

Commento uml GC-45

Nell'UML abbiamo la classe Server che servirà per gestire più partite e inizializzare le varie connessioni con i diversi client. La partita viene gestita dal controller che funziona anche come lobby e creatore del game. Quando viene aggiunto l'ultimo player tramite la add allora il controller chiama startGame. Il client, infatti, chiama le funzioni del Game passando prima per il controller. Game è l'interfaccia sia di un normalGame che di un expertGame e ci sono i metodi di entrambi (pattern decorator) ma se si provano a chiamare i metodi di expertGame in normalGame si lancerà un'eccezione che verrà presa dal controller e rilanciata al client. Per la gestione di 4 giocatori abbiamo creato una classe Team che avrà una lista da 1 o 2 player: questa classe servirà per la gestione del calcolo dell'influenza e delle torri. La gestione dei round avviene in Game tramite nextPlayer, nextPhase, playerOrder e roundIndex e actionPhase (conta in quale sottofase della fase azione sono). La classe astratta GameComponent è la superclasse di tutti i componenti sulle quali posso mettere gli studenti che sono gestiti come un array di 5 byte che conta quanti sono gli studenti di quel colore, dove il colore è l'indice dell'array. La gestione delle isole è fatta tutta in una sola classe secondo il paradigma: "se unisci due isole ottieni sempre un'isola". Cloud e entranceHall anche se sono vuote sono in realtà due implementazioni diverse della classe astratta GameComponent. (anche perché vogliamo bene a cloud adesso e non la togliamo). Per i character li gestiamo in expertGame con 12 classi che fanno l'override del metodo play e ogni carta si salva gli input di cui ha bisogno oppure setta dei boolean nel caso deve modificare influenza o calcolo dei professori.

Per ogni altro dubbio non esitate comunque a contattarci.

Grazie

Andrea Piras, Andrea Sanguineti, Emanuele Santoro