20232810008

Sedanur Gültekin

ECON381 Fall 2024
Homework Assignment 2

Question 1 :

Adding and deleting keys during gaming entails the following actions**:**

Adding a key: When adding a key to the board, it's important to consider the placement and order of the current keys. This could entail moving old switches and installing a new switch in a particular location. For instance, a key must be added and positioned appropriately without interfering with the other keys' order.

Key Removal: When taking a key off the board, it's important to think about which key to take off and how doing so will impact the other keys. This could entail keeping the remaining keys in their original arrangement but removing a specific key. For instance, removing a key can require rearranging other keys or filling the empty space.

1.Adding Keys:

Insertion into Data Structure: The board's representation requires the insertion of a new OkeyKey object. It could be necessary to keep the insertion in a certain order, as by color or number.

Dynamic Allocation: If required, extra memory is set aside to hold the added key in dynamic structures like linked lists.

Complexity:

Array: Finding the ideal location and moving components to make room for the new key may be necessary for the array.

Linked List: To add a new key, the references of neighboring nodes must be updated.

2.Removing Keys:

Search for Key: Determine which key has to be taken out. A search operation is necessary for this.

Deletion: Take the data structure's key out.

Complexity:

Array: Shifting items are needed in an array to fill the space left by the deleted key.

Linked List: To get around the deleted key, update the pointers of nearby nodes.

3.Key Reordering: To create valid blocks, keys frequently need to be rearranged. This entails arranging or switching keys according to their characteristics, including color and number.

4.Validation after operations: Validation may be necessary for each addition or deletion to ensure that the updated configuration complies with the block or pair rules.

Question 2:

The following tests must be made in order to ascertain whether a player has finished the game:

1.Validation of Blocks:

Definition: At least three consecutive numerals of the same color make up a block.The same number in several hues.

How to Verify:Sort keys by number or color.Within each group, sort the keys.Check to see if the grouped keys create legitimate blocks.

2.Validation of Pairs:

A pair is defined as two keys that are the same color and number.

How to Verify:Count how many times each key appears.Verify that there are precisely seven pairs, making sure that no keys are missing or excessive.

3.Complete Key Utilization:

Make that there are no leftovers and that all 14 keys are a part of legitimate blocks or pairs.

4.Edge Situations:

Check to make sure no regulations—like the minimum block size—are being broken.

Verifying that the user has completed creating:

-The block creation: Verifying that all 14 keys form blocks of three or more sizes is essential. A block may consist of successive numbers in the same color (red 2, 3, 4) or the same number in a variety of colors (11 in three distinct colors, for example). To ascertain if the user is completed, it is crucial to verify that these blocks have been produced appropriately.

-Pair Creation: It is required to verify that there are seven pairs (each with the same color and number). To ascertain whether the user is done, it is crucial to verify that these pairings are established appropriately.

Question 3:

First Choice: Single Fixed-Size Array

Benefits:Having a fixed size makes management easier because there are always 14 keys.
gives constant-time index-based key access.
Effective for easy retrieval and storing.

Drawbacks:Block and pair keys are challenging to dynamically group and rearrange.
Additional logic is needed for validation, sorting, and grouping.

Second Choice : Linked Lists or Multiple Arrays

Benefits:enables keys to be dynamically grouped into pairs or blocks.

Rearranging and validating are made easier with a flexible framework.
Insertions and deletions are made more efficient via linked lists.

Drawbacks:more difficult to administer and execute.
a little cost brought on by linked lists' dynamic memory allocation and pointer management.

Data Structure for Key Storage:

-One Array of Fixed Size: It could be enough to utilize a single fixed-size Java array to store 14 keys. This method of handling keys is straightforward and easy. You may simply maintain track of the keys' layout and order by utilizing a single array.

-Numerous Arrays or Linked List: It may be more effective to handle blocks using numerous arrays or linked lists. Group key management and organization may become simpler as a result. For instance, you may increase the flexibility of block generation and administration by utilizing a different array or linked list for every block.

Preferred Method
Multiple linked lists, each of which represents a block or pair, are what I would want to use. Given that the keys must be regularly moved into groups, this strategy fits in nicely with the gaming mechanics. Effective insertion and deletion are made possible using linked lists, which is crucial for dynamically changing the key layout while playing the game.

In conclusion, the usage of numerous linked lists is the best option for depicting the game board since it improves flexibility and reflects the dynamic nature of Okey gaming.