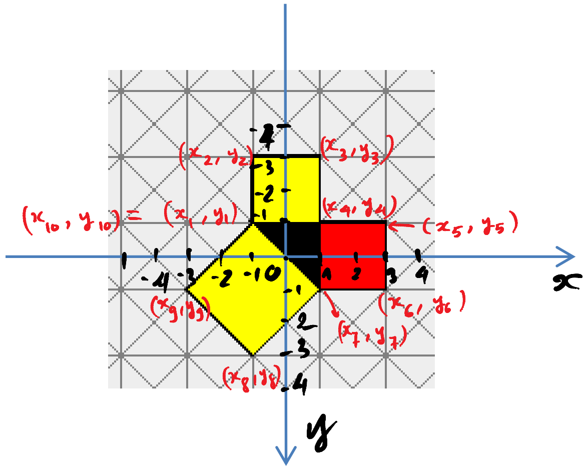

Throughout the game, data structure for a colorful piece is represented as following way. Each piece consists of 10 points in normal xy-coordinators system: p1 = (x1, y1), p2 = (x2, y2) … p9 = (x9, y9), p10 = p1. For easy manipulation, the y-axis is reversed in the same way it is represented on computer monitor. 10 points of each piece create the piece border as shown in above figure. Information about the squares and triangle of the piece can be extracted from these 10 points. These 10 points are ordered in clockwise manner such that the first four points are always vertices of one small square in the piece. The 4th to the 7th points are vertices of the second small square. The 7th to the 10th are vertices of the large square. The 4th, the 7th and the 10th (or the 1st) points are vertices of the triangle. The 10th point is actually the 1st point. It is added to make for ease of extracting information in programming.

Center of a piece is defined as the mid-point of the triangle hypotenuse. Coordinators of above 10 points are relative coordinators to the center. By this way, the 10 points coordinators can be kept the same while the piece is moving. Moving a piece can be performed by simply changing the center coordinators of the piece. All the coordinators used in this program are normalized such that each unit in this xy-coordinators system equals to half of the small square width. Rotating a piece can be performed easily by changing each point from (x, y) to (-y, x).

Using above normalized coordinators is also useful for determining relationship between two pieces positions. By taking advantage of the fact that all the coordinators are normalized such that the small square width is equal to two units in coordinators system and all the coordinator are integers. Consider two squares with their coordinators are (x11, y11) (x12, y12) (x13, y13) (x14, y14) and (x21, y21) (x22, y22) (x23, y23) (x24, y24) respectively. If they are both small squares, relationship of them is determined by following formula

$$D = \left|\sum_{i=1}^{4} x_{1i} - \sum_{i=1}^{4} x_{2i}\right| + \left|\sum_{i=1}^{4} y_{1i} - \sum_{i=1}^{4} y_{2i}\right|$$

These two small square are overlapped if D = 0, and are adjacent if D =8.

For two large squares, consider

$$D1 = \left| \sum_{i=1}^{4} x_{1i} - \sum_{i=1}^{4} x_{2i} \right|$$

$$D2 = \left| \sum_{i=1}^{4} y_{1i} - \sum_{i=1}^{4} y_{2i} \right|$$

If D1 = D2 = 8 then the two squares are adjacent. In other case, if D1 + D2 < 16 then the two squares are interfered.

In case of a small square and a large square, they are interfering if and only if one point of the small square is the center of the large square.

In order to deal with scoring and determining winner, an efficient way of recognizing large square strips is necessary. In this game, each strip is represented by a linked list of pieces. When two large squares placed next together, two pieces that the two squares belong to will be put into a linked list. Depending on context, new linked list can be created or two linked list can be merged. Number of linked list existing at the end of the game will indicate number of strips at that time. Length of linked list conveys number of pieces that form the strip. The linked list is chosen because it is flexible to append two of them together.

In conclusion, a normalized coordinators system is used for manipulating pieces and game board while linked list is used for easy scoring and winning condition determining. This data structure allows two pieces relationship quickly determined by only adding and subtracting the coordinators. Moving a piece is done by changing its center location. Rotating is as simple as swapping two numbers.