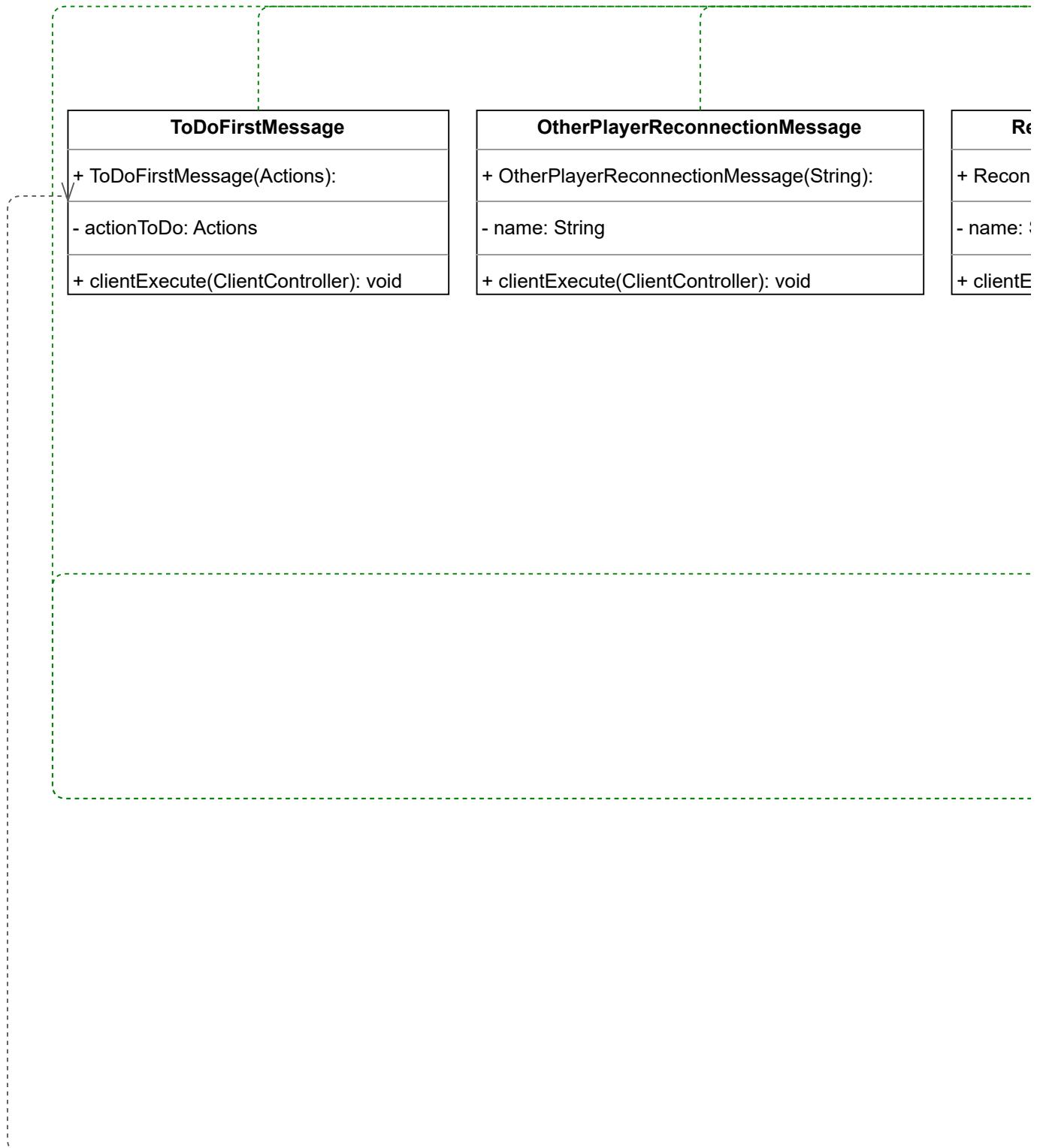
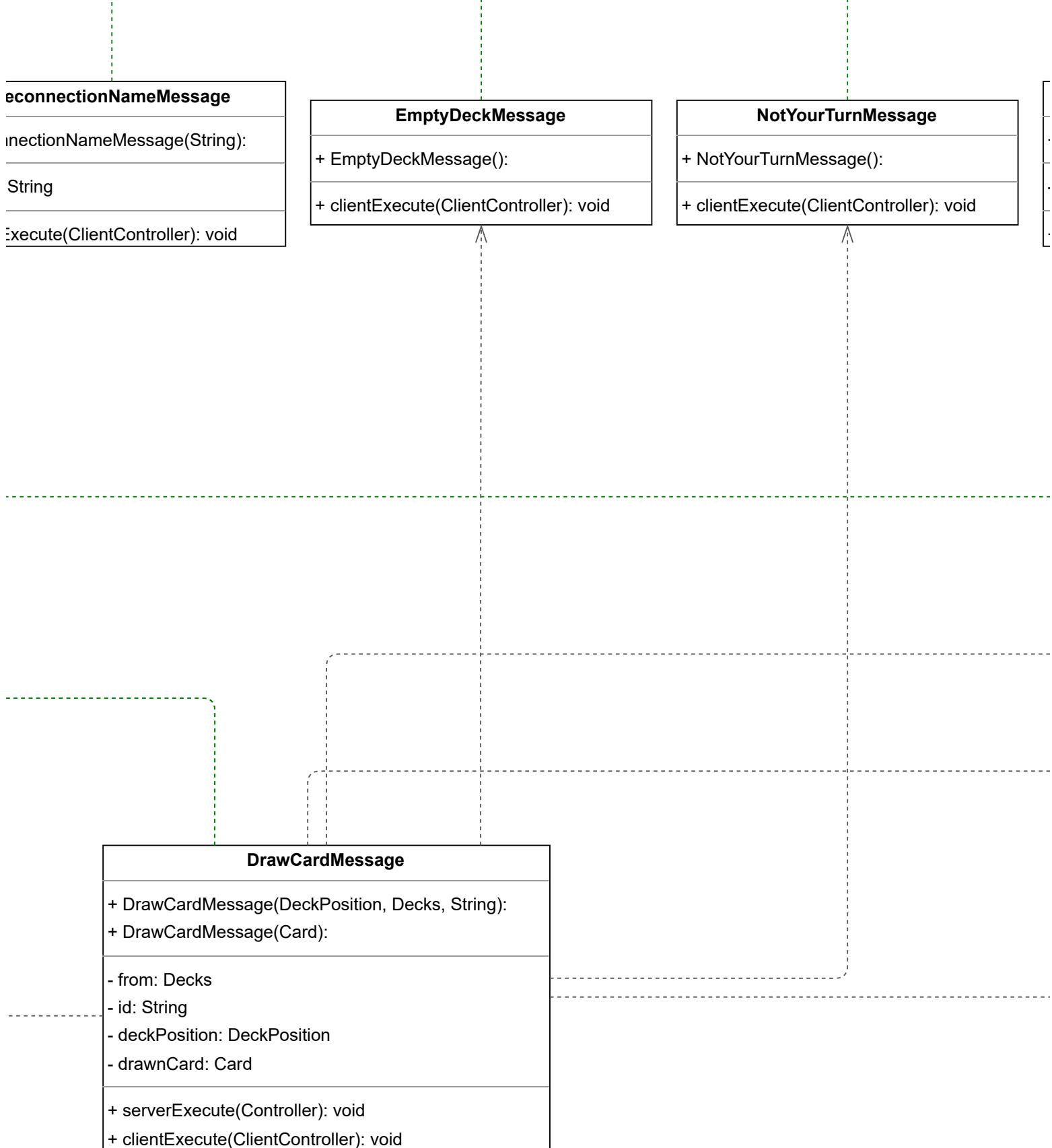
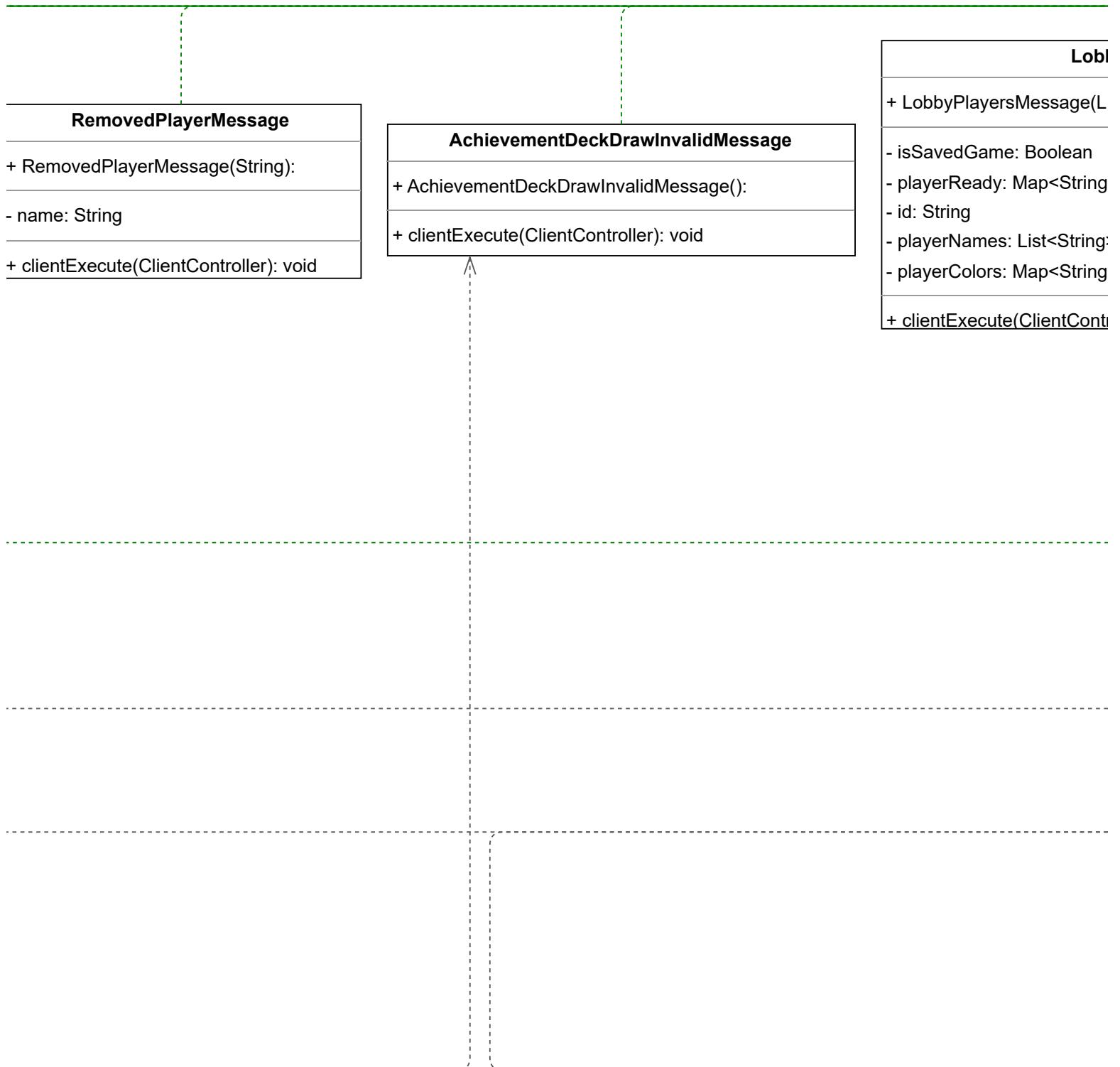
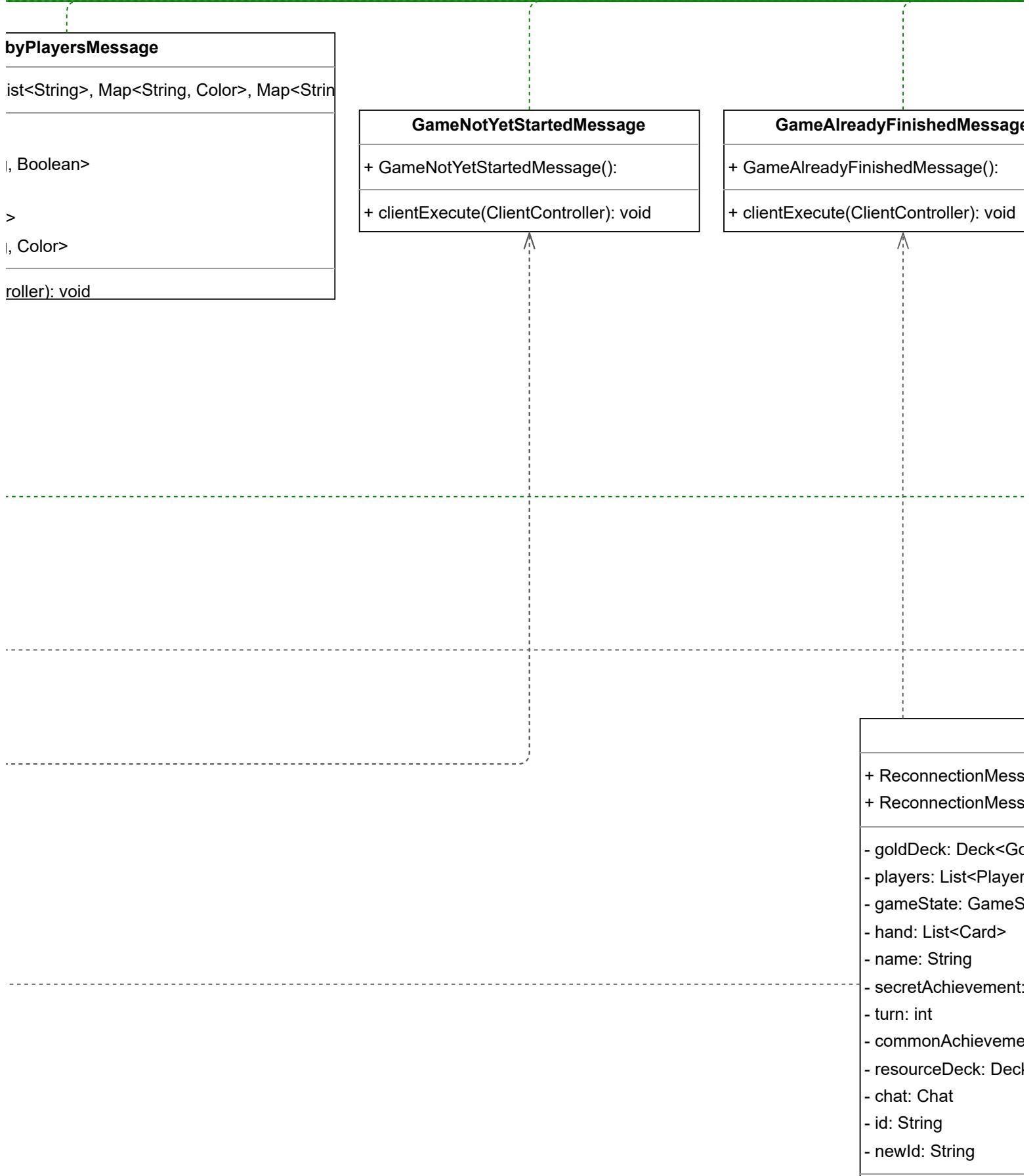


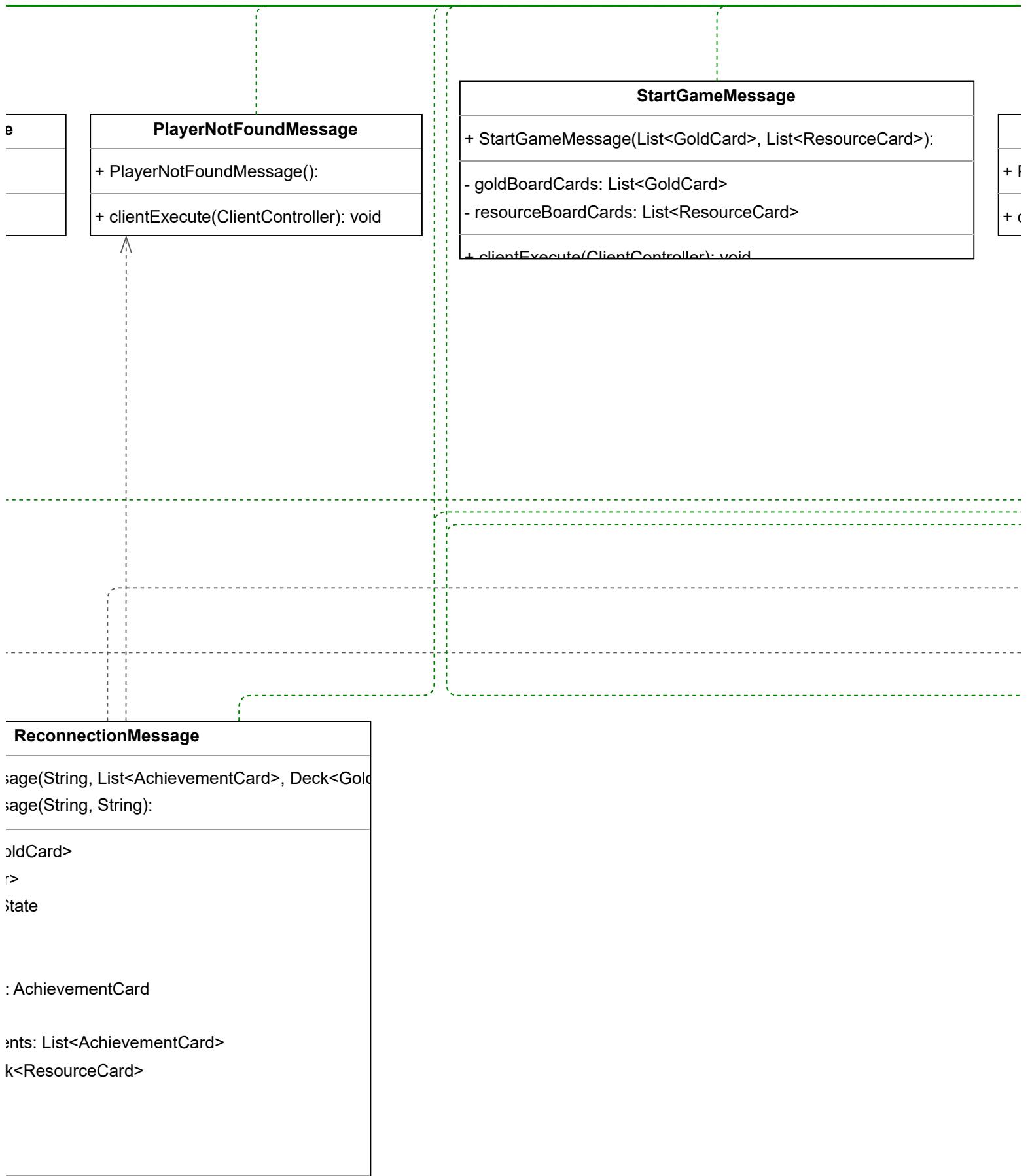
MESSAGES

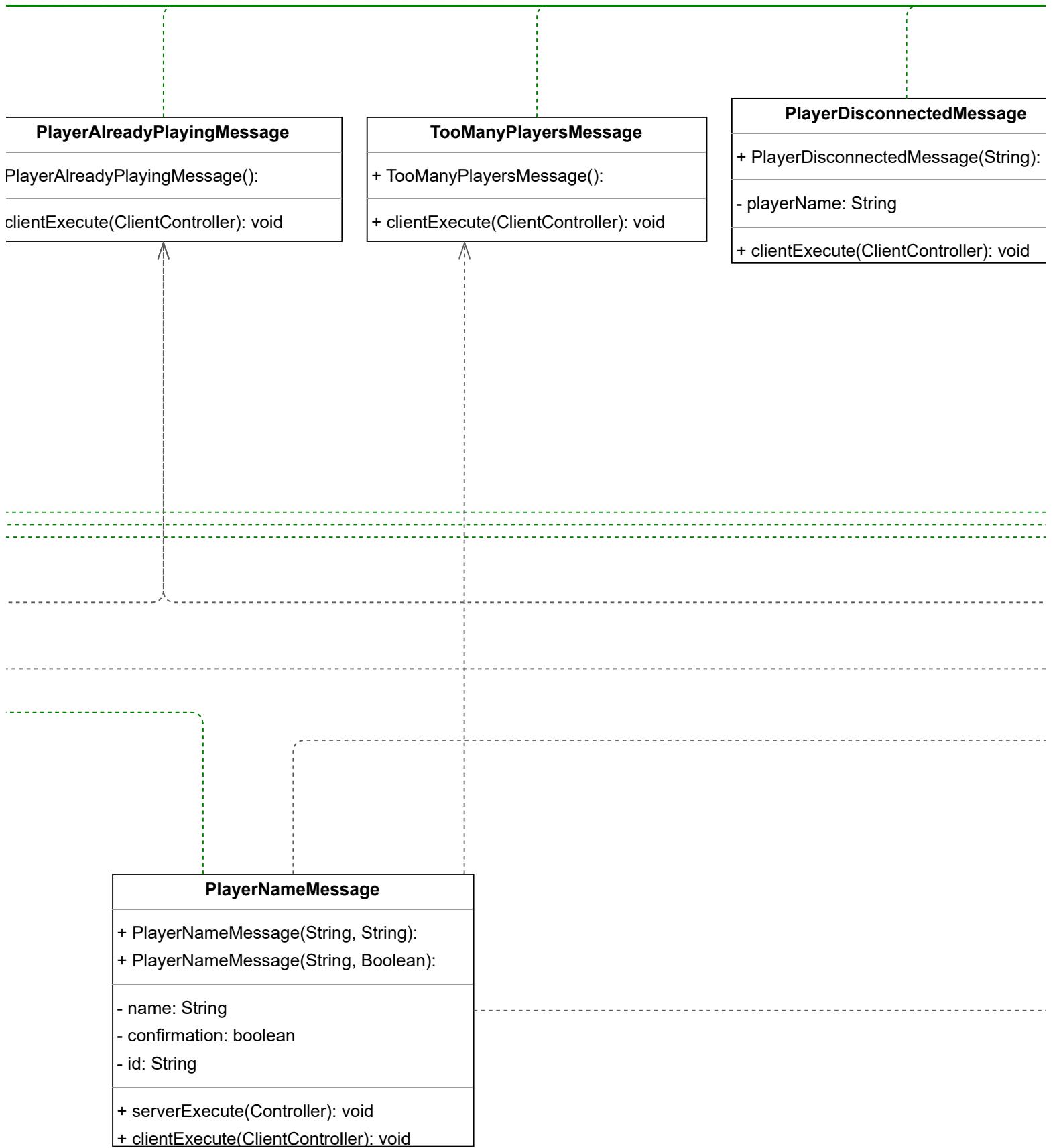


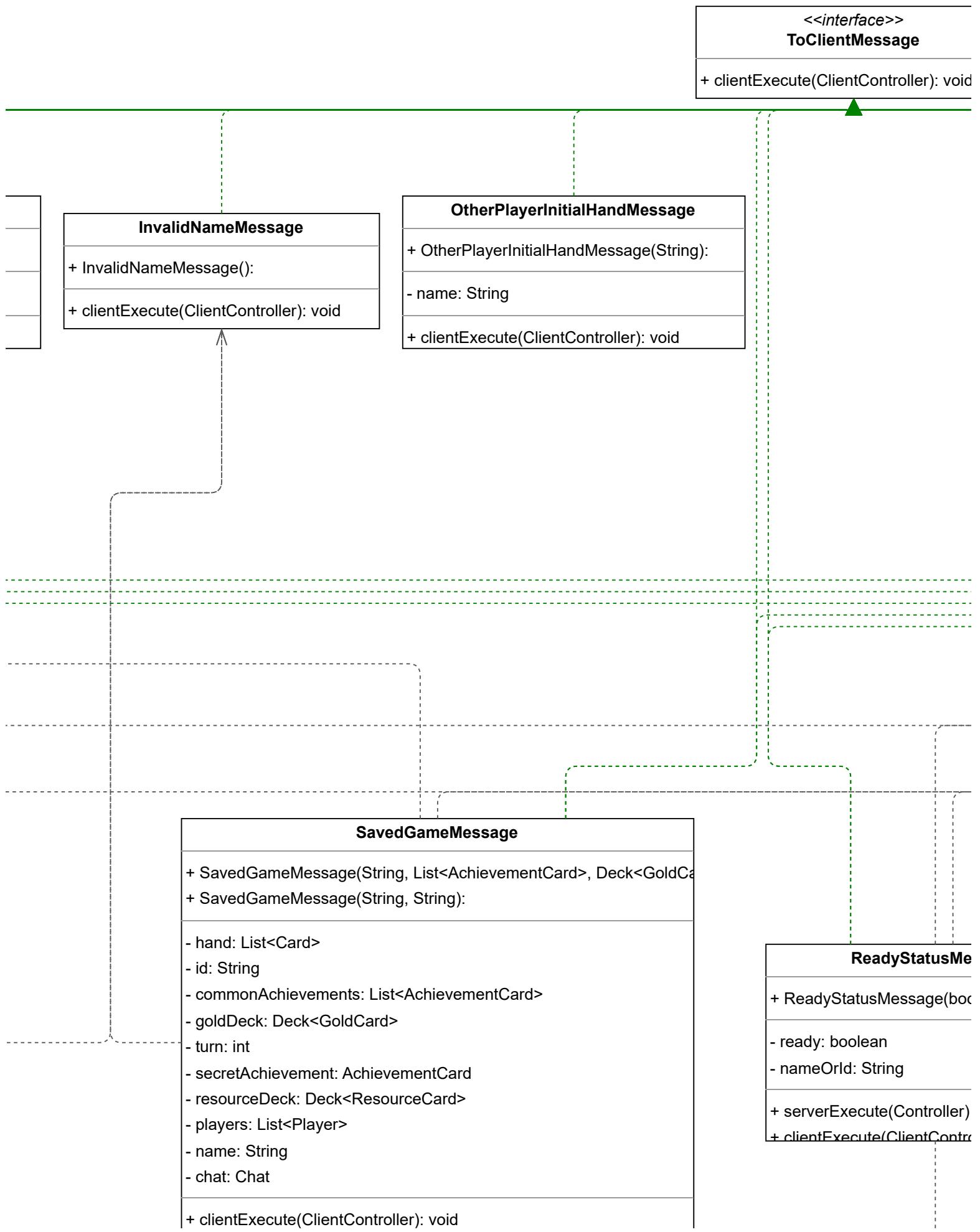


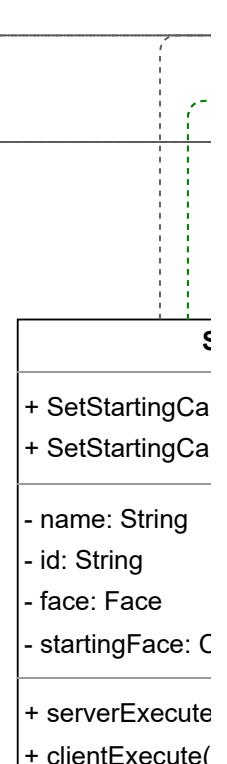
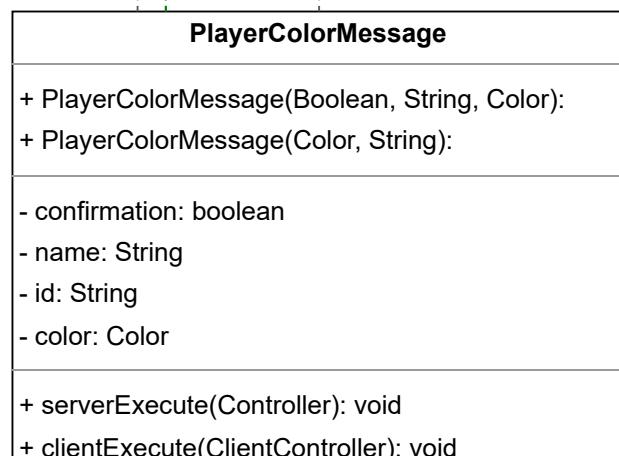
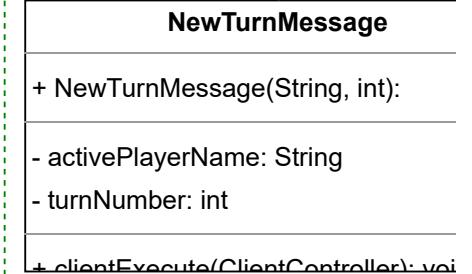


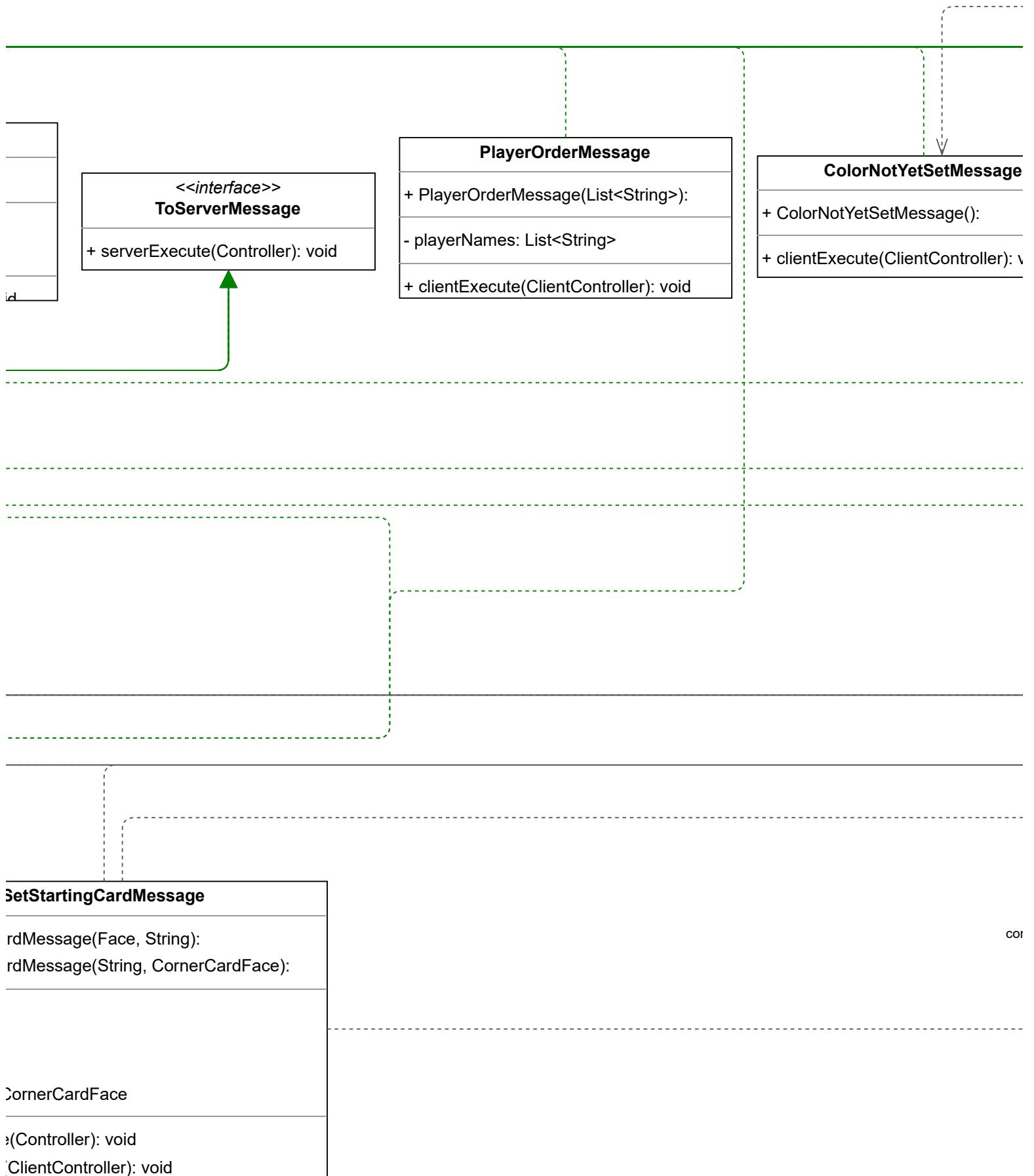


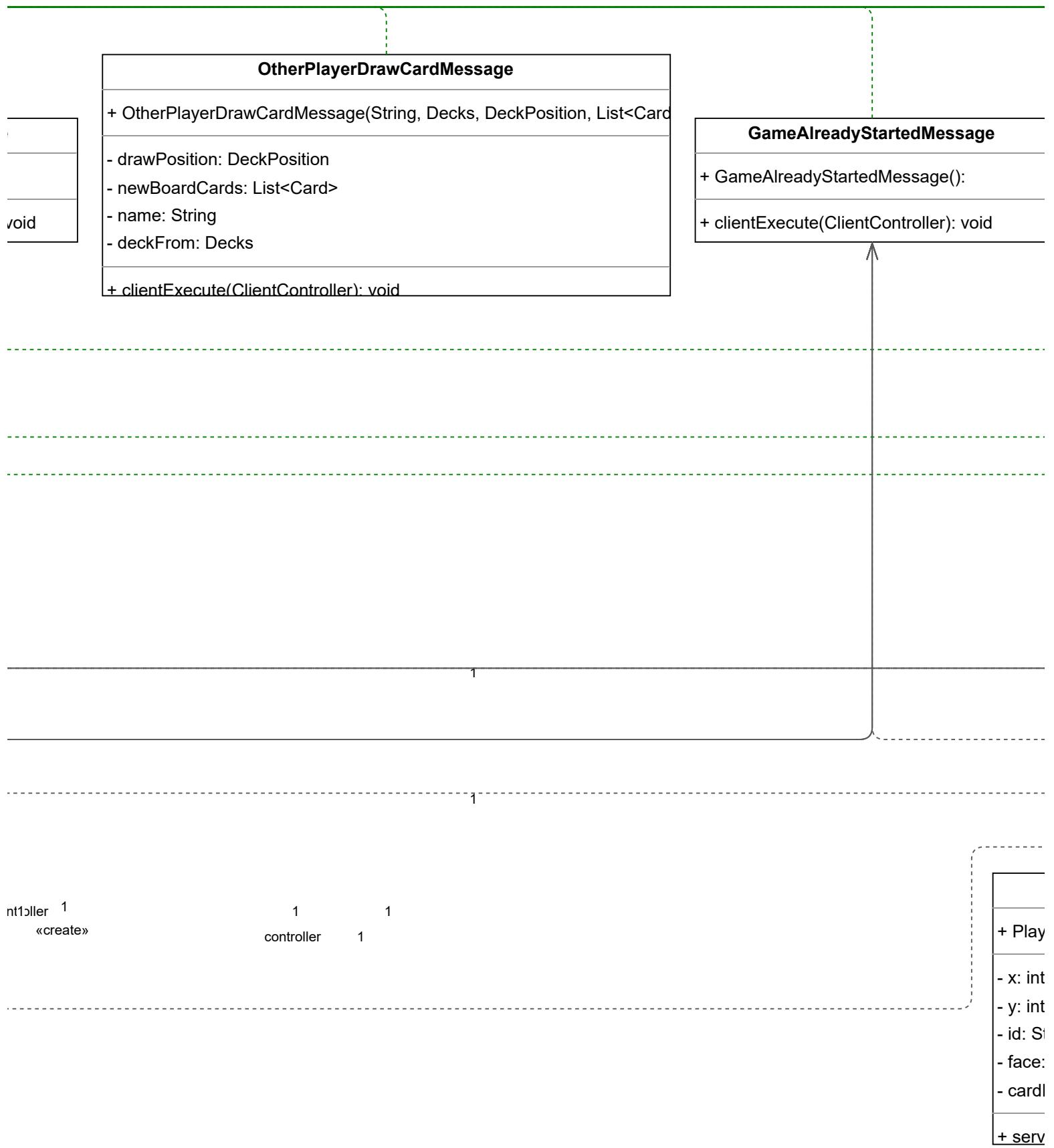


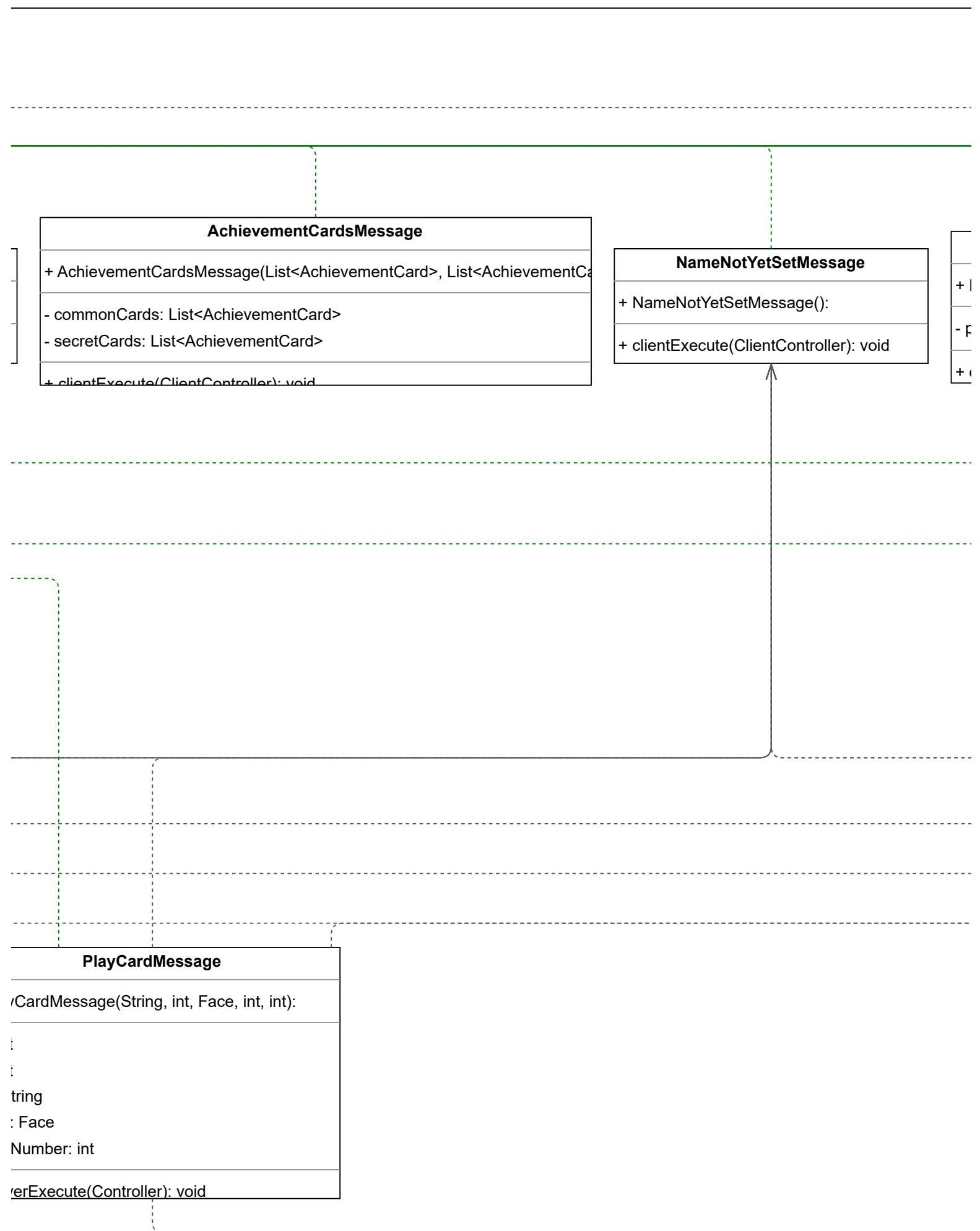


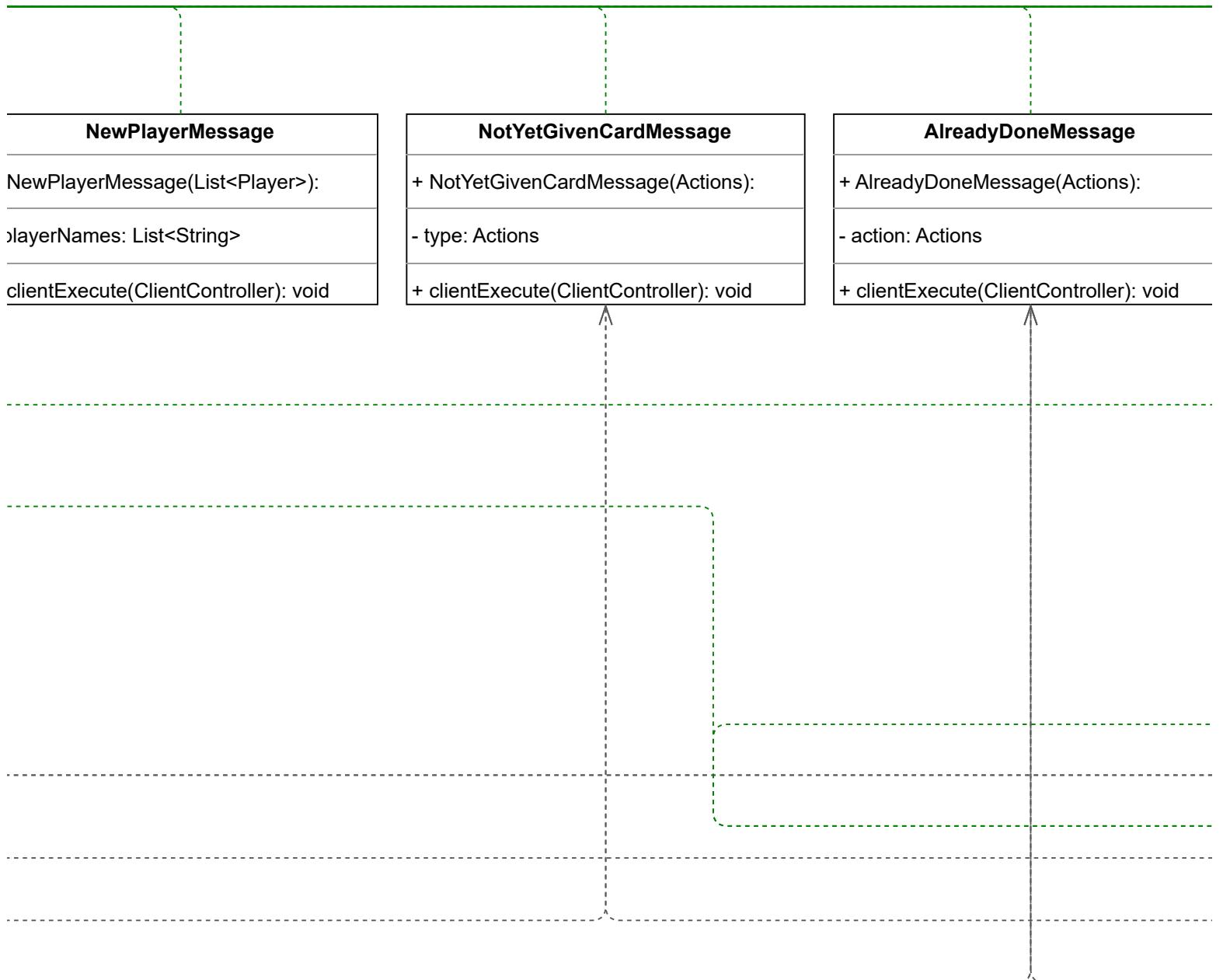












OtherPlayerPlayCardMessage

+ OtherPlayerPlayCardMessage(String, CornerCardFace, int, int, int)

~ placedCardFace: CornerCardFace
~ playerName: String
~ x: int
~ y: int
~ obtainedPoints: int

+ clientExecute(ClientController): void

StartingCardsMessage

+ StartingCardsMessage(Card):

- card: Card

+ clientExecute(ClientController): void

+ Invalid

- cardTy

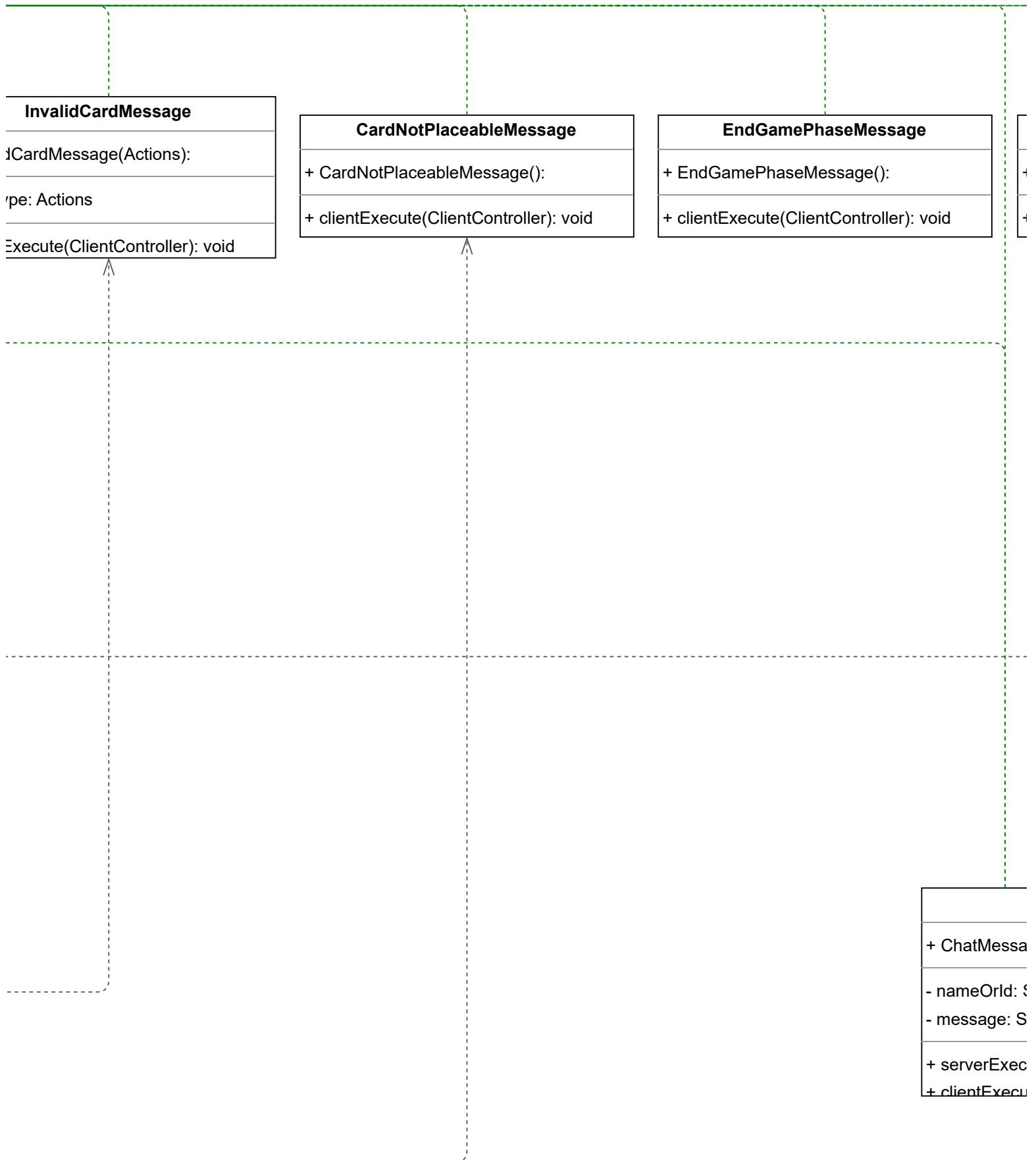
+ clientE

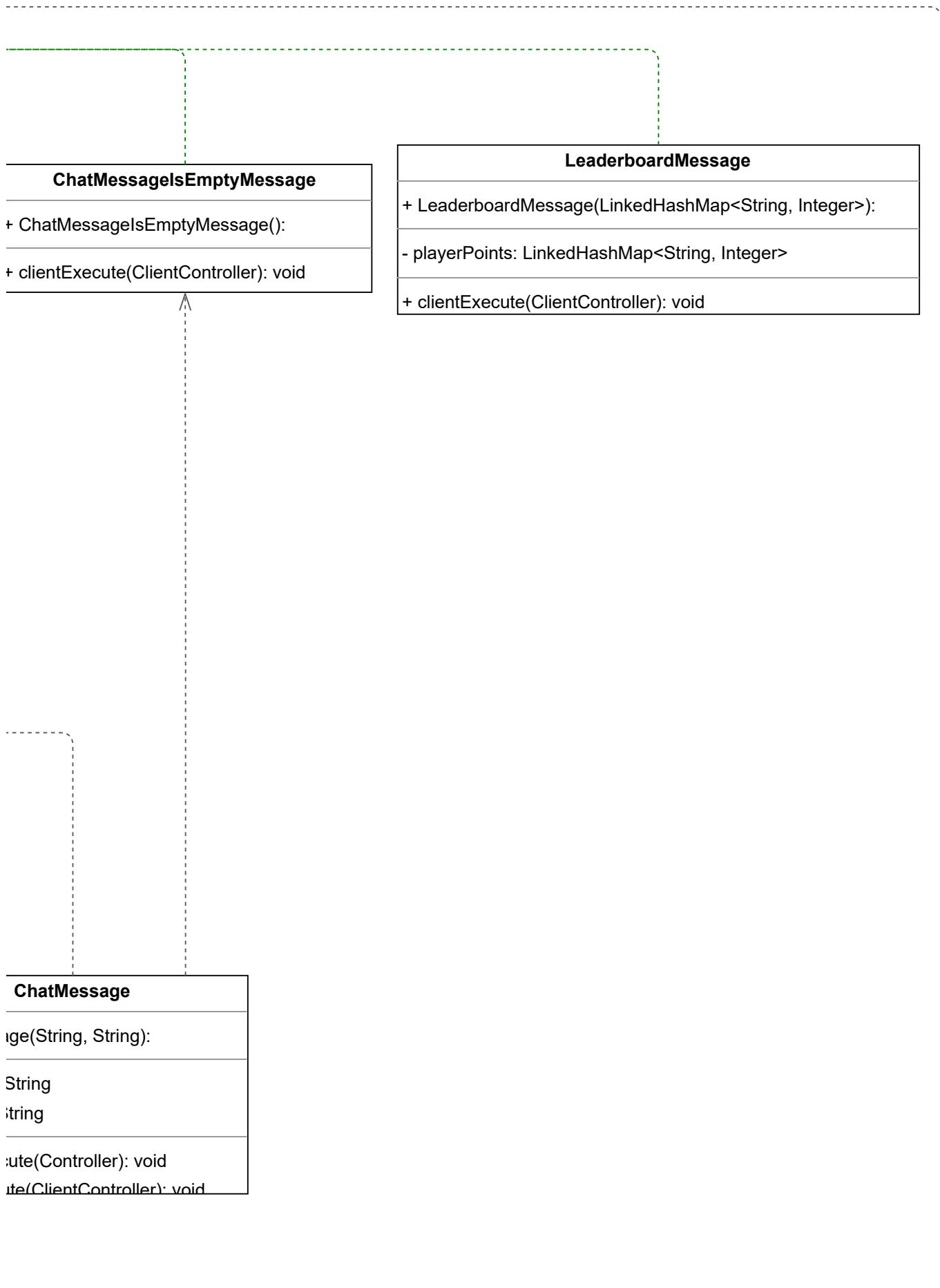
SetSecretCardMessage

+ SetSecretCardMessage(String):
+ SetSecretCardMessage(int, String):

- idOrName: String
- chosenCard: Integer

+ serverExecute(Controller): void
+ clientExecute(ClientController): void

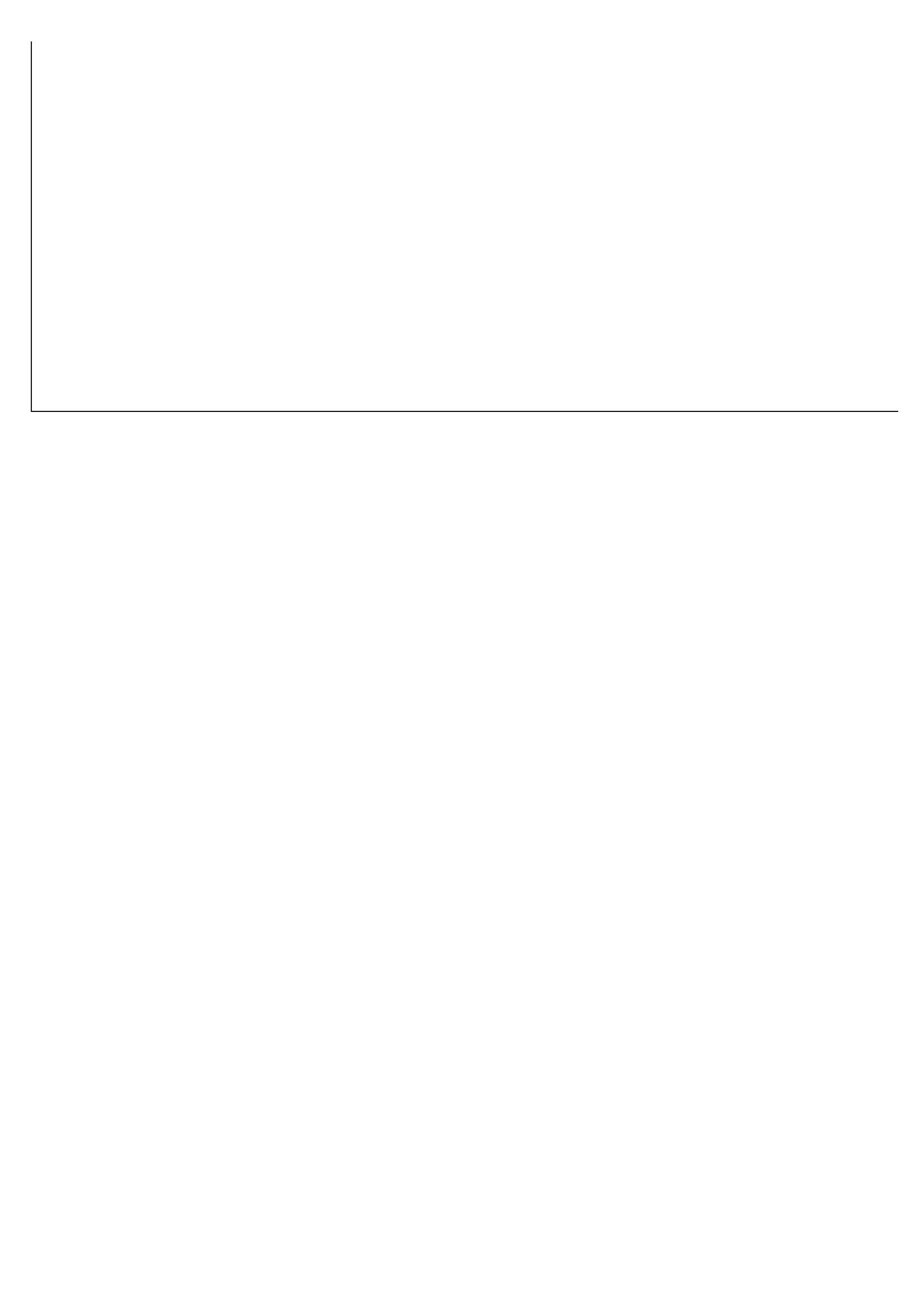




CLIENT

<<interface>>
ClientController

```
+ giveInitialHand(List<Card>): void
+ otherPlayerReconnected(String): void
+ nameNotYetSet(): void
+ drawOtherPlayer(String, Decks, DeckPosition, List<Card>): void
+ chooseSecretAchievement(int): void
+ getLeaderboard(): Map<String, Integer>
+ playerAlreadyPlaying(): void
+ playerRemoved(String): void
+ setName(Boolean): void
+ getPlayerNames(): List<String>
+ main(UI): void
+ setColor(boolean, Color): void
+ sendChatMessage(String): void
+ getDeckSize(Decks): int
+ displayLeaderboard(LinkedHashMap<String, Integer>): void
+ clear(): void
+ getChat(): List<Message>
+ giveDrawnCard(Card): void
+ updatePlayerColors(Color, String): void
+ getHand(): List<Card>
+ setSecretCard(int): void
+ clearUI(): void
+ getPotentialSecretAchievements(): List<AchievementCard>
+ askSetColor(Color): void
+ getMyId(): String
+ loadLobbyInfo(String, List<String>, Map<String, Color>, Map<String, Boolean>)
+ askPlayCard(int, Face, int, int): void
+ chooseStartingCardFace(Face): void
+ setMyColor(Color): void
+ setRMIMode(boolean): void
+ setEndGame(): void
```



SE

:RVER

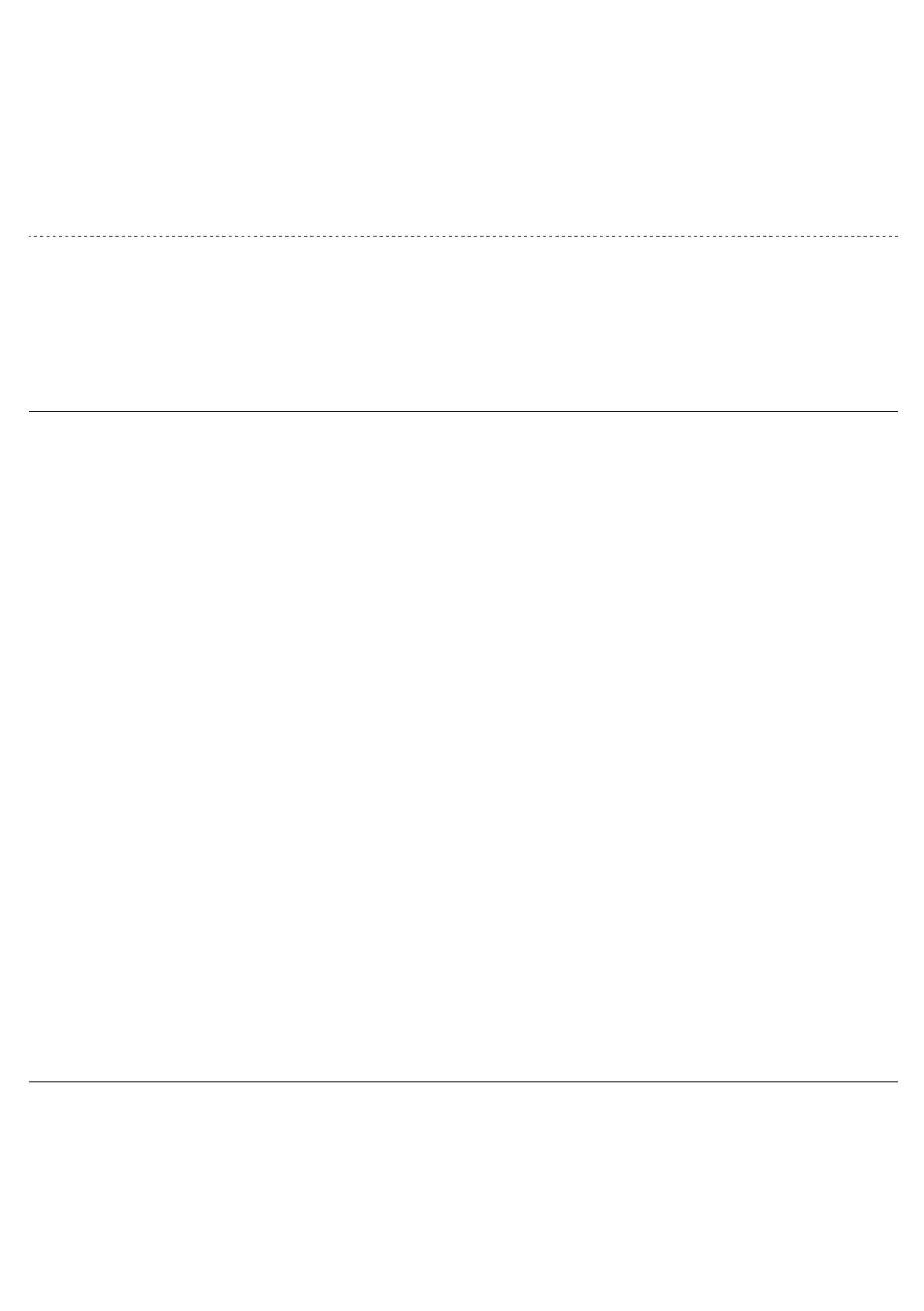
+ clientExecute(Clien
+ serverExecute(Con



```
ntController): void  
ntroller): void
```

1
controller

+ serverExecute(Controller): void

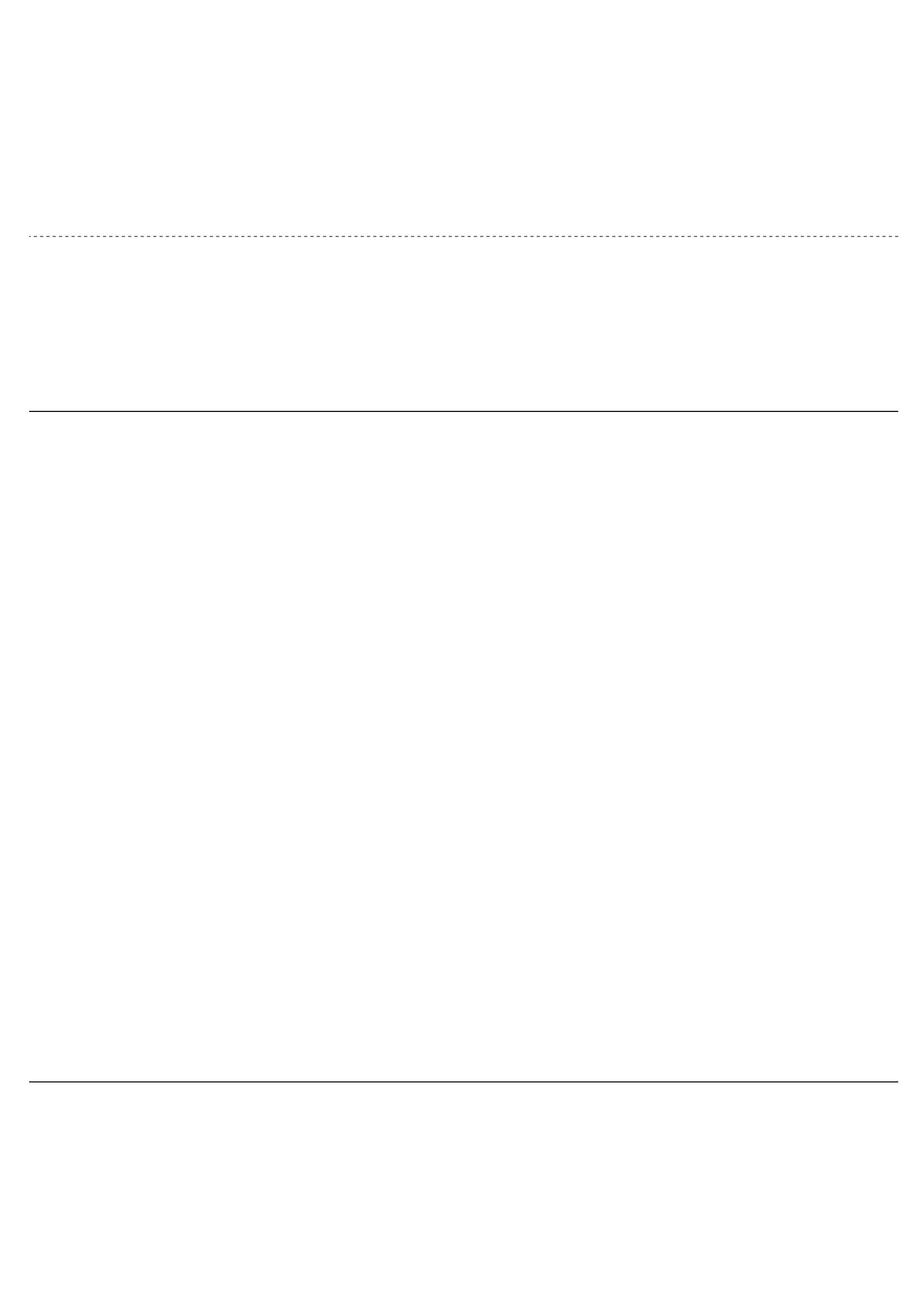


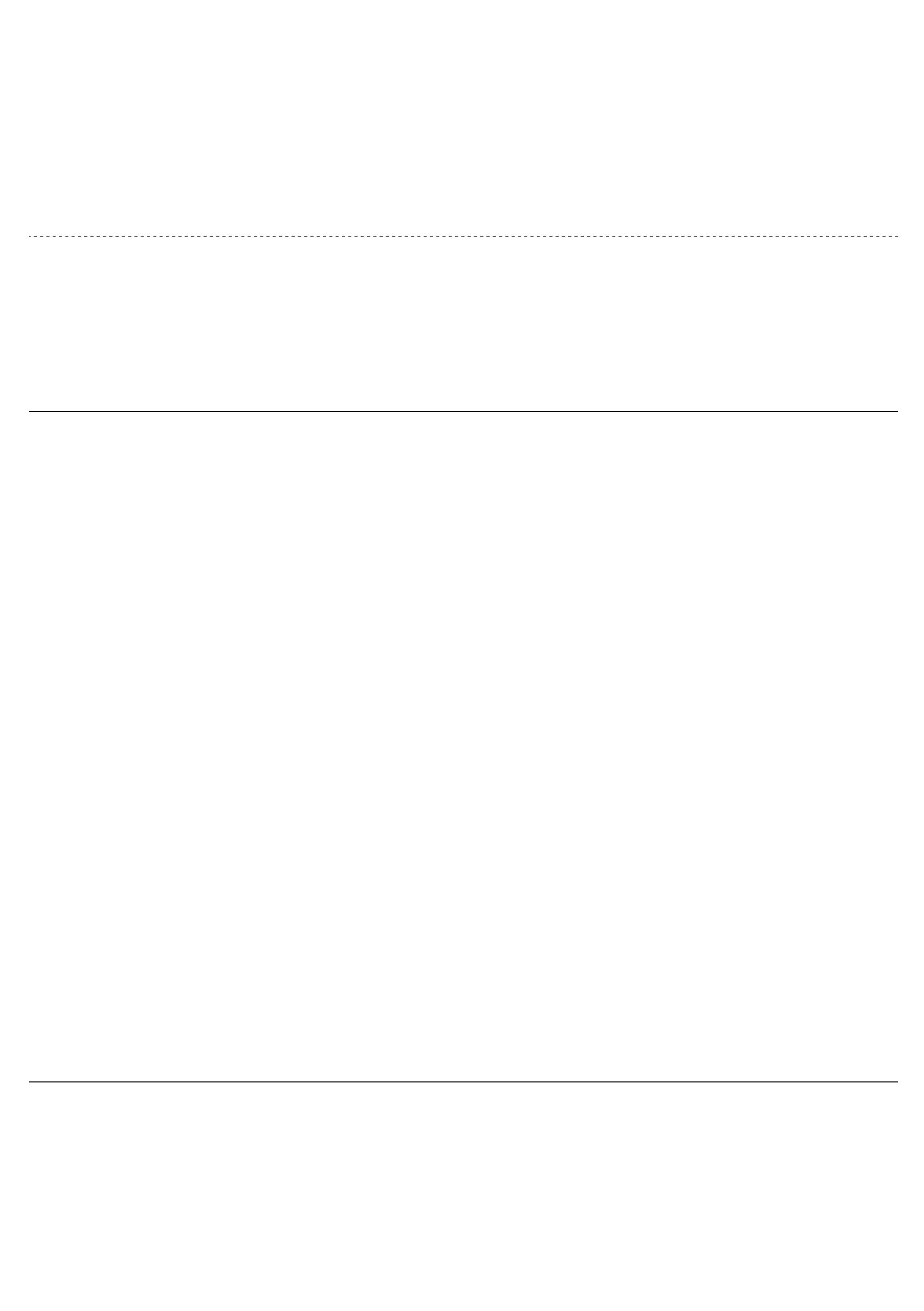
«create»

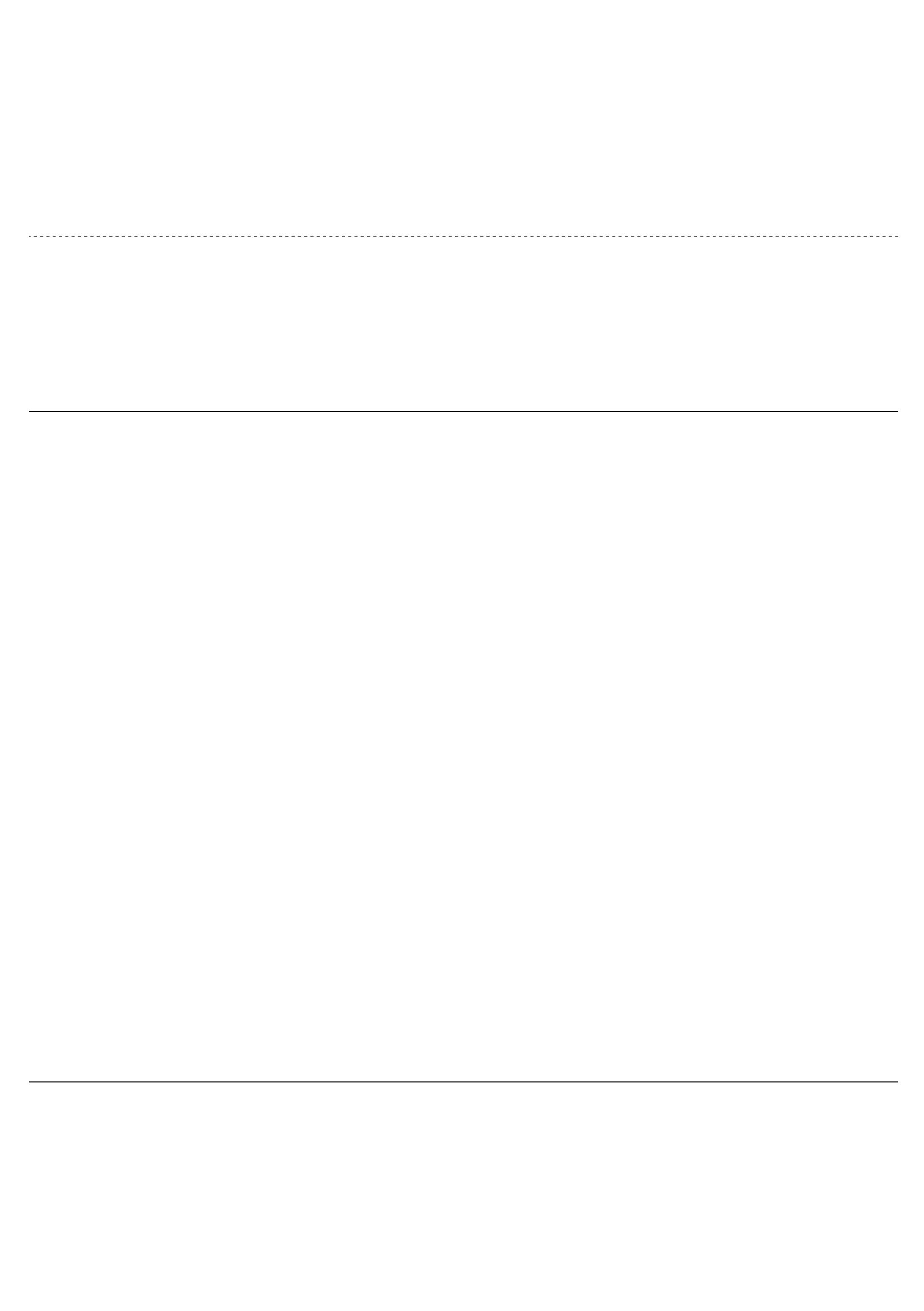
«create»

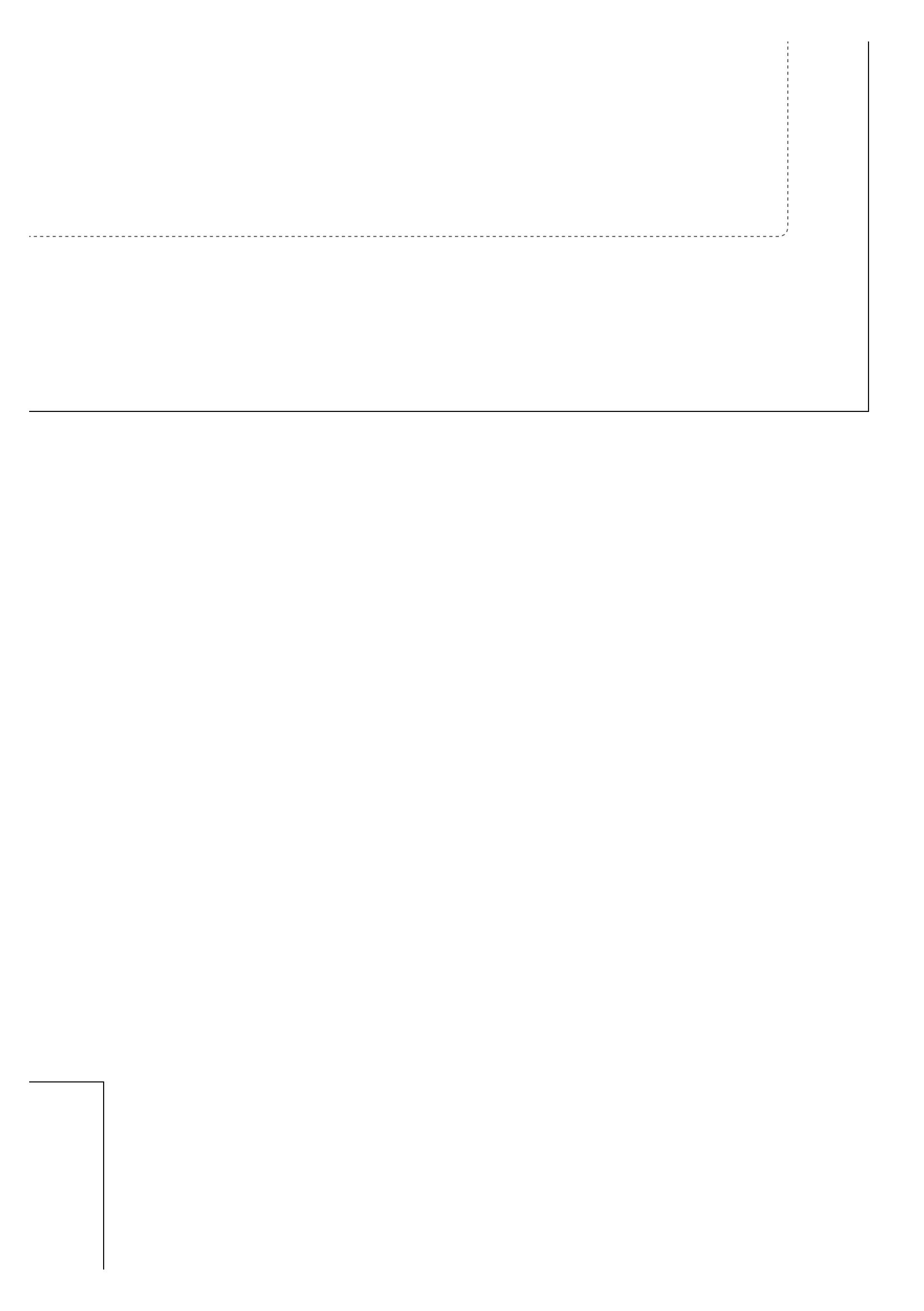
*

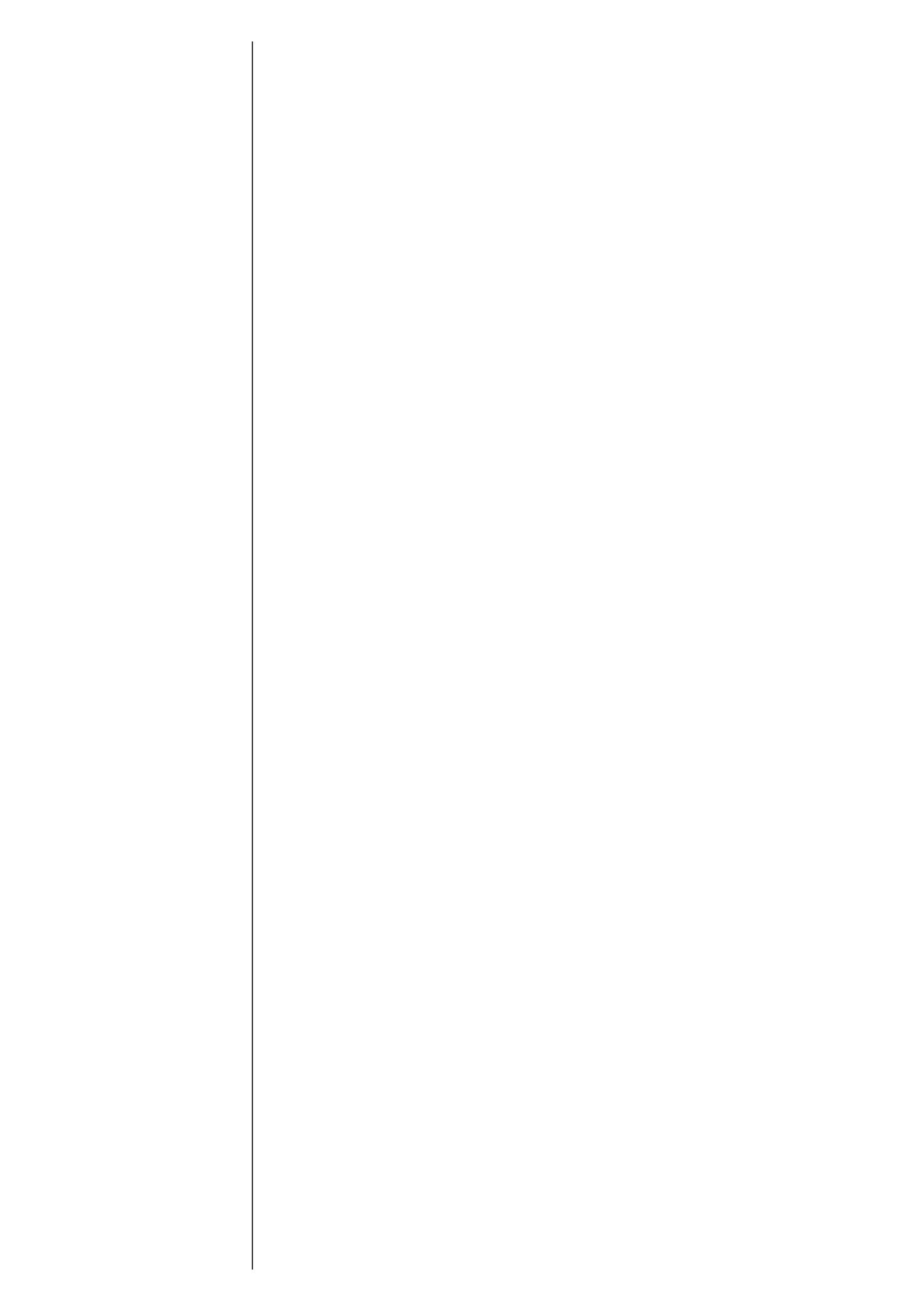
1



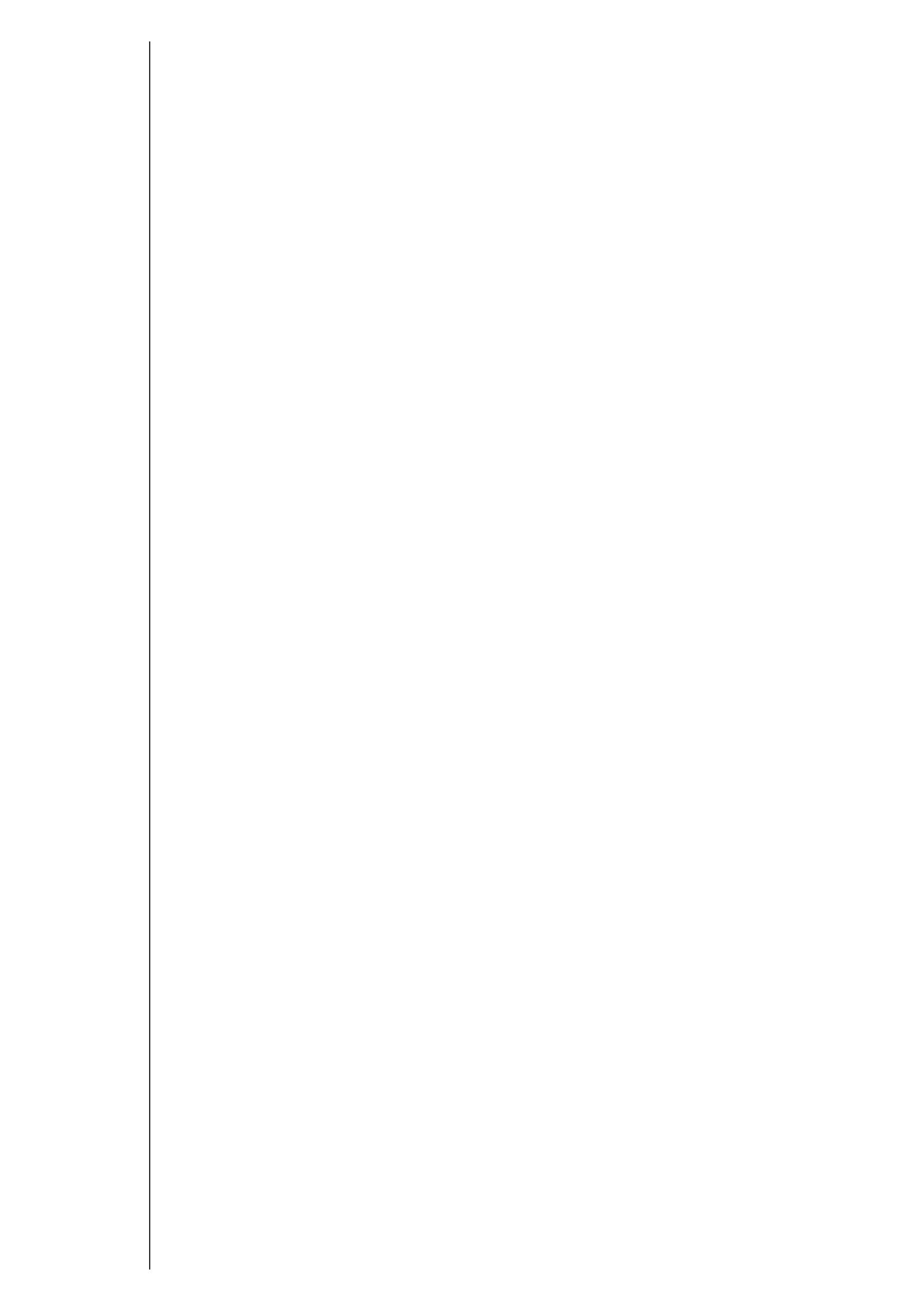









```
+ setGameInfo(String, List<AchievementCard>, Deck<GoldCard>, Deck<ResourceCard>): void
+ getMyName(): String
+ invalidName(): void
+ startingCardChosen(String, CornerCardFace): void
+ startGame(List<GoldCard>, List<ResourceCard>): void
+ askDrawCard(Decks, DeckPosition): void
+ addChatMessage(Message): void
+ notYetGivenCard(Actions): void
+ getActivePlayer(): Player
+ placeCard(String, CornerCardFace, int, int, int): void
+ getPlayers(): List<Player>
+ getPlayerByName(String): Player
+ gameAlreadyStarted(): void
+ getSecretAchievement(): AchievementCard
+ playerDisconnected(String): void
+ reconnect(String): void
+ getTurn(): int
+ setId(String): void
+ emptyDeck(): void
+ ToDoFirst(Actions): void
+ achievementDeckDrawInvalid(): void
+ getIndexofSecretAchievement(): int
+ getChosenHandCard(): Integer
+ chatMessageIsEmpty(): void
+ invalidCard(Actions): void
+ giveOtherPlayerInitialHand(String): void
+ updatePlayerOrder(List<String>): void
+ gameNotYetStarted(): void
+ loadGame(List<AchievementCard>, Deck<GoldCard>, Deck<ResourceCard>): void
+ gameAlreadyFinished(): void
+ tooManyPlayers(): void
+ giveAchievementCards(List<AchievementCard>, List<AchievementCard>): void
+ setReady(): void
+ alreadyDone(Actions): void
+ serverDisconnected(): void
+ setName(String): void
+ askSetName(String): void
+ getCommonAchievements(): List<AchievementCard>
+ getGameState(): GameState
+ giveStartingCard(Card): void
+ newPlayer(List<String>): void
+ setChosenHandCard(Integer): void
+ colorNotYetSet(): void
+ notYourTurn(): void
+ cardNotPlaceable(): void
+ joinSavedGame(String): void
+ newTurn(String, int): void
+ getMyColor(): Color
+ getAvailableColors(): List<Color>
+ idNotInGame(): void
+ isSavedGame(): boolean
+ joinServer(String, int): void
```

EXCEPTION

PlayerNotInAnyServerConnectionHandlerException

+ PlayerNotInAnyServerConnectionHandlerException():

PlayerNotFoundByNameException

+ PlayerNotFoundByNameException(String):

IncorrectDeckPositionException

+ IncorrectDeckPositionException(String):

TooManyElementsException

+ TooManyElementsException(String):

TooFewElementsException

+ TooFewElementsException(String):

TooManyPlayersException

+ TooManyPlayersException(String):

AlreadyFinishedException

+ AlreadyFinishedException(String):

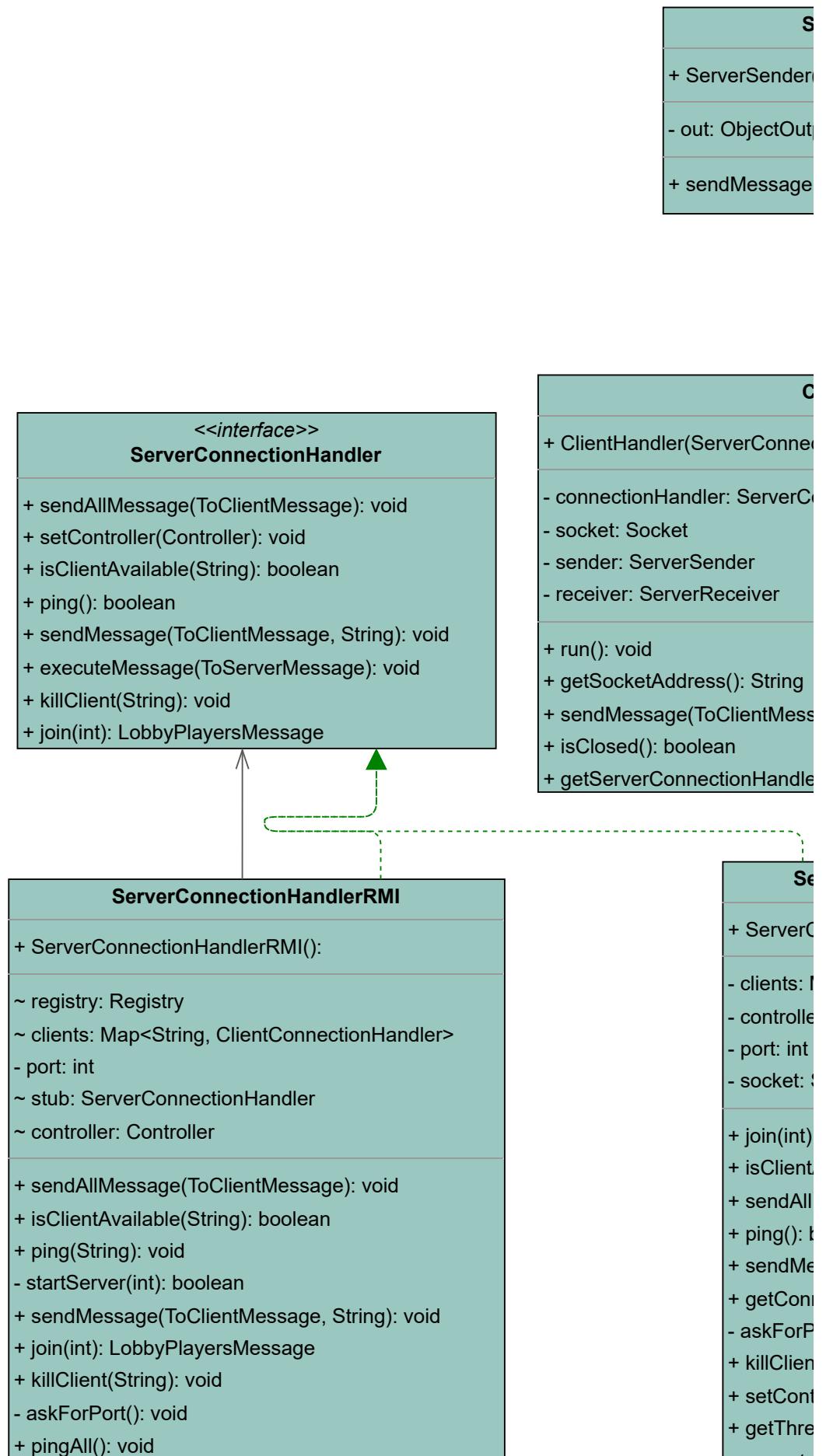
AlreadyStartedException

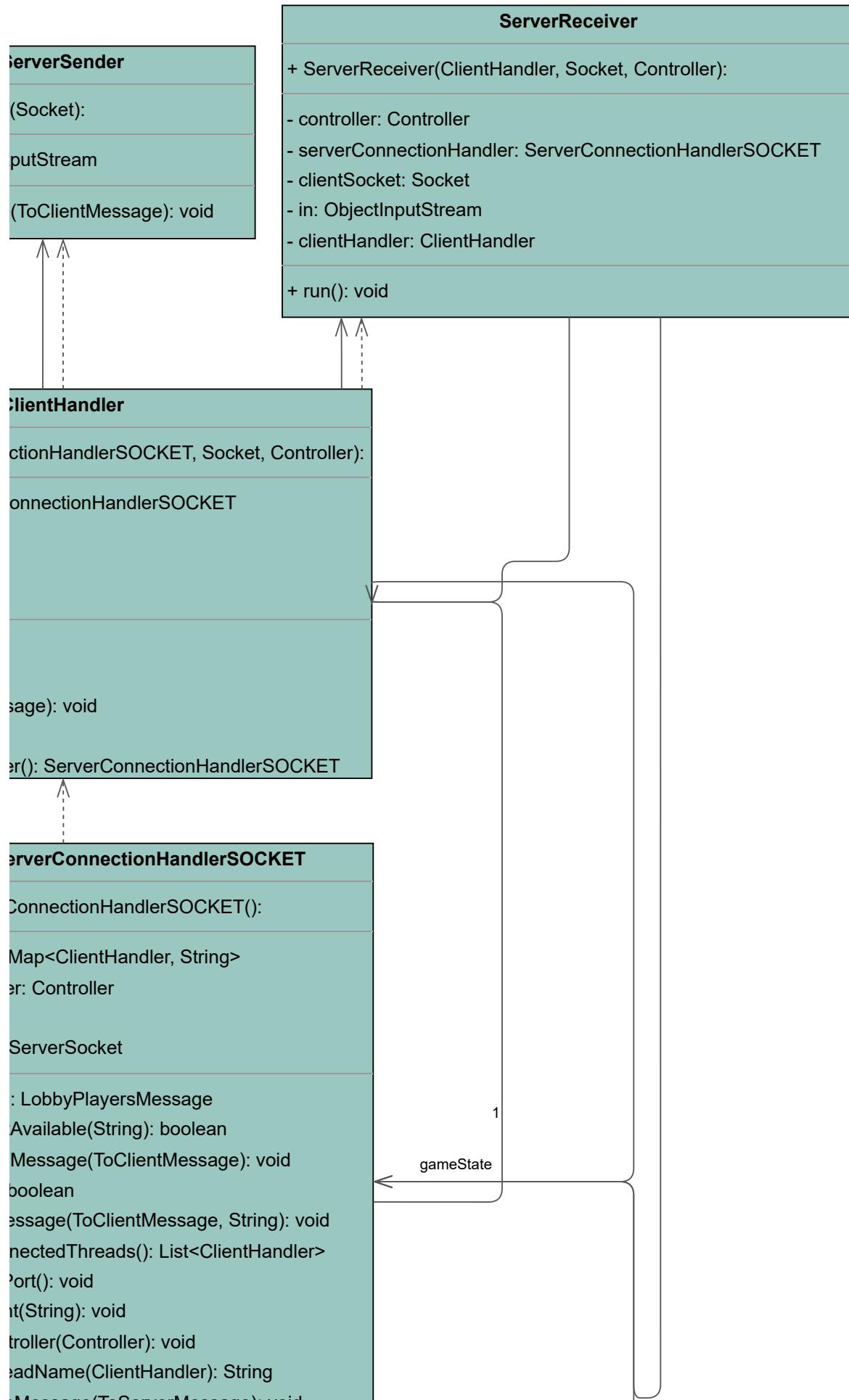
+ AlreadyStartedException(String):

NotYetStartedException

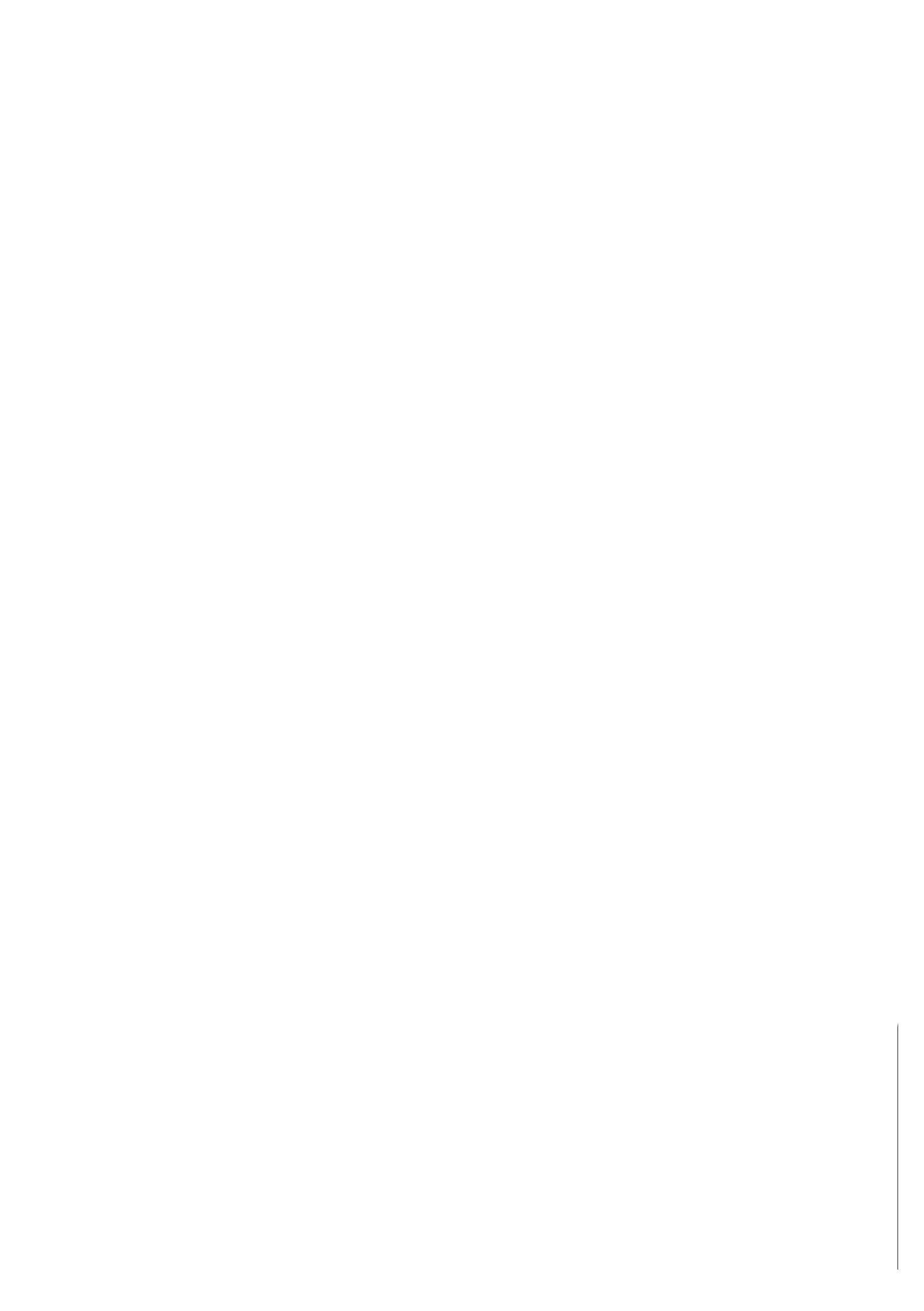
+ NotYetStartedException(String):

SERVER CONNECTION









<<interface>>
ServerState

+ removePlayer(Player): void
+ reactToDisconnection(String): void
+ isInSavedGameLobby(): boolean
+ addSavedPlayer(String, String): void
+ drawCard(Player, DeckPosition, Decks): void
+ addPlayer(String, String): void
+ playCard(Player, int, int, int, CornerCardFace): void
+ reconnect(String, String, Player, Map<Player, List<AchievementCard>>



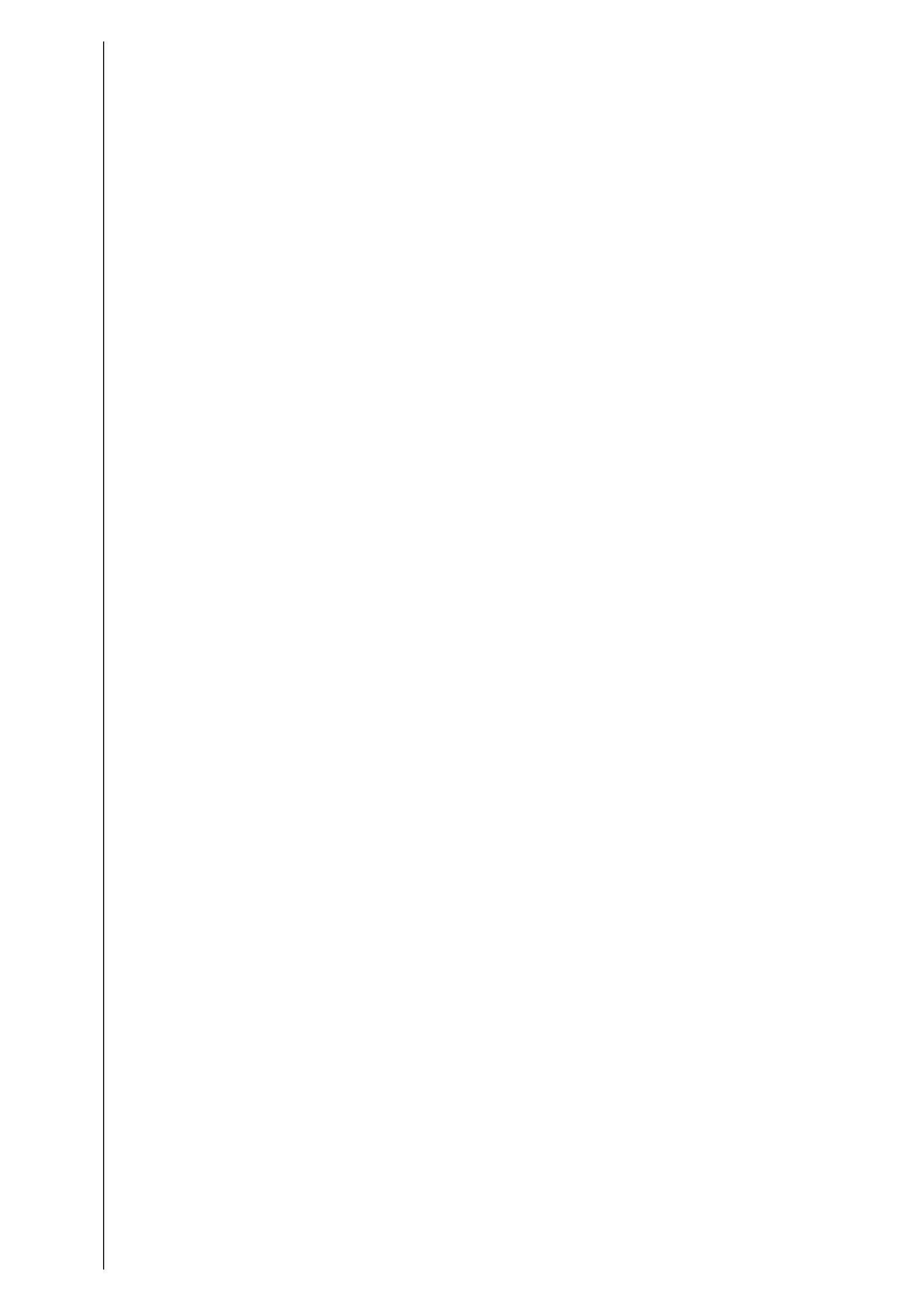
Dis
+ DisconnectionTimer(Cor
~ t: Timer

«create»

sconnectionTimer
ntroller, GeneralServerConnectionHandler,

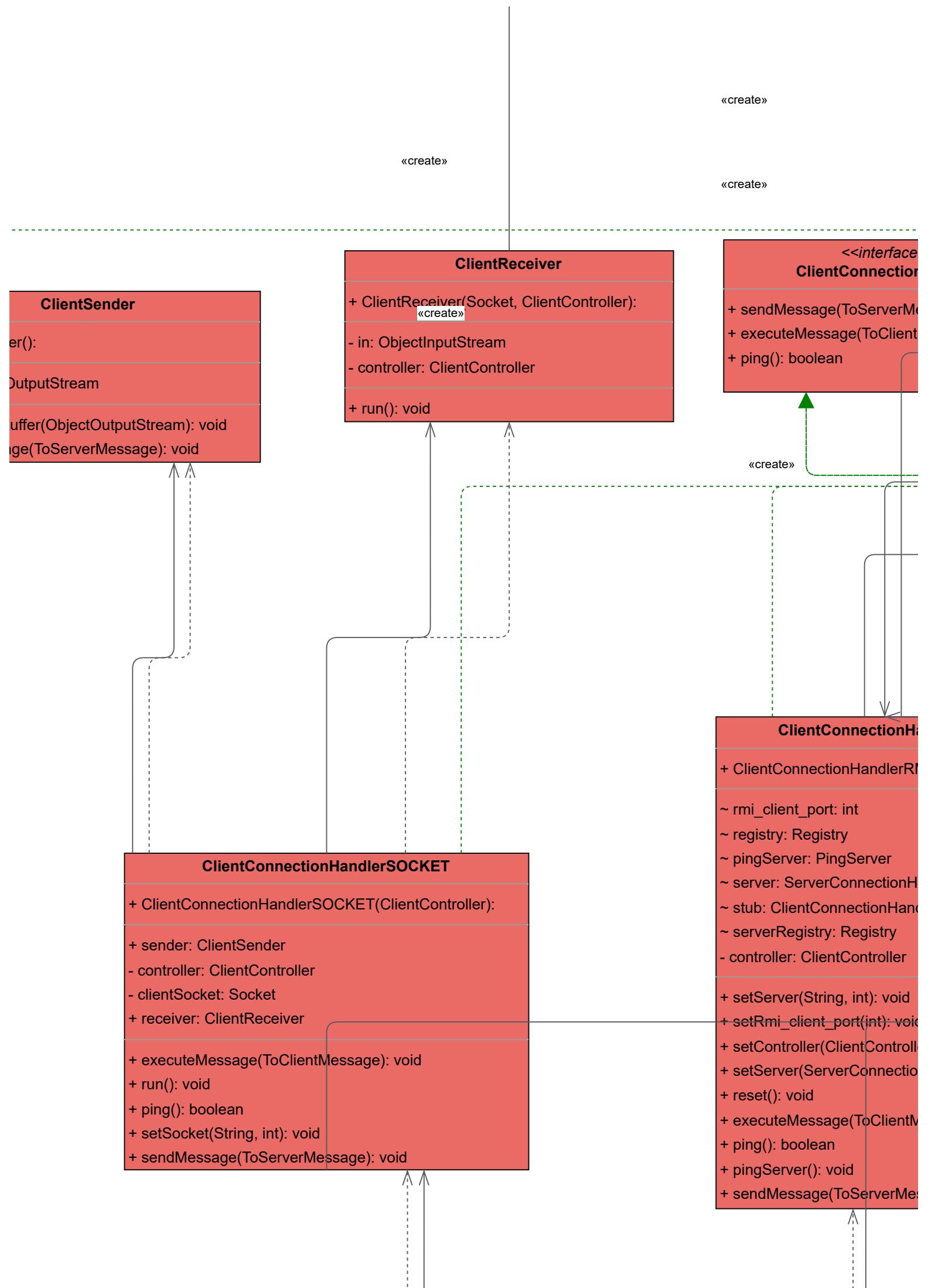


0

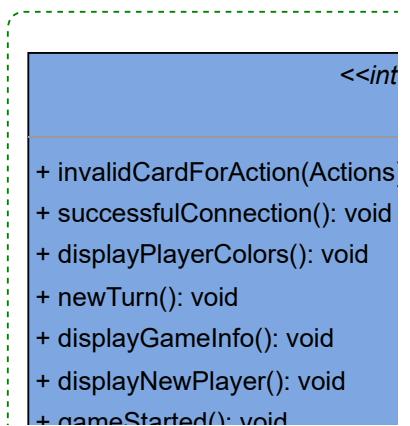
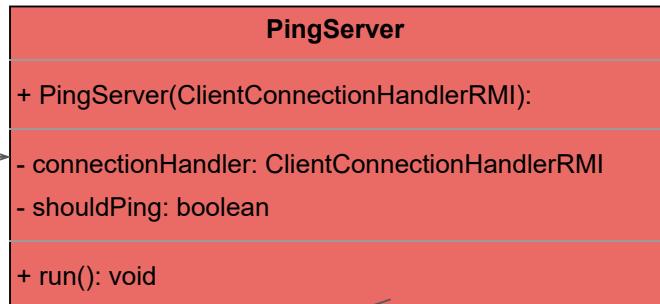
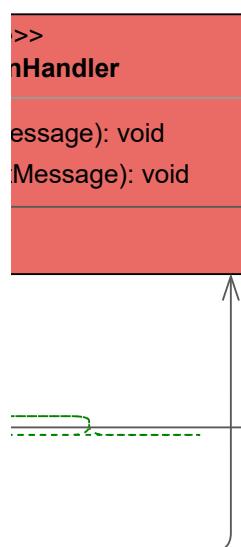


```
classDiagram\n    class Client {\n        +ClientSender()\n        -out: ObjectConsumer\n        +setOutputBuffer()\n        +sendMessage()\n    }
```

+ ClientSender()
- out: ObjectConsumer
+ setOutputBuffer()
+ sendMessage()



+ setSecretCard(String): void
+ getBoardCards(Decks): List<Card>
+ updatePlayerReady(boolean, String): void



«create»

OnBoardCard
+ OnBoardCard(Image, int, int, MainBoardSceneController):
+ BOTTOM_RIGHT: Button
+ onBoardCards: List<OnBoardCard>
+ y: int
+ image: ImageView
+ BOTTOM_LEFT: Button
+ TOP_LEFT: Button
+ TOP_RIGHT: Button
+ x: int
- sceneController: MainBoardSceneController
+ place(Group): void
~ playCardBottomLeft(): void
~ playCardBottomRight(): void
~ playCardTopLeft(): void
~ playCardTopRight(): void

«create»

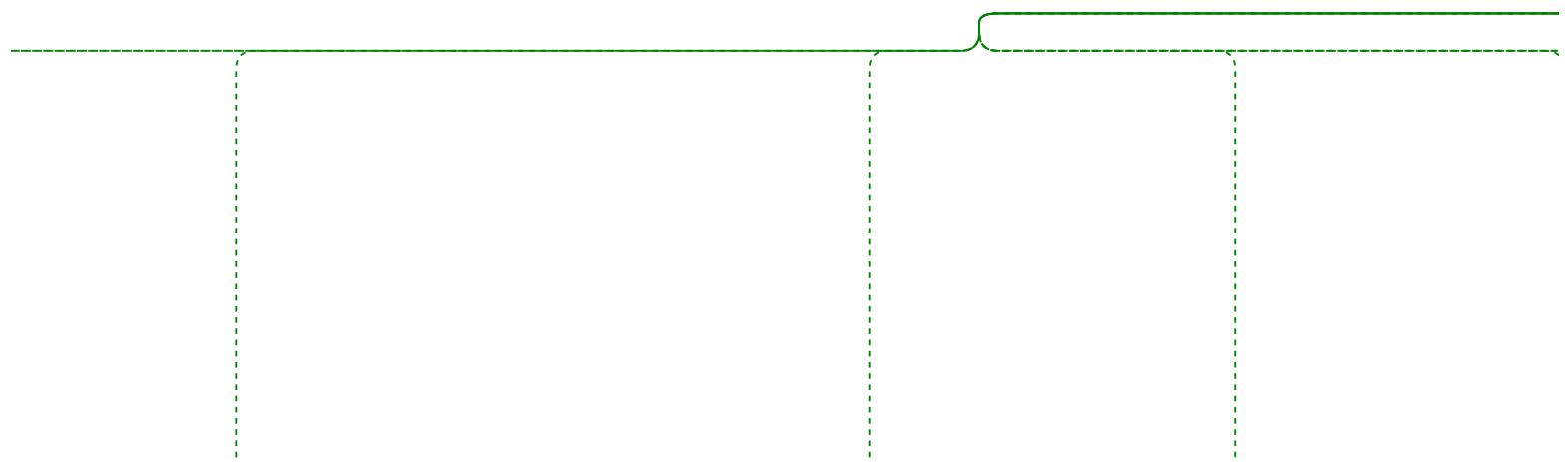
OtherPlayerTab
+ OtherPlayerTab(String):
+ scrollPane: ScrollPane
+ group: Group
+ onBoardCardList: List<OnBoardCard>
+ tab: Tab
...

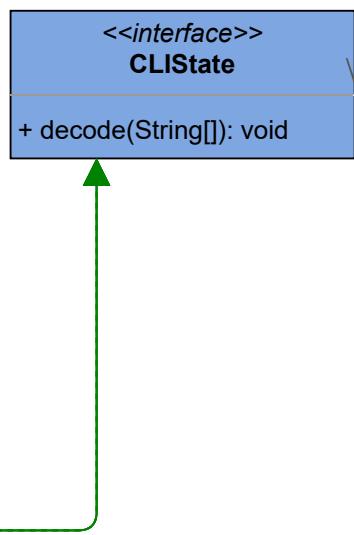
erface>>

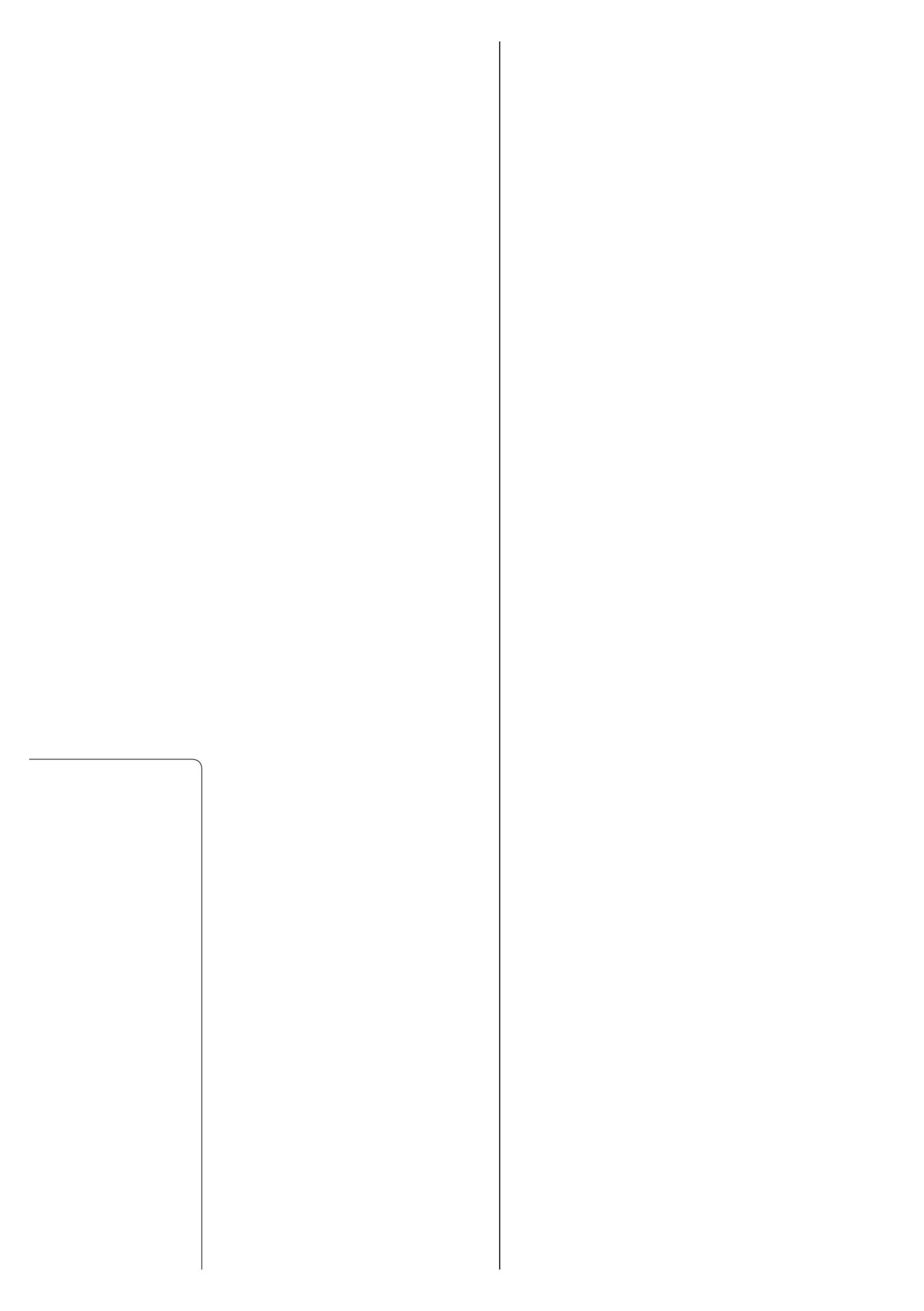
UI

): void

«create»







ENU

+

+

+

+

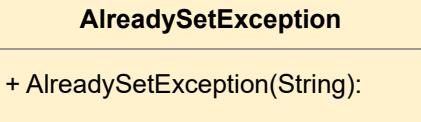
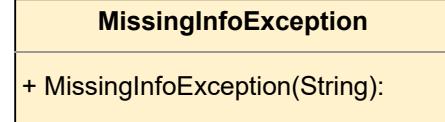
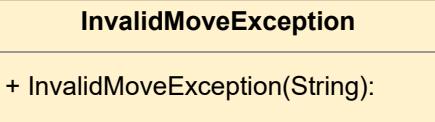
+

+

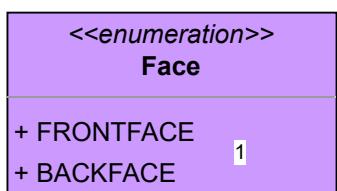
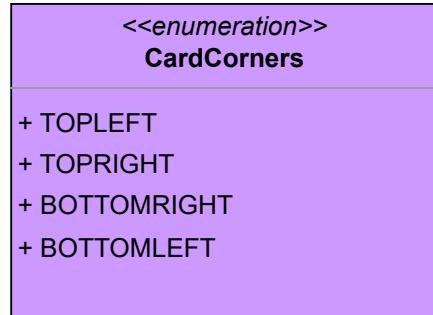
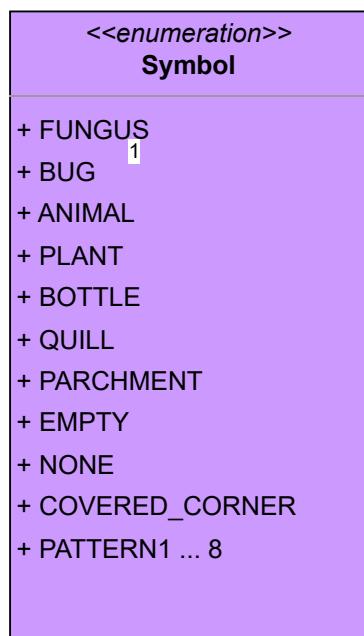
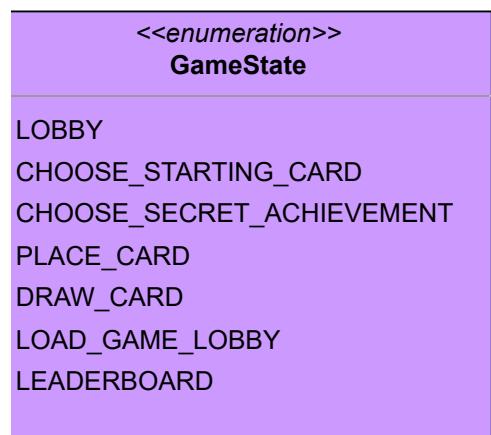
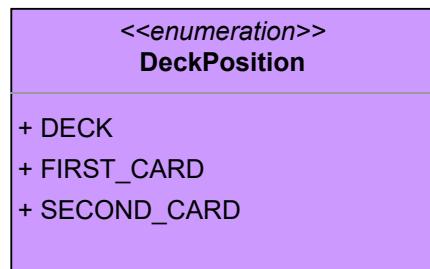
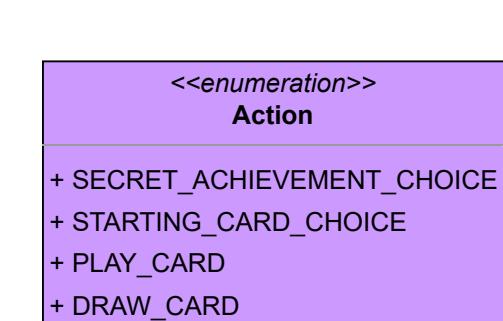
+

+

+



IM



1


```
+ ping(): boolean  
+ setController(Controller): void  
+ executeMessage(ToServerMessage): void
```

```
+ execute  
+ getCont  
+ run(): v  
- startSer
```



GeneralServerConnectionHandler

```
+ GeneralServerConnectionHandler():  
  
- disconnectedPlayerIds: List<String>  
- serverConnectionHandlerRMI: ServerConnectionHandlerRMI  
- serverConnectionHandlerSOCKET: ServerConnectionHandlerSOCKET  
- playerID: Map<String, String>  
  
+ removePlayerByName(String): void  
+ getServerConnectionHandlerRMI(): ServerConnectionHandlerRMI  
+ clear(): void  
+ setController(Controller): void  
+ changePlayerId(String, String): void  
+ getPlayerNameByID(String): String  
+ sendAllMessage(ToClientMessage): void  
+ setOnline(String): void  
+ getIdByName(String): String  
+ sendMessage(ToClientMessage, String): void  
+ isIdConnectedToName(String): boolean  
+ start(): void  
+ isNameConnectedToId(String): Boolean  
+ setOffline(String): void  
+ isInDisconnectedList(String): boolean  
+ getServerConnectionHandler(String): ServerConnectionHandler  
+ addPlayerByID(String, String): void  
+ getServerConnectionHandler(): ServerConnectionHandler  
+ getDisconnectedList(): List<String>
```



PingPong

```
+ PingPong(GeneralServerConnectionHandler):  
  
- connectionHandler: GeneralServerConnectionHandler  
  
+ run(): void
```

```
>Message(ToServerMessage): void  
troller(): Controller  
oid  
ver(int): boolean
```

```
erRMI  
HandlerSOCKET
```

```
HandlerRMI
```

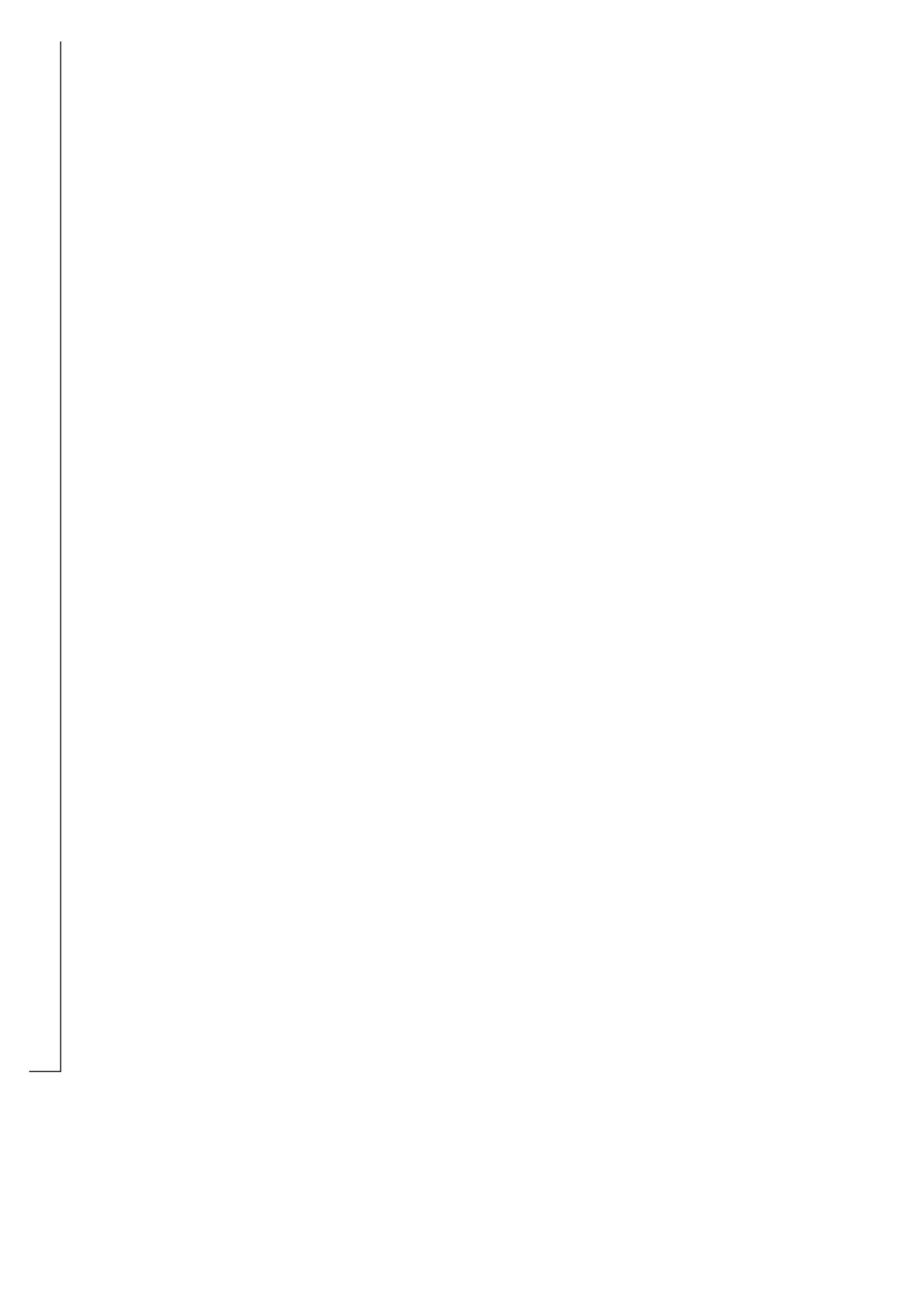
```
onHandler
```

```
dler
```

```
Handler
```

1

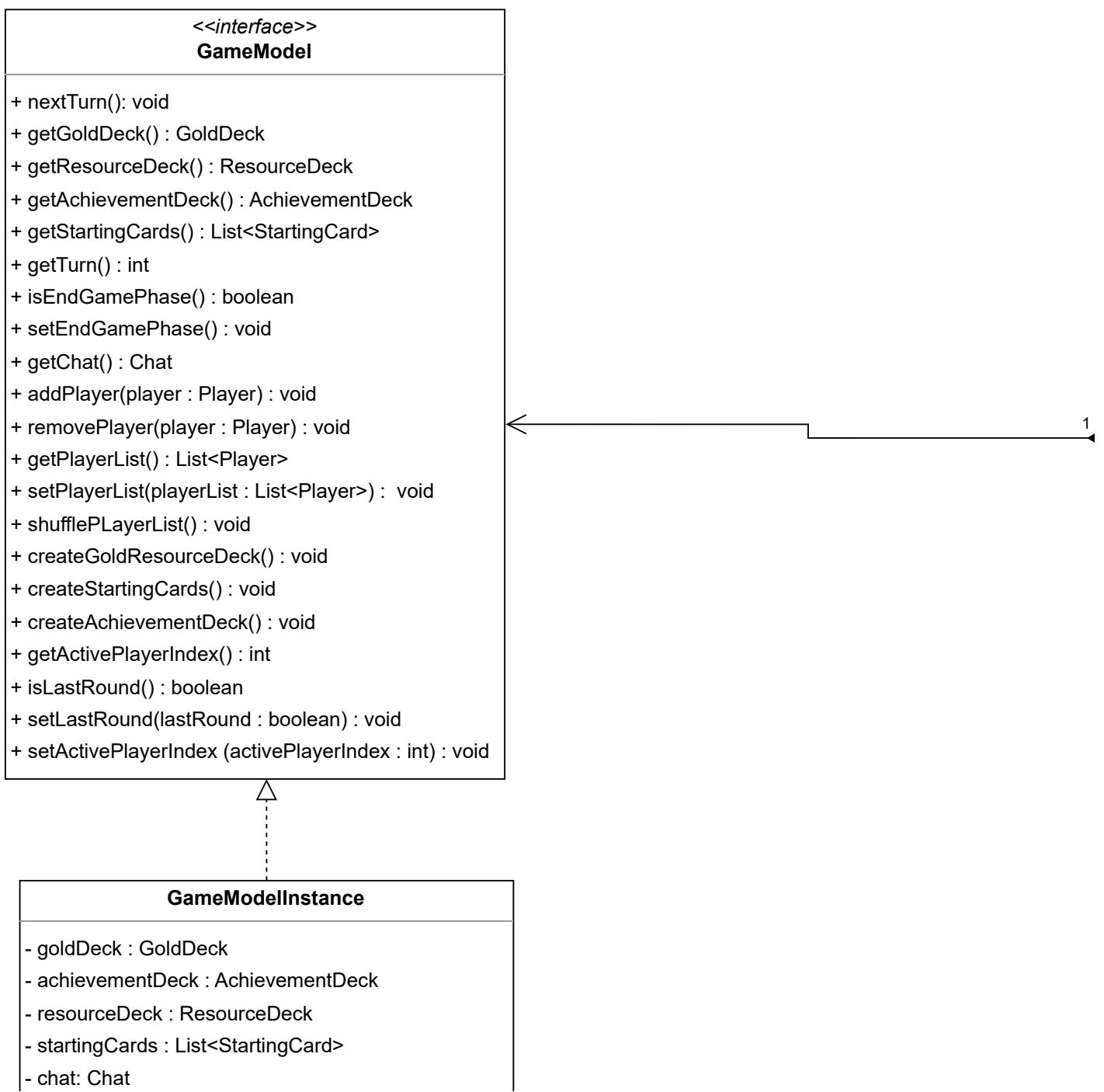
1



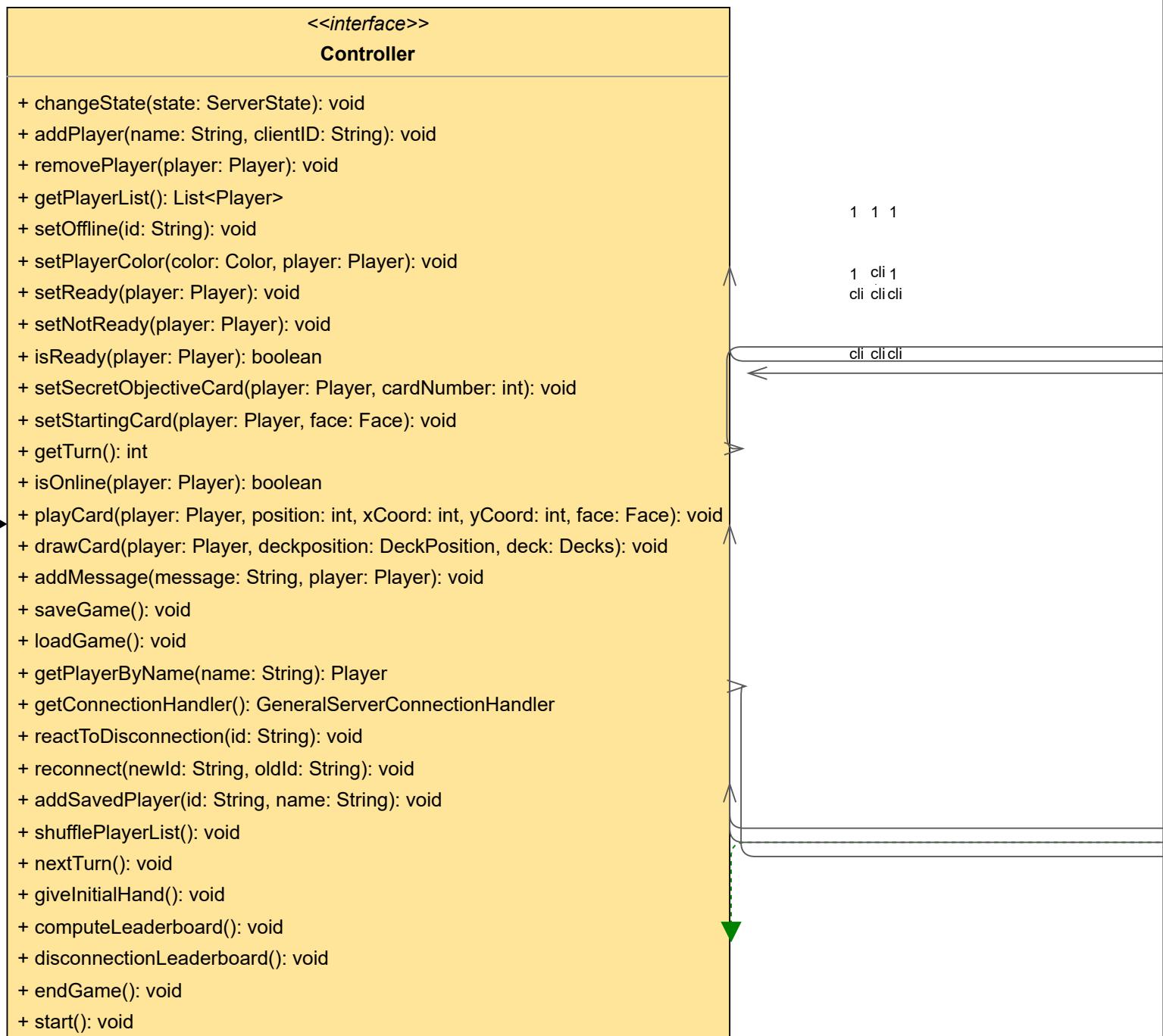
«create»

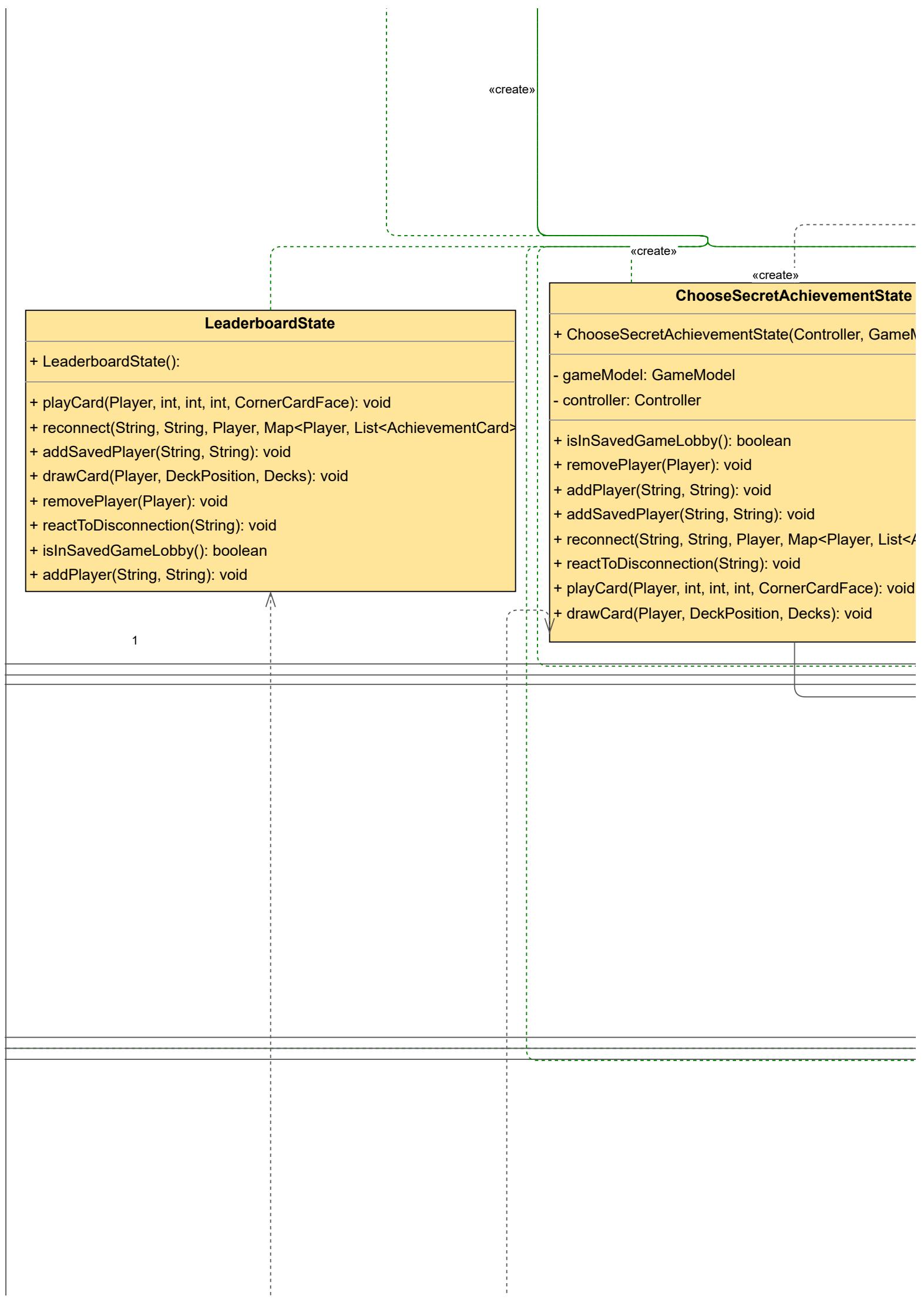
1

package



«create»





Model):

AchievementCard>

LobbyState

+ LobbyState(GameModel, GeneralServerConnectionHandler, Controller)

- connectionHandler: GeneralServerConnectionHandler

- gameModel: GameModel

- controller: Controller

+ reactToDisconnection(String): void

+ drawCard(Player, DeckPosition, Decks): void

+ removePlayer(Player): void

+ addPlayer(String, String): void

↳ + isInSavedGameLobby(): boolean

+ playCard(Player, int, int, int, CornerCardFace): void

+ addSavedPlayer(String, String): void

+ reconnect(String, String, Player, Map<Player, List<AchievementCard>>): void

ChooseStart

+ ChooseStartingCardState(Controller)

- gameModel: GameModel

- controller: Controller

+ addSavedPlayer(String, String): void

+ playCard(Player, int, int, int, CornerCardFace): void

+ reconnect(String, String, Player, Map<Player, List<AchievementCard>>): void

+ removePlayer(Player): void

+ isInSavedGameLobby(): boolean

+ drawCard(Player, DeckPosition, Decks): void

+ reactToDisconnection(String): void

+ addPlayer(String, String): void

ingCardState
ller, GameModel):

void
erCardFace): void
Map<Player, List<AchievementCard>
Decks): void
d

DrawCardState

+ DrawCardState(Controller, GameModel):

- controller: Controller
- gameModel: GameModel

+ reactToDisconnection(String): void
+ isInSavedGameLobby(): boolean
+ addPlayer(String, String): void
+ playCard(Player, int, int, int, CornerCardFace): void
+ removePlayer(Player): void
+ drawCard(Player, DeckPosition, Decks): void
+ reconnect(String, String, Player, Map<Player, List<AchievementCard>
+ addSavedPlayer(String, String): void

PlaceCardState

+ PlaceCardState(Controller, GameModel):

- gameModel: GameModel
- controller: Controller

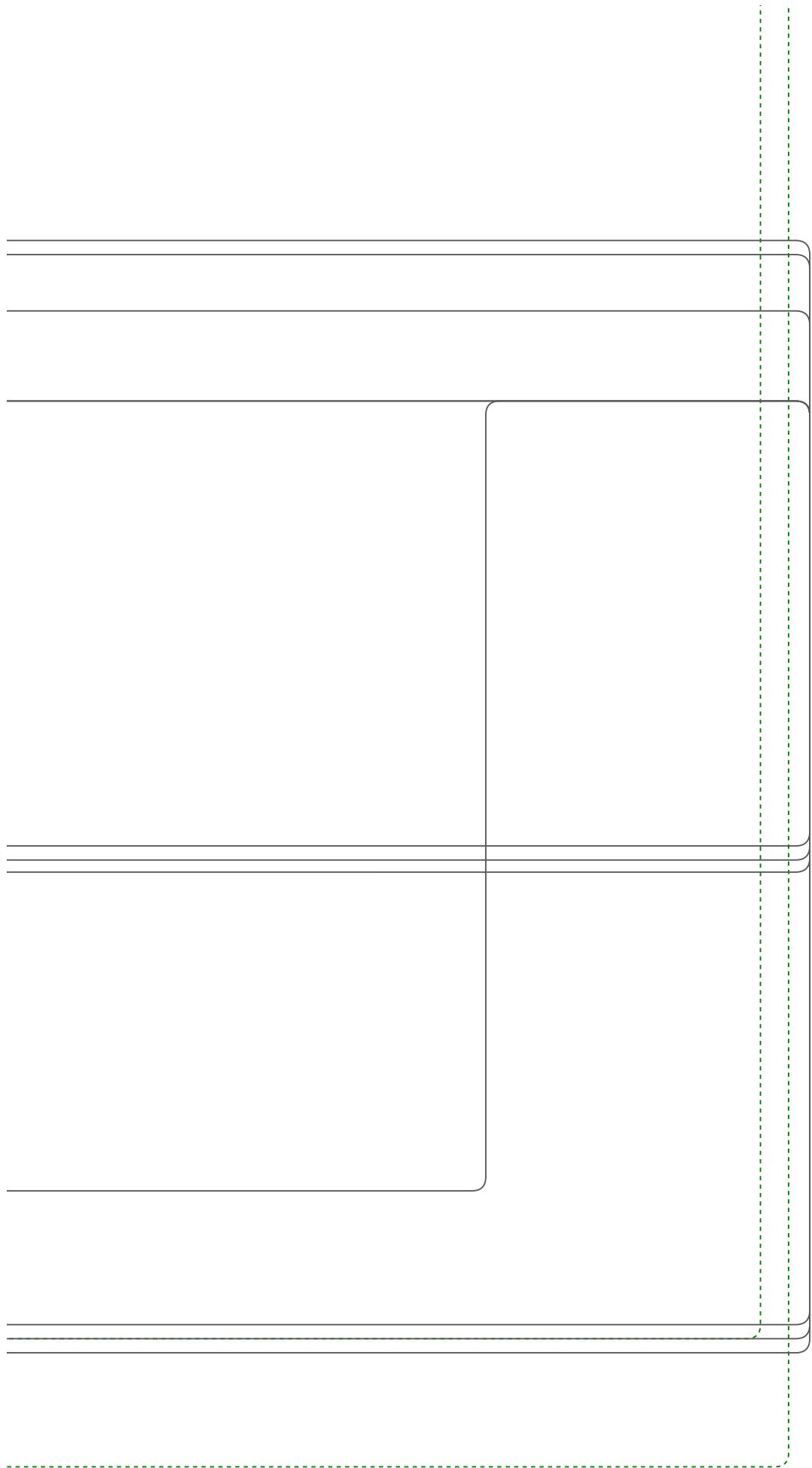
+ addSavedPlayer(String, String): void
+ removePlayer(Player): void
+ reactToDisconnection(String): void
+ drawCard(Player, DeckPosition, Decks): void
+ playCard(Player, int, int, int, CornerCardFace): void
+ reconnect(String, String, Player, Map<Player, List<AchievementCard>
+ isInSavedGameLobby(): boolean
+ addPlayer(String, String): void

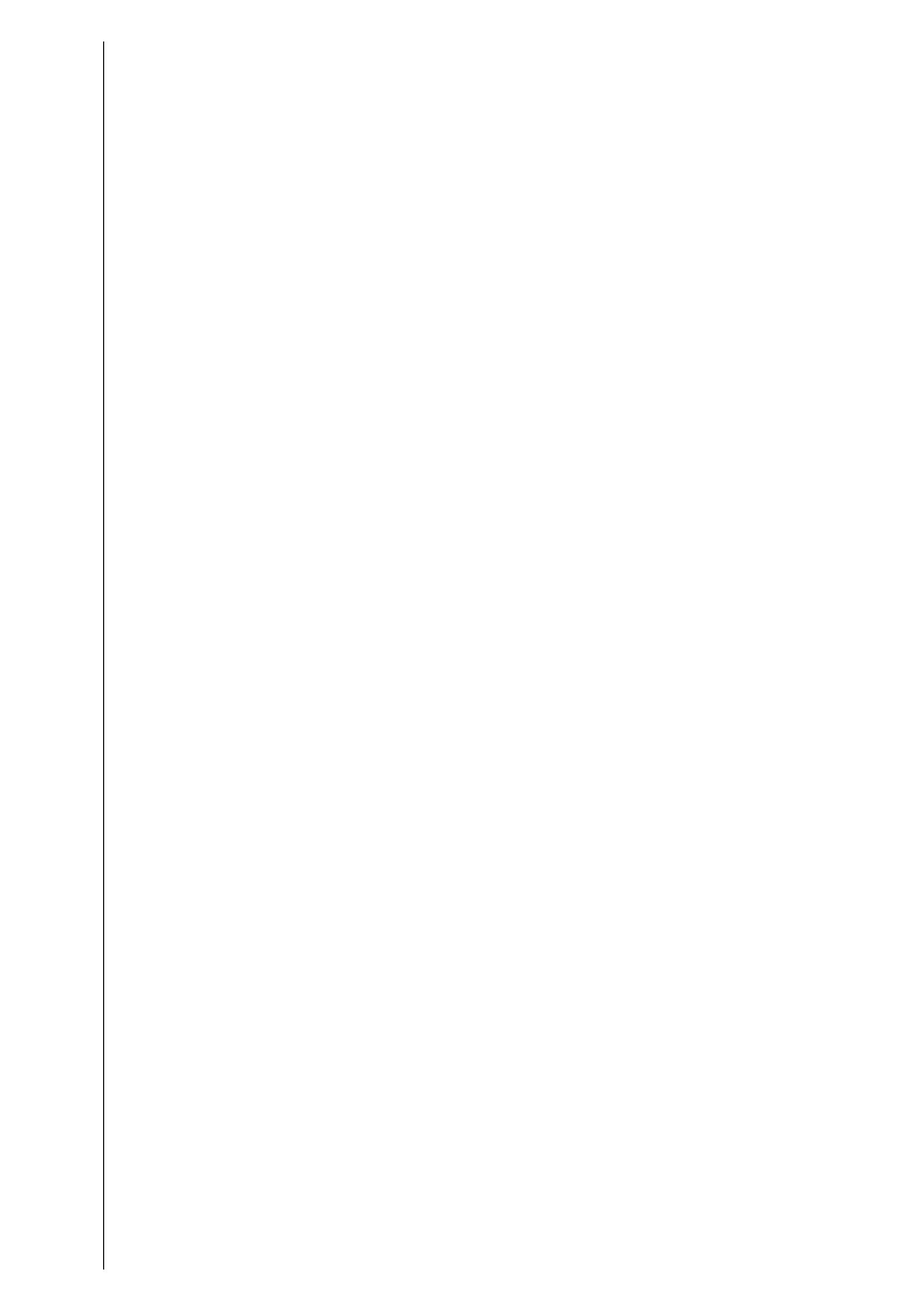
LoadGameLobbyState

+ LoadGameLobbyState(GeneralServerConnectionHandler, GameModel,

- connectionHandler: GeneralServerConnectionHandler
- gameModel: GameModel
- controller: Controller

+ addSavedPlayer(String, String): void
+ playCard(Player, int, int, int, CornerCardFace): void





Player	
+ Player(String, int, int, boolean, Color,	
+ Player(String):	
- achievementPoints: int	
- handSize: int	
- color: Color	
- ready: boolean	
- manuscriptPoints: int	
- manuscript: Manuscript	
- name: String	
- active: boolean	
+ setReady(boolean): void	
+ getManuscript(): Manuscript	
+ setHandSize(int): void	
+ getName(): String	
+ isReady(): boolean	
+ setActive(boolean): void	
+ isActive(): boolean	
+ getColor(): Color	
+ setAchievementPoints(int): void	
+ getHandSize(): int	
+ getPoints(): int	
+ setColor(Color): void	
+ getAchievementPoints(): int	
+ addManuscriptPoints(int): void	
+ addCardToManuscript(int, int, CornerCardFace): void	
+ initializeManuscript(CornerCardFace)	

Player

+ Player(String, int, int, boolean, Color,
+ Player(String):

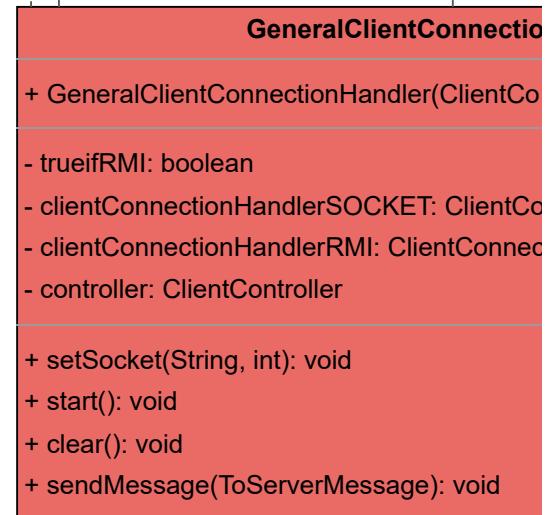
- achievementPoints: int
- handSize: int
- color: Color
- ready: boolean
- manuscriptPoints: int
- manuscript: Manuscript
- name: String
- active: boolean

+ setReady(boolean): void
+ getManuscript(): Manuscript
+ setHandSize(int): void
+ getName(): String
+ isReady(): boolean
+ setActive(boolean): void
+ isActive(): boolean
+ getColor(): Color
+ setAchievementPoints(int): void
+ getHandSize(): int
+ getPoints(): int
+ setColor(Color): void
+ getAchievementPoints(): int
+ addManuscriptPoints(int): void
+ addCardToManuscript(int, int, CornerCardFace): void
+ initializeManuscript(CornerCardFace)



Manuscript):

CardFace, int): void
(): void



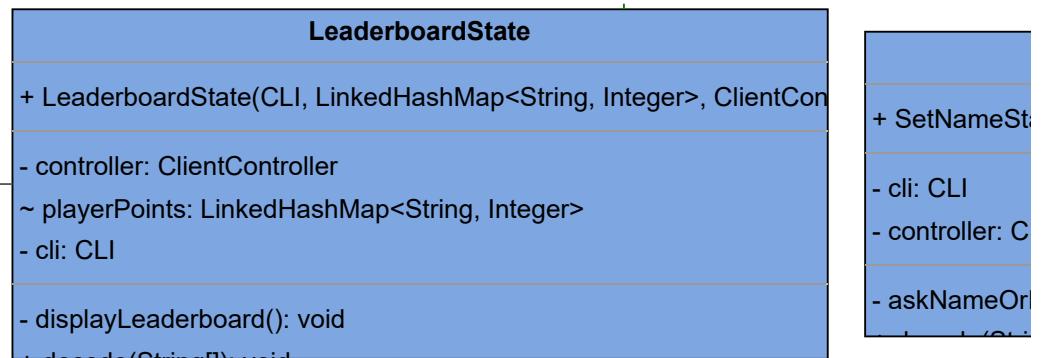
ConnectionHandler

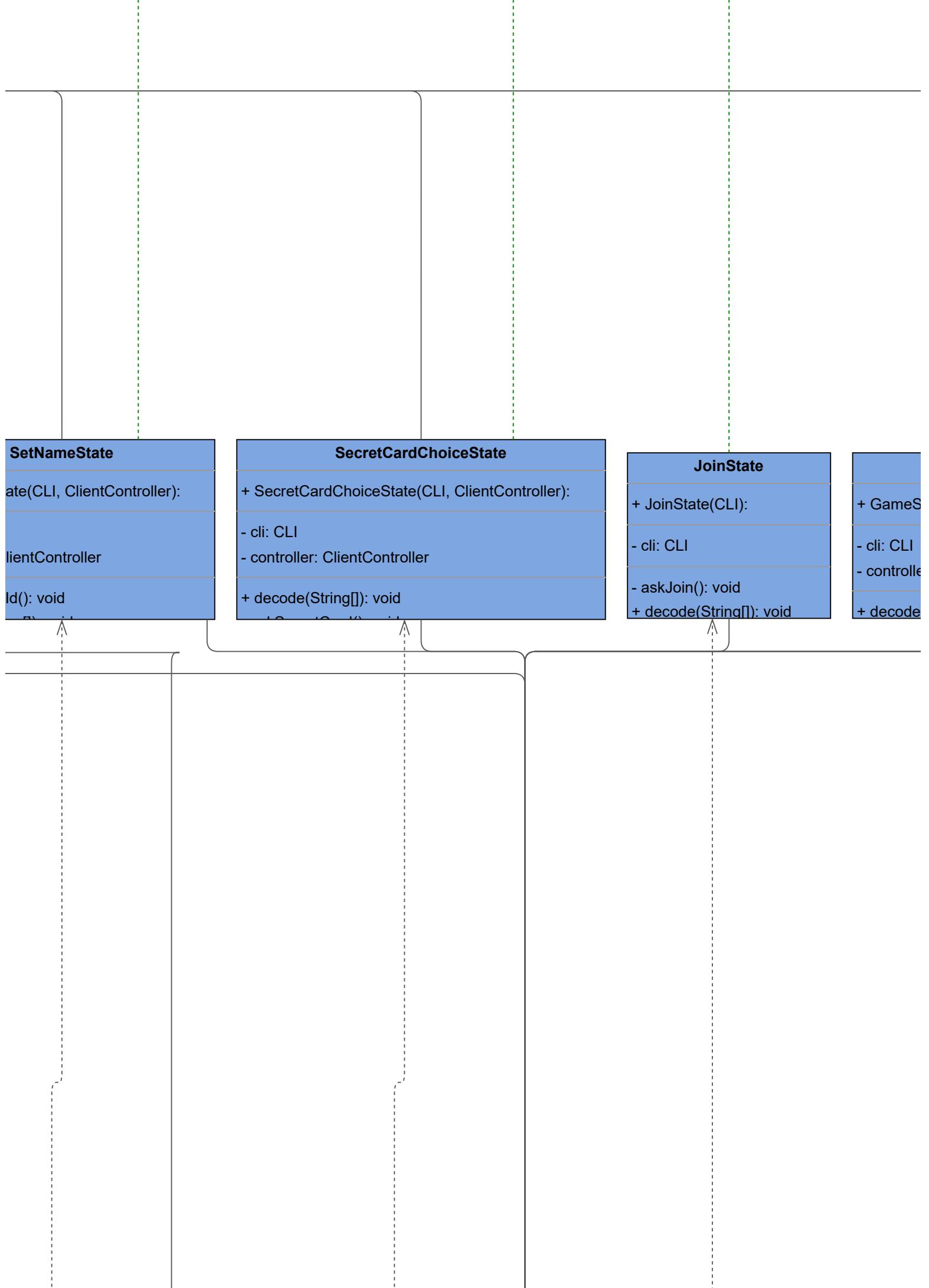
 + controller, boolean):

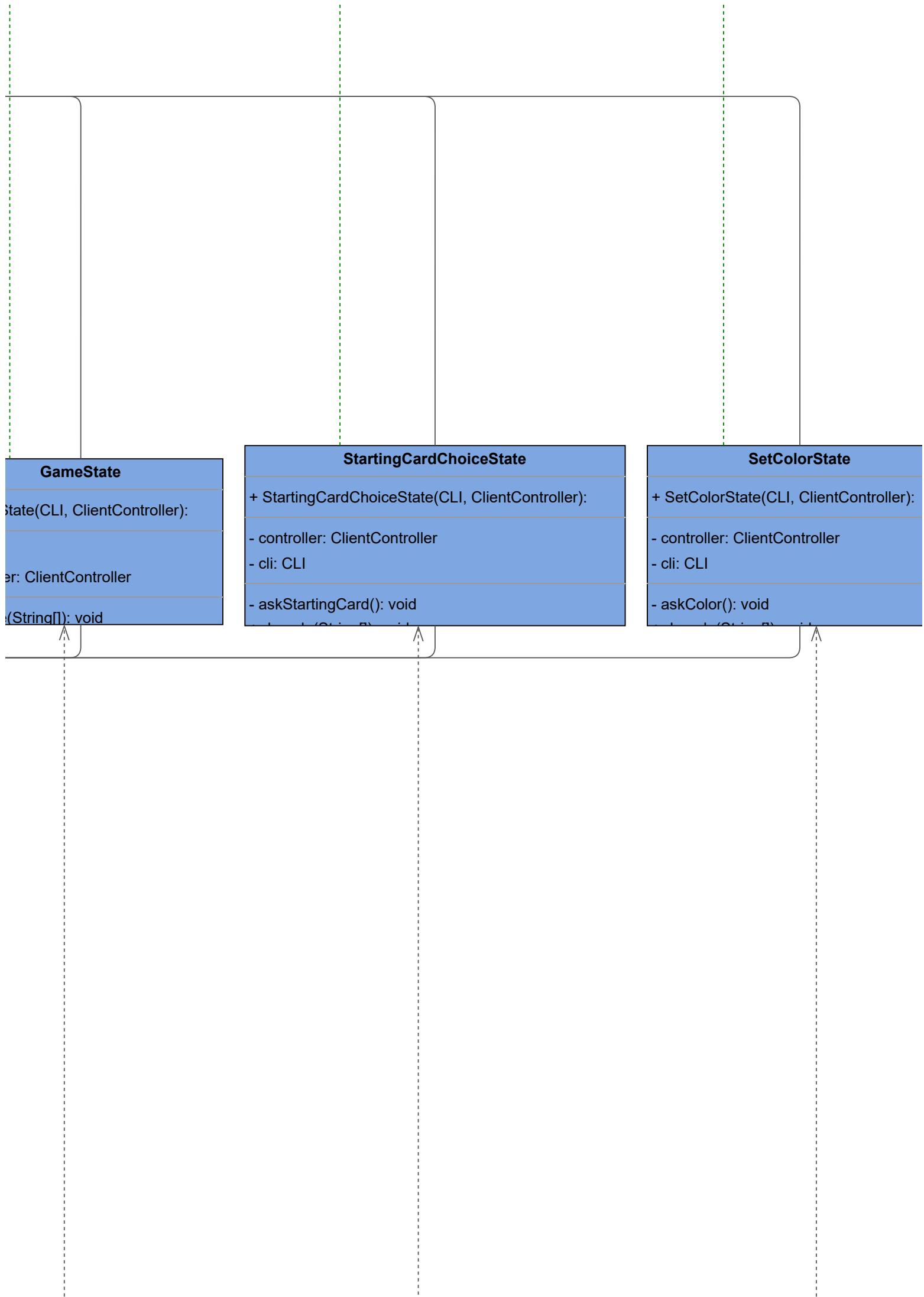
 + connectionHandlerSOCKET
 + connectionHandlerRMI

+ gameStarted(): void
+ unavailableColor(): void
+ otherPlayerReconnected(String): void
+ colorChangeFailed(): void
+ displayBoardCards(): void
+ secretAchievementChosen(String): void
+ nameChanged(): void
+ updateReady(String, boolean): void
+ alreadyDone(Actions): void
+ chooseStartingCardFace(CardFace): void
+ colorChanged(): void
+ chat(Message): void
+ connectionFailed(): void
+ chooseSecretAchievement(List<String>): void
+ idNotInGame(): void
+ gameAlreadyStarted(): void
+ needNameOrColor(): void
+ otherPlayerInitialHand(String): void
+ startingCardChosen(String): void
+ needColor(): void
+ clear(): void
+ displayNewCardInHand(): void
+ chatMessagesIsEmpty(): void
+ gameAlreadyFinished(): void
+ nameChangeFailed(): void
+ notYetGivenCard(Actions): void
+ notYourTurn(): void
+ displayCommonAchievement(): void
+ cantDrawAchievementCards(): void
+ displayPlayerOrder(): void
+ doFirst(Actions): void
+ playerRemoved(String): void
+ cardNotPlaceable(): void
+ deckIsEmpty(): void
+ cardPlaced(String, CornerCard): void
+ otherPlayerDraw(String, Deck): void
+ needName(): void
+ displayPlayerPoints(String): void
+ playerDisconnected(String): void
+ displayId(): void
+ tooManyPlayers(): void
+ invalidColor(): void
+ gameNotYetStarted(): void
+ playerAlreadyPlaying(): void
+ endGameStarted(): void
+ askConnectionMode(): void
+ displayHand(): void
+ displayLeaderboard(LinkedHashMap<String, Integer>): void
+ displayLobby(): void

```
ng): void  
  
tring): void  
  
): void  
  
d): void  
  
ist<AchievementCard>): void  
  
): void  
void  
  
id  
  
oid  
  
s()): void  
(): void  
  
rdFace, int, int): void  
ks, DeckPosition): void  
  
void  
void  
  
ashMap<String, Integer>): void
```









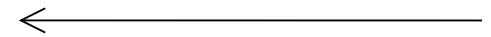
<ul style="list-style-type: none">- score: int- scoreRequirement: int
<ul style="list-style-type: none">+ Achievement(): void+ scoreRequirement(): int+ getScore(): int+ getCornerScore(): int+ getCenterScore(): int+ getScoreRequirement(): int+ getPlacementRequirement(): int+ getXCoordinate(): int+ getYCoordinate(): int+ setPlacementRequirement(int requirement)+ setXCoordinate(int coordinate)+ setYCoordinate(int coordinate)+ getPlacementRequirement(): int+ getKingdom(): Kingdom

<ul style="list-style-type: none">- scoreRequirement: int- placementRequirement: int
<ul style="list-style-type: none">+ GoldFrontFrontier(): void+ getScoreRequirement(): int+ getPlacementRequirement(): int+ getCenterScore(): int+ toString(): String- placeRequirement(): void- scoreRequirement(): int

AchievementFrontFace

```
rements : Map<Symbol, Integer>

ntFrontFace(imageURI : String, scoreRequirements : Map<Symbol, Integer>, score : int)
uirements() : List<Symbol>
 : int
ymbols() : Map<CardCorners, Symbol>
ymbols() : List<Symbol>
equirements() : Map<Symbol, Integer>
entRequirements() : Map<Symbol, Integer>
() : int
() : int
entTurn(placementTurn : int) : void
(xCoord : int) : void
(yCoord : int) : void
entTurn() : int
n() : Symbol
```



GoldFrontFace

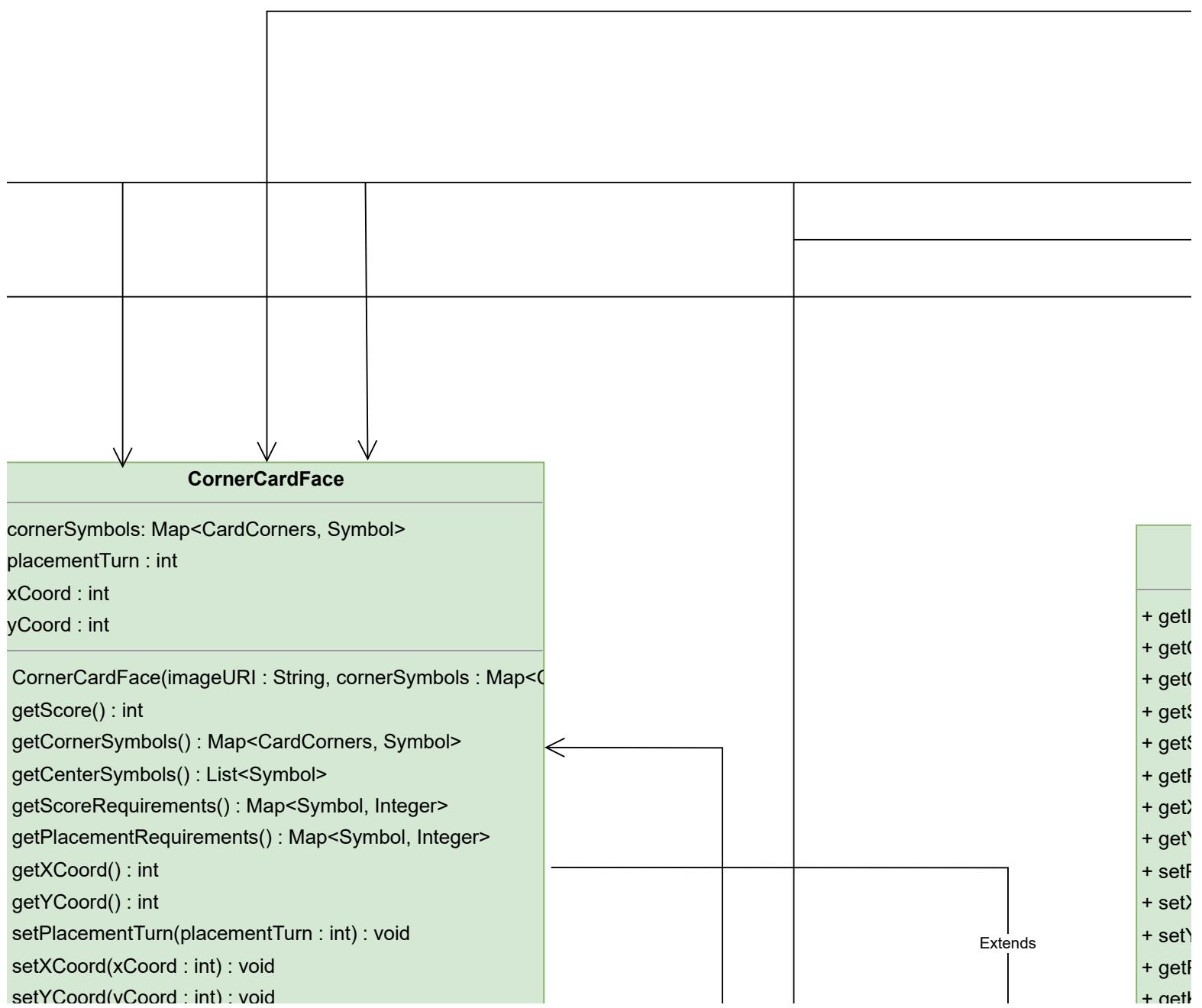
```
rements : Map<Symbol, Integer>
Requirements : Map<Symbol, Integer>

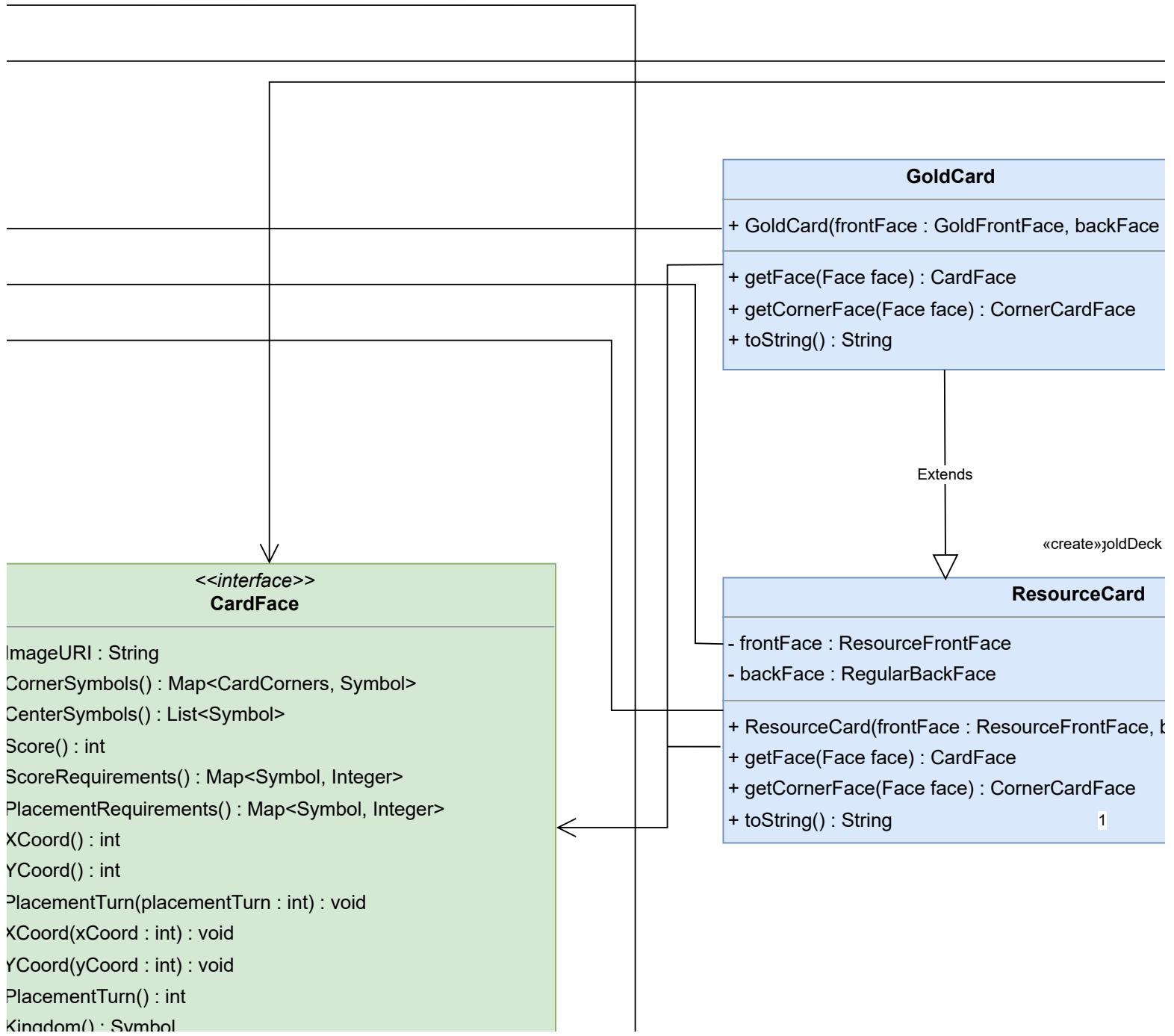
Face(imageURI : String, cornerSymbols : Map<CardCorners, Symbol>, score : int, placementRequirements: Map<Symbol, Integer>, sc
equirements() : Map<Symbol, Integer>
entRequirements() : Map<Symbol, Integer>
ymbols() : List<Symbol>
String
orPrint() : String
orPrint() : String
```

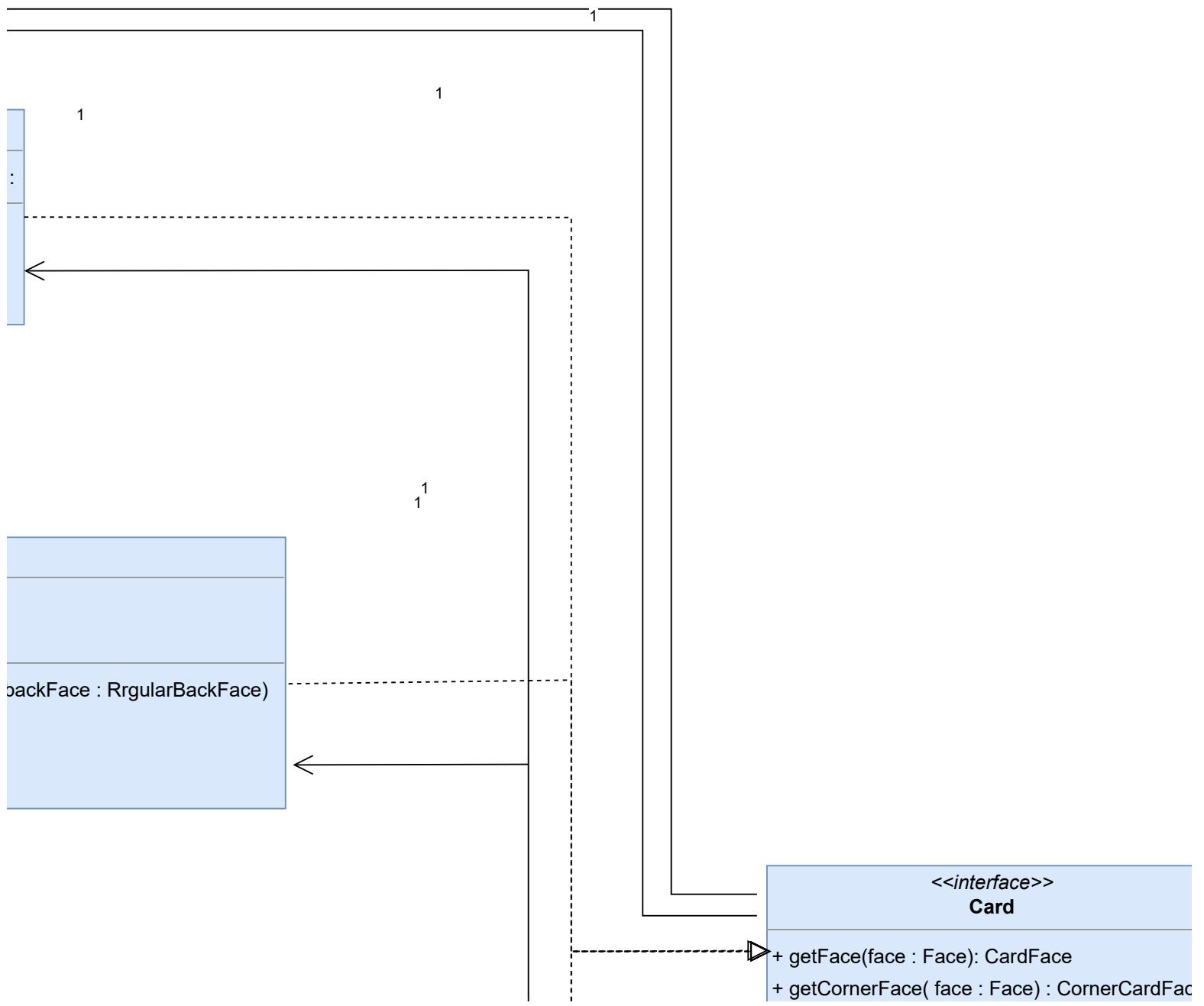
coreRequirements: Map<Symbol, Integer>, kingdom: Symbol)



- - - + + + + + + + + + + +







Chat

- messages : List<Message>

+ Chat()

+ addMessage(message : Message) : void

+ addMessage(message : String, player : Player) : void

+ addMessage(message : String, name : String) : void

+ getMessages() : List<Message>

Message

- timestamp: Timestamp

- message: String

- sender: Player

+ Message(message : String, name : String)

+ getMessage() : String

+ getName() : String

+ getTimestamp() : Timestamp

<<interface>>

Deckable

+ addCard(card : Card, position : DeckPosition) : void

+ popCard(position : DeckPosition) : Card

+ getTopCardNoPop() : Card

```

- isEndGamePhase : boolean
- turn : int
- playerList : List<PlayerInstance>
- activePlayerIndex : int
- lastRound : boolean

+ GameModellInstance()
+ getActivePlayerIndex() : int
+ isLastRound() : boolean
+ setLastRound(lastRound : boolean) : void
+ setActivePlayerIndex(activePlayerIndex : int) : void
+ generateStartingCards() : void
+ nextTurn(): void
+ setIsEndGame() : void
+ getGoldDeck() : GoldDeck
+ getResourceDeck() : ResourceDeck
+ getAchievementDeck() : AchievementDeck
+ getTurn() : int
+ getStartingCards() : List<StartingCard>
+ getChat() : Chat
+ isEndGamePhase() : boolean
+ setEndGamePhase() : void
+ getPlayerList() : List<Player>
+ setPlayerList(playerList : List<Player>) : void
+ addPlayer(player : Player) : void
+ removePlayer( player : Player) : void
+ shufflePlayerList() : void
+ createGoldResourceDecks() : void
+ createstartingCards() : void
+ createAchievementDeck() :void

```

<<interface>>

Player

```

+ addCardToHand(card : Card) : void
+ addPoints(int): void
+ initializeManuscript(StartingCard, Face): void
+ removeCardFromHand(position : int)

+ getName() : String
+ getColor() : Color
+ setColor(color : Color)
+ getPoints() : int
+ addPoints(num :int)
+ initializeManuscript(startingCard : Card, face : Face)
+ getSecretObjective() : AchievementCard
+ setSecretObjective(achievementCard : AchievementCard)
+ getManuscript(): Manuscript
+ setReady(ready : boolean)
+ isReady() : boolean

```

PlayerInstance

```

- points: int
- name: String
- manuscript: Manuscript
- secretObjective: AchievementCard
- color: Color

+ PlayerInstance(name : String)
+ initializeManuscript(StartingCard, Face): void
+ addPoints(int): void
+ addCardToHand(card : Card): void
+ removeCardFromHand(position : int): void
+ getName() : String
+ getColor() : Color
+ setColor(color : Color)

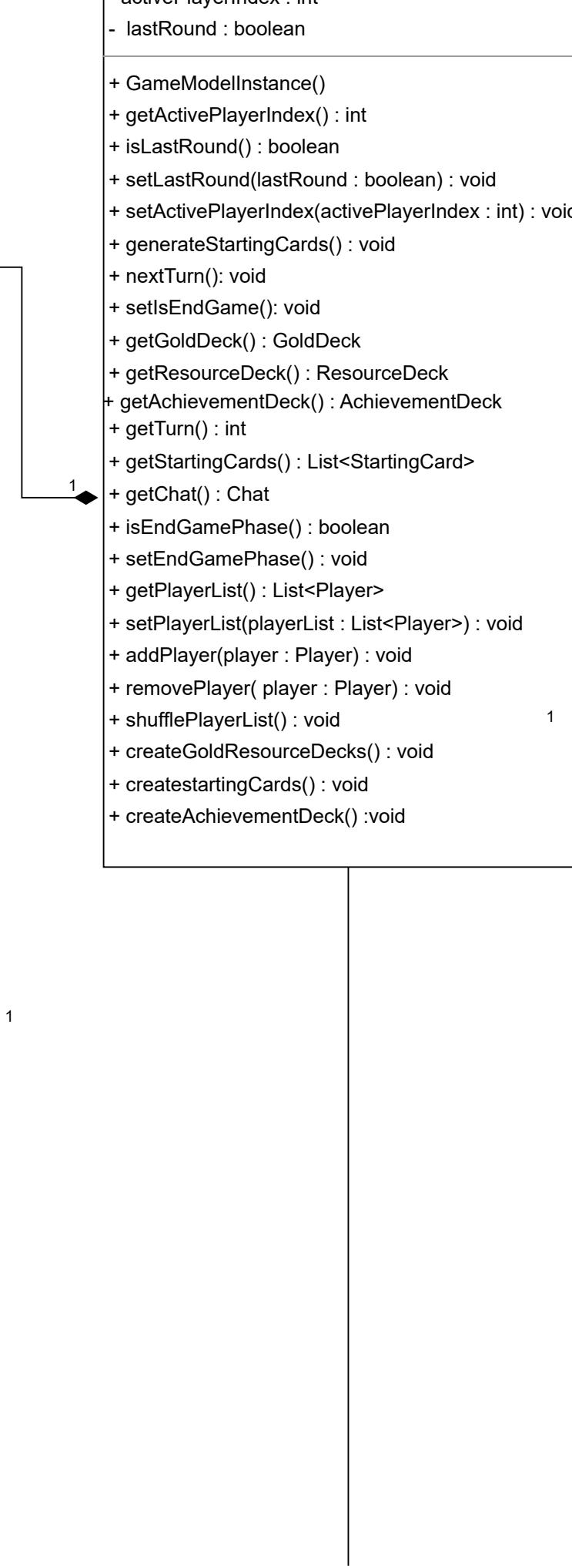
```

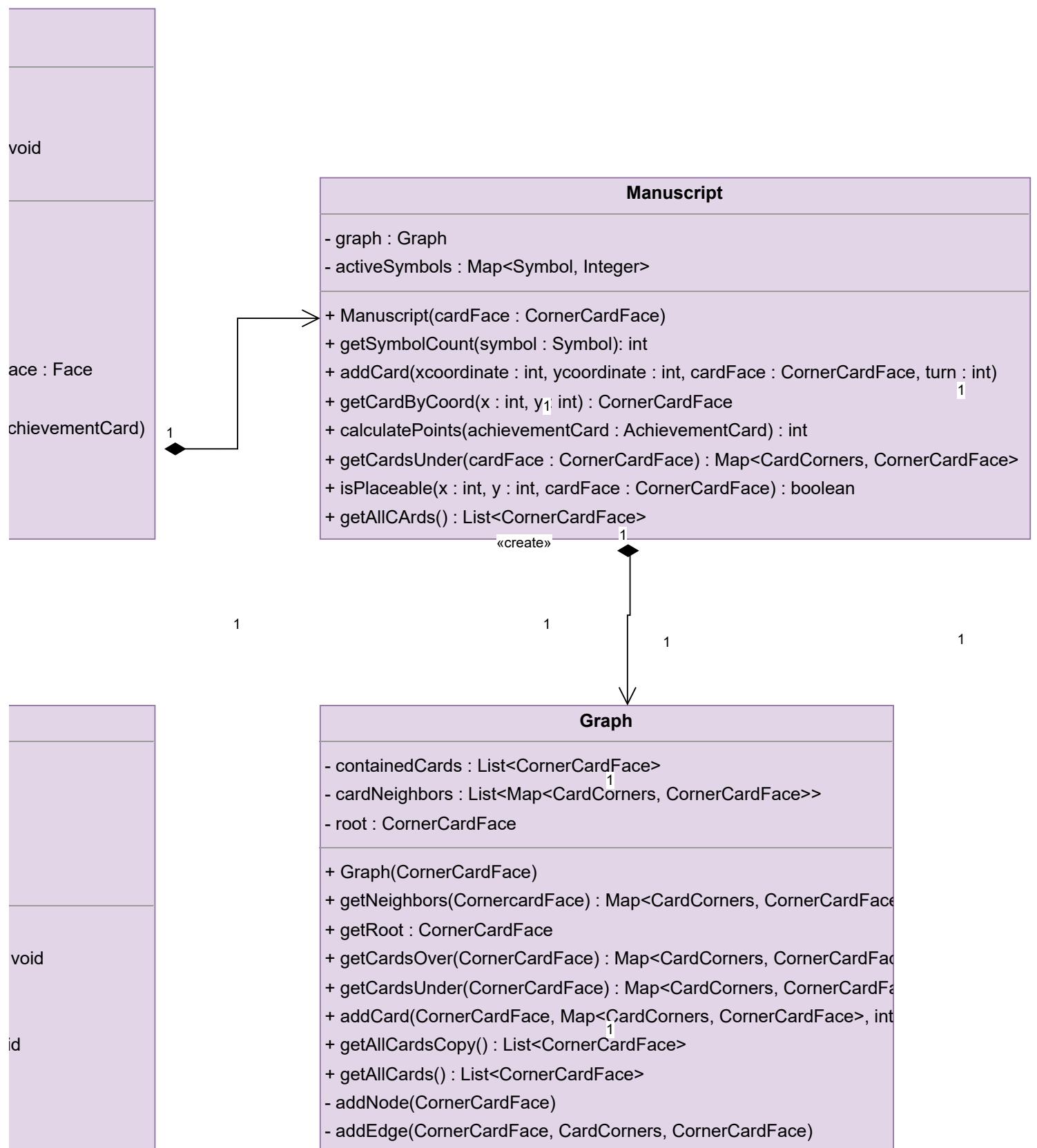
1

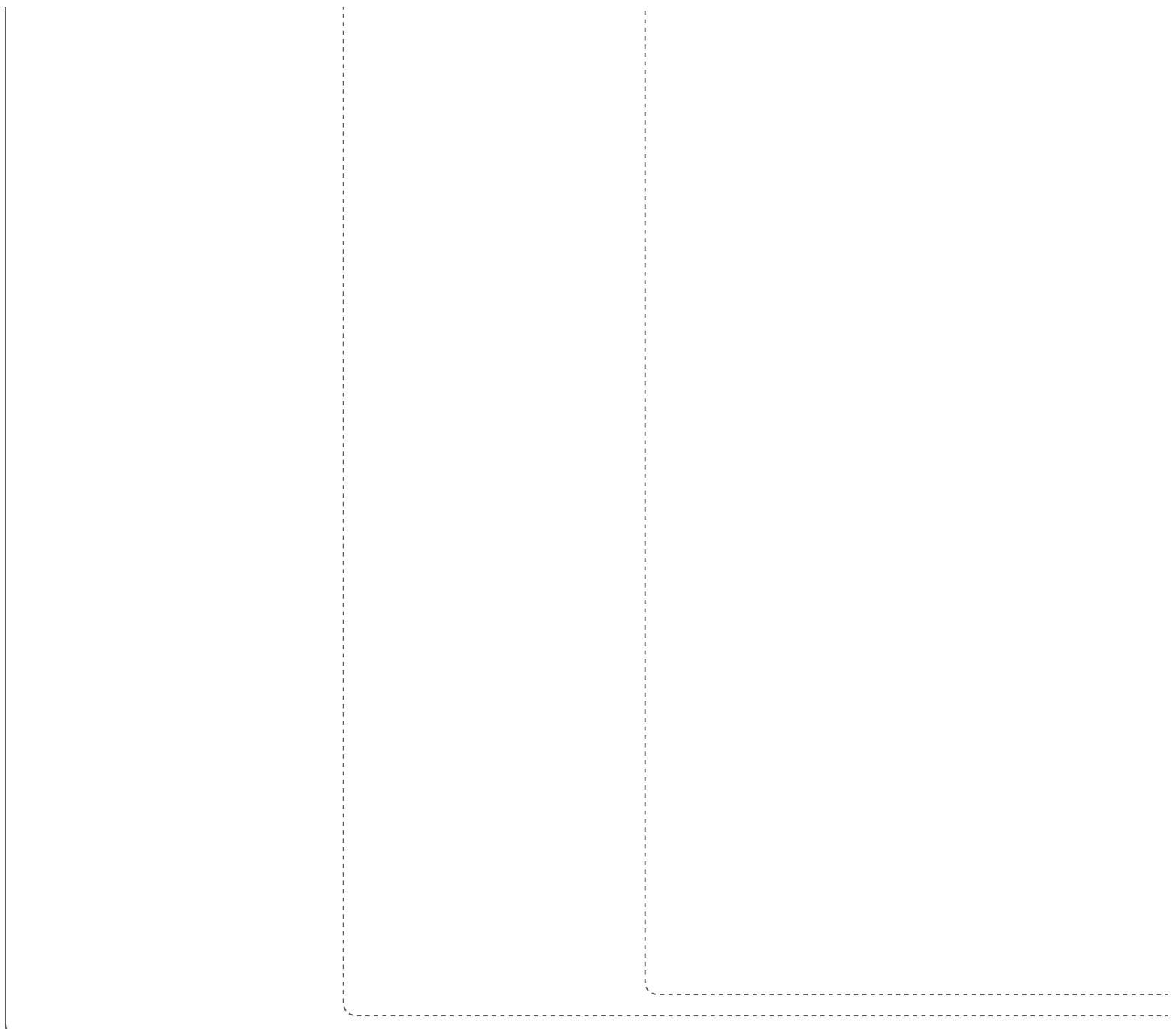
1

1

△



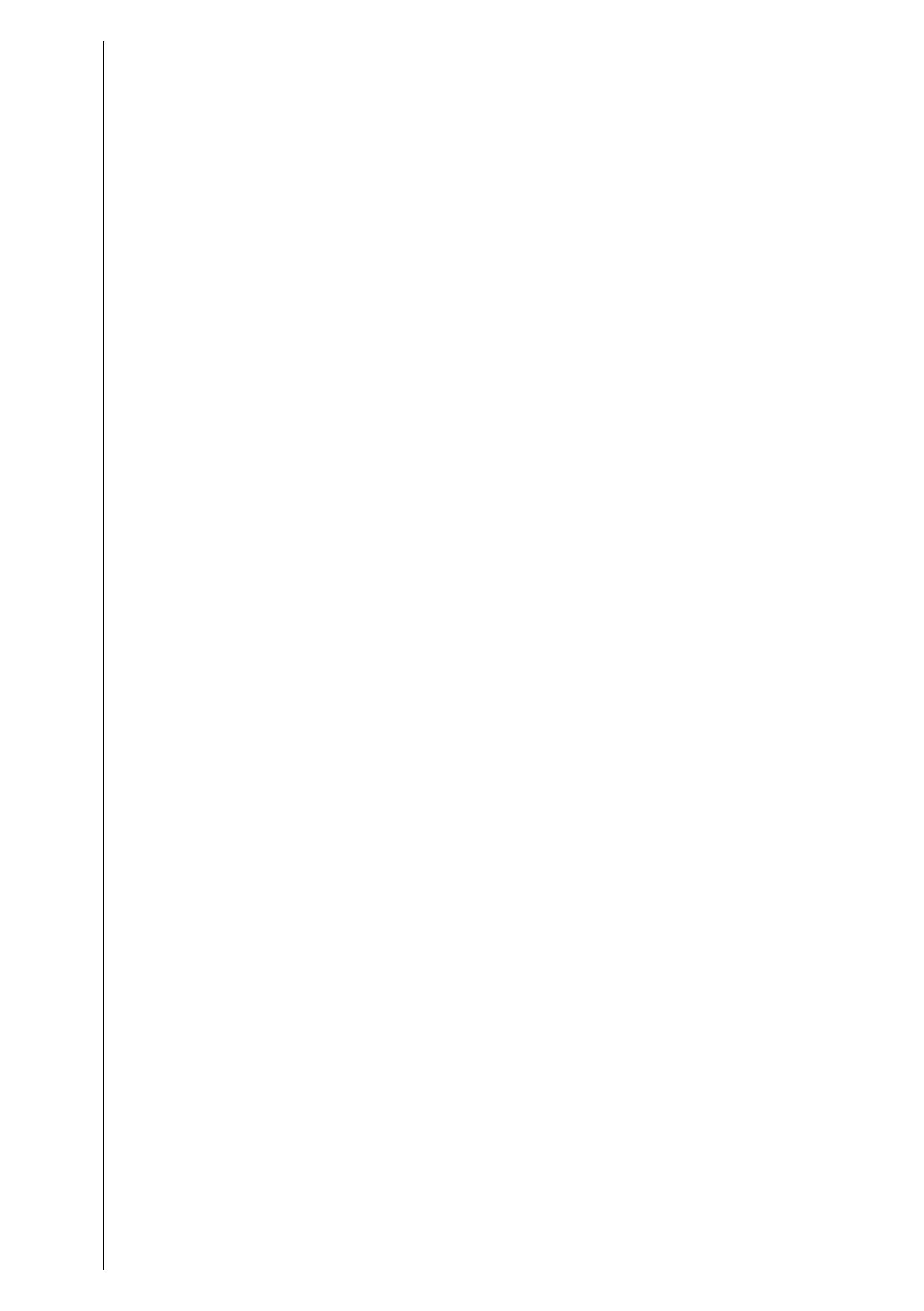


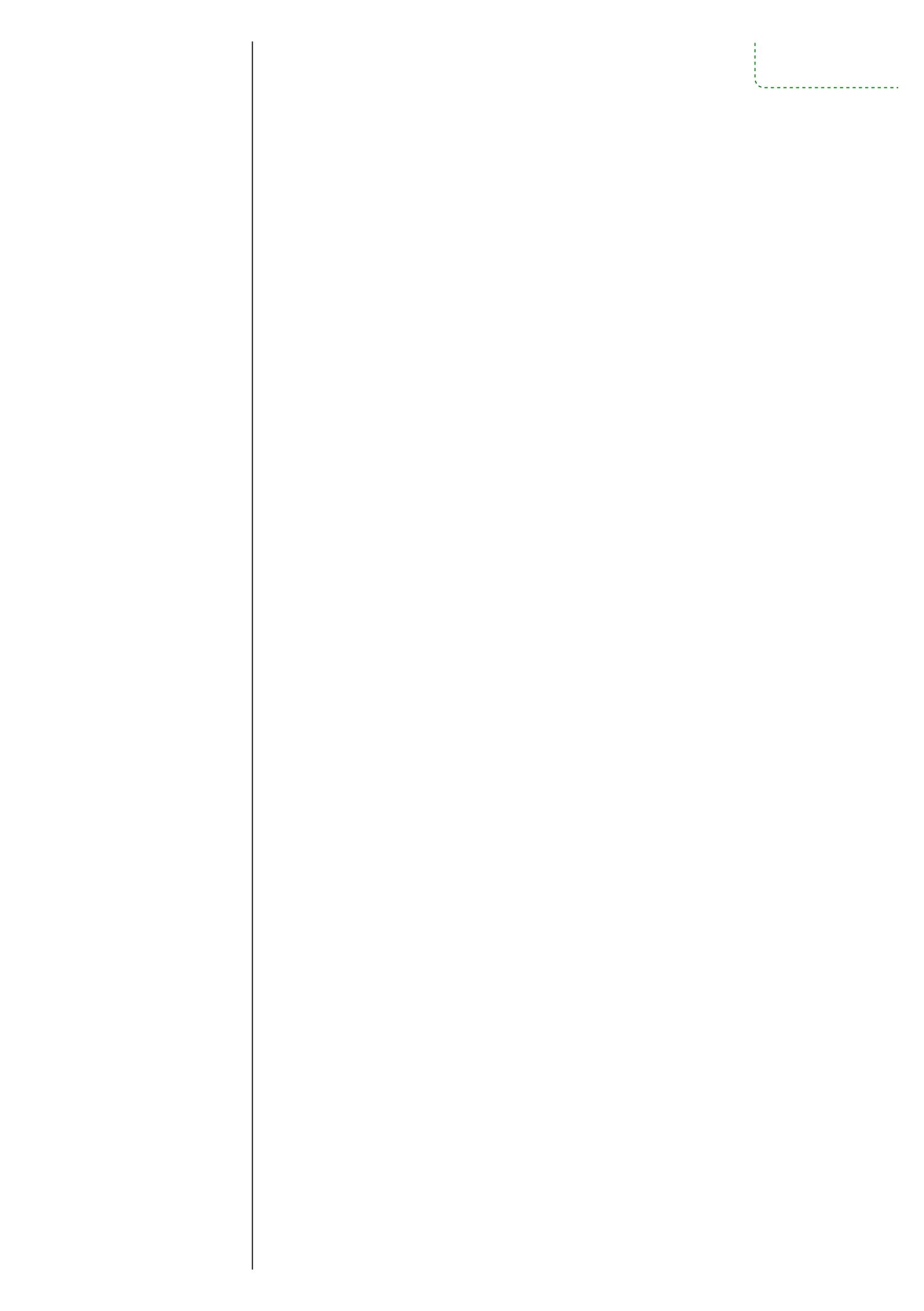


| ControllerInstance |
|---|
| + ControllerInstance(GeneralServerConnectionHandler): |
| - gameState: ServerState |
| - givenStartingCards: Map<Player, StartingCard> |
| - gameModel: GameModel |
| - givenSecretObjectiveCards: Map<Player, List<AchievementCard>> |
| - connectionHandler: GeneralServerConnectionHandler |
| + removePlayer(Player): void |
| + start(): void |
| + isInSavedGameLobby(): boolean |
| + giveInitialHand(): void |
| + setReady(Player): void |
| + setSecretObjectiveCard(Player, int): void |
| + addPlayer(String, String): void |
| + setStartingCard(Player, Face): void |
| + getConnectionHandler(): GeneralServerConnectionHandler |
| + reactToDisconnection(String): void |
| + computeLeaderboard(): void |
| + setNotReady(Player): void |
| + nextTurn(): void |
| + getPlayerList(): List<Player> |
| + drawCard(Player, DeckPosition, Decks): void |
| + isReady(Player): boolean |
| + disconnectionLeaderboard(): void |
| + shufflePlayerList(): void |
| + addMessage(String, Player): void |
| + isOnline(Player): boolean |
| + changeState(ServerState): void |
| + getPlayerByName(String): Player |
| + playCard(Player, int, int, int, Face): void |
| + saveGame(): void |
| + setPlayerColor(Color, Player): void |
| - giveStartingCards(): void |
| + getTurn(): int |
| + addSavedPlayer(String, String): void |
| + setOffline(String): void |
| + endGame(): void |
| + clear(): void |
| + loadGame(): void |
| - giveSecretObjectiveCards(): void |
| - createDecks(): void |
| + reconnect(String, String): void |

```
+ isInSavedGameLobby(): boolean  
+ drawCard(Player, DeckPosition, Decks): void  
+ removePlayer(Player): void  
+ reactToDisconnection(String): void  
+ reconnect(String, String, Player, Map<Player, List<AchievementCard>>  
+ addPlayer(String, String): void
```

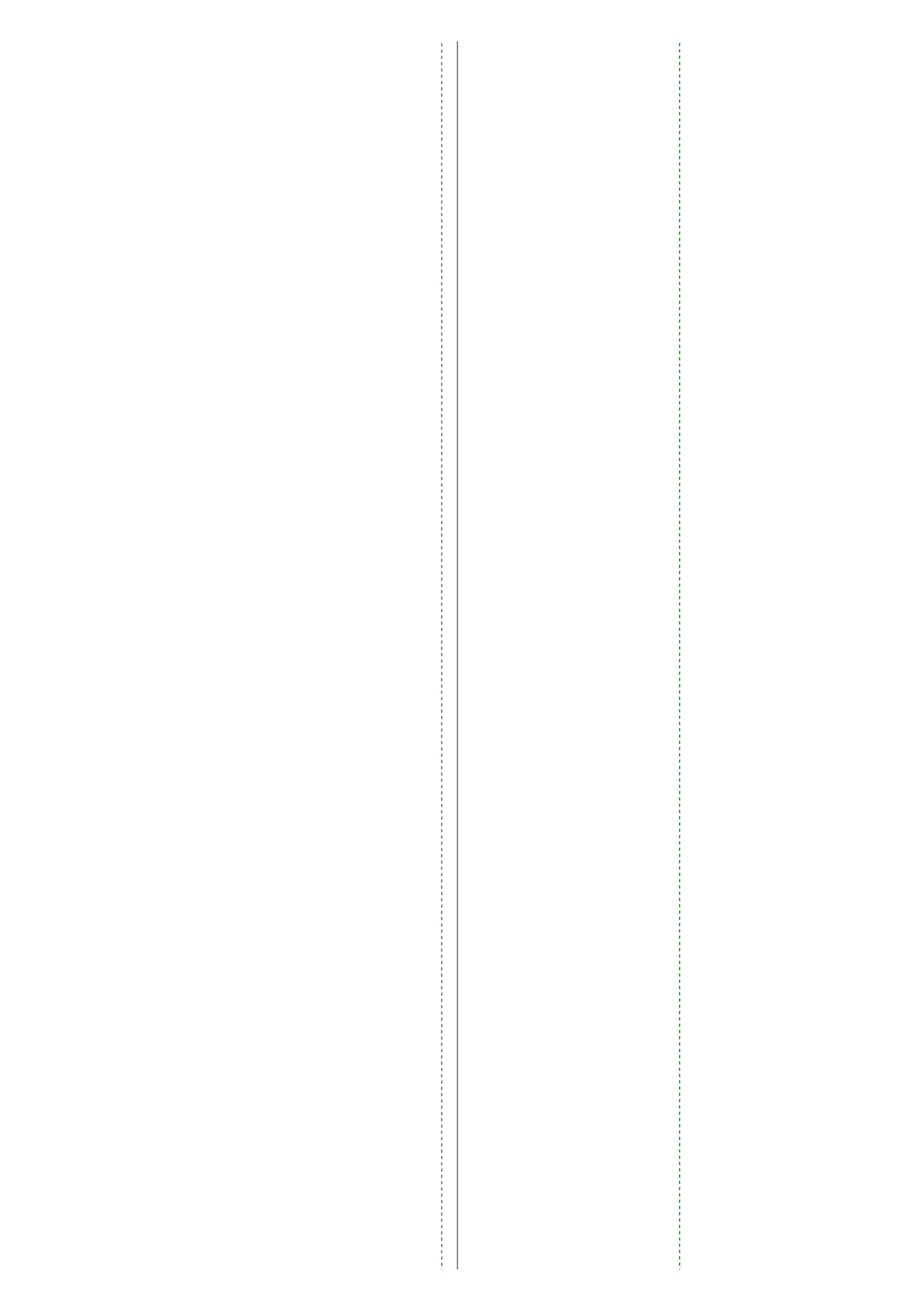






```
- clientConnectionHandler: GeneralClientConnectionHandler
- players: List<Player>
- proposedName: String
- id: String
- ui: UI
- secretAchievement: AchievementCard
- gameState: GameState
- isSavedGame: Boolean
- goldDeck: Deck<GoldCard>
- resourceDeck: Deck<ResourceCard>
- turn: int
- indexofSecretAchievement: int
- myName: String

+ chooseStartingCardFace(Face): void
+ getDeckSize(Decks): int
+ askSetName(String): void
+ setMyColor(Color): void
+ getMyColor(): Color
+ askPlayCard(int, Face, int, int): void
+ startingCardChosen(String, CornerCardFace): void
+ setEndGame(): void
+ askSetColor(Color): void
+ getMyId(): String
+ giveStartingCard(Card): void
+ reconnect(String): void
+ giveAchievementCards(List<AchievementCard>, List<AchievementCard>): void
+ updatePlayerOrder(List<String>): void
+ placeCard(String, CornerCardFace, int, int, int): void
+ getPlayerNames(): List<String>
+ getSecretAchievement(): AchievementCard
+ clearUI(): void
+ playerAlreadyPlaying(): void
+ getChosenHandCard(): Integer
+ gameAlreadyFinished(): void
+ tooManyPlayers(): void
+ giveInitialHand(List<Card>): void
+ updatePlayerColors(Color, String): void
+ colorNotYetSet(): void
+ otherPlayerReconnected(String): void
+ getBoardCards(Decks): List<Card>
+ getGameState(): GameState
+ addChatMessage(Message): void
+ newPlayer(List<String>): void
+ getIndexofSecretAchievement(): int
+ askDrawCard(Decks, DeckPosition): void
+ clear(): void
+ joinServer(String, int): void
+ joinSavedGame(String): void
+ setColor(boolean, Color): void
+ displayLeaderboard(LinkedHashMap<String, Integer>): void
+ serverDisconnected(): void
+ playerDisconnected(String): void
```





+ CLI(ClientController):

- in: Scanner

- controller: ClientController

- scene: Scene

+ idNotFound(): void

+ tooManyPlayers(): void

+ nameChanged(): void

+ displayDeck(): void

+ deckIsEmpty(): void

- manuscript(): void

+ gameStarted(): void

+ updateName(): void

+ needName(): void

+ cantDraw(): void

+ startingCardChosen(): void

+ nameChanged(): void

+ displayHand(): void

+ chooseCard(): void

+ cardNotAvailable(): void

+ otherPlayerConnected(): void

+ needConnectionMode(): void

+ invalidCard(): void

+ endGame(): void

+ printOnDeck(): void

+ displayDeck(): void

+ gameNotYetStarted(): void

+ gameAlreadyStarted(): void

+ colorChanged(): void

+ chat(Message): void

+ otherPlayerDisconnected(): void

+ displayHand(): void

+ backToDeck(): void

+ playerDisconnected(): void

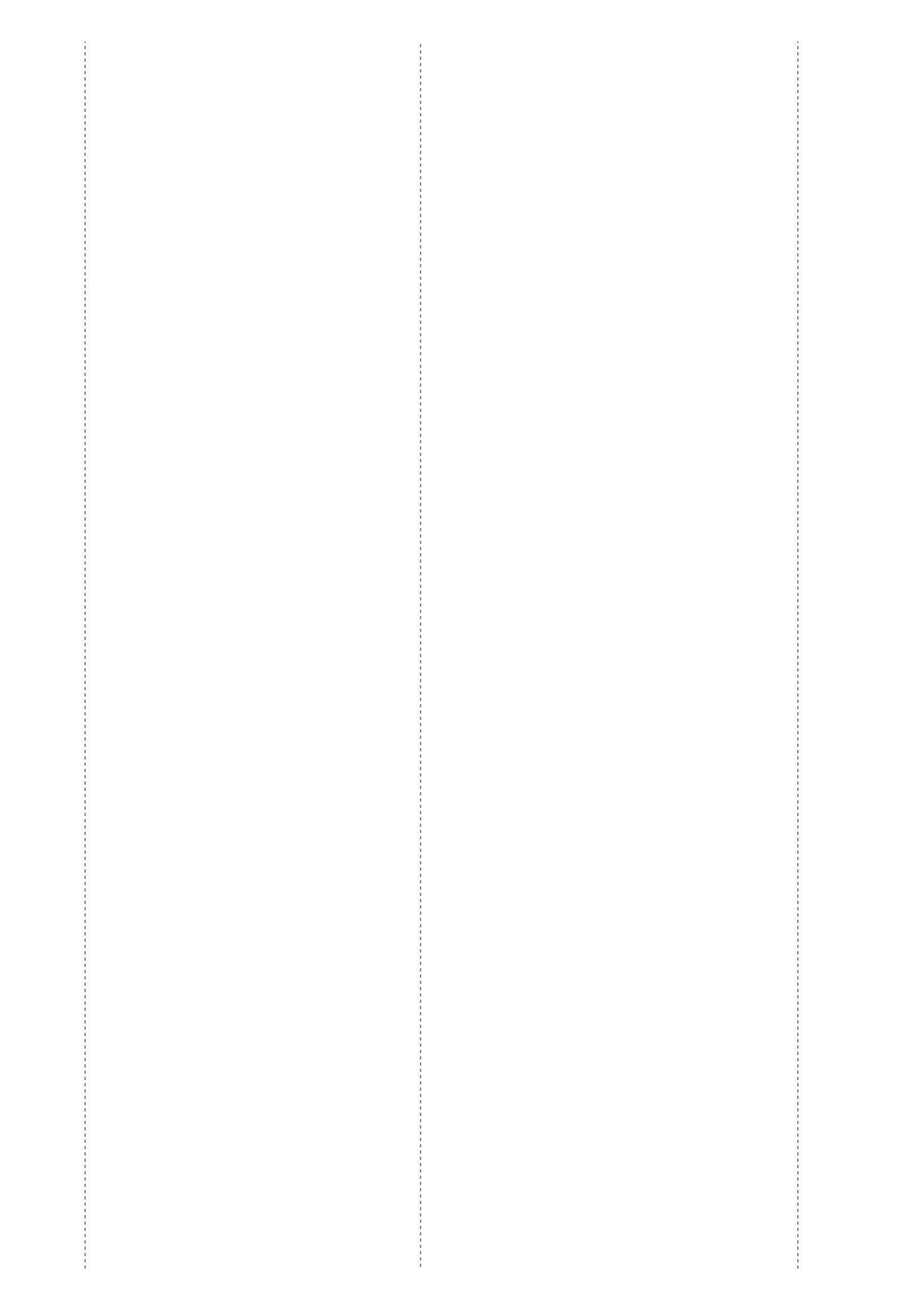
+ displayDeck(): void

+ displayHand(): void

+ playerDisconnected(): void

CLI

```
entController):  
  
nner  
er: ClientController  
CLISate  
  
Game(): void  
nyPlayers(): void  
hangeFailed(): void  
CommonAchievements(): void  
Empty(): void  
criptCoordString(int): String  
tarted(): void  
Ready(String, boolean): void  
ame(): void  
awAchievementCards(): void  
CardChosen(String): void  
hanged(): void  
Card(String, int, int): void  
SecretAchievement(List<AchievementCard>): void  
tPlaceable(): void  
ayerInitialHand(String): void  
olor(): void  
Color(): void  
meStarted(): void  
nNewLine(String): void  
NewCardInHand(): void  
otYetStarted(): void  
lreadyStarted(): void  
hanged(): void  
essage): void  
ayerDraw(String, Decks, DeckPosition): void  
PlayerColors(): void  
ameOrColor(): void  
Hand(): void  
SecretAchievement(): void  
Lobby(): void  
Disconnected(String): void  
Id(): void  
Turn(): void  
Removed(String): void
```



- score : int
- kingdom : S

+ ResourceF
+ getCenterS
+ getKingdon
+ getScoreRe
+ getPlaceme
+ getScore()
+ toString() :

+ StartingFro
- centerSymb

+ StartingFro
+ getCenterS
+ getKingdon
+ getScoreRe
+ getPlaceme
+ getScore()
+ toString() :

ResourceFrontFace

Symbol

frontFace(imageURI : String, cornerSymbols : Map<CardCorners, Symbol>, score : int, kingdom : Symbol)

symbols() : List<Symbol>

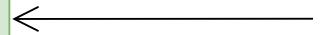
n() : Symbol

equirements() : Map<Symbol, Integer>

entRequirements() : Map<Symbol, Integer>

: int

String



StartingFrontFace

ntFace(String, Map<Integer, Symbol>, List<Symbol>):

ools: List<Symbol>

ntFace(imageURI : String, cornerSymbols : Map<CardCorners, Symbol>, centerSymbols : List<Symbol>)

symbols() : List<Symbol>

n() : Symbol

equirements() : Map<Symbol, Integer>

entRequirements() : Map<Symbol, Integer>

: int

String



+

+

-

-

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

setCoordinateX(int) : void

getPlacementTurn() : int

getKingdom() : Symbol

RegularBackFace

centerSymbols: List<Symbol>

kingdom : Symbol

RegularBackFace(imageURI : String, centerSymbol : List<Symbol>)

getCenterSymbols() : List<Symbol>

getKingdom() : Symbol

getScoreRequirements() : Map<Symbol, Integer>

getPlacementRequirements() : Map<Symbol, Integer>

getScore() : int

toString() : String

- imag

+ Emp

+ getl

+ gets

+ getc

+ geto

+ gets

+ getf

+ getp

+ getv

+ setf

+ seto

+ setv

+ getf

+ getp

+ getv

+ setf

+ seto

+ setv

+ getf

+ getp

+ getv

+ setf

+ seto

+ setv

+ getf

+ getp

+ getv

+ setf

+ seto

+ setv

+ getf

+ getp

+ getv

+ setf

+ seto

+ setv

+ getf

+ getp

+ getv

+ setf

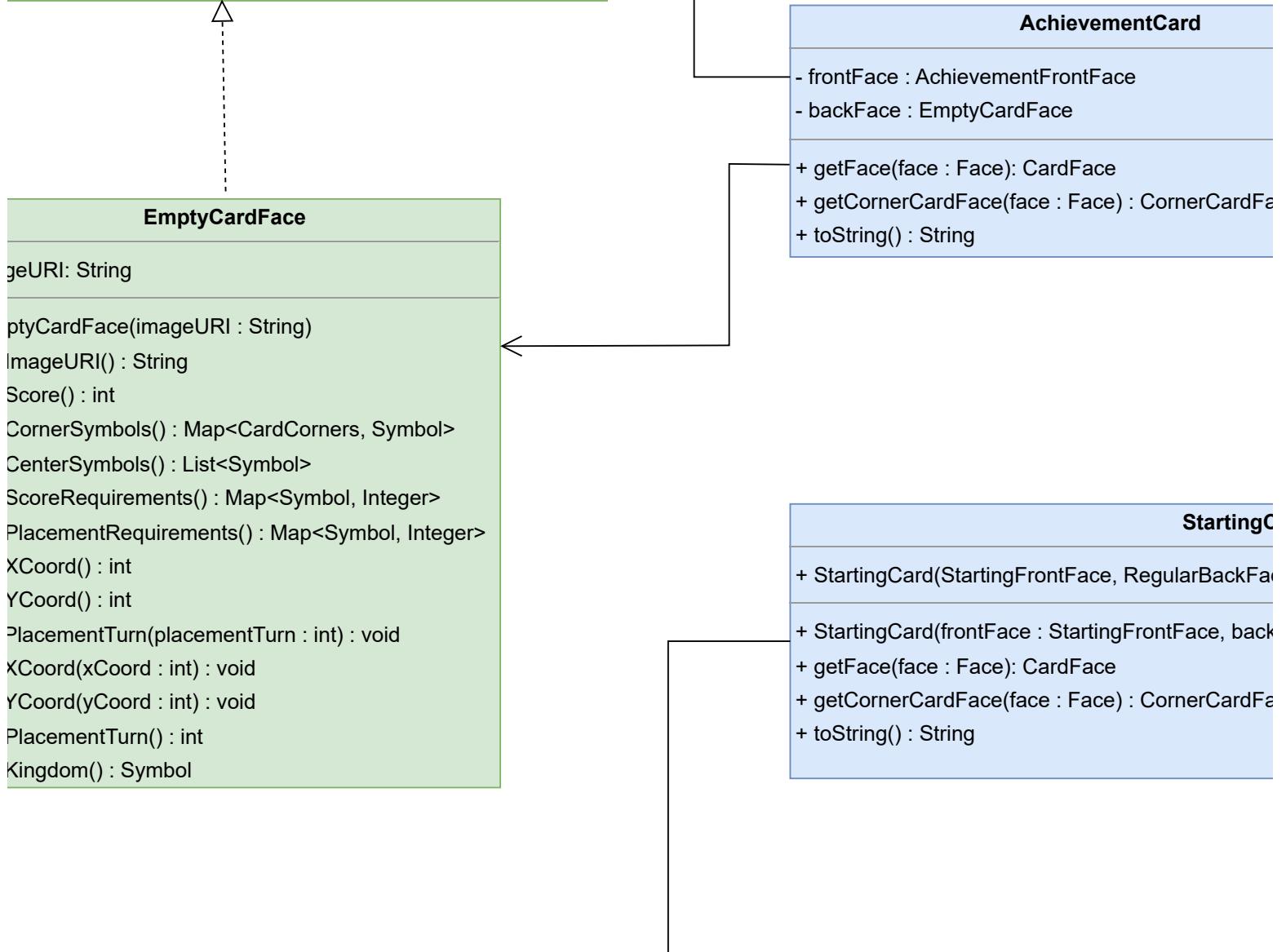
+ seto

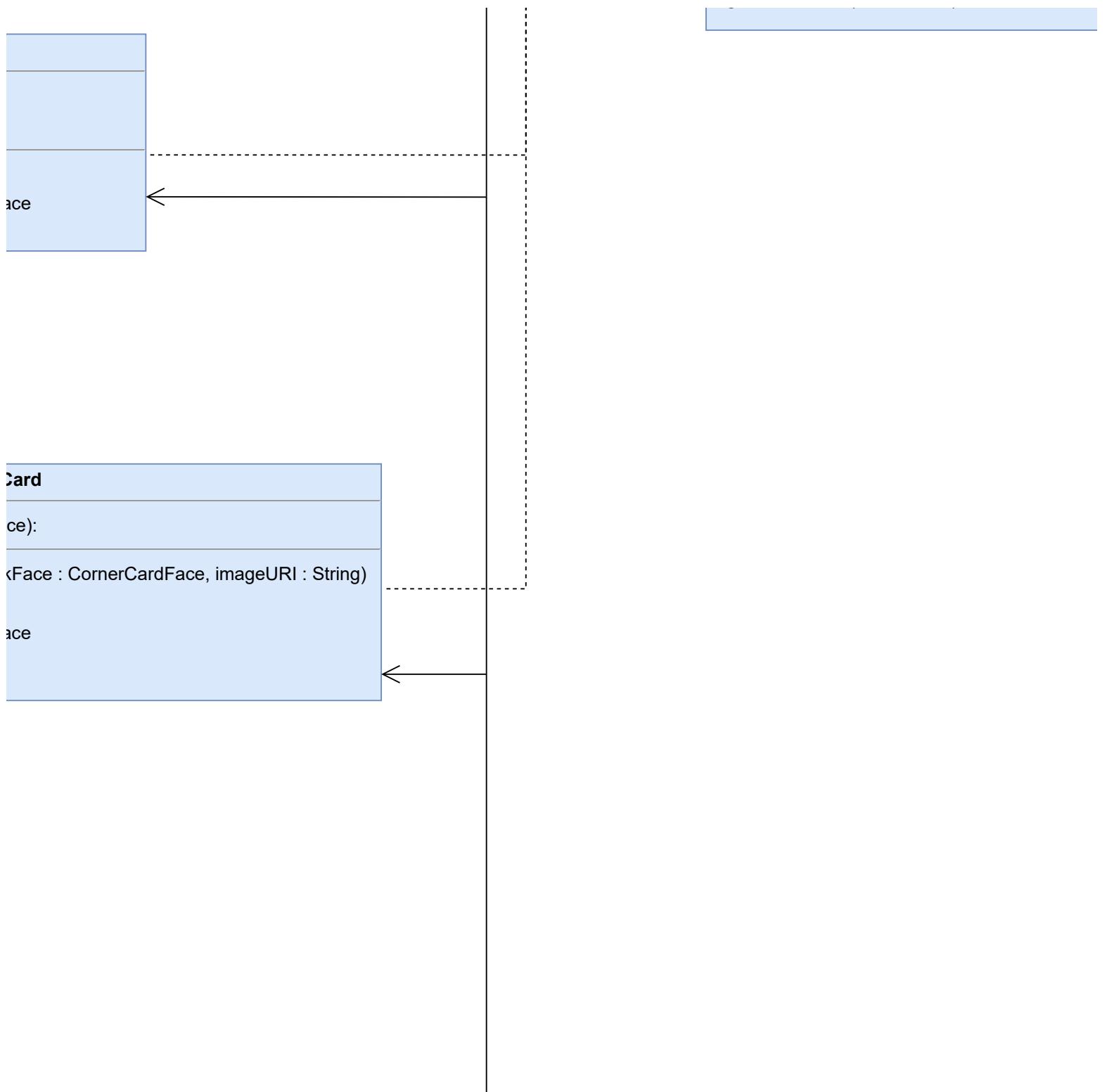
+ setv

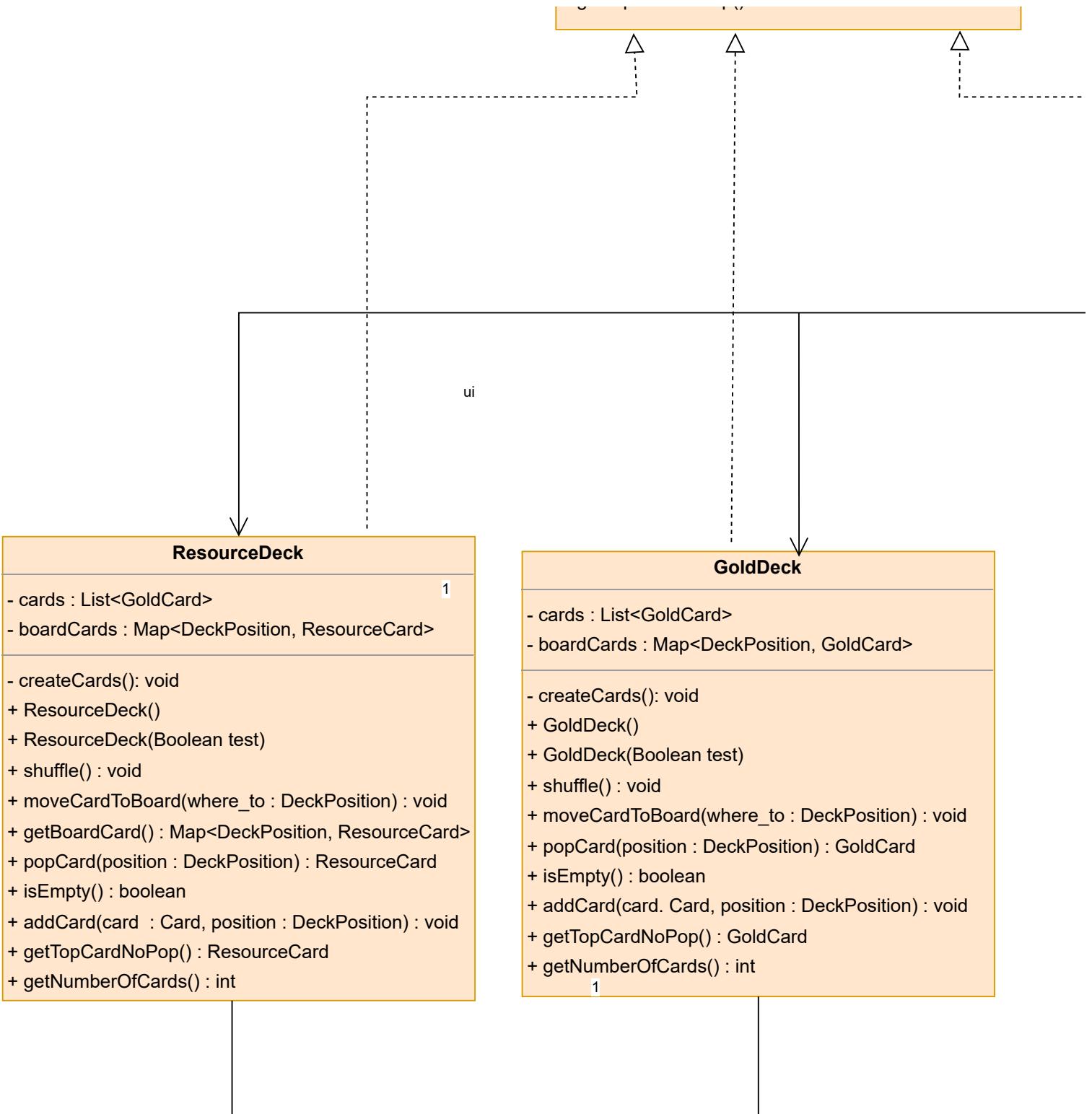
+ getf

+ getp

+ getv







```
+ getPoints() : int  
+ addPoints(num : int)  
+ getHand : List<Card>  
+ getSecretObjective: AchievementCard  
+ setSecretObjective(achievementCard : AchievementCard)  
+ getManuscript() : Manuscript  
+ setReady(ready : boolean)  
+ isReady() : boolean
```

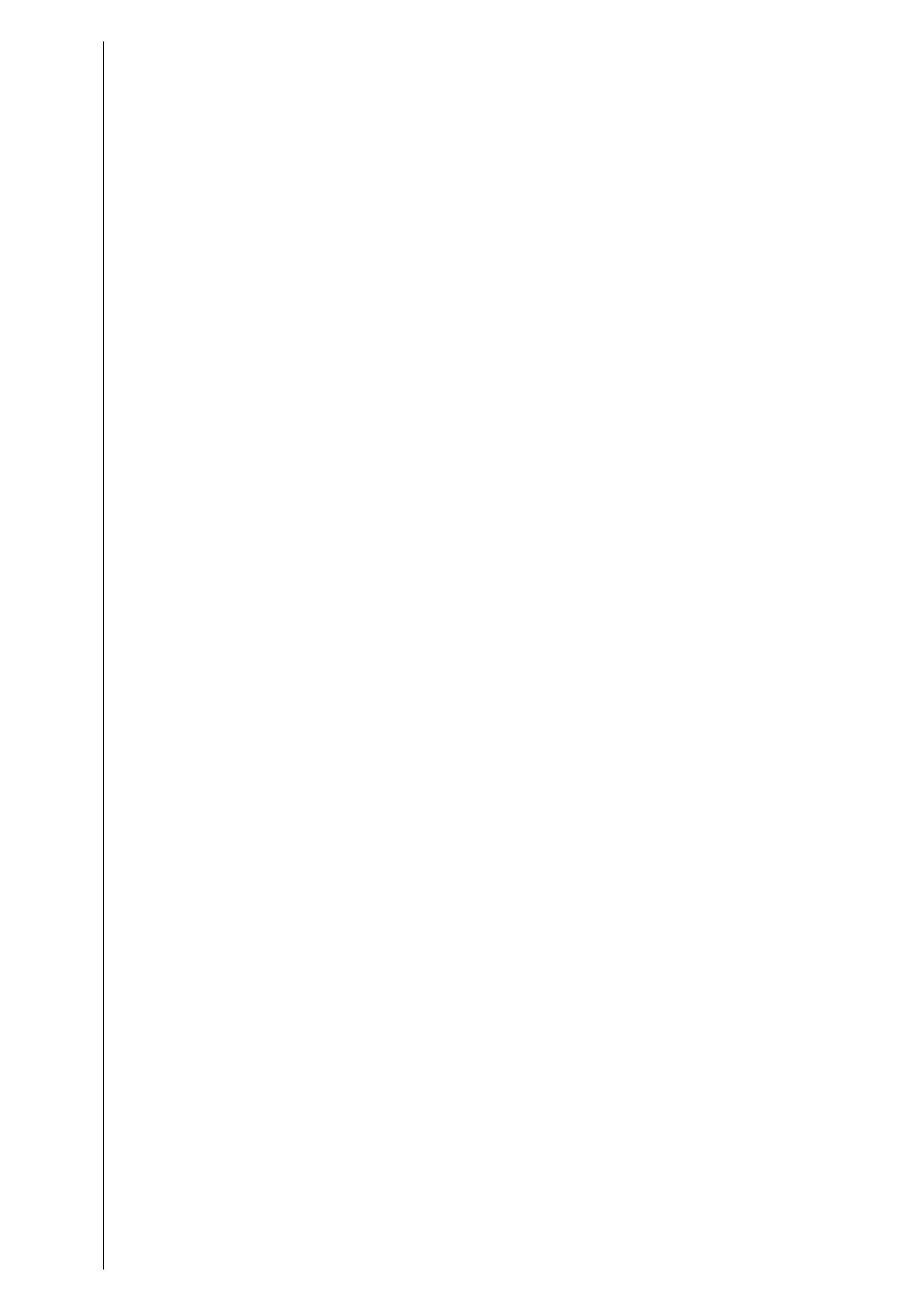
AchievementDeck

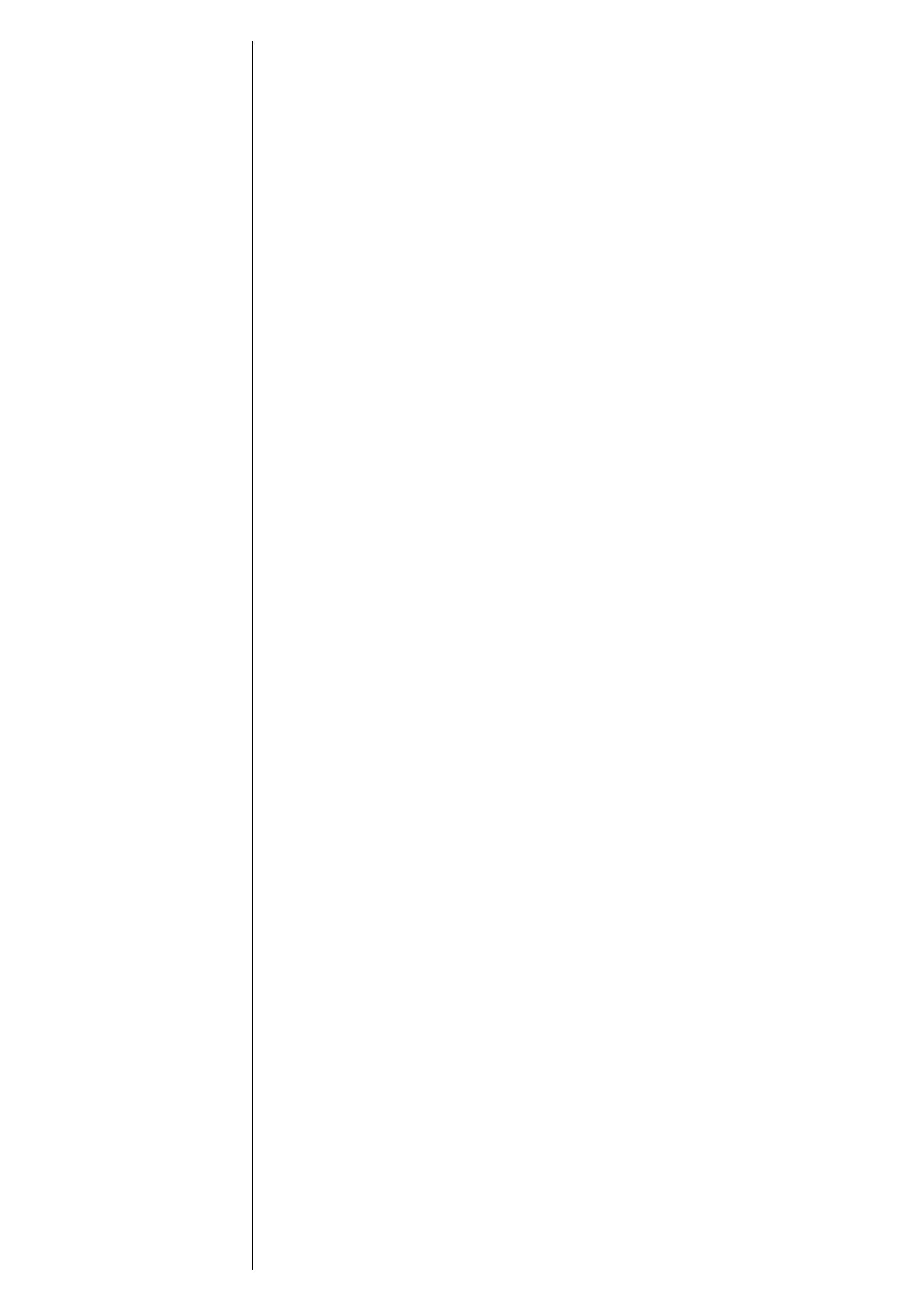
```
- cards : List<AchievementCard>  
- boardCards : Map<DeckPosition, AchievementCard>  
  
- createCards() : void  
+ AchievementDeck()  
+ AchievementDeck(Boolean test)  
+ shuffle() : void  
+ moveCardToBoard(where_to : DeckPosition) : void  
+getBoardCard() : Map<DeckPosition, AchievementCard>  
+ isEmpty() : boolean  
+ addCard(card : Card, position : DeckPosition) : void  
+ getTopCardNoPop() : Card  
+ getNumberOfCards() : int
```

1

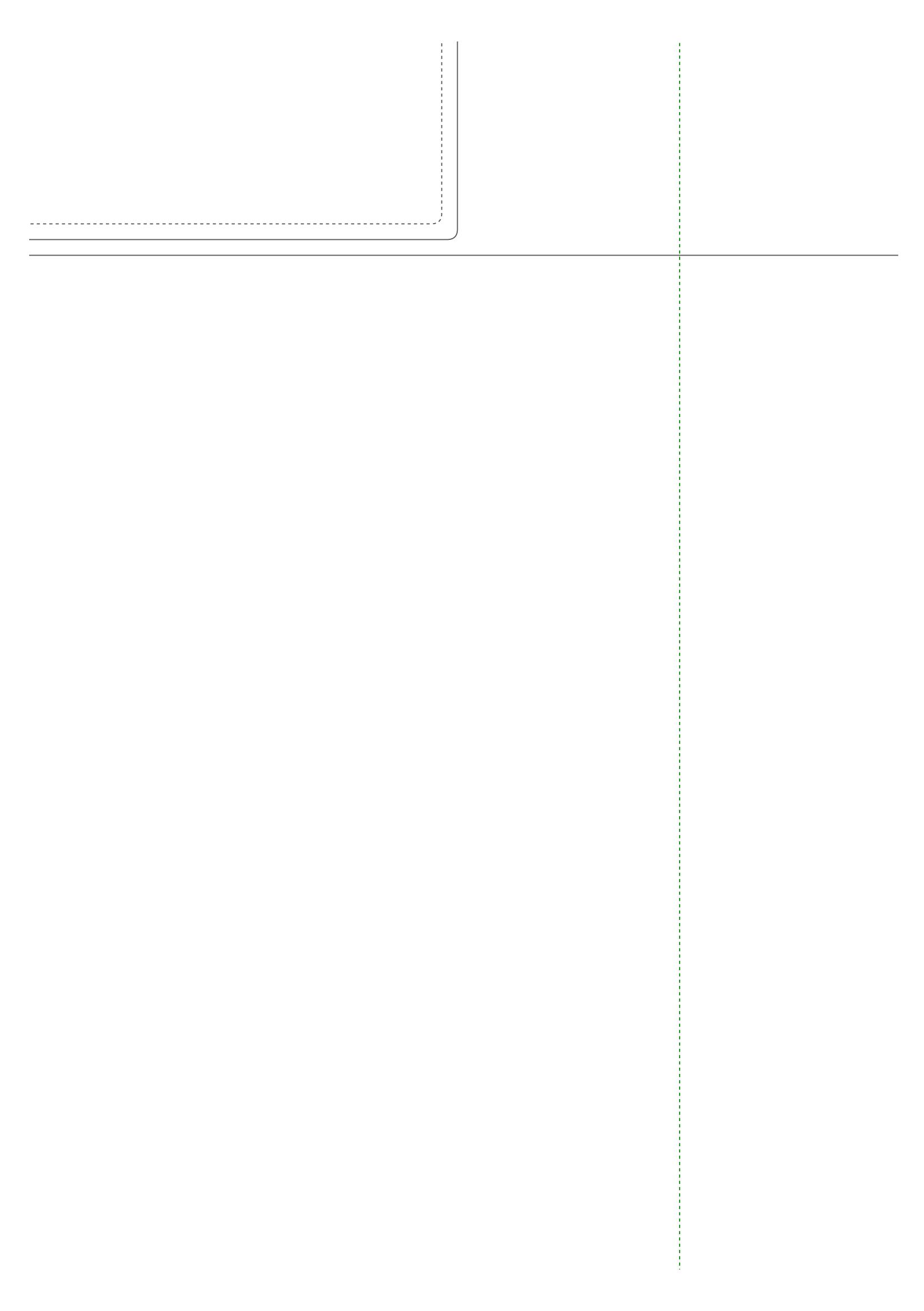


[REDACTED]





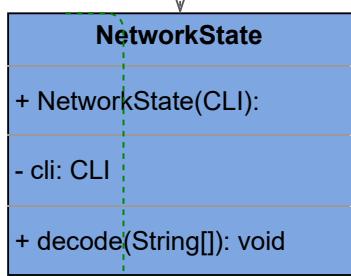
```
+ giveDrawnCard(Card): void
+ invalidCard(Actions): void
+ getActivePlayer(): Player
+ setChosenHandCard(Integer): void
+ loadGame(List<AchievementCard>, Deck<GoldCard>, Deck<ResourceCard>
+ idNotInGame(): void
+ getChat(): List<Message>
+ main(UI): void
+ getPlayers(): List<Player>
+ alreadyDone(Actions): void
+ gameNotYetStarted(): void
+ setId(String): void
+ setSecretCard(String): void
+ notYourTurn(): void
+ chooseSecretAchievement(int): void
+ getHand(): List<Card>
+ isSavedGame(): boolean
+ getTurn(): int
+ updatePlayerReady(boolean, String): void
+ getPlayerByName(String): Player
+ notYetGivenCard(Actions): void
+ invalidName(): void
+ getMyName(): String
+ startGame(List<GoldCard>, List<ResourceCard>): void
+ setName(Boolean): void
+ getLeaderboard(): Map<String, Integer>
+ achievementDeckDrawInvalid(): void
+ playerRemoved(String): void
+ setSecretCard(int): void
+ loadLobbyInfo(String, List<String>, Map<String, Color>, Map<String, Boolean>
+ nameNotYetSet(): void
+ setReady(): void
+ cardNotPlaceable(): void
+ drawOtherPlayer(String, Decks, DeckPosition, List<Card>): void
+ setGameInfo(String, List<AchievementCard>, Deck<GoldCard>, Deck<ResourceCard>
+ getCommonAchievements(): List<AchievementCard>
+ giveOtherPlayerInitialHand(String): void
+ getAvailableColors(): List<Color>
+ emptyDeck(): void
+ newTurn(String, int): void
+ ToDoFirst(Actions): void
+ getPotentialSecretAchievements(): List<AchievementCard>
+ chatMessageIsEmpty(): void
+ sendChatMessage(String): void
+ setRMIMode(boolean): void
+ setName(String): void
+ gameAlreadyStarted(): void
```

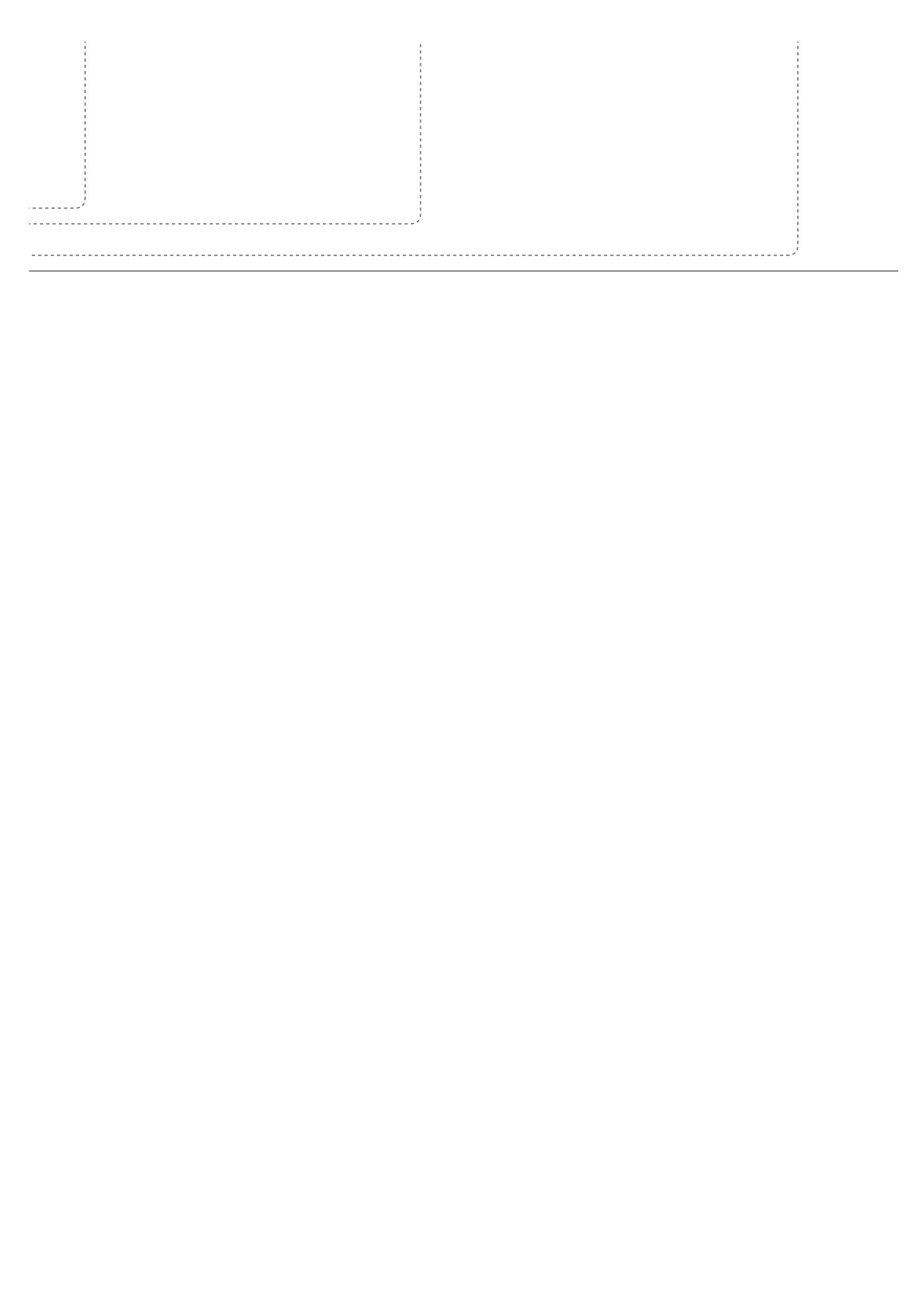


```
+ displayName(): void  
+ displayGameInfo(): void  
+ unavailableColor(): void  
+ notYourTurn(): void  
+ displayNewPlayer(): void  
+ cardPlaced(String, CornerCardFace, int, int): void  
+ tooManyPlayers(): void  
+ idNotInGame(): void  
+ chooseStartingCardFace(Card): void  
+ invalidColor(): void  
+ secretAchievementChosen(String): void  
+ otherPlayerInitialHand(String): void  
- regenerateManuscript(): void  
+ nameChanged(): void  
+ needName(): void  
+ deckIsEmpty(): void  
+ chooseSecretAchievement(List<AchievementCard>): void  
+ cardNotPlaceable(): void  
+ getScene(SceneName): Scene  
+ displayBoardCards(): void  
+ otherPlayerDraw(String, Decks, DeckPosition): void  
+ displayLobby(): void  
+ connectionFailed(): void  
+ updateReady(String, boolean): void  
+ displayId(): void  
+ clear(): void  
+ displayLeaderboard(LinkedHashMap<String, Integer>): void  
+ notYetGivenCard(Actions): void  
+ nameChangeFailed(): void
```

```
+ success(): void  
+ invalidColor(): void  
+ chatMessage(): void  
+ connectionFailed(): void  
+ displayBoardCards(): void  
+ playerAdded(): void  
+ colorChanged(): void  
+ alreadyConnected(): void  
+ displayLobby(): void  
+ displaySecretAchievement(): void  
+ doFirstTurn(): void  
+ secretAchievementChosen(): void  
+ gameAccepted(): void  
+ displaySecretAchievement(): void  
+ changeName(): void  
+ notYetGivenCard(): void  
+ newTurn(): void  
+ notYourTurn(): void  
+ joinServer(): void  
+ clear(): void  
+ displayBoardCards(): void  
+ printProcedure(): void  
+ displayLobby(): void  
+ chooseStartingCardFace(): void  
+ otherPlayerInitialHand(): void  
+ run(): void  
+ unavailableColor(): void  
+ setRMIAddress(): void  
+ displaySecretAchievement(): void  
+ displayBoardCards(): void  
+ askCornerCardFace(): void  
+ cardPlaced(): void
```

```
sfulConnection(): void  
CardForAction(Actions): void  
essageIsEmpty(): void  
ctionFailed(): void  
Lobby(): void  
lreadyPlaying(): void  
angeFailed(): void  
/Done(Actions): void  
PlayerOrder(): void  
BoardCards(): void  
(Actions): void  
chievementChosen(String): void  
lreadyFinished(): void  
DeckSizes(): void  
Scene(CLIStrate): void  
GivenCard(Actions): void  
rn(): void  
rTurn(): void  
ver(String, int): void  
void  
PlayerPoints(String): void  
omptLine(): void  
Leaderboard(LinkedHashMap<String, Integer>): void  
StartingCardFace(Card): void  
ayerReconnected(String): void  
oid  
ableColor(): void  
Mode(boolean): void  
Manuscript(String): void  
NewPlayer(): void  
GameInfo(): void  
nnectionMode(): void  
aced(String, CornerCardFace, int, int): void
```



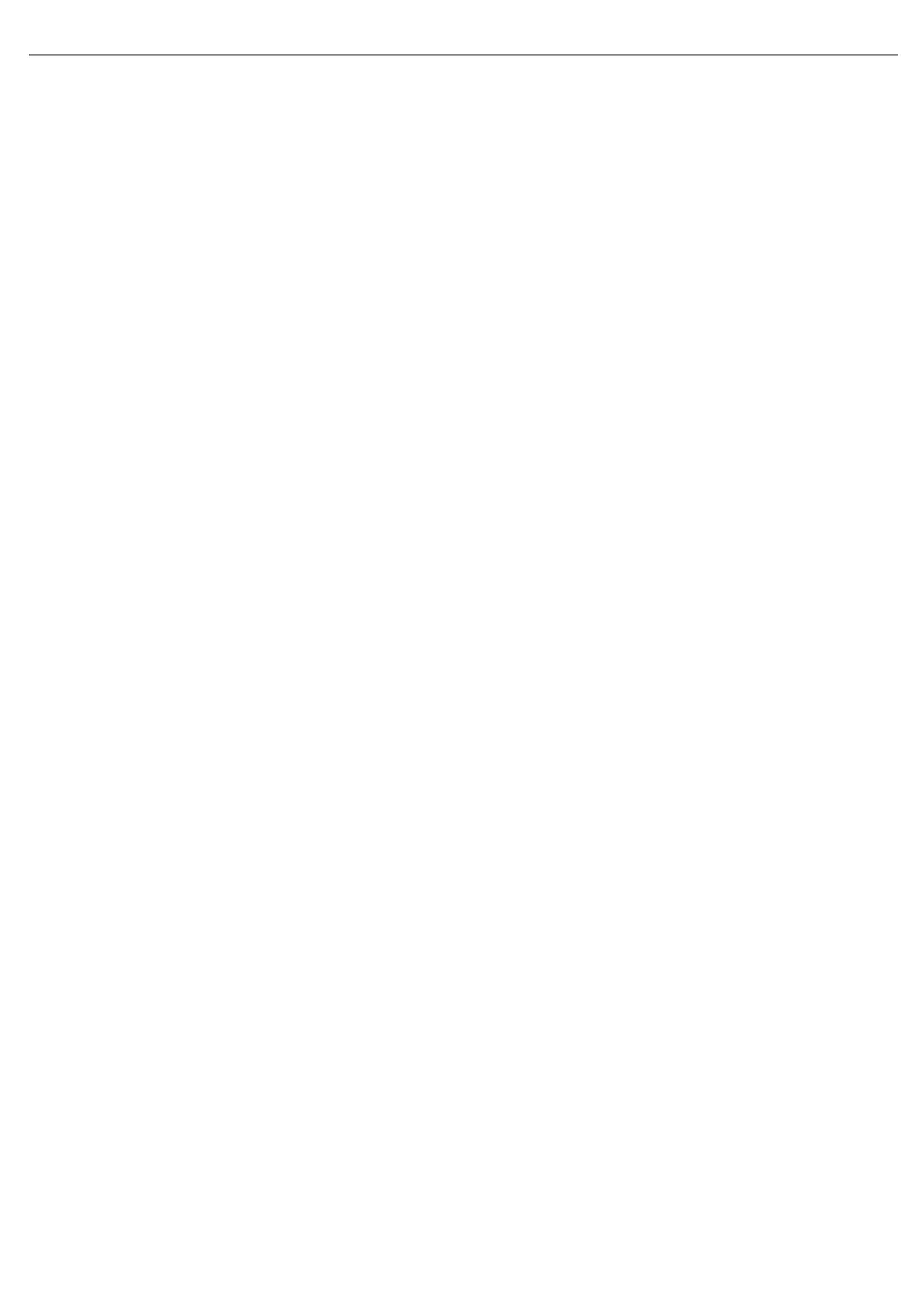














«create»



