

# My Shelfie

Phil Walker-Jarling  
& Matthew Dunstan

**Xavier Adrian Pacheco Murgueitio**  
**Maria Diletta Screpis**  
**Alfonso Shytani**  
**Lorenzo Vaninetti**



POLITECNICO  
MILANO 1863

# 1 Presentazione UML

Metodi la cui funzione è auto esplicativa, quali **getName()** e/o **setFinalToken()**, sono esenti dalla descrizione sintattica dell'UML proposta, al fine di rendere il testo citato il più agevole possibile a coloro che non hanno lavorato in modo diretto al progetto e consentire una lettura scorrevole di quest'ultimo. Il testo, per quanto semplice possibile, focalizza l'attenzione sugli aspetti fondamentali per la comprensione dell'implementazione del gioco in corso. L'UML del Model sviluppata presenta nove classi pubbliche principali e due enumerazioni (nello specifico **CommonGoalType** & **ItemTileType**). Le classi proposte per lo sviluppo funzionale del gioco sono le seguenti:

1. GameModel;
2. GameBoard;
3. Player;
4. ItemBag;
5. Shelf;
6. ItemTile;
7. LivingRoom;
8. CommonGoal;
9. Personal Goal;

Le entità principali dello schema si identificano nel **GameModel** e **GameBoard**: alla prima classe spetta l'arduo compito di impostare il numero massimo di giocatori, verificare il numero effettivo di essi, impostare la **chair**, la quale determina l'ordine con cui i giocatori approcceranno la partita, stabilire il vincitore della partita corrente e richiamare la plancia di gioco; la GameBoard, traduzione di plancia da gioco, implementa tutti quei metodi necessari all'effettiva costruzione del gioco fisico: richiama la **LivingRoom**, costruita come matrice di tiles, e la **ItemBag**, rispettivamente il salotto su cui piazzare le **Tiles**, le quali sono identificate univocamente da un id univoco e un colore specifico, e la sacca per l'estrazione di # delle 132 tessere disponibili. Al fine di consentire un flusso di gioco continuo la LivingRoom controlla che vi siano abbastanza pedine sul tabellone, in caso contrario effettua un pescaggio dal Bag ed inserisce le tessere nelle caselle libere in modo del tutto casuale. Le tiles, le quali rappresentano il cuore del gioco, possiedono due attributi fondamentali per consentire lo scorrimento della partita: **availability** e **adjacency**, rispettivamente la reperibilità della pedina (secondo le regole del gioco) e l'adiacenza.

Oltre alle classi descritte, il **Player** rappresenta l'oggetto di maggiore rilevanza: viene qui usata per identificare i giocatori ammessi alla partita, ai quali viene assegnato/richiesto un nickname univoco e associato il numero di punti totali ottenuti durante lo svolgimento di tutto il gioco. Secondo le regole di MyShelfie, ogni giocatore potrà accedere alla consultazione dei due CommonGoal stabiliti alla creazione del GameBoard, al fine di soddisfarle per accumulare punti e vincere la partita. Viene inoltre presentato un attributo descritto come **finalToken**, utile per consentire l'analisi dell'ultimo turno di gioco e decretare la fine di quest'ultimo. All'inizio del gioco, ad ogni giocatore viene assegnata una propria libreria chiamata **Shelf**, definita come matrice di itemTile, sulla quale ogni giocatore potrà impilare le proprie tessere, pescate ad ogni turno, e disporle secondo le regole stabilite dal regolamento ufficiale di MyShelfie, al fine di riempirla prima degli avversari ed, in parallelo, conseguire l'adempimento delle **CommonGoal** e delle **PersonalGoal** assegnate. Come anticipato, le CommonGoal, le quali sono 12 all'interno del gioco, sono state listate attraverso un'enumerazione ed alla creazione della partita vengono selezionate in modo casuale e poste sulla LivingRoom. Discorso simile vale per il PersonalGoal: una volta creata la plancia e stabilito il numero di giocatori, vengono assegnati i rispettivi PersonalGoal, definiti qui come attributi veri e propri della classe Player. Alla fine di ogni turno, il GameModel effettuerà un check su tutti i giocatori per verificare l'adempimento dei propri obiettivi e di quelli comuni, con lo scopo di assegnare in modo corretto i punti stabiliti e non creare disambiguità sullo stato del gioco.