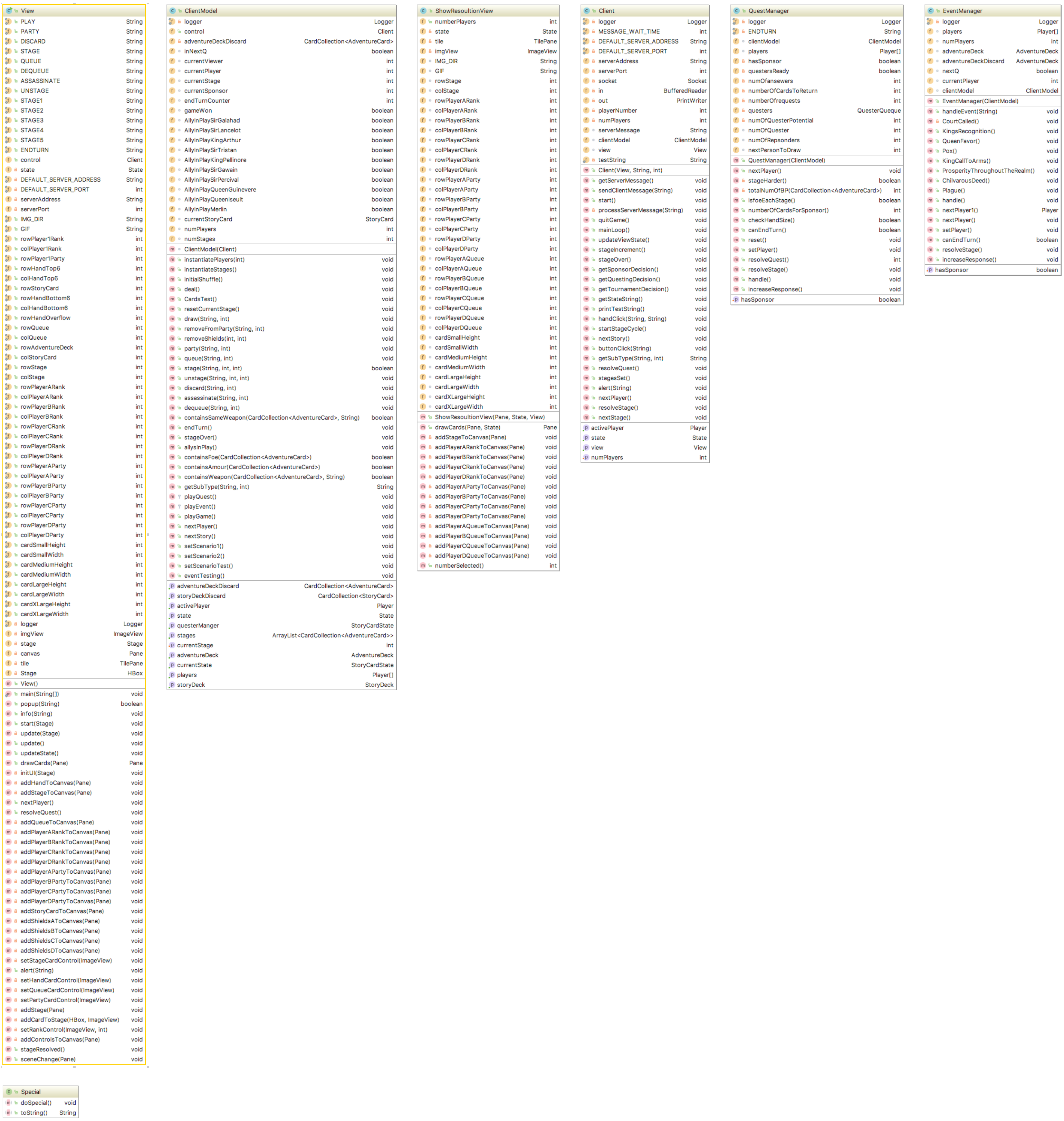

View	
PLAY	String
PARTY	String
DISCARD	String
STAGE	String
QUEUE	String
DEQUEUE	String
ASSASSINATE	String
UNSTAGE	String
STAGE1	String
STAGE2	String
STAGE3	String
STAGE4	String
STAGE5	String
ENDTURN	String
control	Client
state	State
DEFAULT_SERVER_ADDRESS	String
DEFAULT_SERVER_PORT	int
serverAddress	String
serverPort	int
IMG_DIR	String
GIF	String
rowPlayer1Rank	int
colPlayer1Rank	int
rowPlayer1Party	int
rowHandTop6	int
colHandTop6	int
rowStoryCard	int
rowHandBottom6	int
colHandBottom6	int
rowHandOverflow	int
rowQueue	int
colQueue	int
rowAdventureDeck	int
colStoryCard	int
rowStage	int
colStage	int
rowPlayerARank	int
colPlayerARank	int
rowPlayerBRank	int
colPlayerBRank	int
rowPlayerCRank	int
colPlayerCRank	int
rowPlayerDRank	int
colPlayerDRank	int
rowPlayerAParty	int
colPlayerAParty	int
rowPlayerBParty	int
colPlayerBParty	int
rowPlayerCParty	int
colPlayerCParty	int
rowPlayerDParty	int
colPlayerDParty	int
cardSmallHeight	int
cardSmallWidth	int
cardMediumHeight	int
cardMediumWidth	int
cardLargeHeight	int
cardLargeWidth	int
cardXLargeHeight	int
cardXLargeWidth	int
logger	Logger
imgView	ImageView
stage	Stage
canvas	Pane
tile	TilePane
Stage	HBox
View()	
main(String[])	void
popup(String)	boolean
info(String)	void
start(Stage)	void
update(Stage)	void
update()	void
updateState()	void
drawCards(Pane)	Pane
initUI(Stage)	void
addHandToCanvas(Pane)	void
addStageToCanvas(Pane)	void
nextPlayer()	void
resolveQuest()	void
addQueueToCanvas(Pane)	void
addPlayerARankToCanvas(Pane)	void
addPlayerBRankToCanvas(Pane)	void
addPlayerCRankToCanvas(Pane)	void
addPlayerDRankToCanvas(Pane)	void
addPlayerAPartyToCanvas(Pane)	void
addPlayerBPartyToCanvas(Pane)	void
addPlayerCPartyToCanvas(Pane)	void
addPlayerDPartyToCanvas(Pane)	void
addStoryCardToCanvas(Pane)	void
addShieldsAToCanvas(Pane)	void
addShieldsBToCanvas(Pane)	void
addShieldsCToCanvas(Pane)	void
addShieldsDToCanvas(Pane)	void
setStageCardControl(ImageView)	void
alert(String)	void
setHandCardControl(ImageView)	void
setQueueCardControl(ImageView)	void
setPartyCardControl(ImageView)	void
addStage(Pane)	void
addCardToStage(HBox, ImageView)	void
setRankControl(ImageView, int)	void
addControlsToCanvas(Pane)	void
stageResolved()	void
sceneChange(Pane)	void

Special	
doSpecial()	void
toString()	String

ClientModel	
logger	Logger
control	Client
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
AllyInPlayQueenIseult	boolean
AllyInPlayMerlin	boolean
currentStoryCard	StoryCard
numPlayers	int
numStages	int
ClientModel(Client)	
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
queue(String, int)	void
stage(String, int, int)	boolean
unstage(String, int, int)	void
discard(String, int)	void
assassinate(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
questerManger	StoryCardState
stages	ArrayList<CardCollection<AdventureCard>>
currentStage	int
adventureDeck	AdventureDeck
currentState	StoryCardState
players	Player[]
storyDeck	StoryDeck

ShowResoultionView	
numberPlayers	int
state	State
tile	TilePane
imgView	ImageView
IMG_DIR	String
GIF	String
rowStage	int
colStage	int
rowPlayerARank	int
colPlayerARank	int
rowPlayerBRank	int
colPlayerBRank	int
rowPlayerCRank	int
colPlayerCRank	int
rowPlayerDRank	int
colPlayerDRank	int
rowPlayerAParty	int
colPlayerAParty	int
rowPlayerBParty	int
colPlayerBParty	int
rowPlayerCParty	int
colPlayerCParty	int
rowPlayerDParty	int
colPlayerDParty	int
rowPlayerAQueue	int
colPlayerAQueue	int
rowPlayerBQueue	int
colPlayerBQueue	int
rowPlayerCQueue	int
colPlayerCQueue	int
rowPlayerDQueue	int
colPlayerDQueue	int
cardSmallHeight	int
cardSmallWidth	int
cardMediumHeight	int
cardMediumWidth	int
cardLargeHeight	int
cardLargeWidth	int
cardXLargeHeight	int
cardXLargeWidth	int
ShowResoultionView(Pane, State, View)	
drawCards(Pane, State)	Pane
addStageToCanvas(Pane)	void
addPlayerARankToCanvas(Pane)	void
addPlayerBRankToCanvas(Pane)	void
addPlayerCRankToCanvas(Pane)	void
addPlayerDRankToCanvas(Pane)	void
addPlayerAPartyToCanvas(Pane)	void
addPlayerBPartyToCanvas(Pane)	void
addPlayerCPartyToCanvas(Pane)	void
addPlayerDPartyToCanvas(Pane)	void
addPlayerAQueueToCanvas(Pane)	void
addPlayerBQueueToCanvas(Pane)	void
addPlayerCQueueToCanvas(Pane)	void
addPlayerDQueueToCanvas(Pane)	void
numberSelected()	int

Client	
logger	Logger
MESSAGE_WAIT_TIME	int
DEFAULT_SERVER_ADDRESS	String
DEFAULT_SERVER_PORT	int
serverAddress	String
serverPort	int
socket	Socket
in	BufferedReader
out	PrintWriter
playerNumber	int
numPlayers	int
serverMessage	String
clientModel	ClientModel
view	View
testString	String
Client(View, String, int)	
getServerMessage()	void
sendClientMessage(String)	void
start()	void
processServerMessage(String)	void
quitGame()	void
mainLoop()	void
updateViewState()	void
stageIncrement()	void
stageOver()	void
getSponsorDecision()	void
getQuestingDecision()	void
getTournamentDecision()	void
getStateString()	void
printTestString()	void
handClick(String, String)	void
startStageCycle()	void
nextStory()	void
buttonClick(String)	void
getSubType(String, int)	String
resolveQuest()	void
stagesSet()	void
alert(String)	void
nextPlayer()	void
resolveStage()	void
nextStage()	void
activePlayer	Player
state	State
view	View
numPlayers	int

QuestManager	
logger	Logger
ENDTURN	String
clientModel	ClientModel
players	Player[]
hasSponsor	boolean
questersReady	boolean
numOfansewers	int
numberOfCardsToReturn	int
numberOfrequests	int
questers	QuesterQueque
numOfQuesterPotential	int
numOfQuester	int
numOfRepsonders	int
nextPersonToDraw	int
QuestManager(ClientModel)	
nextPlayer()	void
stageHarder()	boolean
totalNumOfBP(CardCollection<AdventureCard>)	int
isfoeEachStage()	boolean
numberOfCardsForSponsor()	int
checkHandSize()	boolean
canEndTurn()	boolean
reset()	void
setPlayer()	void
resolveQuest()	int
resolveStage()	void
handle()	void
increaseResponse()	void
hasSponsor	boolean

EventManager	
logger	Logger
players	Player[]
numPlayers	int
adventureDeck	AdventureDeck
adventureDeckDiscard	AdventureDeck
nextQ	boolean
currentPlayer	int
clientModel	ClientModel
EventManager(ClientModel)	
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