

Polyomino Tangram

By Subramanya N

The project is a web app written using Flask.

- Languages used: [python, HTML, CSS, javascript, jQuery]
- Instructions to run the code is given in readme

Task 1: Basic Game Play

Notes:

1. Polyomino - Generating Polyomino code is borrowed and modified to act as an API from open source project. (<https://github.com/tesseractis/polyomino>).
polyomino.py has modified code

```
import polyomino as _mino

@app.route('/allpieces/<n>')
def getAllpieces(n):
    minos = []
    for i in range(1, int(n)+1):
        minos.extend(_mino.one_sided(_mino.generate(int(i))))
    return jsonify(minos)
```

2. Random number of index generator, used native random number generator of javascript and added to a set until I got the required number of random number generated.

Sudo Code:

Step 1: Create Board of 10x10 table

Step 2: Get all configurations of polyomino possible using the API from polyomino.py

- Example all polyomino **until 5 unit squares** are about 29 in number.
- Randomly select a number in range of 2 to (Total Number of possible polyomino / 2)
 - Why 2? if you have 0 or 1 polyomino no fun in game

- Why (Total Number of possible polyomino / 2)? Just a threshold that is big enough but doesn't cover all possible polyomino
- Populate polyomino list to play.

Step 3: construct a matrix of board configuration and **initialize them with false**

- Why **matrix of board config**? So that we can updated locations where the polyomino have been placed.

Step 4: Use drag and drop functionality part of jQuery, to drag and drop elements.

Step 5: On every polyomino check if a rows or columns of the placed polyomino has been filled. If so end the game by flashing Winner :). If not continue.

Step 6: As part of step 4 updated the used pieces and check if any more pieces are left. If there are continue the game. If not end game by flashing looser :P.

To-Do:

If you can place any place the pieces that are available on the board we need to end game.

Task 2: Rotate the Piece

Notes:

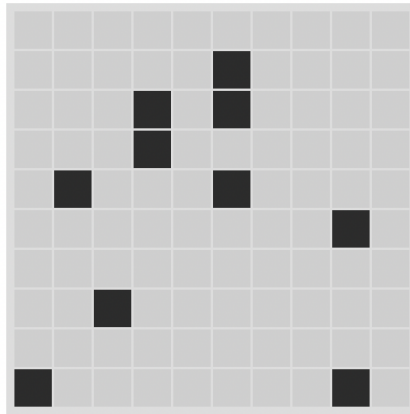
Double Click to select and deselect to rotate the Polyomino.

Sudo code:

Step 1. Make the Polyomino selectable with some sort of visual feedback

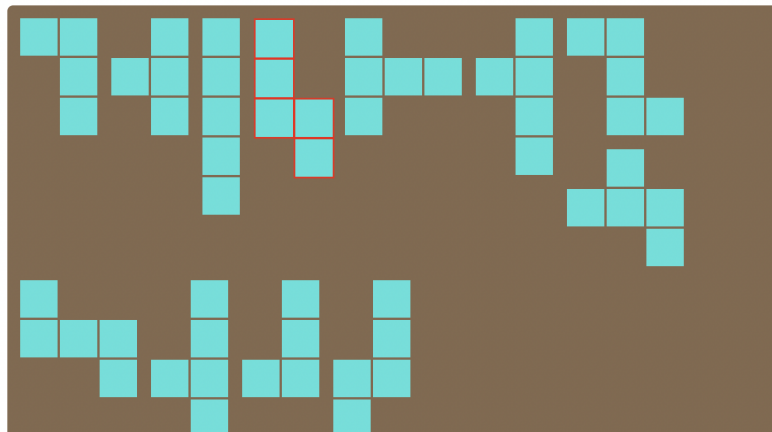
Step 2. Simple $[p[1], (\text{polyomino.height()}/34-1)-p[0]]$ algo for rotate for each square of a Polyomino. (p is each square of a polyomino, 34 because I use 34px squares)

```
var id = selected.attr("data-id");
for (var i = 0; i < pieces[id].length; i++) {
    var c = pieces[id][i];
    pieces[id][i] = [c[1], (selected.height()/34-1)-c[0]];
}
```

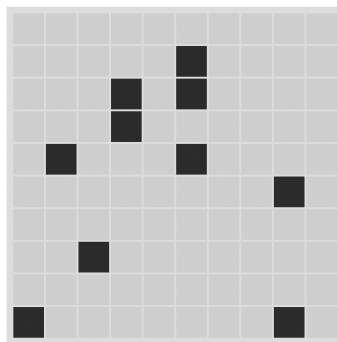


Rotate Clock-Wise

Toggle Blocks

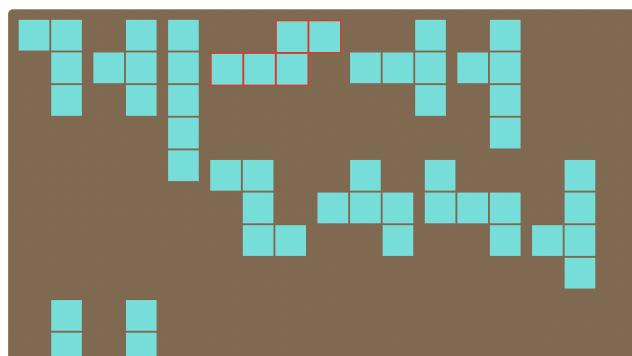


Red Border - When Polyomino is selected. (Double click)



Rotate Clock-Wise

Toggle Blocks



Same Polyomino as above rotated.

Task 3: Generate 10 Blocked out blocks.

Notes:

Same random function used as told in task 1.

Sudo code:

Step 1: Generate 10 random number in the range of 100 as we have 10x10 board

Step 2: Use the same method used in task 1 where if a Polyomino is already on in that location don't place it. Treat there 10 generated block as Polyomino already placed.

```
for (var j = 0; j < 10; j++) {
  if(noInput!=null){
    if(noInput.includes((i*10)+j)){
      tr.append("<td id='b" + i + "-" + j + "' class='noBlock'></td>");
      matrix[i][j] = true;
    }
    else{
      tr.append("<td id='b" + i + "-" + j + "'></td>");
      matrix[i][j] = false;
    }
  }
  else{
    tr.append("<td id='b" + i + "-" + j + "'></td>");
    matrix[i][j] = false;
  }
}
```

```
for (var j = 0; j < 10; j++) {
  if(noInput!=null){
    if(noInput.includes((i*10)+j))
      tr.append("<td id='b" + i + "-" + j + "' class='noBlock'></td>");
    else
      tr.append("<td id='b" + i + "-" + j + "'></td>");
  }
  else
    tr.append("<td id='b" + i + "-" + j + "'></td>");
  matrix[i][j] = false;
}
```

The above code is used to generate table and updating the matrix.

Case 1: If a row or a column has blocked squares then filling up the remaining spaced in the row or column will win the game

Case 2: If a row or a column has blocked squares then that row or column will never win the game.

If you have any queries please feel free to reach out.

Thank you!