

Peer-Review 1: UML

Santandrea, Panceri, Ruzza, Villa

Gruppo 33

UML - MODEL:

- 1. Gestione delle Carte:** Abbiamo suddiviso le carte in due categorie: ObjectiveCard e PlaceableCard. Le carte obiettivo utilizzano uno strategy pattern per calcolare i punti di risoluzione, mentre le PlaceableCard gestiscono la logica degli angoli (occupato/aperto, visibile/non visibile) e lo stato (fronte/retro) sul tavolo. Abbiamo anche gestito le StartingCard all'interno delle PlaceableCard separandole dalle ResourceCard e GoldCard; le seconde utilizzando il strategy pattern per il calcolo dei punti a piazzamento (Cfr. *Cards diagram*);
- 2. Gestione dei Mazzi di Carte:** Abbiamo progettato una classe per i Deck per gestire i mazzi di carte, definendo le classi ObjectiveDeck, che contiene gli obiettivi comuni dei giocatori e non può essere pescato durante il gioco, e PlaceableDeck, che unifica i mazzi di carte normali e oro utilizzando un pattern Abstract Factory per generare le carte (Cfr. *Decks diagrams*);
- 3. Classe Board:** Abbiamo progettato una classe Board che funge da punto centrale del gioco, fornendo un'interfaccia per i giocatori e contenendo i mazzi di gioco e le aree di gioco di ciascun giocatore (cfr. *Gametable diagram*);
- 4. Gestione dei Giocatori:** La classe Player rappresenta ciascun giocatore, con informazioni come nome, colore e numero di giocatore. Contiene la mano del giocatore (PlayerHand) e i relativi punteggi (cfr. *Gametable diagram*);
- 5. Gestione dell'Area di Gioco:** La classe PlayArea contiene la mappa delle carte e una mappa delle risorse visibili per ogni giocatore, aggiornata durante il piazzamento delle carte. Inoltre, gestisce una lista di angoli liberi, anch'essa aggiornata durante il gioco. Per implementare la mappa, abbiamo utilizzato una classe di matrice dinamica che sfrutta due hashmap innestate (cfr. *Gametable diagram*);
- 6. Pattern Observable e Observer:** Abbiamo implementato il pattern Observable e Observer per notificare alla View e quindi al client le modifiche del model durante il gioco(cfr. *View-Model diagram*).