

# OSSP 주간보고서

이번 주  
주요 내용

비파괴 피사체 / 개수조절 / 비행체 속도 제어 / 화면 크기 조절 / 시작화면 이미지

교과목명	오픈소스 소프트웨어 프로그래밍			담당교수	김동호 교수님
과제명	오픈소스 소프트웨어 게임			팀명	무히려좋아
일자	2021/05/23	시간	22:00	장소	webex

## - 진행상황 보고

### 1) 파괴할 수 없는 피사체 등장 및 개수조절

- 기존에 위에서 아래로 내려오는 파괴가능한 피사체만 내려왔음
- 왼쪽에서 오른쪽으로 이동하는, 파괴할 수 없고 피해야만 하는 장애물 생성
- 각각의 객체를 만들고 오른쪽 게임화면 끝으로 사라지면 메모리 효율을 위해 객체를 삭제함
- 랜덤함수를 사용하여 선인장의 사이즈가 결정되게 코딩함

내용



<새로 구현한 피사체>

```
# 장애물 등장
if random.random() > Speed.create_rate_r:
    # 장애물 객체 생성
    block = obj()
    block.put_img('SourceCode/Image/CATUS.png')
    random_size = random.randint(Size.min_size, Size.block_max_size)
    block.change_size(random_size, random_size)
    # block.change_size(Size.block_size, Size.block_size)
    block.x = 10
    block.y = random.randint(0, size[0] - block.sx - round(ss.sx/2))
    block.move = 2 + (Util.score//100)
    Util.block_list.append(block)

d2_list=[]
for i in range(len(Util.block_list)):
    b = Util.block_list[i]
    b.x += b.move
    if b.x >= size[0]:
        d2_list.append(i)

d2_list.reverse()
for d2 in d2_list:
    del Util.block_list[d2]
```

<선인장 피사체 구현 코드>

```

# 비행체 보여주기
ss.show()
# 미사일 보여주기
for m in Util.m_list:
    m.show()
# 피사체 보여주기
for a in Util.a_list:
    a.show()
# 선인장 장애물 보여주기
for d in Util.block_list:
    d.show()

```

- 리스트로 받은 피사체가 게임화면 상에  
보여질 수 있도록 구현한 for loop

## 2) 플레이어 비행체 속도 제어

- 과거엔 score가 높아짐에 따라 피사체들과 장애물의 속도도 빨라졌으나 플레이어의 비행체의 속도는 그대로였음.
- Score 400점마다 비행체의 속도가 1씩 증가하도록 수정
- 점수가 높아짐에 따라 많아지는 피사체를 피하고, 파괴하기 용이해짐

### (코드정보)

```

# score 400점마다 비행체의 속도 1씩 증가
Speed.s_speed = Speed.s_speed + Util.score//400

```

Find related code in 2021-1-OSSPC-MUHIRY0-4

### 3) 마우스로 화면 크기 조절

- 화면 비율을 직접 조절할 수 없었고, 게임이 플레이어의 디스플레이 비율에 맞게 실행되었음
- 플레이어가 화면의 크기를 마우스나 패드의 드래그로 조절할 수 있게 되었음
- 게임창이 축소되거나 확대되었을 때, 게임상 존재하는 모든 객체의 사이즈가 변경됨
- change\_size\_rate라는 이름으로 함수를 만들어 구현. 파라미터로 size를 받음
- 사이즈를 변경함에 있어 발생하는 추가적 문제는 추후에 수정할 예정

```
# 2-1 고정된 화면 크기
# size = [700,800]
# screen = pygame.display.set_mode(size)

# 2-2 플레이어의 컴퓨터 환경에 맞춘 화면의 크기
infoObject = pygame.display.Info()
# 896 * 1020
size = [infoObject.current_w//2,infoObject.current_h*8//9]
screen = pygame.display.set_mode(size,pygame.RESIZABLE)

Find related code in 2021-1-0SSPC-MUHTRY0-4 You
```


```
elif event.type == pygame.VIDEORESIZE:
    width, height = event.w, event.h
    size = [width,height]
    window = pygame.display.set_mode(size, pygame.RESIZABLE)
    Move.position = True

# 마우스로 인해 화면이 작아지면 다른 객체들의 사이즈도 전부 변경
if Move.position is True:
    change_size_rate(size)
```

<화면크기 조정 구현 코드>

```
def change_size_rate(size):
    Size.a_size = size[0]//18
    Size.a_y_size = size[1]//13
    Size.m_size = size[0]//179
    Size.m_y_size = size[1]//68
    Size.min_size = (size[0]//50 + size[1]//50)*2//3
    Size.max_size = (size[0]//30 + size[1]//30)*2//3
    Size.block_max_size = size[0]//10

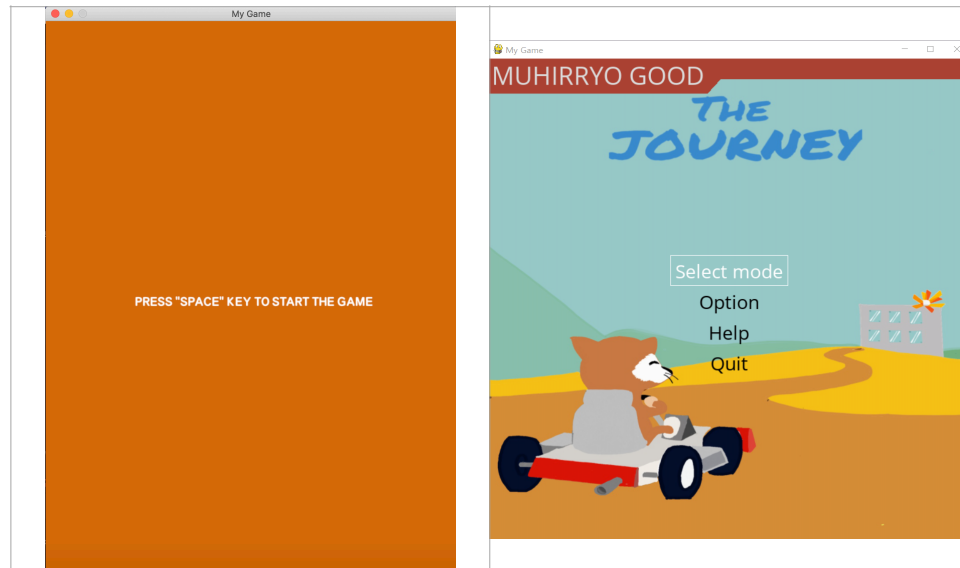
    # 오즈의 갈 나무의 크기를 줄일 시 객체가 화면 밖으로 못나가게 제한 함
    if ss.x + ss.sx > size[0]:
        ss.x = size[0] - ss.sx
    # 바닥 위에서 크기를 줄일 시 객체가 화면 밖으로 못나가게 제한 함
    if ss.y + ss.sy > size[1]:
        ss.y = size[1] - ss.sy
    # 마우스로 객체의 사이즈 변경
    try:
        ss.change_size(Size.a_size, Size.a_y_size)
    except:
        pass
    try:
        # 지금 현재 리사일을 보낼수까지 않는 상태 일 수도 있기 때문
        mm.change_size(Size.m_size, Size.m_y_size)
    except:
        pass
    # You, a day ago - Update Main.py
    try:
        # 함수가 아직 도달하지 못하여 mm2객체가 만들어지지 않았을 수도 있음
        mm2.change_size(Size.m_size, Size.m_y_size)
    except:
        pass
    try:
        random_size = random.randint(Size.min_size, Size.block_max_size)
        block.change_size(Size.block_max_size, Size.block_max_size)
    except:
        pass
    try:
        aa.put_img("SourceCode/Image/scorpion1-removebg-preview.png")
    except:
        pass
    try:
        random_size = random.randint(Size.min_size, Size.max_size)
        aa.change_size(random_size, random_size)
    except:
        pass
```



<화면 크기 조절 영상>

#### 4) 시작화면 이미지 제작 및 메뉴기능 구현

- 기존 단색 바탕화면이었던 시작화면에 이미지를 적용해 구현
- 기본적인 메인메뉴 인터페이스 구현
- 메뉴 각 기능이 게임과 연동될 수 있도록 구현할 예정



<변경 전 시작화면 이미지>

<변경 후 시작화면 이미지>

