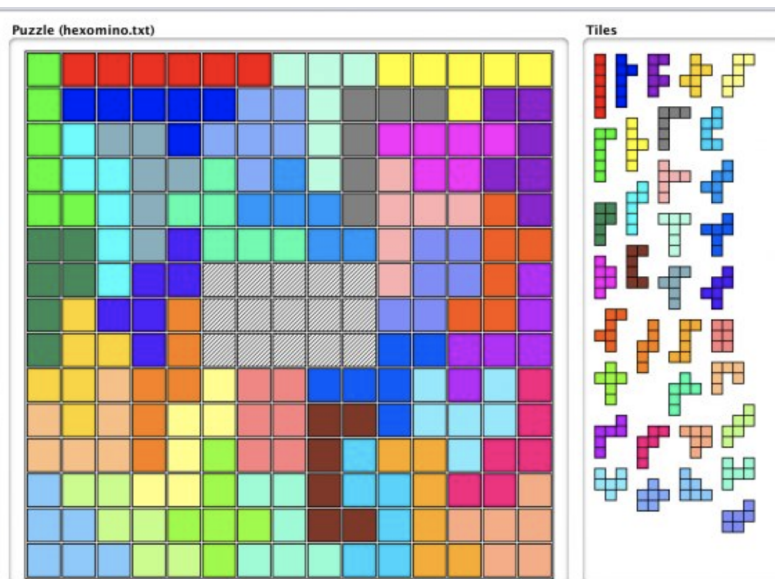


Polyomino Tangram

The Tangram is a puzzle where the user must arrange geometric shapes to completely fill a square. In this task, you have to implement Tangram but using Polyomino blocks. Polyominoes are geometric figures that are formed by joining---edge to edge---squares of unit dimensions. They have different names based on the number of unit squares they have. For example, monomino is just a single square of dimension 1 x 1, domino is a polyomino with two squares of dimension 1 x 2, and so on. A polyomino with three squares can be of two possible shapes: an L-shaped polyomino, or a 1 x 3 rectangular-shaped polyomino formed by three squares placed side-by-side. As you must have seen, such polyominoes form the building blocks of famous games such as the computer game Tetris, and various other tiling puzzles.

Task 1- Create an interface using the programming language of your choice to visualize the puzzle. The interface should have the following abilities -

1. When the game starts, a set of random polyominoes is displayed on the screen along with a 10x10 square-grid playing board to place the polyominoes on. These polyominoes can have up to 5 squares, i.e., each polyomino can be as large as up to pentomino. Each time the game is started, your program should randomly pick how many polyominoes are displayed on the screen, and each of their sizes and shapes. Please describe the underlying random selection rule you have employed to pick the polyominoes, to make the game interesting (see the remaining rules of the game below). An example interface is shown below -



(source: <https://favpng.com/>)

2. The player should then be able to drag and drop the polyominoes into any one of the currently-unoccupied empty spaces (of the 100 spaces) in the playing board. When they drag a polyomino into the playing board, the angular orientation of the polyomino should not change, but they can move it up/down or sideways to place it in an empty space in the playing board. A move cannot be reversed. The player continues to drag and drop polyominoes from the provided set into empty spaces in the playing board until: (a)

either they run out of polyominoes, or (b) at least one column or one row of the playing board has all squares occupied.

3. The game should stop and declare victory as soon as all 10 spaces of at least one column or row in the playing board gets occupied.
4. The game should stop and declare failure if the player runs out of polyominoes before the objective of filling up one column or row was accomplished successfully.
5. Write a brief pseudocode for this task.

Task 2 – Modify the interface developed in Task 1 to add a clockwise rotation button, such that the user can select a polyomino from the given set and then click on the rotation button (multiple times if they feel necessary) to get the desired orientation of the block, before they place it in the playing board. The rest of the rules of the game stay as in Task 1. Modify the pseudocode for Task 1, to show the new step(s).

Task 3 – Now modify the interface such that before the game begins, 10 randomly-chosen squares (of the 100) in the playing board are blocked out, which cannot be occupied. You may paint them black. You may choose (and describe in the pseudocode) how you select the squares to block. If the user drops a polyomino such that one or more squares of that polyomino would occupy any of the blocked squares, the polyomino should immediately jump back to the displayed set at the top of the screen. The rest of the rules of the game stay as in Task 2. Modify the pseudocode for Task 2, to show the new step(s).

Submission deadline - Monday, Feb 7, 8am MST.

Submission format - Submit a .zip file named 'YourFirstName_EWD.zip' containing

- a pdf with the answers. Include your simulation results as well in this file.
- the code files

If you have any questions, please contact ashlesha@email.arizona.edu.

Additional Notes: You may use open-source repositories available on the internet (such as [PolyominoBlocks](#) and [Tiling Polyominoes Game](#)) as your starting point provided that they are properly cited. If you cannot complete a task, write down the steps you have taken to implement it.