

pygame을 활용한 퍼즐게임(테트리스) 만들기

튜터 지준섭 gaon0403@gmail.com

KAIST 전산학부 및 해킹 동아리 GoN 소속

A large, semi-transparent watermark of the Python logo (a stylized snake) is centered in the background.

1.

Python이 뭔가요?

왜 굳이 Python을 쓰나요?

Language Rank	Types	Spectrum Ranking
1. C	📱💻CHIP	100.0
2. Java	🌐📱💻	98.1
3. Python	🌐💻	98.0
4. C++	📱💻CHIP	95.9
5. R	💻	87.9
6. C#	🌐📱💻	86.7
7. PHP	🌐	82.8
8. JavaScript	🌐📱	82.2
9. Ruby	🌐💻	74.5
10. Go	🌐💻	71.9

Python은
세계적으로 가장 널리 쓰이는
프로그래밍 언어 중 하나입니다.

다른 언어와 구별되는
파이썬의 장점

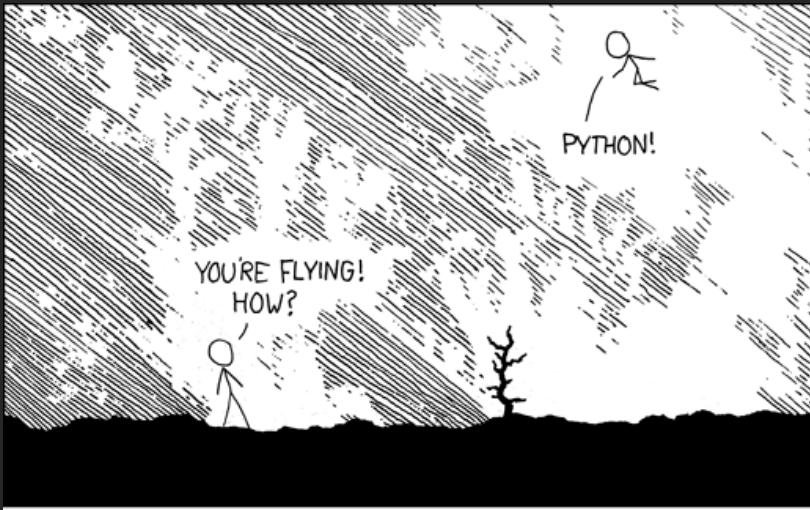


1. 생산성

2. 디자인

3. 만능성

1. 생산성



- 타 언어와 비교했을 때 매우 우수한 생산성
- 우수한 자료형, 다양한 모듈
- C로 2년 동안 개발 = Python으로 한 달 개발?!

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
    }  
}
```

JAVA

```
#include <stdio.h>  
int main(void) {  
    printf("Hello world");  
}
```

C

```
print 'hello world!'
```

P

<각 언어로 “Hello, world!”를 출력하는 코드>

개발기간 단축 = Python !!

2. 디자인

- 다른 언어와 구별되는 Python의 디자인 철학
“가장 아름다운 하나의 답이 존재한다”

```
for (i = 0; i < n; i++) {  
    /* source */  
    /* source */  
}
```

```
for (i = 0; i < n; i++)  
{  
    /* source */  
}
```

<기본적인 indent style마저도 사람마다 다른 C>

```
for i in range(n):  
    # source #
```

<오직 하나로만 통일된 Python의 indentation>

3. 만능성

- 굉장히 다양하고 방대한 Python의 패키지
- 뭔가 하고 싶은데, 뭘 해야 할지 모르겠다?
→ Google에 python을 붙여서 검색!
 - Ex) python blind SQL injection, python web, …
- 여러분을 도와줄 라이브러리들이 언제나 준비되어 있습니다
- 우리가 쓸 pygame도 그 중 하나



2. **Python**의 실습

직접 써 봅시다

개발환경 세팅

- 총 세 가지를 내려받아 설치해야 합니다
 - Python 2.7 (가장 중요)
 - pygame (매우 중요)
 - Wing IDE 101 (다른 IDE나, 텍스트 에디터를 이용해도 가능)
- <http://youthcamp.gaonnr.me>
- 위 링크를 통해 다운로드 링크를 볼 수 있습니다
- 사실 Google에 검색하면 다 나와요

간단한 예제

- Python은 대강 쓸 줄 알죠?
- 이제 pygame을 써 봅시다
- 첫 줄에 다음을 적어넣습니다

```
import pygame
```

- 이제 pygame에 구현되어 있는 모든 것들을 쓸 수 있습니다
- 나머지 step 및 예제는 <http://youthcamp.gaonnr.me>에 설명되어 있습니다
- 페이지 만드느라 힘들었어요

객체 지향 프로그래밍(OOP)



- 이미지 여러 개를 불러와 각기 따로 움직이게 하려면?
- 좌표를 담당하는 변수를 따로따로 다 만들 수도 있겠지만 그러면 손이 너무 많이 가요
- Class를 사용합시다

Class가 뭐에요?

- 변수, 상수, 그리고 함수들을 묶어놓은 집합
- **새로운 자료형**을 만드는 것이라고 보시면 이해가 될지도?
사실 자료형 + 함수까지 들어가니 더 큰 개념입니다
- 예를 들어 Bird라는 class를 새로 만든다고 합시다
- 우리에게 필요한 변수와 함수가 뭐가 있죠?
 - 새의 이미지 파일
 - 새의 위치
 - 새의 속도
 - 새를 화면 위에 보여주는 함수
 - 새를 움직여 주는 함수 (위치를 업데이트 시켜주는 함수)

```
class Bird:  
    def __init__(self, imageName, location, spd):  
        self.imageName = imageName  
        self.location = location  
        self.spd = spd  
  
        self.image = pygame.image.load(imageName)  
  
    def move(self):  
        X, Y = self.location  
        X += self.spd  
        self.location = (X, Y)  
  
    def show(self):  
        screen.blit(self.image, self.location)  
        self.move()
```

```
newBird = Bird(image, (0, 0), 3)
```

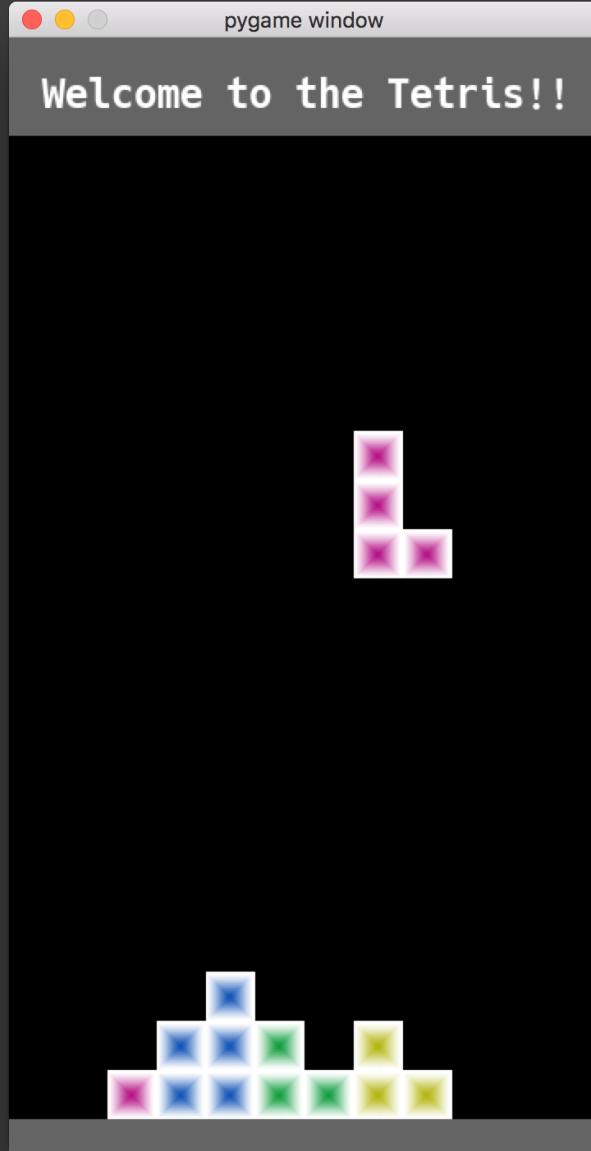
3. **Tetris** 만들기

본격적으로 테트리스를 만들어 봅시다

시작하기 전에….

- 처음부터 너무 고퀄리티를 지향하지 맙시다
- 시중에 나오는 테트리스 게임의 퀄리티는 수업시간 내에 따라잡기 힘들어요
- 일단은 간단한 구현으로 만족하고,
차차 기능을 덧붙여 나갑시다





(실제로 만든 게임)
이 정도만 되어도 괜찮아요

그럼 시작해 봅시다

- <https://github.com/GAONNR> 로 가주세요
 - 화면 상단의 Repositories 클릭
 - 두 번째에 있는 pygame_tetris 클릭
 - 화면 중간즈음에 있는 Branch: master 버튼을 눌러서 Branch: tetris-skeleton 으로 변경해 주세요
 - 이제 파일들을 다운받아 압축을 풁니다
- tetris.py 파일과 blockClass.py 파일을 작성하면 됩니다
- <http://www.pygame.org/tags/tetris>
여기서 다른 테트리스 코드를 참고하셔도 좋아요

```
1 import sys, os, math, random
2 import pygame
3 from pygame.locals import *
4 from blockClass import *

5
6 FPS = 3
7 GRAY = (100, 100, 100)
8 WHITE = (255, 255, 255)
9 BLACK = (0, 0, 0)
10 BLOCK_WIDTH = 30
11 SCREEN_SIZE = (360, 680)

12
13 GRID_SIZE_X = 12 ### X GRID 12
14 GRID_SIZE_Y = 20 ### Y GRID 20
15
16 GRAVITY = (0, 1)
17 MOVE_LEFT = (-1, 0)
18 MOVE_RIGHT = (1, 0)
19
20 LOSE = False
21
22 pygame.init()
```

도움이 될 모듈들
+ blockClass의 import

FPS는 3,
블록의 크기는 가로 30px
화면의 크기는 360 x 680

가로 12칸, 세로 20칸입니다

```
24 def handleEvents(block, screen):
25     for event in pygame.event.get():
26         if event.type == pygame.QUIT:
27             return False
28
29         if event.type == KEYDOWN:
30             if event.key == K_RIGHT:
31                 block.move(screenArr, BLOCK_WIDTH, MOVE_RIGHT)
32             elif event.key == K_LEFT:
33                 block.move(screenArr, BLOCK_WIDTH, MOVE_LEFT)
34             elif event.key == K_DOWN:
35                 block.moveDown(screen, screenArr, BLOCK_WIDTH, GRAVITY)
36             elif event.key == K_UP:
37                 block.rotate(screenArr, BLOCK_WIDTH)
38             else:
39                 continue
40
41     return True
```

나중에 이벤트를 핸들링해 줄 함수에요

종료 버튼을 누르면 종료, 오른쪽과 왼쪽 방향키를 누르면 이동
아래 방향키를 누르면 맨 아래로 바로 이동, 위 방향키를 누르면 회전
…을 하게 해 줄 함수를 부릅니다

함수 내용은 이제 여러분이 짜셔야….

```
43 def loadImages():
44     colors[0] = pygame.image.load('Block1.png')
45     colors[1] = pygame.image.load('Block2.png')
46     colors[2] = pygame.image.load('Block3.png')
47     colors[3] = pygame.image.load('Block4.png')
```

제가 만들어 둔 블록 이미지를 불러오는 함수에요
이미지 파일은 전부 30px짜리 정사각형이지만 색깔이 다릅니다

colors[0]은 파란색, [1]은 빨강색, [2]는 노랑색, [3]은 초록색

```
49 screen = pygame.display.set_mode(SCREEN_SIZE) ### MAKE SCREEN
50 clock = pygame.time.Clock()
51
52 background = pygame.Surface(SCREEN_SIZE) ### BACKGROUND LAYER
53 background.fill(GRAY) ### BLACK COLOR
54
55 titleFont = pygame.font.Font('Hack-Bold.ttf', 24) ### LOAD FONT
56 titleSurface = titleFont.render('Welcome to the Tetris!!', True, WHITE) ### Welcome to the Tetris!!
57
58 gameWindow = pygame.Surface((360, 600)) ### MAIN WINDOW FOR GAME
59 gameWindow.fill(BLACK)
60
61 background.blit(gameWindow, (0, 60))
62 background.blit(titleSurface, (20, 20))
63 screen.blit(background, (0, 0))
64 pygame.display.update()
```

스크린이 어떻게 보일지를 설정하고 있네요
pygame.font를 사용해서 상단에 문구도 출력합니다

```
colors = [0,0,0,0]
loadImages() ### load images of blocks. 0: Blue, 1: Red, 2: Yellow, 3: Green

screenArr = Screen(GRID_SIZE_X, GRID_SIZE_Y)

shapes = ['I', 'J', 'L', 'O', 'S', 'T', 'Z']
FIRSTLOCATION = (5, -4)

randomColor = random.randrange(4)
currentBlock = Block(randomColor, random.choice(shapes), FIRSTLOCATION)
```

블록의 이미지를 `loadImages()`로 불러오고,
`screenArr`이라는 새 `Screen` 클래스를 선언합니다
`Screen` 클래스가 뭐하는 건지는 나중에 보여드립니다

블록의 모양에는 I, J, L, O, S, T, Z가 있고,
블록이 처음 시작하는 위치는 (5, -4) 위치네요

랜덤한 컬러를 가진 `currentBlock`이라는 `Block`을 선언했습니다

```
77 while not LOSE:
78     if not handleEvents(currentBlock, gameWindow):
79         pygame.quit()
80         exit(0)
81
82     gameWindow.fill(BLACK)
83
84     ### move currentBlock. it it can't move anymore, make new block.
85     ### also check line clearing.
86     #####
87
88     ###      Write Here      ###
89
90     #####
91
92     screenArr.show(gameWindow, BLOCK_WIDTH, colors)
93     background.blit(gameWindow, (0, 60))
94     screen.blit(background, (0, 0))
95
96     pygame.display.update()
97     clock.tick(FPS)
98
99     pygame.quit()
100    exit(0)
```

적을 공간이 없네요

```
1 import sys, os, math, random  
2 import pygame  
3 from pygame.locals import *  
4 from blockRotation import *  
5  
6 pygame.init()  
7
```

이제 blockClass 파일을 봅시다
처음은 별 내용이 없구요
blockRotation이란 녀석을 import하는데
여기는 건드릴 필요는 없습니다

```
8 class Block:
9     def __init__(self, color, shape, location):
10        self.color = color
11        self.location = location
12        self.static = False
13        self.STATE = 0
14        self.SHAPE = shape
15
16        self.blockArr = [[0, 0, 0, 0],
17                         [0, 0, 0, 0],
18                         [0, 0, 0, 0],
19                         [0, 0, 0, 0]]
20
21        self.blockArr = rotateBlock(shape, self.STATE)
22
```

`__init__`에서 볼 수 있듯이 `color`(0~3의 숫자),
`shape`(한 글자짜리 문자열), `location`(좌표)를 인자로 받습니다
`static`은 이 블록이 더 움직일 수 없는지를 뜻합니다
`STATE`는 블록의 현재 모양의 상태를 말해주구요
`STATE`에 따른 현재 블록의 모양이 `blockArr`에 저장됩니다
0은 빈칸, 1은 블록이 있어야 할 곳이에요

```
23     def move(self, screenArr, BLOCK_WIDTH, DIRECTION):
24         if self.static:
25             return True
26
27         x, y = self.location
28         dx, dy = DIRECTION
29
30         ### remove the block in screenArr
31         #####
32
33         ###      Write Here      ###
34
35         #####
36
37         ### check if block is fit for new location. if it doesn't fit, self.static == True
38         #####
39
40         ###      Write Here      ###
41
42         #####
```

Move함수는 screenArr(스크린의 상태를 담고 있는 Screen객체), BLOCK_WIDTH(30), DIRECTION(움직일 방향)을 인자로 받네요

(x, y)가 현재의 위치를 받아오고, (dx, dy)가 방향을 받아옵니다
이제 screenArr에서 지금 블록을 지우고, 새로이 움직일 데에
블록이 이미 차있지 않은지, 혹은 움직일 수 없는 공간인지 확인해 줍시다

```
44         ### if block fits for new location, update location
45         #####
46
47         ###      Write Here      ###
48
49         #####
50
51         ### write new location
52         #####
53
54         ###      Write Here      ###
55
56         #####
57
58         ### prevent block stopping when there's no place to move in X direction
59         if DIRECTION != (0, 1):
60             self.static = False
61             return False
62
63         if self.static:
64             return True
65         else:
66             return False
```

만약 움직일 수 있으면, location을 업데이트하고, screenArr에 바뀐 스크린의 상태를 써 줍시다.

그 밑에 있는 코드는 이 블록을 계속 신경을 써야 하는지, 아니면 새 블록을 떨어뜨려야 하는지를 결정해서 판단해주는 코드예요

```
68     def moveDown(self, screen, screenArr, BLOCK_WIDTH, GRAVITY):
69         while True:
70             if self.move(screenArr, BLOCK_WIDTH, GRAVITY):
71                 return
```

moveDown함수는 move함수를 블록이 더 이상 움직일 수 없을 때까지 반복하는 함수입니다.

```
73     def rotate(self, screenArr, BLOCK_WIDTH):
74         tempArr = rotateBlock(self.SHAPe, (self.STATE + 1) % 4)
75
76         x, y = self.location
77
78         ### remove the block in screenArr
79         #####
80
81         ###      Write Here      ###
82
83         #####
84
85         ### check if block is fit for new shape(rotated). if it doesn't fit, return
86         #####
87
88         ###      Write Here      ###
89
90         #####
91
92         ### write new shape
93         #####
94
95         ###      Write Here      ###
96
97         #####
98
99         self.blockArr = tempArr
100        self.STATE += 1
101        self.STATE %= 4
```

rotate 함수의 기본 원리는 move와 비슷합니다
다음 STATE의 배열을 불러와서 들어갈 수 있는지 검사하죠

```
103     def getY(self):  
104         return self.location[1]
```

이름만큼 정직한 함수입니다

```
107 class Screen:  
108     def __init__(self, X, Y):  
109         self.arr = [[-1 for col in range(X)] for row in range(Y)]  
110         self.WIDTH = X  
111         self.HEIGHT = Y  
112  
113     def write(self, x, y, color):  
114         self.arr[y][x] = color  
115  
116     def getXY(self, x, y):  
117         return self.arr[y][x]  
118
```

Screen 클래스는 게임 화면의 상태를 배열로 저장하고 (self.arr) gameWindow에 배열을 실제 위치와 이미지로 변환하여 blit, 또한 라인 클리어도 담당해주는 함수들이 포함되어 있습니다.

Block 클래스와 달리, 배열에서 -1이 빈칸이며 0~3까지는 그곳에 위치한 블록의 색깔을 나타냅니다

write함수는 arr의 좌표에 인자로 받은 컬러를 써 주는 함수이며, Block 클래스를 짤 때 활용하시면 됩니다
getXY함수는 arr의 좌표의 컬러값을 알려주는 함수겠군요

```

119     def show(self, screen, BLOCK_WIDTH, images):
120         for i in range(self.HEIGHT):
121             for j in range(self.WIDTH):
122                 if self.arr[i][j] < 0:
123                     continue
124
125                 y = i * BLOCK_WIDTH
126                 x = j * BLOCK_WIDTH
127
128                 screen.blit(images[self.arr[i][j]], (x, y))
129
130     def lineClear(self, lineIdx):
131         if lineIdx >= self.HEIGHT:
132             return
133         for X in range(self.WIDTH):
134             if self.arr[lineIdx][X] < 0:
135                 return
136
137         tempArr = [[-1 for col in range(self.WIDTH)] for row in range(self.HEIGHT)]
138         for i in range(lineIdx + 1, self.HEIGHT):
139             for j in range(self.WIDTH):
140                 tempArr[i][j] = self.arr[i][j]
141         for i in range(lineIdx):
142             for j in range(self.WIDTH):
143                 tempArr[i + 1][j] = self.arr[i][j]
144
145         self.arr = tempArr
146
147     def getWidth(self):
148         return self.WIDTH
149     def getHeight(self):
150         return self.HEIGHT

```

설명은 여기까지

- 이제 직접 코딩해 보세요
- 너무 어렵다면….
 - 아까 스켈레톤 코드를 다운로드 받은 곳에 힌트가 있을지도 몰라여
 - 그래도 일단은 최대한 자기 힘으로 해결해 봅시다
 - 제가 드린 코드의 방식이 너무 구리다고 생각되면 고치셔도 돼요



3. 창의적인 Tetris 만들기

개량해 봅시다

3. 창의적인 Tetris 만들기

개량해 봅시다



3. 창의적인 뭔가를 만들기

바꿨어요



여기서 끝내면 좋겠지만
사람들은 여러분의 결과물을 원합니다….

여태껏 배운 것들로 창의적인 뭔가를 만들어 봅시다

플래시 애니메이션 같은 것도 좋고
뭔가를 시뮬레이션해도 좋고
원래대로 게임을 만들어도 좋아요

아무튼 배운 것들
(이미지 표시하기, 움직이기, 객체 지향)
을 기반으로 뭔가 보기 좋은 ‘움직이는 것’을
만들어 봅시다

몇 가지 함수가 더 있어요

- Surface.get_rect() 함수
- pygame.transform.scale() 함수
- pygame.transform.rotate() 함수
- pygame.transform.flip() 함수



4. **발표 준비하기**

우리의 결과물이 이렇게 잘났다! 를 보여주세요

역할 분담을….

- 발표 자료 제작자, 발표자 등 역할을 나눕시다
- 한 역할을 여러명이 해도 되고, 한 명이 여러 역할을 해도 OK
- 근데 들러리가 있으면 안 돼요

발표 내용의 구성

- 자신들이 개발한 것의 특징과 장점을 위주로 발표합니다
- 개발 과정을 넣으셔도 돼요 (구현하던 중 만난 문제점, 해결 과정 등)
- 구현이 실패하셨더라도 어떤 방식으로 구현하려고 했는지 설명해 주시면 좋습니다
- 구현이 된 부분까지 게임 시연을 준비해 주세요
 - 게임을 직접 실행해서 시연하셔도 좋고
 - 작동 영상을 녹화해서 준비하셔도 좋습니다

