

Peer-Review 1: UML

Andrea Somaini, Agnese Taluzzi, Aaron Tognoli

Gruppo GC-04

Valutazione del diagramma UML delle classi del gruppo GC-14.

Lati positivi

UML e relazione molto precisi e ordinati, con una relazione esaustiva senza essere inutilmente verbosa.

Apprezziamo l'uso di un file JSON per tenere traccia delle configurazioni standard delle partite rispetto alle diverse modalità di gioco, evitando così di inserire numerose costanti nel codice.

Forte utilizzo del paradigma object oriented, con una divisione ben caratterizzata delle diverse classi e una separazione opportuna delle loro responsabilità.

Lati negativi

Factory method over ingegnerizzato: per esempio la classe IslandFactory poteva benissimo essere implementata direttamente con un metodo privato del Model, senza dover scrivere una classe apposita.

Per la classe Assistant i campi dati sono entrambi delle enumerazioni quando potrebbero tranquillamente essere degli interi, al limite si potrebbe controllare a runtime se questi valori interi appartengono all'intervallo di quelli validi.

Non è ben chiaro come hanno intenzione di implementare le carte character che modificano le regole del calcolo dell'influenza, come Knight Centaur e MushroomSeller. Si potrebbe pensare per esempio ad inserire dei flag nel model per notificare l'attivazione di una di queste carte oppure si potrebbe implementare un pattern strategy su calculateInfluence.

Confronto tra le architetture

Noi avevamo deciso di fare scegliere agli utenti la posizione iniziale di madre natura, mentre GC-14 ha deciso di metterla sempre nell'isola zero. Siccome le regole sono vaghe da questo punto di vista, crediamo che entrambi gli approcci siano corretti.

GC-14 ha implementato la lista di isole come lista di “gruppi di isole” mentre nel nostro caso il concetto di isola comprende sia l’isola singola che 2 o più isole unite tra di loro. La differenza è il calcolo dell’influenza, che a nostro avviso è un po’ più complicata rispetto al nostro approccio.

Ragionamento inverso per i players, che noi teniamo separati mentre loro li uniscono a squadre. La differenza sta nel calcolo dell’influenza, nel nostro caso dobbiamo lavorare sugli indici nell’array che contiene i players mentre loro li hanno già raggruppati a squadre, a discapito però di un approccio che a nostra opinione è un po’ macchinoso.