Lencho Burka Isabel Baltazar Raeciel Reyes Kimi Zhou

Etude 10 - Pentominoes Data Structure Proposal

The Pentomino Puzzle

Given an empty board and 12 Pentomino pieces, find a way to fill the board using the pentomino pieces and the pieces cannot overlap.

Representing the Pentomino Board

The board will be represented through a 2D array of Piece enumerated types. The board is initialized to have all unfilled spaces as empty.

Representing Pentominoes

Each pentomino is made up of 5 squares which are be represented as a 2D array.
There are 12 unique pentominoes labelled O-Z, and each pentomino can be rotated or flipped to show a mirror representation.

The R, Q, S, P, Y pentominoes have 6 rotations and mirror flips.
The T,U ,V, W, Z pentominoes have 4 rotations.
The O Pentomino has 2 rotations.
The X Pentomino has 1 rotation.

Including all rotations and mirror flips, there are 66 possible pentomino placements.
We will represent each pentomino using a Piece enumerated type, a Pentomino class and a class that implements the Piece interface.

The Piece enumerated type

The `Piece` enumerated type represents a pentomino rotation and its `String` representation. Each `Piece` type will represent a pentomino piece.

The `Piece` enumerated type also has `EMPTY` and `INVALID` types, which are used to represent unfilled spaces on the pentomino puzzle board and invalid spaces which should not be filled in.

The Pentomino class

The `Pentomino` class contains one static method, `placeOf(),` which takes a `Piece` as an argument.

`placeOf()` has a switch statement that returns an array of coordinates associated with the `Piece` parameter.

For example,

```
O O O O
- - - -
- - - -
- - - -
```

This board represents the `Piece.O` pentomino, which is the O-pentomino and its horizontal rotation. The `Piece.O1's String` representation is O.

```
- - X -
- X X X
- - X -
- - - O
```

This board represents the `Piece.X` pentomino, which is the X-pentomino. The `Piece.X's String` representation is X.

For example,

```
- - - -
-     -
-     -
- - - -
```

This puzzle board has invalid spaces at (1,1), (1,2), (2,1), and (2,2), so that we can represent a 4 x 4 board with a rectangular square in the middle. The rest of the board is represented through the `EMPTY` enum type.

Lencho Burka Isabel Baltazar Raeciel Reyes Kimi Zhou

Finding The Solution

The puzzle can be solved using Depth First Search. If we find all possible pentomino placements on the board, then we can form an adjacency list of all pentominoes that can fit at the same time.

Two pentomino placements are adjacent if neither of the two contain the same coordinate pairs.

For example,
A = ((0,1), (0,2), (0,3))
B = ((0,1), (0,5))
C = ((0,4),(0,6))

AC and BC are adjacent because neither sets contain a coordinate pair already present in the other set.
AB are not adjacent because (0,1) is represented in both sets.

We can use Depth First Search on the adjacency list to find a solution to the puzzle.