**MSU CSC 540/640**

**Project: Jigsaw Puzzle**

**Stage 1. Identify corner and edge pieces**

*Prematurely given: Fri., Sept. 7, Due: Mon. Sept. 17*

**Revised and re-issued: Wed. Oct. 10, Due: Mon. Oct. 22, 11:59pm**

**Group work is OK – each person turns in a report**

This project, in several stages, will assemble a jigsaw puzzle. The puzzle has 550 pieces of various irregular shapes and sizes, approx. 3-5 cm. The completed puzzle is 61 cm wide and 45.7 cm high.

For this stage, you are given a Python function that returns a sequence of *(x, y)* coordinates of each piece, all around the piece in a clockwise direction back to the origin of the coordinate sequence. There is no provided indication or estimate of what might be interpreted as a “side” of the piece. Each piece will be given a random, unique, sequential identifying number.

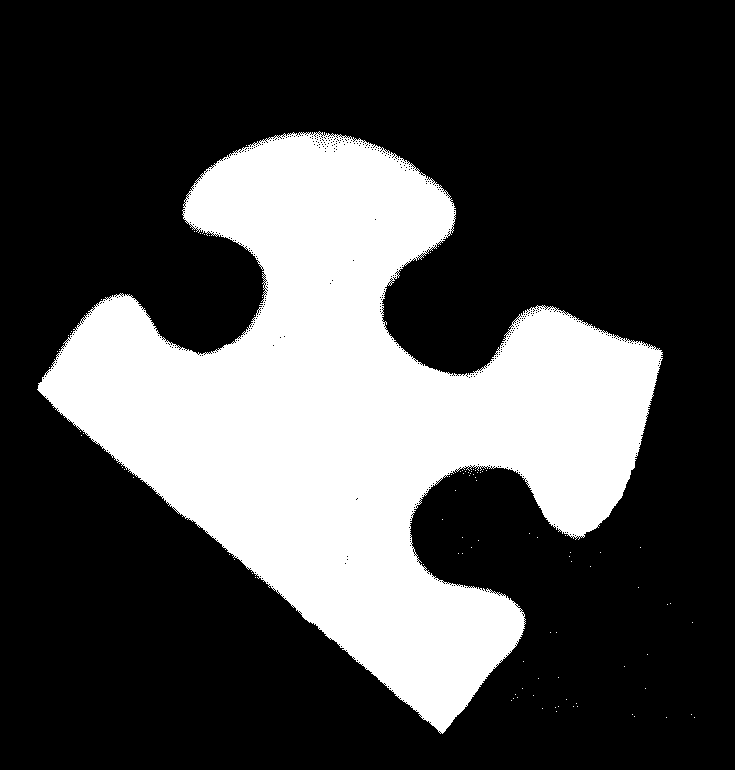
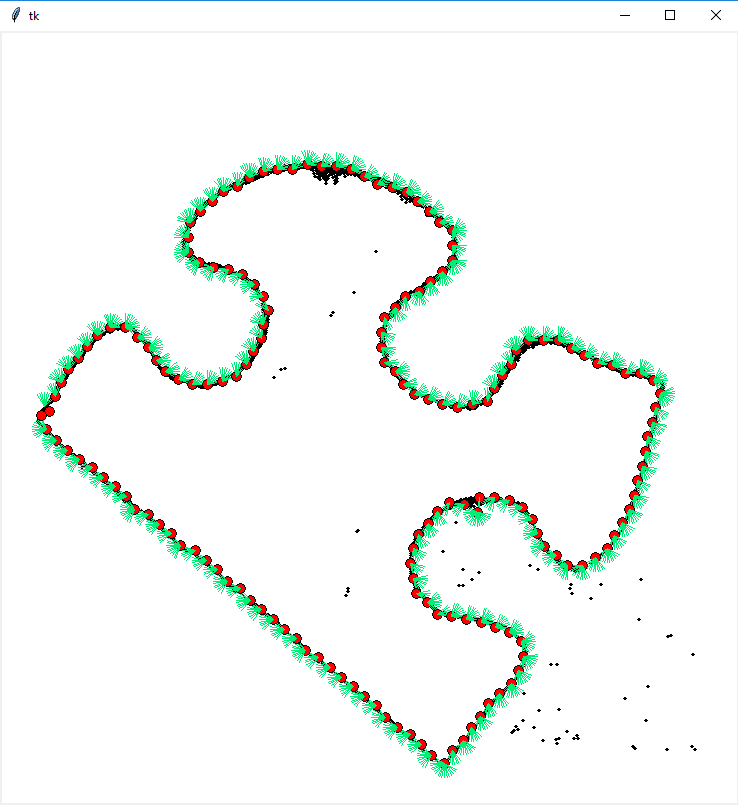
The puzzle piece edge coordinates are the last step of a series of stages based on image processing operations.

**Provided:** Black-and-white silhouette data files of five puzzle pieces, and a Python program that finds points on the color boundary in the Black-and-white silhouette image. The Python program displays the boundary graphically and as a printed list of values.

*Run the program with the input data file in the command line (or rewrite to run differently).*

*There are five input data files, P?edges.png*

**python ReadEdgeFromImage\_v4.py P1edges.png**

**Turn in:** Each person turns in to BlackBoard a **one-page** Word document and some Python program.

* First, do the Python program, because it produces some analysis of the shape of the puzzle piece boundary. *(For example, for a piece with a straight edge, the program enables you to conclude that the N points between HERE and THERE are in a straight line.)* The Python program could be a modification of the provided program, or a totally separate program that interprets the list of coordinates obtained from the provided program.
* Then, do the **one-page** writeup. Describe how you recognized straight lines.