Peer-Review 1: UML

Peer-Review 1: UML

Davide Grazzani, Luca Muroni, Sara Mucci

Gruppo 46

Valutazione del diagramma UML delle classi del gruppo 47

Nota Introduttiva

Se un metodo/attributo di una classe non viene citato né positvamente né negativamente allora significa che lo reputiamo consono all'implementazione da voi fornita

Lati positivi

- **Game**
  - Ottimo il salvataggio del numero di giocatori, l'ordine di gioco e il giocatore corrente
  - Gestione consona dei Wizard come enum
- **Cloud**
  - La size di una Cloud è sicuramente una scelta ottima per poter continuare a riempire le nuvole con la giusta dimensione
- **Player**
  - Buona idea mettere il colore delle torri come attributo del giocatore
- **Assistant**
  - Piaciuta l'idea di utilizzare un enum per rappresentare le carte asistente
- **Arcipelago**
  - Interessante unire due isole in una classe Arcipelago

Lati negativi

- **Generale**
  - Nessuna separazione tra attributi e metodi da esperto rispetto a quelli base
  - Meglio specificare più cardinalità (UML)
- **Character, effect e gli effetti**
  - Non capiamo il motivo per cui Character debba essere una classe astratta

- **Game**
  - Manca un Wizard (sono 5 in totale)
  - Metodo start() potrebbe essere da controller se non utilizzato come inizzializatore
- **Player**
  - Ci sono dei metodi che sembrano da Controller (tipo endturn())
  - Per noi è sbaglito utilizare tanti attributi&metodi boolean per la gestione del flusso di gioco(canchoseAssistant())
- **Bag**
  - A cosa serve il metodo put? Non può la borsa generare da sola gli studenti?
- **Board**
  - scomodo l'utilizzo di int per mappare i colori di professori e studenti
  - difficoltà nel chimare mergeIsland. Come ottengo le isole da passare?
  - metodi move sembrano mancare di parametri necessari al funzionamento dei metodi stessi
- **Arcipelago**
  - Viene utilizzata in una maniera molto complicata, difficile stabilire solo da UML il corretto funzionamento (non entriamo nel dettaglio proprio per tale motivo)

Confronto tra le architetture

- Conferma di un'idea pregressa di mettere il colore delle torri come attributo della classe Player
- Spostata la gestione del currentPlayer e dell'array di tutti i giocatori nella classe Game come attributo quindi del package Model
- Aggiunta di int per indicare la dimensione delle nuvole che va mantenuta costante a runtime
- Punto di forza è la classe arcipelago, piace molto l'idea di avere una classe unica che gestisce più isole che sono state unite