

ServerModel	
logger	Logger
server	Server
questManager	StoryCardState
eventManager	StoryCardState
currentState	StoryCardState
adventureDeckDiscard	CardCollection<AdventureCard>
inNextQ	boolean
currentViewer	int
currentPlayer	int
currentStage	int
currentSponsor	int
endTurnCounter	int
gameWon	boolean
AllyinPlaySirGalahad	boolean
AllyinPlaySirLancelot	boolean
AllyinPlayKingArthur	boolean
AllyinPlaySirTristan	boolean
AllyinPlayKingPellinore	boolean
AllyinPlaySirGawain	boolean
AllyinPlaySirPercival	boolean
AllyinPlayQueenGuinevere	boolean
AllyinPlayQueenIsleut	boolean
AllyinPlayMerlin	boolean
numStages	int
ServerModel(Server)	
instantiatePlayers(int)	void
instantiateStages()	void
initialShuffle()	void
deal()	void
CardsTest()	void
resetCurrentStage()	void
draw(String, int)	void
removeFromParty(String, int)	void
removeShields(int, int)	void
party(String, int)	void
stage(String, int, int)	boolean
unstage(String, int, int)	void
discard(String, int)	void
assassinate(String, int)	void
queue(String, int)	void
dequeue(String, int)	void
containsSameWeapon(CardCollection<AdventureCard>, String)	boolean
endTurn()	void
stageOver()	void
allysinPlay()	void
containsFoe(CardCollection<AdventureCard>)	boolean
containsAmour(CardCollection<AdventureCard>)	boolean
containsWeapon(CardCollection<AdventureCard>, String)	boolean
getSubType(String, int)	String
playQuest()	void
playEvent()	void
playGame()	void
nextPlayer()	void
nextStory()	void
setScenario1()	void
setScenario2()	void
setScenarioTest()	void
eventTesting()	void
adventureDeckDiscard	CardCollection<AdventureCard>
storyDeckDiscard	CardCollection<StoryCard>
activePlayer	Player
state	State
stages	ArrayList<CardCollection<AdventureCard>>
currentStage	int
adventureDeck	AdventureDeck
numPlayers	int
currentState	StoryCardState
players	Player[]
storyDeck	StoryDeck

EventManager	
logger	Logger
players	Player[]
numPlayers	int
adventureDeck	AdventureDeck
adventureDeckDiscard	CardCollection<AdventureCard>
nextQ	boolean
currentPlayer	int
serverModel	ServerModel
EventManager(AbstractModel)	
EventManager(ServerModel)	
setPlayers()	void
handleEvent(String)	void
CourtCalled()	void
KingsRecognition()	void
QueenFavor()	void
Pox()	void
KingCallToArms()	void
ProsperityThroughoutTheRealm()	void
ChilvarousDeed()	void
Plague()	void
handle()	void
nextPlayer1()	Player
nextPlayer()	void
setPlayer()	void
canEndTurn()	boolean
resolveStage()	void
increaseResponse()	void
hasSponsor	boolean

QuestManager	
logger	Logger
ENDTURN	String
serverModel	ServerModel
players	Player[]
questersReady	boolean
numOfansewers	int
numberOfCardsToReturn	int
numberOfRequests	int
questers	QuesterQueue
numOfQuesterPotential	int
numOfQuester	int
numOfRepsponders	int
nextPersonToDraw	int
QuestManager(ServerModel)	
handle()	void
nextPlayer()	void
stageHarder()	boolean
totalNumOfBP(CardCollection<AdventureCard>)	int
isfoeEachStage()	boolean
numberOfCardsForSponsor()	int
checkHandSize()	boolean
canEndTurn()	boolean
reset()	void
setPlayer()	void
resolveQuest()	int
resolveStage()	void
increaseResponse()	void
hasSponsor	boolean

Server	
logger	Logger
DEFAULT_PORT	int
DEFAULT_PLAYERS_PER_TABLE	int
maxPlayers	int
serverPort	int
playersPerGame	int
clientThreads	ArrayList<ServerThread>
serverModel	ServerModel
Server(int, int)	
start()	void
sendMessage(String)	void
sendMessageToOne(String, int)	void
sendMessageToAllButOne(String, int)	void
getSponsorDecision()	void
getQuesterDecision()	void
main(String[])	void
alert(String)	void
updateViewState()	void
resolveQuest()	void
resolveStage()	void
update()	void
getTournamentDecision()	void
model	ServerModel

TournamentManger	
logger	Logger
ENDTURN	String
serverModel	ServerModel
players	Player[]
questersReady	boolean
numOfansewers	int
numberOfCardsToReturn	int
numberOfRequests	int
questers	QuesterQueue
numOfQuesterPotential	int
numOfQuester	int
numOfRepsponders	int
nextPersonToDraw	int
TournamentManger(ServerModel)	
handle()	void
nextPlayer()	void
setPlayer()	void
canEndTurn()	boolean
resolveStage()	void
increaseResponse()	void
hasSponsor	boolean

ServerThread	
logger	Logger
socket	Socket
server	Server
out	PrintWriter
in	BufferedReader
ServerThread(Socket, Server, int)	
run()	void
getClientMessage(String)	void
playerNumber	int

MainMenu	
logger	Logger
IMG_DIR	String
GIF	String
SQUIRE	String
KNIGHT	String
CHAMPION_KNIGHT	String
numberPlayers	int
MainMenu(View, Pane)	
numberSelected()	int