ECON381 Fall 2024
Homework Assignment 2

1. During game play you will add and remove keys to the board. What kind of operations would that mean? Please elaborate.
2. To determine if a user is done, what kinds of checks would you need to do? Please elaborate.
3. Given the OkeyKey class available and given your discussion for the above two topics,would you rather hold the 14 keys in the Okey board in a single fixed size Java array?Or would you have multiple arrays or linked lists to hold the blocks? Please elaborate.

2.1

Adding Tiles:

- Tiles can be added using the `addKey` method in the `OkeyBoard` class.
- This process involves adding tiles to a movable collection (e.g., an `ArrayList`).

Checks:

- If the board is full (limit of 14 tiles), no tile can be added. In this case, an `IllegalStateException` is thrown.

Removing Tiles:

- Tiles can be removed using the `removeKey` method.
- This involves removing a specific tile from the collection.

Checks:

- If the tile does not exist on the board, an `IllegalArgumentException` is thrown.

Example Operations:

1. At the start of the game, tiles are added to the board in sequence.
2. During the game, a player can remove a tile from the board (e.g., to form a block of tiles).
3. When joker rules are added, replacing a joker works similarly to adding and removing tiles.

2.2

Determining If a Player Has Won:
The following checks must be performed to determine if a player has completed the game:

1. **Are All Tiles Grouped Into Blocks?**
    - All tiles must form blocks:
        - Consecutive numbers of the same color (e.g., 2-3-4).
        - Same number in different colors (e.g., red 11, blue 11, yellow 11).
    - To verify this:
        - Sort tiles by color and number.
        - Check if consecutive tiles form valid blocks.
        - Ensure all tiles are included in at least one block.
2. **Seven Pairs Check:**
    - Does the player have 7 pairs of tiles (e.g., seven pairs of the same number and color)?
    - To verify pairs:
        - Group tiles by number and color.
        - Check if each group contains pairs.

Additional Challenges:

- **Joker Tiles:**
    - The joker can represent any tile, so its optimal use must be determined during win checks.
- **Fake Joker Tiles:**
    - Fake jokers neutralize the joker tile, which impacts the checks.

2.3

Preferred Data Structures:
I prefer using multi-dimensional arrays or linked lists because grouping tiles into blocks is a fundamental part of the game.

Dynamic Block Management:

- Since tiles in Okey are constantly added or removed, each block can be represented as a separate list. For example:
    - A group of consecutive numbers (e.g., red 2-3-4) can be stored in one list.
    - A group of same numbers with different colors (e.g., red 11, blue 11, yellow 11) can be stored in another list.

Easy Block Validation:

- To check if the player has completed the game, we need to verify that all tiles are grouped into blocks. Representing each block as a separate list makes this process simple and efficient.

Flexibility for Jokers and Fake Jokers:

- The joker tile can be added to any block, so it might need to move between blocks. Using lists for blocks allows dynamic movement of tiles.
- When the fake joker replaces the actual joker, adjustments in blocks are required, which is easier with lists.

Limitations of Fixed-Size Arrays:

- A single fixed-size array is sufficient for storing all tiles. However, since dynamic block management is a core requirement of the game, multi-dimensional arrays or linked lists are clearly more advantageous.
- These structures enhance modularity, maintainability, and performance of the code.

Barış PALA
20232810024