

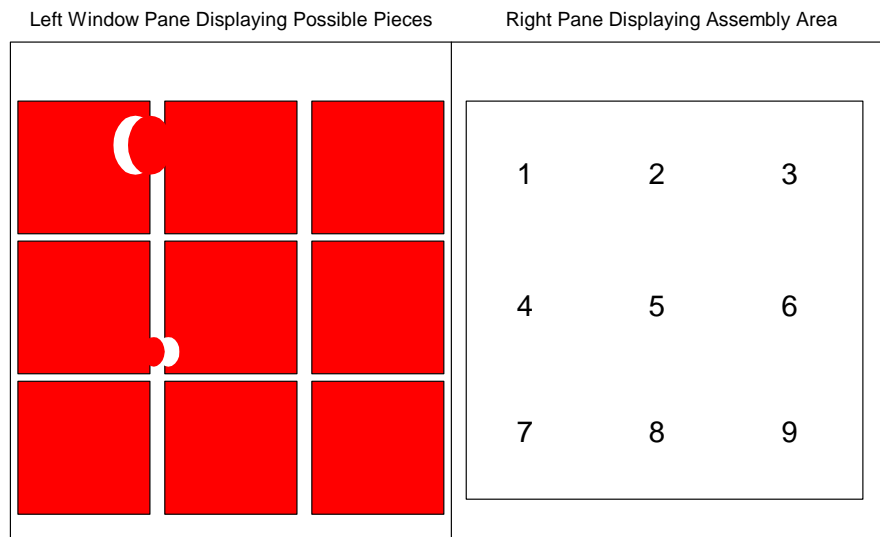
EGR 106 Project Proposal: Puzzle Creator

Objective: Create a puzzle game using a user's specified photo.

Input: The only input to the game is a picture file (.jpg, etc.). For example the one shown below:



Output: Create a program that will display at least 9 puzzle pieces produced from the photo. The pieces are displayed in random order on a left window pane in the figure window. A working puzzle assembly board is displayed in a right window pane. The user can drag and drop pieces one at a time, into their proper place on the assembly board on the right to recreate their photo. The program indicates when the pieces are placed properly.



Core Skill: The core problem is to pull apart matrices of data representing the color of each pixel in the imported photo using matrix manipulation and assignment operations.

Other functions: imread, figure, randomization, if-then and loop statements to execute sections of the code effectively

One advanced function: determine an effective method to cut and remove portions of square puzzle pieces (array data) to create non-straight edge puzzle pieces. Creating a semi-circle shape as shown above, for instance, you must remove a certain “cut” of data from a square matrix.

Suggested Steps

1. Input a color photo of any size, using the `imread` command. This command creates a 3-D array of data for the 2-D photo, where each element represents a value for Red, Green, or Blue for each pixel in the photo. For example, to determine the color of the pixel at (2,3), you would look at the RGB triplet stored in (2,3,1:3). Suppose (2,3,1) contains the value 0.5176, (2,3,2) contains 0.1608, and (2,3,3) contains 0.0627. The color for the pixel at (2,3) is 0.5176 0.1608 0.0627
2. First build puzzle pieces from the photo with non-intricate sides, just straight.
 - a. Divide the photo 3-D array into 9 equal parts, vertically and horizontally. This step creates 9 different 3-D arrays. This involves counting the number of elements on each dimension of the array and dividing by 3. Once the size for the new 9 arrays are determined, the large array can be pulled apart using matrix manipulation with values assigned to 9 new arrays.
 - b. Create a figure window that will display the 9 individual blocks of the puzzle randomly on the left hand side window pane.
 - c. Create the right window pane as shown above which outlines a workspace and shows background numerical markers for guides for pieces.
 - d. Allow the user to interactively grab and drop pieces on the puzzle. Each piece can only have two positions, the one where it was grabbed from and the one where it is dropped. If the piece is removed from the puzzle workspace and placed back on the left hand side, then the piece is dropped back into its original position on the left
 - e. Write the code to check whether pieces are placed in the proper spot by the user. Once all the pieces are in the proper spot, a message should appear in the figure window telling the user congratulations.
3. Extend this program by adding a least one non-straight edge to each edge of each puzzle piece. And re-create the puzzle game.