

MODEL

Abbiamo creato una classe centrale chiamata Game, che si interfaccia con il Controller. Contiene i principali metodi di gestione della partita.

La classe Game contiene come attributi le varie classi che rappresentano i vari componenti del gioco.

La classe Bookshelf rappresenta la bookshelf del singolo player. E' composta da una ArrayList di Column.

La classe Column è a sua volta una ArrayList di Tiles.

La classe Tile rappresenta la singola tile, è caratterizzata da una posizione, da un colore e da un tipo (gatto, vaso ...)

Il tipo e il colore sono delle ENUM, salvate in TileColor e TileType.

Per la posizione abbiamo optato per creare una classe dedicata per maggior chiarezza.

Per mappare le posizioni dei tile abbiamo creato una mappa.

Abbiamo poi creato un'altra mappa per rappresentare le Tiles sulla board

Abbiamo creato due classi CommonGoalToken e EndToken per rappresentare i token che fanno guadagnare punti durante e alla fine del gioco

Per i Personal Goal abbiamo utilizzato un file json contenente la descrizione di tutte le carte. Per ciascuna carta sono salvate le coordinate delle tiles richieste e il loro colore.

Per i Common Goal abbiamo creato una classe Padre e fatto 12 sottoclassi che overrideano il metodo check. I Common Goal vengono scelti randomicamente all'inizio della partita.

Per ultimo abbiamo una classe Bag, che contiene le tiles per refillare la board

CONTROLLER

La Virtual View è la rappresentazione remota della view. Contiene tutti i metodi ereditati dall'interfaccia IGameObserver, sono metodi che vengono chiamati quando il game notifica i propri observer, che sono le virtual view connesse a quella particolare istanza del gioco.

IGameObserver è l'interfaccia Observer che serve alla Virtual View per ereditare i metodi che le permettono di mettersi in ascolto del gioco

IGame è l'interfaccia Observable dalla quale eredita il game, eredita metodi che servono per notificare, registrare e rimuovere gli observer (Virtual View)

La classe Client Handler si occupa della gestione dei messaggi, arrivano i messaggi provenienti dal client e che passano per un message adapter, e in base a tale messaggio, il Client Handler chiamerà metodi specifici del GameController.

La Lobby si occupa di gestire tutti i Client Handler creati dal Server quando i giocatori si collegano; mette i client handler in una coda e quando raggiunge il numero corretto di giocatori, crea il game ed il relativo controller

Il Controller si occupa della gestione del game attraverso il GameController, che vien creato proprio dal controller

il GameController si occupa della gestione del game attraverso dei metodi che chiamano i metodi corrispondenti del game, quindi sostanzialmente agisce sul game chiamandone i metodi che ne modificano lo stato