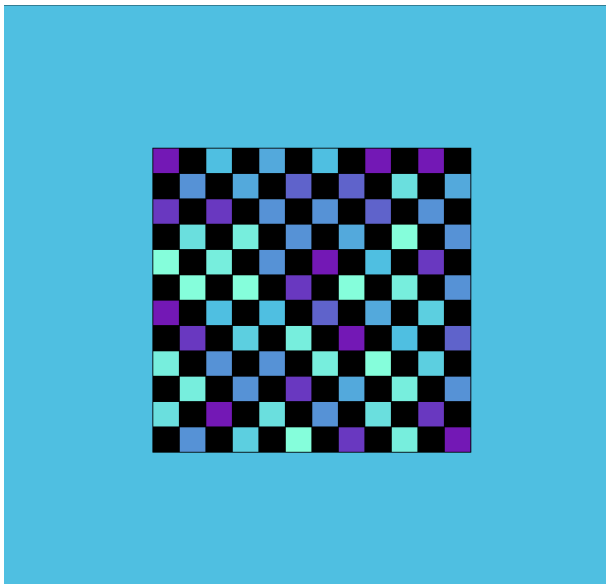
Ryan Dennie
PC04-Generative Art-Pseudocode



This pseudocode and final code describes how to draw a generative chessboard and a generative background color to go with it:

1.) Import:
import turtle
import random

2.) Define checkerboard's width and size
    start_x = width*size/2.0
    start_y = width*size/2.0

3.) Set turtle's speed:
turtle.speed(10)

4.) Pick the pen up

5.) Set the turtle's position and

6.) Set up a for loop for the fill color to alternate between black and generative color using random and RGB values from https://coolors.co:

```
 for i in range(width):
       for j in range(width):
          turtle.begin_fill()
          if fill % 2 == 0:
             turtle.fillcolor(0,0,0)
          else:
             turtle.fillcolor(random.choice([(116, 0, 184), (105, 48, 195), (94, 96, 206), (83, 144, 217), (78, 168, 222), (72, 191, 227),(86, 207, 225),(100, 223, 223), (114, 239, 221), (128, 255, 219)]))
```

7.) Draw chessboard using following:

```
            turtle.pendown()
        turtle.forward(size)
        turtle.left(90)
        turtle.forward(size)
        turtle.left(90)
        turtle.forward(size)
        turtle.left(90)
        turtle.forward(size)
        turtle.left(90)
        turtle.end_fill()
        fill += 1
        x, y = turtle.pos()
        turtle.setpos(x+size, y)
```

8.) Use an if statement to accomplish a “true/false” set:

```
if width % 2 == 0:
            fill += 1
      turtle.penup()
      x, y = turtle.pos()
      turtle.setpos(x-length,y+size)
```

9.) Import colormode:

```
turtle.colormode(255)
```

10.) Change checkerboard’s size as needed:

```
turtle.tracer(0)
checkerboard(12, 30)
```

11.) turtle.up()

12.) Set the generative background color using random:

```
panel.bgcolor(random.choice([(XXXXX)]))
```

13.) turtle.done

End Code:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Sep 21 19:14:31 2021

@author: Skippy3
"""

import turtle
import random

def checkerboard(width, size):
```

```
    turtle.speed(10)
    start_x = width*size/2.0
    start_y = width*size/2.0
    turtle.penup()
    x, y = turtle.pos()
    turtle.setpos(x - start_x, y - start_y)
    length = width*size

    fill = 0
    turtle.fillcolor(0,0,0)
    for i in range(width):
        for j in range(width):
            turtle.begin_fill()
            if fill % 2 == 0:
                turtle.fillcolor(0,0,0)
            else:
                turtle.fillcolor(random.choice([(116, 0, 184), (105, 48, 195), (94, 96, 206), (83, 144, 217), (78, 168, 222),
(72, 191, 227),(86, 207, 225),(100, 223, 223), (114, 239, 221), (128, 255, 219)]))
            turtle.pendown()
            turtle.forward(size)
            turtle.left(90)
            turtle.forward(size)
            turtle.left(90)
            turtle.forward(size)
            turtle.left(90)
            turtle.forward(size)
            turtle.left(90)
            turtle.end_fill()
            fill += 1
            x, y = turtle.pos()
            turtle.setpos(x+size, y)

        if width % 2 == 0:
            fill += 1
        turtle.penup()
        x, y = turtle.pos()
        turtle.setpos(x-length,y+size)

turtle.colormode(255)

# Create a panel to draw on.
panel = turtle.Screen()
w = 700 # width of panel
h = 700 # height of panel
panel.setup(width=w, height=h) #600 x 600 is a decent size to work on.
#You can experiment by making it the size of your screen or super tiny!

#=======Add your code here======
turtle.tracer(0)
```

```
checkerboard(12, 30)

#==============================
# This section allows for clean execution of the code! (Ignore it)
turtle.up()


#=======Clean up code (do not change)======
# this code ensures that your script runs correctly each time.
panel.bgcolor(random.choice([(116, 0, 184), (105, 48, 195), (94, 96, 206), (83, 144, 217), (78, 168, 222), (72, 191,
227),(86, 207, 225),(100, 223, 223), (114, 239, 221), (128, 255, 219)]))
turtle.done()
```