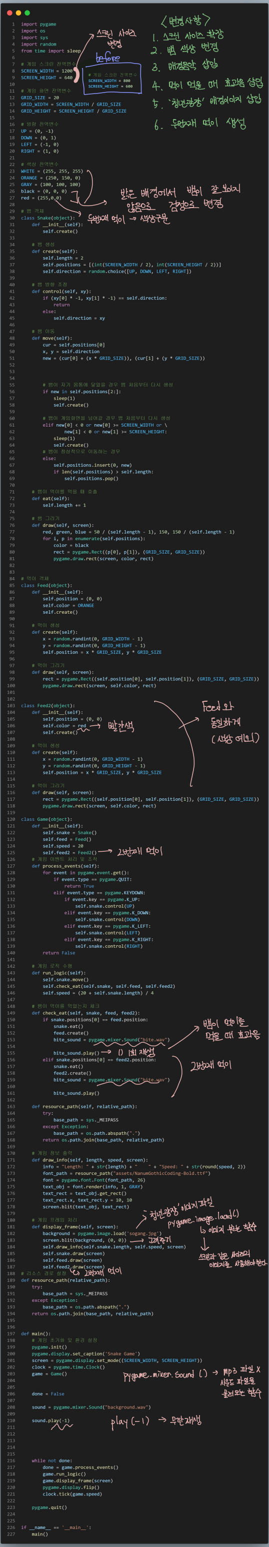


〈변경사항〉

1. 스크린 사이즈 확장
2. 뱀 색상 변경
3. 배경음악 삽입
4. 먹이 먹을 때 효과음 삽입
5. '청년광장' 배경이미지 삽입
6. 두번째 먹이 생성

```python
import pygame
import os
import sys
import random
from time import sleep

# 게임 스크린 전역변수
SCREEN_WIDTH = 1200
SCREEN_HEIGHT = 640

# 게임 화면 전역변수
GRID_SIZE = 20
GRID_WIDTH = SCREEN_WIDTH / GRID_SIZE
GRID_HEIGHT = SCREEN_HEIGHT / GRID_SIZE

# 방향 전역변수
UP = (0, -1)
DOWN = (0, 1)
LEFT = (-1, 0)
RIGHT = (1, 0)

# 색상 전역변수
WHITE = (255, 255, 255)
ORANGE = (250, 150, 0)
GRAY = (100, 100, 100)
black = (0, 0, 0)
red = (255,0,0)
```

스크린 사이즈 변경

before

```python
# 게임 스크린 전역변수
SCREEN_WIDTH = 800
SCREEN_HEIGHT = 600
```

밝은 배경에서 뱀이 잘 보이지 않음으로 검정으로 변경

두번째 먹이 → 색상구분

```python

# 뱀 객체
class Snake(object):
    def __init__(self):
        self.create()

    # 뱀 생성
    def create(self):
        self.length = 2
        self.positions = [(int(SCREEN_WIDTH / 2), int(SCREEN_HEIGHT / 2))]
        self.direction = random.choice([UP, DOWN, LEFT, RIGHT])

    # 뱀 방향 조정
    def control(self, xy):
        if (xy[0] * -1, xy[1] * -1) == self.direction:
            return
        else:
            self.direction = xy

    # 뱀 이동
    def move(self):
        cur = self.positions[0]
        x, y = self.direction
        new = (cur[0] + (x * GRID_SIZE)), (cur[1] + (y * GRID_SIZE))



        # 뱀이 자기 몸통에 닿았을 경우 뱀 처음부터 다시 생성
        if new in self.positions[2:]:
            sleep(1)
            self.create()

        # 뱀이 게임화면을 넘어갈 경우 뱀 처음부터 다시 생성
        elif new[0] < 0 or new[0] >= SCREEN_WIDTH or \
                new[1] < 0 or new[1] >= SCREEN_HEIGHT:
            sleep(1)
            self.create()
        # 뱀이 정상적으로 이동하는 경우
        else:
            self.positions.insert(0, new)
            if len(self.positions) > self.length:
                self.positions.pop()

    # 뱀이 먹이를 먹을 때 호출
    def eat(self):
        self.length += 1

    # 뱀 그리기
    def draw(self, screen):
        red, green, blue = 50 / (self.length - 1), 150, 150 / (self.length - 1)
        for i, p in enumerate(self.positions):
            color = black
            rect = pygame.Rect((p[0], p[1]), (GRID_SIZE, GRID_SIZE))
            pygame.draw.rect(screen, color, rect)


# 먹이 객체
class Feed(object):
    def __init__(self):
        self.position = (0, 0)
        self.color = ORANGE
        self.create()

    # 먹이 생성
    def create(self):
        x = random.randint(0, GRID_WIDTH - 1)
        y = random.randint(0, GRID_HEIGHT - 1)
        self.position = x * GRID_SIZE, y * GRID_SIZE

    # 먹이 그리기
    def draw(self, screen):
        rect = pygame.Rect((self.position[0], self.position[1]), (GRID_SIZE, GRID_SIZE))
        pygame.draw.rect(screen, self.color, rect)


class Feed2(object):
    def __init__(self):
        self.position = (0, 0)
        self.color = red
        self.create()
```

빨간색

Feed와 동일하게 (색상 예외)

```python

    # 먹이 생성
    def create(self):
        x = random.randint(0, GRID_WIDTH - 1)
        y = random.randint(0, GRID_HEIGHT - 1)
        self.position = x * GRID_SIZE, y * GRID_SIZE

    # 먹이 그리기
    def draw(self, screen):
        rect = pygame.Rect((self.position[0], self.position[1]), (GRID_SIZE, GRID_SIZE))
        pygame.draw.rect(screen, self.color, rect)

class Game(object):
    def __init__(self):
        self.snake = Snake()
        self.feed = Feed()
        self.speed = 20
        self.feed2 = Feed2()
```

2번째 먹이

```python
    # 게임 이벤트 처리 및 조작
    def process_events(self):
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                return True
            elif event.type == pygame.KEYDOWN:
                if event.key == pygame.K_UP:
                    self.snake.control(UP)
                elif event.key == pygame.K_DOWN:
                    self.snake.control(DOWN)
                elif event.key == pygame.K_LEFT:
                    self.snake.control(LEFT)
                elif event.key == pygame.K_RIGHT:
                    self.snake.control(RIGHT)
        return False

    # 게임 로직 수행
    def run_logic(self):
        self.snake.move()
        self.check_eat(self.snake, self.feed, self.feed2)
        self.speed = (20 + self.snake.length) / 4

    # 뱀이 먹이를 먹었는지 체크
    def check_eat(self, snake, feed, feed2):
        if snake.positions[0] == feed.position:
            snake.eat()
            feed.create()
            bite_sound = pygame.mixer.Sound("bite.wav")

            bite_sound.play()
        elif snake.positions[0] == feed2.position:
            snake.eat()
            feed2.create()
            bite_sound = pygame.mixer.Sound("bite.wav")

            bite_sound.play()
```

뱀이 먹이를 먹을 때 효과음

() 1회 재생

2번째 먹이

```python

    def resource_path(self, relative_path):
        try:
            base_path = sys._MEIPASS
        except Exception:
            base_path = os.path.abspath(".")
        return os.path.join(base_path, relative_path)

    # 게임 정보 출력
    def draw_info(self, length, speed, screen):
        info = "Length: " + str(length) + "    " + "Speed: " + str(round(speed, 2))
        font_path = resource_path("assets/NanumGothicCoding-Bold.ttf")
        font = pygame.font.Font(font_path, 26)
        text_obj = font.render(info, 1, GRAY)
        text_rect = text_obj.get_rect()
        text_rect.x, text_rect.y = 10, 10
        screen.blit(text_obj, text_rect)

    # 게임 프레임 처리
    def display_frame(self, screen):
        background = pygame.image.load('sogang.jpg')
        screen.blit(background, (0, 0))
        self.draw_info(self.snake.length, self.speed, screen)
        self.snake.draw(screen)
        self.feed.draw(screen)
        self.feed2.draw(screen)
```

'청년광장' 이미지파일
pygame.image.load()
↳ 이미지 부르는 함수
↓
스크린과 같은 사이즈의 이미지를 사용해야한다

그려주기

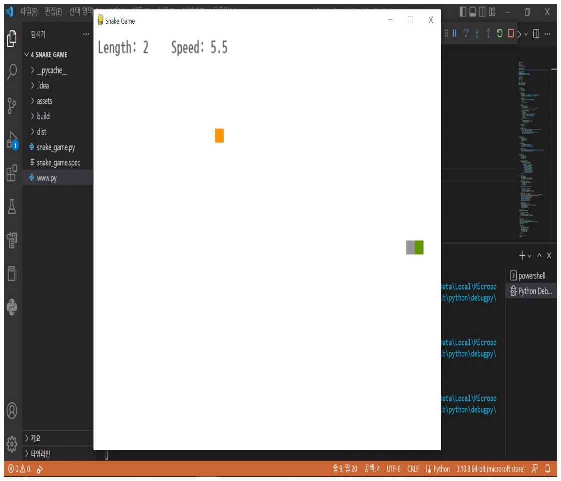
↳ 2번째 먹이

```python
# 리소스 경로 설정
def resource_path(relative_path):
    try:
        base_path = sys._MEIPASS
    except Exception:
        base_path = os.path.abspath(".")
    return os.path.join(base_path, relative_path)


def main():
    # 게임 초기화 및 환경 설정
    pygame.init()
    pygame.display.set_caption('Snake Game')
    screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
    clock = pygame.time.Clock()
    game = Game()


    done = False

    sound = pygame.mixer.Sound("background.wav")

    sound.play(-1)
```

pygame.mixer.Sound () → mp3 파일 X
사운드 파일을 불러오는 함수

play (-1) → 무한재생

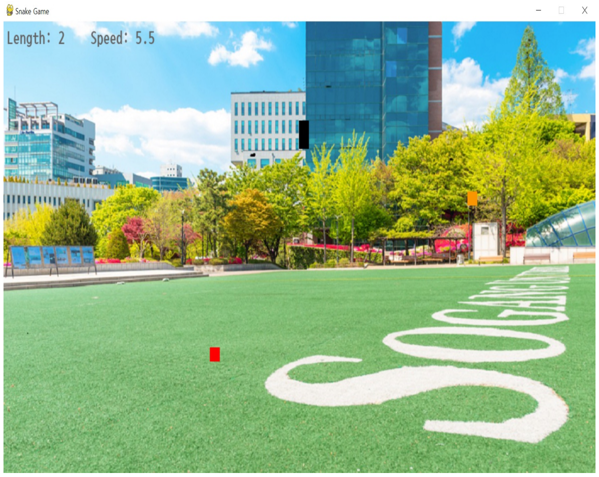
```python




    while not done:
        done = game.process_events()
        game.run_logic()
        game.display_frame(screen)
        pygame.display.flip()
        clock.tick(game.speed)

    pygame.quit()


if __name__ == '__main__':
    main()
```



Snake Game

Length: 2   Speed: 5.5

Before



Snake Game

Length: 2   Speed: 5.5

After