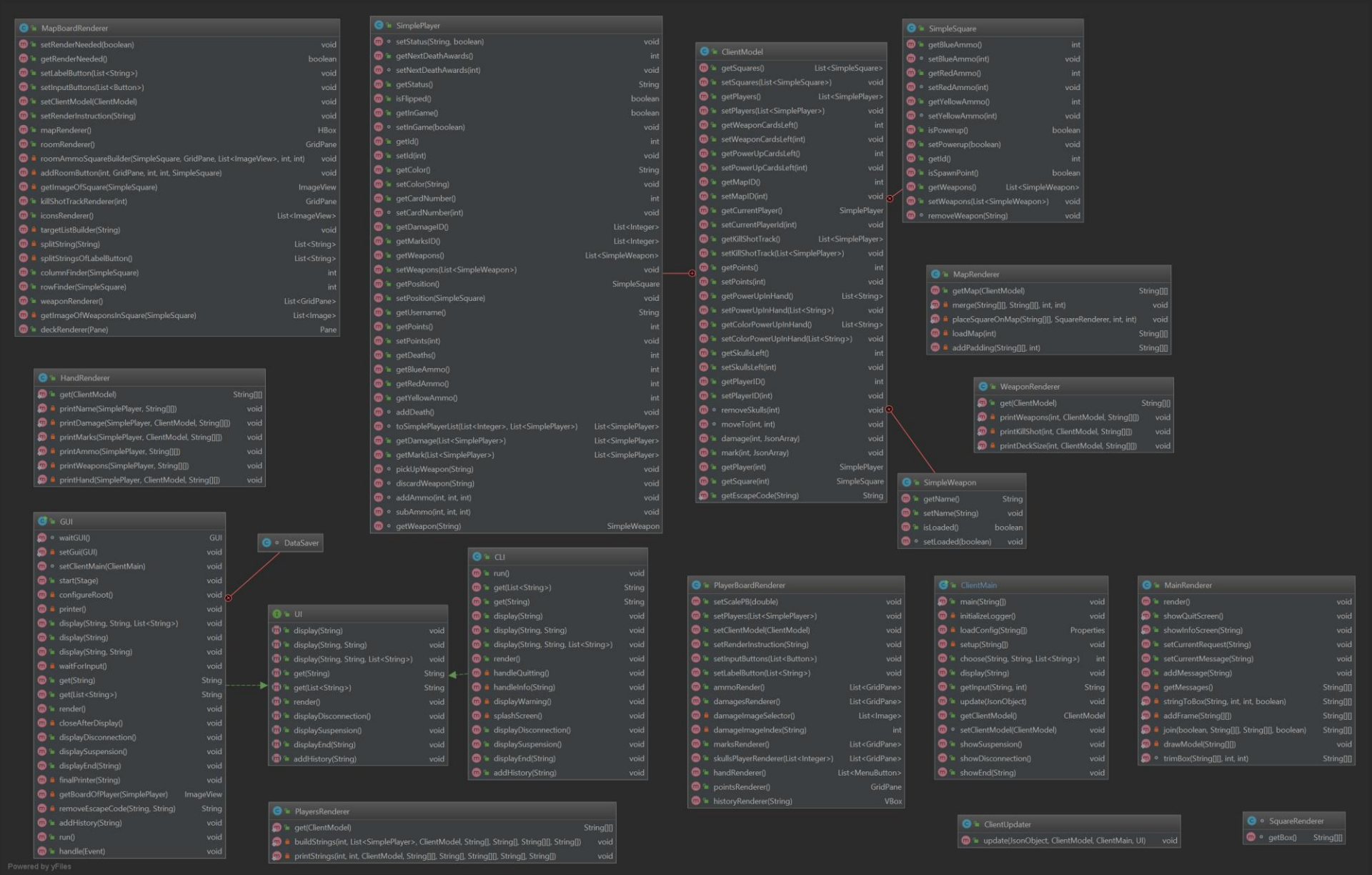
Timer

over	boolean
start	long
duration	long
pausedAt	long
running	boolean

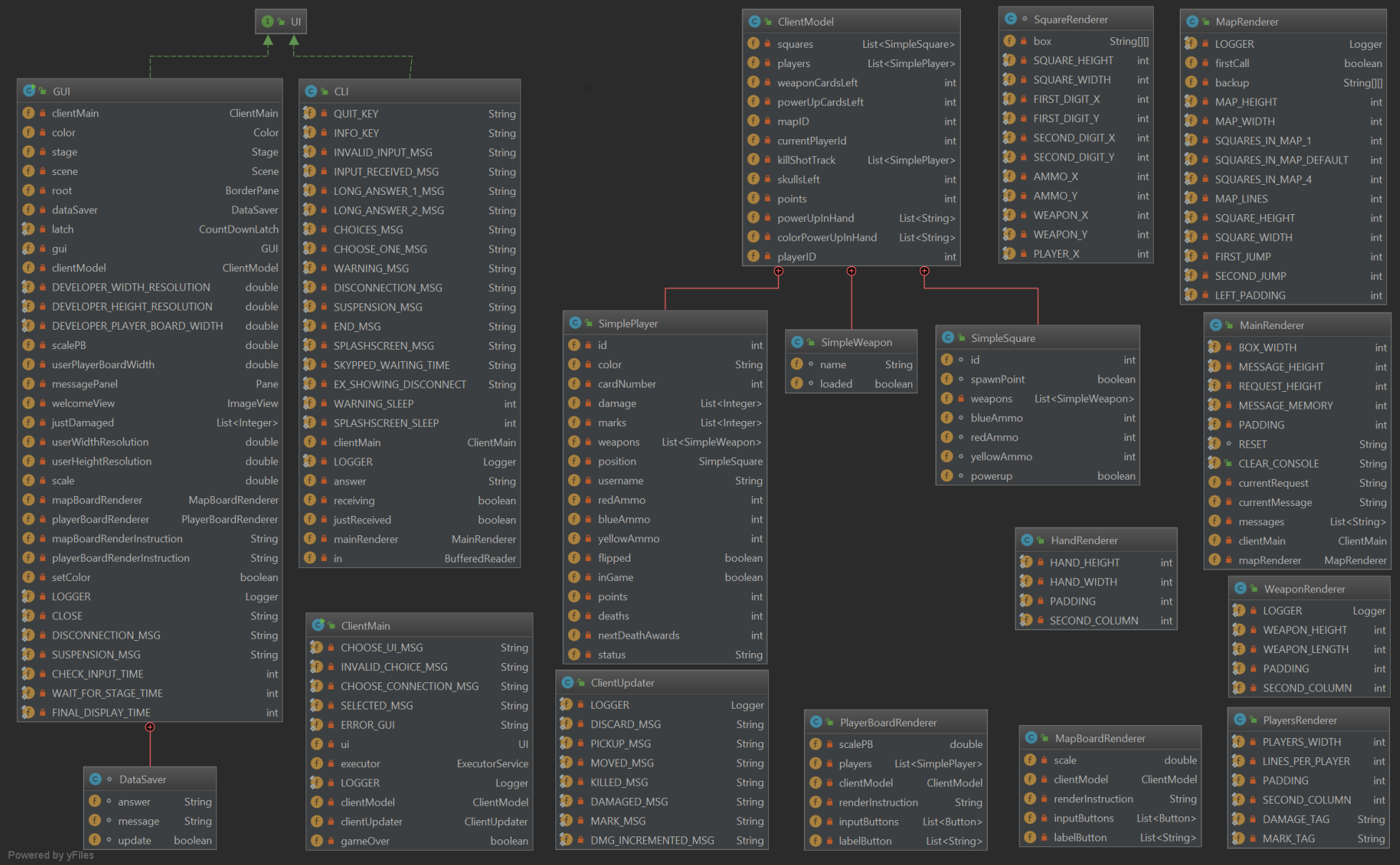
WeaponFactory

board	Board

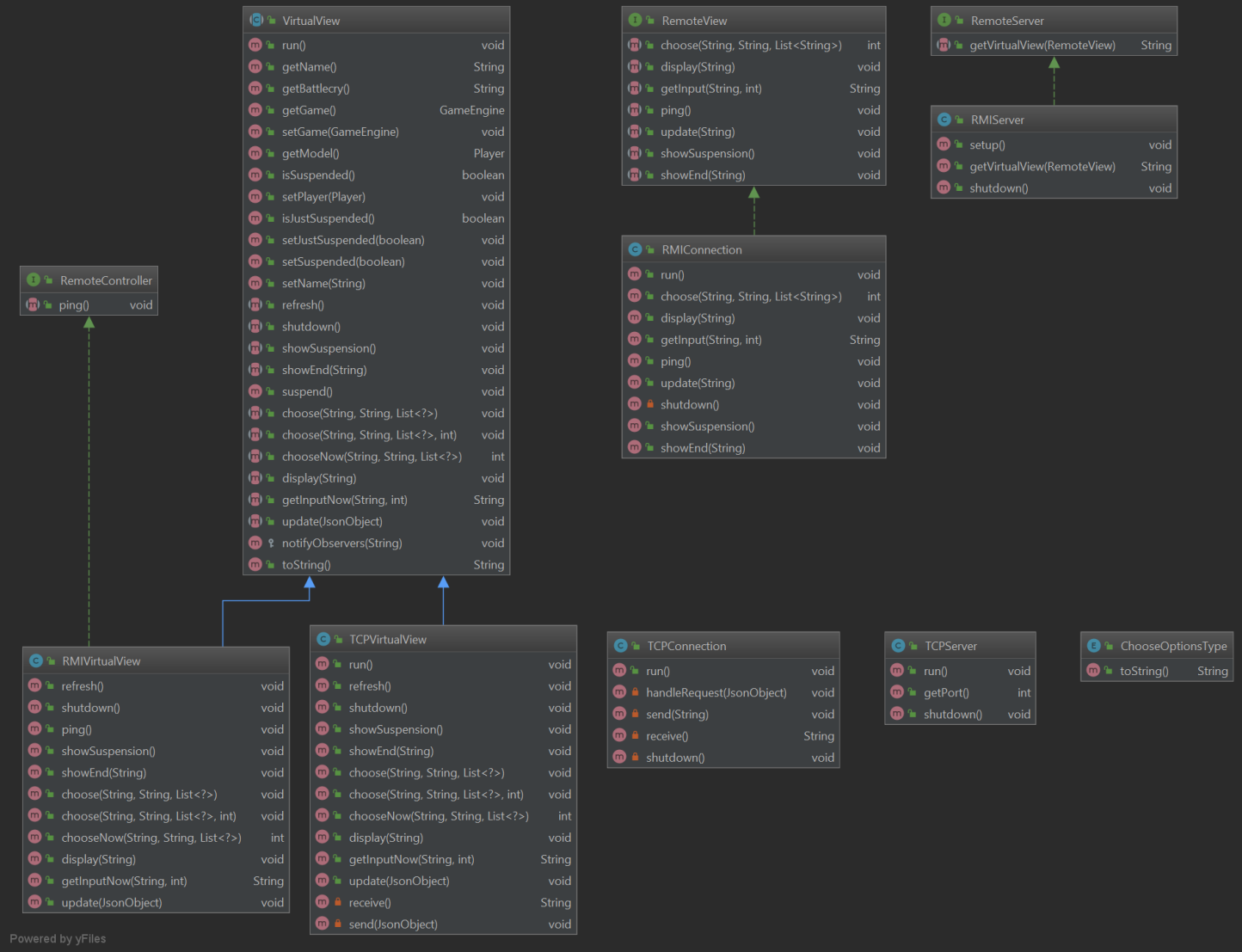
Methods summary: view



Attributes summary: view



Methods summary: network



Attributes summary: network