

+ GameModel(...): costruttore al quale viene passata una lista di nickname dei giocatori e una lista dei corrispettivi tokens. Inoltre il costruttore chiama anche il costruttore di match in modo che vengano inizializzati i mazzi.

+ fetchPickables(): restituisce una lista di 6 CardPlayable, 4 da mostrare scoperte e 2 coperte ().

+ fetchStarter(Player player): restituisce al player la CardStarter che il model gli ha assegnato.  
+ fetchStarter(Player Side): metodo vuoto che indica al model quale lato della CardStarter il giocatore ha scelto

+ fetchHandPlayable(Player): restituisce una lista di 3 CardPlayable che identificano le carte in mano del player (meno la carta obiettivo).  
+ fetchHandObjective(Player): restituisce l'obiettivo in mano al dato player.

+ fetchCommonObjectives(): restituisce una lista di 3 carte obiettivo della sezione Decks (2 scoperte e una coperta).  
+ fetchPersonalObjectives(Player): restituisce una lista di 2 CardObjective che corrispondono alle 2 carte obiettivo tra cui il player deve scegliere

+ choosePersonalObjectives(Player, CardObjective CodChosenObj): imposta l'obiettivo personale scelto da ogni player.

+ playCard(CardSidePlayable, Coordinates): dice che il currentPlayer gioca la CardSidePlayable sul tabellone alle coordinate Coordinates  
+ pickCard(CardPlayable): indica la carta pescata dal currentPlayer (a scelta tra le 6 perdenti della "zona Decks").

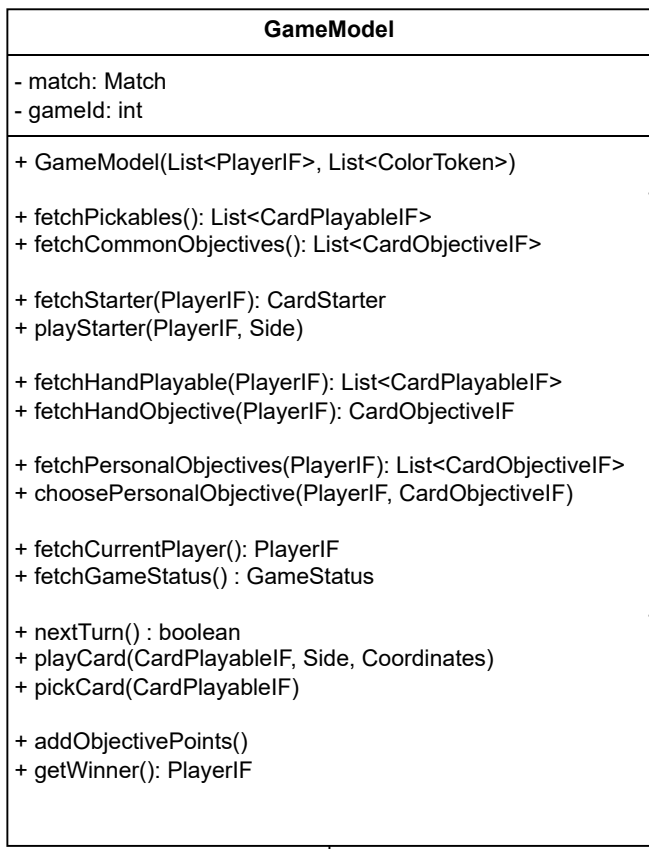
+ fetchCurrentPlayer(): restituisce al controller il currentPlayer. In questo modo se un giocatore fuori turno prova a fare un playCard, il controller controlla che sia uguale al currentPlayer, altrimenti restituisce al client un exception.

+ fetchGameStatus(): il controller può chiamare questo metodo per avere informazioni sullo stato della partita.

+ nextGame(): restituisce true se c'è un turno successivo, altrimenti ritorna false per indicare che siamo nella fase di "CALC\_POINTS"

+ addObjectivePoints(): aggiunge i punti derivanti dagli obiettivi (personali e comuni) ai punti di ciascun player.

+ getWinner(): restituisce il vincitore.



Turn, Async, Game Operations

Metodi chiamabili solo in fase gameStatus=INIT:  
getStarter(Player)  
playStart(Player, Side)  
getPersonalObjectives(Player)  
setPersonalObjective(Player, CardObjective)  
getCurrentPlayer()  
nextTurn()  
playCard()  
pickCard()  
Metodi chiamabili solo in fase gameStatus=IN\_GAME, FINAL\_PHASE  
Metodi chiamabili solo in fase gameStatus=CALC\_POINTS  
addObjectivePoints()  
getWinner()

