Model

Bookshelf

- game_tiles: ItemTile[6][5]
- isFull: Boolean

+ get_bookshelf(): ItemTile[6][5]
+ get_isFull(): Boolean
+ column_is_full(n: Int): Boolean
+ get_isFull_status(): Boolean
+ set_isFull_status(): void
+ set_tiles(column: Int, tiles: List<ItemTile>): void

1 — Owns — 1

Player

- name: String
- points: Int
- goal: PersonalGoal
- first_to_finish: Boolean
- winner: Boolean
- commonOne: Boolean
- commonTwo: Boolean
- bookshelf: Bookshelf

// getter
// setter

1..4

Game

- players: List<Player>
- board: Boardgame
- common_goals: List<CommonGoalCard>
- order: Int[]
- current_turn_player: Int
- last_turn: Boolean

+ get_common_goals(): List<CommonGoalCard>
+ set_common_goals(common_goals: List<CommonGoalCard>): void
+ get_order(): Int[]
+ get_current_turn_player(): Int
+ get_last_turn_status(): Boolean

BoardGame

- board: ItemTile[9][9]
- empy_cells: Int

+ get_board(): ItemTile[9][9]
+ set_tiles(List<ItemTile>): void
+ get_tile(x: Int, y:Int): ItemTile

1..132

ItemTile

- type: ItemType
- in_game: Boolen

+ get_in_game_status(): Boolean

<<enumeration>>
ItemType

Cats
Books
Games
Frames
Trophies
Plants

1..24

{abstract}GoalCard

- id: Int
- explanation: String

PersonalGoalCard

- description: ItemTile[6][5]

+ get_description(): ItemTile[6][5]

CommonGoalCard

- stack: List<ScoringToken>

+ get_stack(): List<ScoringToken>
+ set_stack(scoring_tokens: List<ScoringToken>): void

8

ScoringToken

- roman: RomanType
- points: Int

<<enumeration>>
RomanType

One
Two

Controller

PlayerController

- model: Player
- game: GameController

+ set_goal(model, personal_goal: PersonalGoalCard): void
+ feed_column(model, tiles: List<ItemTile>, column: Int): void
+ check_goal(model, g: GoalCard): void
+ get_tile(game, x: Int, y: Int): ItemTile

1..4

GameController

- model: Game
- name_game: MyShelfie
- players: List<PlayerController>
- board: BoardGame

+ select_first_to_play(): Int
+ set_common_goal_cards(model: Game, name_game: MyShelfie): void
+ associate_scoring_tokens(name_game, n: Int): void
+ set_personal_goal(player: PlayerController, name_game): void
+ restore_board(board, name_game: MyShelfie): void
+ play_turn(player: PlayerController): void
+ start_game(): void
+ end_game(): Player

MyShelfie

- tiles: List<ItemTile>
- personal_goal_card: List<PersonalGoalCard>
- common_goal_card: List<CommonGoalCard>
- scoring_tokens: List<ScoringToken>

// getter
+ select_personal_goal(): PersonalGoalCard
+ select_common_goals(): List<CommonGoalCard>
+ select_scoring_tokens(n: Int): List<ScoringToken>
+ select_tiles(n: Int): List<ItemTile>