

Pygame 1 : Snake Game

20211060 ohseohyun

Github link : https://github.com/hyunneee/hyunneee.github.io/tree/main/%5B22-2%5DVMP_Snake_Game

1. Sound effect : bgm, feed bgm, dead bgm
2. Larger screen : Fullscreen
3. Background image : according to the level
4. Color change : snake, feed, obstacle, text etc.
5. Obstacles

1. Code & Explanation

```
1  import pygame
2  import os
3  import sys
4  import random
5  from time import sleep
6
7  # 게임 스크린 전역변수
8  SCREEN_WIDTH = 800
9  SCREEN_HEIGHT = 600
10
11 # 게임 화면 전역변수
12 GRID_SIZE = 20
13 GRID_WIDTH = SCREEN_WIDTH / GRID_SIZE
14 GRID_HEIGHT = SCREEN_HEIGHT / GRID_SIZE
15
16 # 방향 전역변수
17 UP = (0, -1)
18 DOWN = (0, 1)
19 LEFT = (-1, 0)
20 RIGHT = (1, 0)
21
22 # 색상 전역변수
23 WHITE = (255, 255, 255)
24 RED = (255, 50, 50)
25 GRAY = (100, 100, 100)
26
27 # 잔디, 틀, 물, 모래
28
29 # 뱀 객체
30 class Snake(object):
31     def __init__(self): # 초기 함수
32         self.create() #create() 실행
33         self.hit = pygame.mixer.Sound('/Users/seohyunoh/Documents/python/VMP/4_Snake_Game/assets/hit.wav')
34         self.hit.set_volume(0.5) # hit에 hit.wav를 저장하고 볼륨을 0.5로 조정
35
36
37     # 뱀 생성
38     def create(self):
39         self.length = 2 # 뱀 길이는 2
40         self.positions = [(int(SCREEN_WIDTH / 2), int(SCREEN_HEIGHT / 2))] # 뱀의 위치는 스크린의 가운데
41         self.direction = random.choice([UP, DOWN, LEFT, RIGHT]) # 시작방향은 상하좌우 중 랜덤으로 하나 선택
```

```
43     # 뱀 방향 조정
44     def control(self, xy):
45         if (xy[0] * -1, xy[1] * -1) == self.direction:
46             return
47         else:
48             self.direction = xy
49
50     # 뱀 이동
51     def move(self):
52         cur = self.positions[0] #뱀 머리
53         x, y = self.direction # 방향
54         new = (cur[0] + (x * GRID_SIZE)), (cur[1] + (y * GRID_SIZE)) # 그리드 사이즈 만큼 x,y 이동해주기
55
56         # 뱀이 자기 몸통에 닿았을 경우 뱀 처음부터 다시 생성
57         if new in self.positions[2:]:
58             sleep(1) # 잠시 멈췄다가
59             self.hit.play() # hit 효과음을 넣고
60             self.create() # snake를 새로 만들어준다.
61
62         # 뱀이 게임화면을 넘어갈 경우 뱀 처음부터 다시 생성
63         elif new[0] < 0 or new[0] >= SCREEN_WIDTH or \
64             new[1] < 0 or new[1] >= SCREEN_HEIGHT:
65             sleep(1)
66             self.hit.play()
67             self.create()
68
69         # 뱀이 정상적으로 이동하는 경우
70         else:
71             self.positions.insert(0, new) # 앞으로 이동하는 느낌
72             if len(self.positions) > self.length:
73                 self.positions.pop() # 뒤 꼬리 잘라주는 느낌
74
75     # 뱀이 먹이를 먹을 때 호출
76     def eat(self):
77         self.length += 1
78
79     # 뱀 그리기
80     def draw(self, screen):
81         red, green, blue = 150 / (self.length - 1), 150 / (self.length - 1), 150 # 초기 뱀 색을 150,150,150으로 지정하기
82         for i, p in enumerate(self.positions): # position에 따라 R,G 값을 점점 줄여나갈 예정
83             color = (red * i, green * i, blue)
84             rect = pygame.Rect((p[0], p[1]), (GRID_SIZE, GRID_SIZE)) #그리드 사이즈의 정사각형을 position에 그린다.
85             pygame.draw.rect(screen, color, rect)
```

```
88     # 먹이 객체
89     class Feed(object):
90         def __init__(self):
91             self.position = (0, 0) #초기 위치 설정
92             self.color = RED # 먹이 색
93             self.create() # 먹이 생성
94
95     # 먹이 생성
96     def create(self):
97         x = random.randint(0, GRID_WIDTH - 1) # x를 그리드 위드-1사이의 랜덤 수 설정
98         y = random.randint(0, GRID_HEIGHT - 1) # y를 그리드 위드-1사이의 랜덤 수 설정
99         self.position = x * GRID_SIZE, y * GRID_SIZE # x,y를 그리드 사이즈만큼 곱한 것을 위치로 설정
100
101    # 먹이 그리기
102    def draw(self, screen):
103        rect = pygame.Rect((self.position[0], self.position[1]), (GRID_SIZE, GRID_SIZE))
104        pygame.draw.rect(screen, self.color, rect) # 위치에 그리드 사이즈 만큼의 정사각형 그리기
105
106    # 방해물 객체
107    class Obstacle(object):
108        def __init__(self):
109            self.position = (0, 0) #초기 위치 설정
110            self.color = GRAY # 먹이 색
111            self.create() # 먹이 생성
112
113    # 장애물 생성
114    def create(self):
115        x = random.randint(0, GRID_WIDTH - 1) # x를 그리드 위드-1사이의 랜덤 수 설정
116        y = random.randint(0, GRID_HEIGHT - 1) # y를 그리드 위드-1사이의 랜덤 수 설정
117        self.position = x * GRID_SIZE, y * GRID_SIZE # x,y를 그리드 사이즈만큼 곱한 것을 위치로 설정
118
119    # 장애물 그리기
120    def draw(self, screen):
121        rect = pygame.Rect((self.position[0], self.position[1]), (GRID_SIZE, GRID_SIZE))
122        pygame.draw.rect(screen, self.color, rect) # 위치에 그리드 사이즈 만큼의 정사각형 그리기
```

```

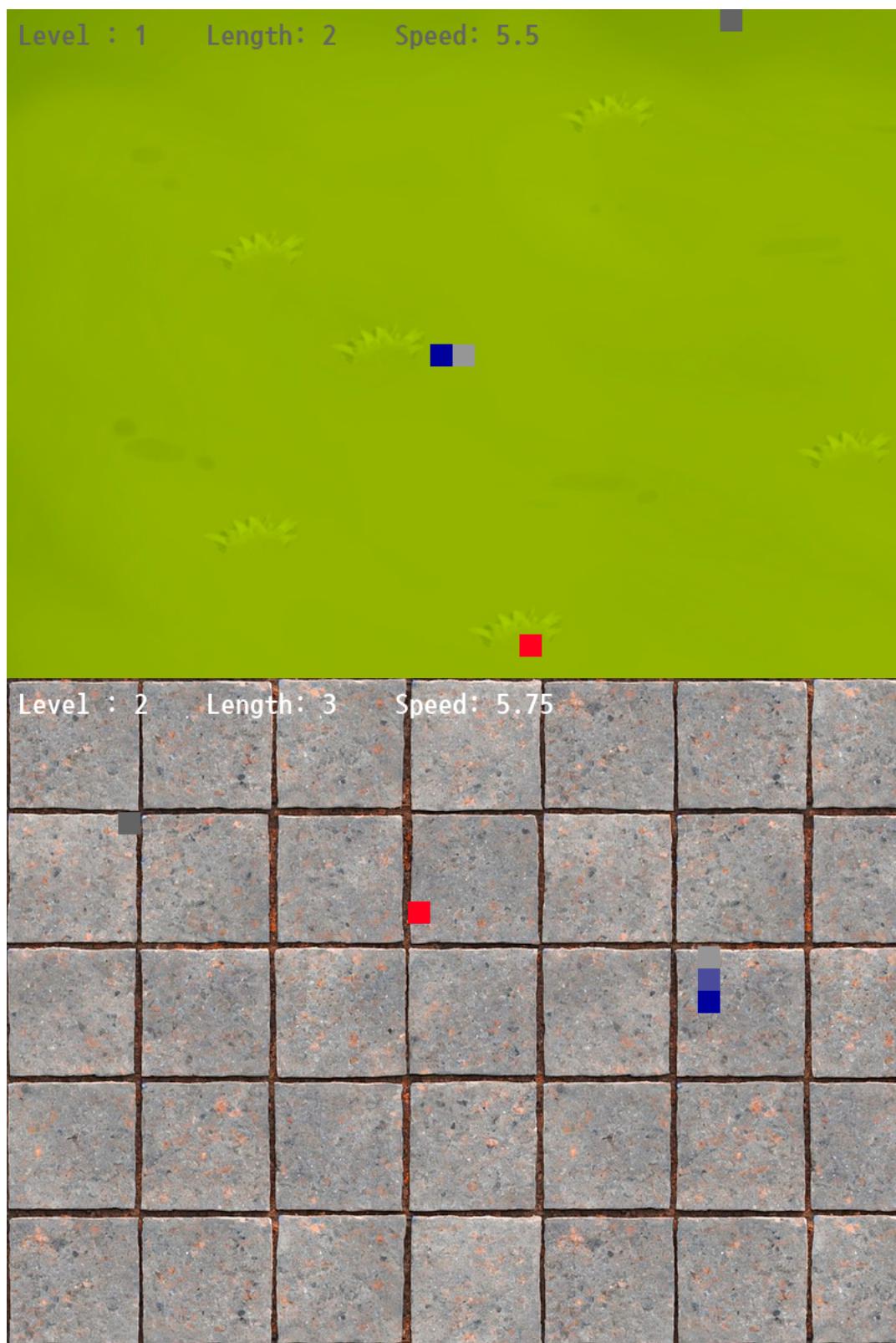
125     # 게임 객체
126     class Game(object):
127         def __init__(self):
128             self.snake = Snake() #초기 snake
129             self.feed = Feed() # 초기 feed
130             self.obt = Obstacle() # 초기 obstacle
131             self.speed = 20 # 초기 스피드
132             self.pop = pygame.mixer.Sound('/Users/seohyunoh/Documents/python/VMP/4_Snake_Game/assets/pop.wav')
133             self.pop.set_volume(0.5) # 먹이 먹을 때 나는 효과음 pop에 저장, 볼륨 0.5로 조정
134             # 효과음
135
136
137         # 게임 이벤트 처리 및 조작
138         def process_events(self):
139             for event in pygame.event.get():
140                 if event.type == pygame.QUIT: # x버튼 눌러서 끄기
141                     return True
142                 elif event.type == pygame.KEYDOWN: # 상하좌우 버튼에 따라 control 움직이기
143                     if event.key == pygame.K_UP:
144                         self.snake.control(UP)
145                     elif event.key == pygame.K_DOWN:
146                         self.snake.control(DOWN)
147                     elif event.key == pygame.K_LEFT:
148                         self.snake.control(LEFT)
149                     elif event.key == pygame.K_RIGHT:
150                         self.snake.control(RIGHT)
151             return False
152
153         # 게임 로직 수행
154         def run_logic(self):
155             self.snake.move() # snake가 움직일때마다
156             self.check_eat(self.snake, self.feed) # 먹었는지,
157             self.check_obt(self.snake, self.obt) # 장애물 닿았는지 체크
158             self.speed = (20 + self.snake.length) / 4 # length에 20을 더하고 4로 나눈 값으로 스피드 조절
159
160         # 뱀이 먹이를 먹었는지 체크
161         def check_eat(self, snake, feed):
162             if snake.positions[0] == feed.position: # 뱀 머리가 feed의 위치와 같으면
163                 snake.eat() # snake.eat 함수 실행
164                 feed.create() # feed.create 함수 실행 ( 먹이 재생성 )
165                 self.obt.create() # obstacle.create 함수 실행 ( 장애물 재생성 )
166                 self.pop.play() # 먹이 먹은 효과음

```

```
168     def check_obt(self, snake, obt):
169         if snake.positions[0] == obt.position: # 뱀 머리가 장애물의 위치와 같으면
170             sleep(1) # 시간 잠시 지연
171             snake.hit.play() # 죽었을 때 효과음
172             snake.create() # 새로 뱀 만들기
173             obt.create() # 장애물 만들기
174
175     def resource_path(self, relative_path): # 상대경로
176         try:
177             base_path = sys._MEIPASS
178         except Exception:
179             base_path = os.path.abspath(".")
180         return os.path.join(base_path, relative_path)
181
182     # 게임 정보 출력
183     def draw_info(self, length, speed, screen):
184         if length < 10: # 총 4단계로 구분해서 길이가 10씩 올라갈때마다 level을 올리고 배경화면을 바꿈
185             bg = pygame.image.load('/Users/seohyunoh/Documents/python/VMP/4_Snake_Game/assets/grass_bg.jpg')
186             level = 1
187         elif length < 20: # 배경화면은 잔디 -> 돌바닥 -> 해수면 -> 모래바닥 순으로 이루어져있음
188             bg = pygame.image.load('/Users/seohyunoh/Documents/python/VMP/4_Snake_Game/assets/rock.jpg')
189             level = 2
190         elif length < 30:
191             bg = pygame.image.load('/Users/seohyunoh/Documents/python/VMP/4_Snake_Game/assets/water.jpg')
192             level = 3
193         else:
194             bg = pygame.image.load('/Users/seohyunoh/Documents/python/VMP/4_Snake_Game/assets/sand.jpg')
195             level = 4
196         info = "Level : " + str(level) + "    " + "Length: " + str(length) + "    " + "Speed: " + str(round(speed))
197         font_path = resource_path("assets/NanumGothicCoding-Bold.ttf")
198         font = pygame.font.Font(font_path, 26)
199         text_obj = font.render(info, 1, WHITE) # 흰색으로 변경
200         text_rect = text_obj.get_rect()
201         text_rect.x, text_rect.y = 10, 10
202         screen.blit(bg,(0,0)) # 배경 로드를 위한 코드
203         screen.blit(text_obj, text_rect) # text 로드를 위한 코드
```

```
205     # 게임 프레임 처리
206     def display_frame(self, screen):
207         self.draw_info(self.snake.length, self.speed, screen) #draw_info 실행하고
208         self.snake.draw(screen) # 뱀 그리기
209         self.feed.draw(screen) # 먹이 그리기
210         self.obt.draw(screen) # 방해물 그리기
211         screen.blit(screen, (0, 0)) # 스크린 띄우기
212
213     # 리소스 경로 설정
214     def resource_path(relative_path):
215         try:
216             base_path = sys._MEIPASS
217         except Exception:
218             base_path = os.path.abspath(".")
219         return os.path.join(base_path, relative_path)
220
221
222     def main():
223         # 게임 초기화 및 환경 설정
224         pygame.init() # 파이게임 초기화
225         pygame.mixer.init() # 미서 초기화
226         pygame.display.set_caption('Snake Game')
227         pygame.mixer.music.load('/Users/seohyunoh/Documents/python/VMP/4_Snake_Game/assets/bgm.mp3')
228         pygame.mixer.music.set_volume(0.7) # music을 이용해 bgm 깔아주기, 볼륨은 0.7
229         pygame.mixer.music.play(-1) # loop 재생
230         screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT),pygame.FULLSCREEN) # 풀스크린으로 설정
231         clock = pygame.time.Clock()
232         game = Game()
233
234         done = False
235         while not done: # 게임 실행
236             done = game.process_events()
237             game.run_logic()
238             game.display_frame(screen)
239             pygame.display.flip()
240             clock.tick(game.speed)
241
242         pygame.quit() # 종료
243
244
245     if __name__ == '__main__':
246         main()
```

2. Screenshot



Level : 3 Length: 5 Speed: 6.25



Level : 4 Length: 7 Speed: 6.75



I died before I reached the level because of my bad game. So I modified the level and took a screenshot.