

# 오픈소스 SW이해와 실습 기말 프로젝트 팀 보고서

2022년 2학기

분반	오픈소스sw이해화 실습(5115004-01)
조 이름	( 3조)

구분	학번	학년	학과	성명
1	2022042011	1	지능로봇공학과	문서영
2	2022042043	1	지능로봇공학과	이지형
3	2022042039	1	지능로봇공학과	서동환
4		1	지능로봇공학과	김다준

## 1. 프로젝트 이름 : HELLO, OUR NEW PLANET

※팀원들이 만들고자 하는 프로젝트 이름을 입력합니다.

## 2. 목적 및 필요성

※이 프로젝트를 선정한 이유와 어디에 사용되면 좋을지 필요성을 설명합니다.

게임산업이 더욱더 성장하며,세계적 발전을 이루고 있는 지금. 게임은 단순한 취미의 영역을 넘어서서, 문화 융합, 소통의 창구의 기능을 하고 있으며, 신드롬의 영역이라고 말할 수 있을 정도로 지대한 영향력을 발휘하고 있다. 따라서 우리조는 1학기때 배운 파이썬을 이용하여, 파이게임을 이용한 게임을 제작해보고자 하였다. 참신한 스토리를 가진 농사게임의 기본을 구현해본다면, 이것을 기반으로 다른 기능을 추가하여 상용가능한 게임의 기반을 닦을 수 있을 것이다.

## 3. 프로젝트 유형

순수 창작	<input type="radio"/>	모델이 될 프로그램을 선정 후 변형	
-------	-----------------------	---------------------	--

※해당 유형에 ○ 표시합니다.

업무용		게임	<input type="radio"/>	교육용		기타	
-----	--	----	-----------------------	-----	--	----	--

※해당 유형에 ○ 표시합니다.

※기타의 경우 유형을 입력합니다.

#### 4. 프로젝트 개발의 모델이 된 프로그램

※개발하고자 하는 프로그램의 모델이 된 프로그램의 이름, 프로그램의 URL, 프로그램의 첫 화면을 캡처하여 넣습니다.

※순수 창작의 경우 “없음”이라고 입력합니다.

없음. 순수창작물임.

## 5. 프로그램의 개요

### 1) 개발하고 있는 프로그램의 첫 화면

※개발하고 있는 프로그램의 첫 화면을 캡처하여 넣어주십시오.

※첫 화면의 설명을 입력합니다.



- 1) 배경화면//정착해야하는 행성의 모습, 농사게임인 만큼 땅의 넓이가 넓게 편집하였다.
- 2) 상점화면//씨앗구매와 게임의 엔딩을 볼 수 있는 상점 이미지, 키보드를 통하여 조작 가능하다.

## 2) 프로그램의 기능 설명

※개발하고 있는 프로그램의 기능을 자세히 설명합니다.

```
import pygame
pygame.init()

screen_width = 1000
screen_height = 1000

screen = pygame.display.set_mode((screen_width, screen_height))
pygame.display.set_caption("THE FARMER")
background = pygame.image.load("배경이미지.png")
background=pygame.transform.scale(background,(1000,300))
clock=pygame.time.Clock()

#이미지 로딩 및 크기 변경
store=pygame.image.load("상점.png")
store=pygame.transform.scale(store,(1000,1000))

player=pygame.image.load("농부.png")
player=pygame.transform.scale(player,(100,100))

path1=pygame.image.load("표지판1.png")
path1=pygame.transform.scale(path1,(100,100))
path2=pygame.image.load("표지판2.png")
path2=pygame.transform.scale(path2,(100,100))
```

```
path3=pygame.image.load("표지판3.png")
path3=pygame.transform.scale(path3,(100,100))
```

```
ground1=pygame.image.load("ground.png")
ground1=pygame.transform.scale(ground1,(100,100))
ground2=pygame.image.load("ground.png")
ground2=pygame.transform.scale(ground2,(100,100))
ground3=pygame.image.load("ground.png")
ground3=pygame.transform.scale(ground3,(100,100))
```

```
plant1=pygame.image.load("plant1.png")
plant1=pygame.transform.scale(plant1,(100,100))
plant2=pygame.image.load("plant2.png")
plant2=pygame.transform.scale(plant2,(100,100))
plant3=pygame.image.load("plant3.png")
plant3=pygame.transform.scale(plant3,(100,100))
```

```
splant1=pygame.image.load("splant1.png")
splant1=pygame.transform.scale(splant1,(100,100))
splant2=pygame.image.load("splant2.png")
splant2=pygame.transform.scale(splant2,(100,100))
splant3=pygame.image.load("splant3.png")
splant3=pygame.transform.scale(splant3,(100,100))
```

```
gplant1=pygame.image.load("gplant1.png")
gplant1=pygame.transform.scale(gplant1,(100,100))
gplant2=pygame.image.load("gplant2.png")
gplant2=pygame.transform.scale(gplant2,(100,100))
gplant3=pygame.image.load("gplant3.png")
gplant3=pygame.transform.scale(gplant3,(100,100))
```

```
ending=pygame.image.load("ending.png")
ending=pygame.transform.scale(ending,(1000,1000))
```

```
#이미지의 Rect 정보를 저장
player_Rect=player.get_rect()
store_Rect=store.get_rect()
background_Rect=background.get_rect()
ground1_Rect=ground1.get_rect()
ground2_Rect=ground2.get_rect()
ground3_Rect=ground3.get_rect()
```

```
path1_Rect=path1.get_rect()
path2_Rect=path2.get_rect()
path3_Rect=path3.get_rect()
```

```
plant1_Rect=plant1.get_rect()
plant2_Rect=plant2.get_rect()
plant3_Rect=plant3.get_rect()
```

```
splant1_Rect=splant1.get_rect()
splant2_Rect=splant2.get_rect()
splant3_Rect=splant3.get_rect()
```

```
gplant1_Rect=gplant1.get_rect()
gplant2_Rect=gplant2.get_rect()
gplant3_Rect=gplant3.get_rect()
```

```
ending_Rect=ending.get_rect()
```

```
ground1_Rect.x=100
ground1_Rect.y=500
ground2_Rect.x=400
ground2_Rect.y=500
ground3_Rect.x=800
```

ground3\_Rect.y=500

path1\_Rect.x=100

path1\_Rect.y=400

path2\_Rect.x=400

path2\_Rect.y=400

path3\_Rect.x=800

path3\_Rect.y=400

plant1\_Rect.x=1500

plant1\_Rect.y=1500

plant2\_Rect.x=1500

plant2\_Rect.y=1500

plant3\_Rect.x=1500

plant3\_Rect.y=1500

splant1\_Rect.x=1500

splant1\_Rect.y=1500

splant2\_Rect.x=1500

splant2\_Rect.y=1500

splant3\_Rect.x=1500

splant3\_Rect.y=1500

gplant1\_Rect.x=1500

gplant1\_Rect.y=1500

gplant2\_Rect.x=1500

gplant2\_Rect.y=1500

gplant3\_Rect.x=1500

gplant3\_Rect.y=1500

store\_Rect.x=1500

store\_Rect.y=1500

background\_Rect.x=0



```
background_Rect.y=0
```

```
ending_Rect.x=1500
```

```
ending_Rect.y=1500
```

```
player_Rect.centerx=round(screen_width/2) #플레이어 초기위치
```

```
player_Rect.centery=round(screen_height/2)
```

```
dx=0
```

```
dy=0
```

```
a=0 #작물생장
```

```
b=0
```

```
c=0
```

```
q=0 #상점열리는거
```

```
class HUD_coin(pygame.sprite.Sprite):
```

```
    def __init__(self, x, y):
```

```
        self.stage_0_img = pygame.image.load('coin.png')
```

```
        pygame.sprite.Sprite.__init__(self)
```

```
        self.image = self.stage_0_img
```

```
        self.rect = self.image.get_rect()
```

```
        self.rect.topleft = x,y
```

```
class HUD_water(pygame.sprite.Sprite):
```

```
    def __init__(self, x, y):
```

```
        self.stage_1_img = pygame.image.load('water.png')
```

```
        pygame.sprite.Sprite.__init__(self)
```

```
        self.image = self.stage_1_img
```

```
        self.rect = self.image.get_rect()
```

```
        self.rect.topleft = x,y
```

```

class HUD_seed1(pygame.sprite.Sprite):
    def __init__(self, x, y):
        self.stage_2_img = pygame.image.load('seed1.png')

        pygame.sprite.Sprite.__init__(self)
        self.image = self.stage_2_img
        self.rect = self.image.get_rect()
        self.rect.topleft = x,y

```

```

class HUD_seed2(pygame.sprite.Sprite):
    def __init__(self, x, y):
        self.stage_3_img = pygame.image.load('seed2.png')

        pygame.sprite.Sprite.__init__(self)
        self.image = self.stage_3_img
        self.rect = self.image.get_rect()
        self.rect.topleft = x,y

```

```

class HUD_seed3(pygame.sprite.Sprite):
    def __init__(self, x, y):
        self.stage_4_img = pygame.image.load('seed3.png')

        pygame.sprite.Sprite.__init__(self)
        self.image = self.stage_4_img
        self.rect = self.image.get_rect()
        self.rect.topleft = x,y

```

```

object_list =pygame.sprite.Group()
coin_png =HUD_coin(500,880)
object_list.add(coin_png)
water_png = HUD_water(600,880)

```

```

object_list.add(water_png)
seed1_png = HUD_seed1(700,880)
object_list.add(seed1_png)
seed2_png = HUD_seed2(800,880)
object_list.add(seed2_png)
seed3_png = HUD_seed3(900,880)
object_list.add(seed3_png)

```

```

font_test= pygame.font.SysFont(None,30)

```

```

coin=100
water =100
seed1=1
seed2=0
seed3=0

```

```

running = True

```

```

while running:

```

```

    for event in pygame.event.get(): # running 중 키보드나,마우스 입력값(이벤트)을
    체크해주는것

```

```

        if event.type == pygame.QUIT: # 창이 닫히는 이벤트가 발생하였는지
            running = False # 게임이 진행중이 아님

```

```

        if event.type ==pygame.KEYDOWN:

```

```

            if event.key==pygame.K_LEFT:
                dx =-5
            if event.key==pygame.K_RIGHT:
                dx =5
            if event.key==pygame.K_UP:
                dy =-5

```

```

if event.key==pygame.K_DOWN:
    dy=5
if event.key==pygame.K_p: #상점열리는
    q = 1
    store_Rect.x=0
    store_Rect.y=0
if event.key==pygame.K_z: #상점닫히는
    q=0
    store_Rect.x=1500
    store_Rect.y=1500

if q==1: #상점구매 엔딩여기에
    if event.key==pygame.K_1: #눌린 키가 0일때=(소)씨앗
        if coin>= 10:
            seed1+=1
            coin-=10

    if event.key==pygame.K_2:#눌린 키가 0일때=(소)씨앗
        if coin>= 10:
            seed2+=1
            coin-=100

    if event.key==pygame.K_3: #눌린 키가 0일때=(소)씨앗
        if coin>= 10:
            seed3+=1
            coin-=1000

    if event.key==pygame.K_e: #엔딩구매
        if coin>= 10000:
            ending_Rect.x=0
            ending_Rect.y=0
    if event.key==pygame.K_r:
        if coin>= 100:

```

```

        water -= 100
        coin -= 100

    if player_Rect.x < 150 and player_Rect.x > 50 and player_Rect.y < 550 and
player_Rect.y > 450:    #1번씨앗심는땅
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                if seed1 != -1 :
                    if water != 0:
                        a += 1
                        water -= 1

    if player_Rect.x < 450 and player_Rect.x > 350 and player_Rect.y < 550 and
player_Rect.y > 450:
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                if seed2 != -1:
                    if water != 0:
                        b += 1
                        water -= 1

    if player_Rect.x < 850 and player_Rect.x > 750 and player_Rect.y < 550 and
player_Rect.y > 450:
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                if seed3 != -1:
                    if water != 0:
                        c += 1
                        water -= 1

    if event.type == pygame.KEYUP:

        if event.key == pygame.K_LEFT:
            dx = 0

```

```

        if event.key==pygame.K_RIGHT:
            dx =0
        if event.key==pygame.K_UP:
            dy =0
        if event.key==pygame.K_DOWN:
            dy=0

player_Rect.x+=dx
player_Rect.y+=dy

if player_Rect.left<0:
    player_Rect.left=0
if player_Rect.right>screen_width:
    player_Rect.right=screen_height
if player_Rect.top<0:
    player_Rect.top=0
if player_Rect.bottom>screen_height:
    player_Rect.bottom=screen_height

if a==1:  # 식물1 생장함수
    seed1-=1
    a=2
    ground1_Rect.x=1500
    ground1_Rect.y=1500
    plant1_Rect.x=100
    plant1_Rect.y=500

elif a==3:
    plant1_Rect.x=1500

```

```

    plant1_Rect.y=1500
    plant2_Rect.x=100
    plant2_Rect.y=500

elif a==5:
    plant2_Rect.x=1500
    plant2_Rect.y=1500
    plant3_Rect.x=100
    plant3_Rect.y=500

elif a==7:
    a=0
    coin+=20
    plant3_Rect.x=1500
    plant3_Rect.y=1500
    ground1_Rect.x=100
    ground1_Rect.y=500

if b==1:    # 식물2 생장함수
    seed2-=1
    b=2
    ground2_Rect.x=1500
    ground2_Rect.y=1500
    splant1_Rect.x=400
    splant1_Rect.y=500

elif b==4:
    splant1_Rect.x=1500
    splant1_Rect.y=1500
    splant2_Rect.x=400
    splant2_Rect.y=500

```

```

elif b==8:
    splant2_Rect.x=1500
    splant2_Rect.y=1500
    splant3_Rect.x=400
    splant3_Rect.y=500

elif b==10:
    b=0
    coin+=200
    splant3_Rect.x=1500
    splant3_Rect.y=1500
    ground2_Rect.x=400
    ground2_Rect.y=500

if c==1:  # 식물3 생장함수
    seed3-=1
    c=2
    ground1_Rect.x=1500
    ground1_Rect.y=1500
    gplant1_Rect.x=800
    gplant1_Rect.y=500

elif c==5:
    gplant1_Rect.x=1500
    gplant1_Rect.y=1500
    gplant2_Rect.x=800
    gplant2_Rect.y=500

elif c==10:

```



```

    gplant2_Rect.x=1500
    gplant2_Rect.y=1500
    gplant3_Rect.x=800
    gplant3_Rect.y=500

elif c==15:
    c=0
    coin+=2000
    gplant3_Rect.x=1500
    gplant3_Rect.y=1500
    ground1_Rect.x=800
    ground1_Rect.y=500

    screen.fill((139, 0, 255)) #RGB형식으로 이미지 로드
    #스크린의 원하는 좌표에 이미지 복사하기
    screen.blit(background, background_Rect)

    screen.blit(ground1, ground1_Rect)
    screen.blit(ground2, ground2_Rect)
    screen.blit(ground3, ground3_Rect)

    screen.blit(plant1, plant1_Rect)
    screen.blit(plant2, plant2_Rect)
    screen.blit(plant3, plant3_Rect)

    screen.blit(path1, path1_Rect)
    screen.blit(path2, path2_Rect)
    screen.blit(path3, path3_Rect)

    screen.blit(splant1, splant1_Rect)
    screen.blit(splant2, splant2_Rect)
    screen.blit(splant3, splant3_Rect)

```

```

screen.blit(gplant1, gplant1_Rect)
screen.blit(gplant2, gplant2_Rect)
screen.blit(gplant3, gplant3_Rect)

screen.blit(player, player_Rect)
screen.blit(store, store_Rect)

text_coin =font_test.render(str(coin), True,(0,0,0))
screen.blit(text_coin,(520,980))
text_water =font_test.render(str(water), True,(0,0,0))
screen.blit(text_water,(630,980))
text_maxwater =font_test.render("/100", True,(0,0,0))
screen.blit(text_maxwater,(660,980))
text_seed1 =font_test.render(str(seed1), True,(0,0,0))
screen.blit(text_seed1,(750,980))
text_seed2 =font_test.render(str(seed2), True,(0,0,0))
screen.blit(text_seed2,(850,980))
text_seed3 =font_test.render(str(seed3), True,(0,0,0))
screen.blit(text_seed3,(950,980))

for obj in object_list:
    screen.blit(obj.image,(obj.rect.x, obj.rect.y))
screen.blit(ending, ending_Rect)
pygame.display.flip()
clock.tick(60)
pygame.display.update()

pygame.quit()

```

## 6. 조원의 프로그램 개발 역할

※팀장은 조원의 개인보고서 파일을 받아 [개인보고서 1번]의 내용을 [복사]하여 [붙여넣기] 합니다.

1) 조원 이름 : 문서영

-게임의 전체적인 스토리 라인을 만들었다.

나는 스타듀벨리라는 게임을 모티브 삼아서, 농장 게임을 제안하였다. 개발 기간이 짧기 때문에, 스타듀벨리의 모든 기능(광산, 낚시, 농사)를 모두 구현하지 못하겠다고 생각하였고, 이에 농사만을 구현하기로 팀원과의 합의를 이끌어 내었다. 흔한 농사 게임과는 우리조만의 차별점이 있으면 좋겠다고 생각하여, 외계행성에서의 농사라는 주제를 제안하였다. 외계행성에서의 농사를 성공시켜서, 결론적으로 외계 식민지를 건설하는 것을 이 게임의 엔딩으로 삼았다.

-게임의 그래픽 디자인을 맡았다.

게임의 배경에 어울리는 사진을 찾고, 편집하여 외계행성에서의 농사가 콘셉트인 우리 게임에 맞는 최적의 배경을 만들었다. 또한 상점에서의 배경화면은 오로지 내가 내손으로 그린 것이다. 우리게임에서는 최대 3개의 씨앗을 경작할 수 있는데, 그 씨앗에 관한 부분도 내가 디자인한 사진을 활용하였다.

-코드 작성을 하였다.

내가 맡은 코드 부분은 파이게임이 시작할 때 배경화면이 게시되는 부분/ 상점키를 눌렀을 때 상점화면이 열리는 부분/ 각 씨앗이 지정된 키를 눌렀을 때 보유 씨앗의 수량이 증가하는 부분/ 캐릭터가 x,y축으로 움직이는 부분의 코드를 작성하였다.

통합 코드에서 이미지 파일 삽입을 내가 맡았다. 최종 완성본의 실행 동영상을 만들었다.

2) 조원 이름 : 이지형

제가 담당한 프로젝트에서의 역할은 코드작성입니다.

기획과 시나리오, 디자인 작성은 모든 조원들과 상의하면서 같이했습니다.

제가 담당한 파트는 작물에 대한 성장코드, HUD 코드, 상점 코드, 물뿌리개 사용코드, 타이틀 화면 제작과 모든 조원이 만든 코드를 합쳐 실행되게 만드는 작업을 했습니다.

작물성장코드와 관련해서는 게임에는 구현되지 못했으나 개발기획당시 time 함수를 사용하여 작물의 성장기준을 시간에 두는 방법과 pygame.locals 를 사용하여 키입력에 따른 작물성장방법 두가지 방법을 생각하고 이 부분을 담당한 둘이서 따로 개발하고 더 구현이 잘되는 방법을 사용하고자 하였습니다.

이중 제가 개발하려 했던 파트가 time을 활용한 방법이었습니다.

#### time 함수사용한 작물성장코드설명

땅클래스를 통해 배경에 땅을 구현하고 물을주면 15초간 젖은상태가 되고, 그 시간이 지나면 다시 마른상태가 되게 만듦.

작물클래스는 땅이 마른상태 일 때 물을 주면 다음단계로 성장하고 4단계를 거쳐 마지막단계에서 는 수확이 되게 만듦.

수확시 씨앗과 일정량의 돈을 받고 인벤토리창의 상태를 바꾸는 클래스만듦.

#### HUD 코드 설명

인벤토리 창을 좌상단에 만들어 보유한 돈의양, 씨앗123의 양, 물의 양을 표시하게 만듦.

#### 상점 코드 설명

p를 누르면 상점창이 열리고 거기서 상품을사거나 엔딩을 볼수있게 만들었음. z를 누르면 상점에 서 빠져나가서 원래 농사짓던 화면을 출력하게 만듦.

#### 물뿌리개 코드

spacebar 를 눌러 식물을 성장시킬때마다 1씩소모하고 상점에서 100coin 으로 다시 충전할수있게 만들어 게임의 재미를 추구함

그이외 다른 조원들이 만든 코드를 받아서 변수이름과 함수이름 등을 통일하고 코드를 합친뒤 타이틀화면과 엔딩장면을 추가하여 게임을 완성시킴

#### 3) 조원 이름 : 서동환

작물이 : 키보드를 눌렀을때 일정 수에 도달하면 다음 작물로 넘어가고 다자라면

수확되고 돈이 오르는 것을 구현하는 것을 맡았으며 작물을 그리고 엔딩장면을 그렸다 코.

드를 합칠때 협력하여 여러 오류들을 해결하는 역할을 맡았다

#### 4) 조원 이름 : 김다준

## 7. 개발하고 있는 프로그램의 제한점

※팀장은 조원의 개인보고서 파일을 받아 [개인보고서 2번]의 내용을 [복사]하여 [붙여넣기] 합니다.

※없는 경우 '없음'을 입력합니다.

### 1) 기능상의 제한점

#### 1) 조원 이름 : 문서영

기능구현의 시간이 촉박했다 보니, 외계에서 발생할 수 있는 다양한 변수를 구현하지 못했다는 아쉬움이 가장크다. 예를 들면, 자기장 폭풍, 태양풍, 운석 충돌등 차별화된 외계에서의 농사라는 장점을 살릴 수 있는 요소들이 있었지만, 시간이 부족하여 구현하지 못하였다. 또한 식물이 생명체인 만큼 시간이 지나면서 자라는 것이 당연한데, 시간 함수가 다루기 어려울지라 캐릭터가 돌보아 주었을 때(생장시켜 주었을 때)만 자란다는 아쉬움이 있다.

#### 2) 조원 이름 : 이지형

-작물성장코드

위 설명한 작물성장코드를 구현가능하게 코딩하고 싶었지만 시간이 지남에 따라 상태와 이미지 파일을 바꾸는 코드는 제작하였으나 농작물이 심어졌을때 현재시간을 식물클래스에 변수로 주는 것을 못해서 time 함수 활용을 포기했음

-스토리 진행에따른 땅추가 지급

게임을 진행하며 돈을 벌면 상점에 스토리를 구매할 수 있게하여 스토리 진행을 보여주고, 스토리가 끝나고 다시 게임으로 돌아오면 작물을 심을수있는 땅을 추가로 지급하는 코딩을 하고 싶었으나 개발진행이 더더 구현을 포기함 스토리를 간소화하고 정해진금액을 벌면 바로 엔딩을 볼수있도록 계획을 수정함.

-식물 마지막단계일때 일정시간 수확안하면 상함(수확시 아무것도 없음)

time 함수를 활용하고자했던 이유였으나 기술상 문제로 구현하지 못함

-우물 코드

물뿌리개의 물을 상점에서 구매하는것이 아니라 맵에 우물을 구현하여 그곳에서 상호작용을 통해 물뿌리개를 충전하게 만들고 싶었으나 개발진행이 더더 구현을 포기함

-방해물 코드

time 함수를 사용하지않고 키입력 횟수에 따라 작물을 성장시키다보니 게임의 재미가 떨어질것같아서 맵을 랜덤하게 돌아다니는 장애물을 만들어 피하면서 농작물을 키우게하면 게임의 흥미도를 올릴수있을거 같아서 구현하려했으나 개발진행이 더더 구현을 포기함

3) 조원 이름 : 서동환

타임함수를 이용하여 일정 시간이 지나면 그림이 자동으로 바뀌는 것을 구현하려 했으나 예상치 못한 오류와 루프문에서 시간함수를 다루는 것이 생각보다 어려워 실패했다.

4) 조원 이름 : 김다준

## 8. 프로젝트 후기

※팀장은 조원의 개인보고서 파일을 받아 [개인보고서 3번]의 내용을 [복사]하여 [붙여넣기] 합니다.

※없는 경우 '없음'을 입력합니다.

### 1) 프로젝트 과정에서 좋았던 점

1) 조원 이름 : 문서영

나는 웹소설을 보는 취미가 있는데, 나의 진로가 개발자가 되는 것이다보니. 그에 관련한 소설을 보기도 하였다. 그 중 아마도 스타듀벨리 개발자를 모티브로 한듯한 웹소설도 읽었는데, 그 소설을 보고 감명을 받아 나 또한 작은 인디 게임을 만들어 보고 싶다고 생각하였다. 따라서 게임을 만들자라는 이야기가 나왔을 때 적당히 타협한 작은 플레쉬게임이 아닌 정말 퀄리티가 있는 인디게임을 만들고 싶다고 생각하였다. 생각만큼 결과물이 남들 눈에 있어보이지 않는 것 같다. 아무래도 시간도 촉박하였고, 모두가 처음 시도하는 게임 개발이어서 그런 것 같다. 그러나 우선 꿈꿔왔던 게임을 완성하였다는 것이 의미 있고, 뿌듯하다. 겨울 방학때 게임 개발 능력을 더 길러서, 정말 번듯한 게임 하나 만들어 보고 싶다는 새로운 꿈을 찾았다는 것도 이 과목에 감사하게되는 요인 중 하나인 것 같다.

2) 조원 이름 : 이지형

평소 게임만드는것에 관심이 있었고 다른 플랫폼을 활용해서 만들어본적도있었는데, 파이썬을 올해 처음배우게되었고, 파이썬을 활용하여 게임을 만들수있는 pygame이라는 좋은 게임제작라이브러리를 알게 되어 좋았다.

깃허브 사용법을 배워 코딩협업하는법을 배운것도 유익했다. 카톡등의 다른 연락망으로 코드를 주고 받게되면 코드수정이나 파악이 어려웠는데, 깃허브를 통해 편하게 코딩협업이 가능함을 알게 되었다.

### 3) 조원 이름 : 서동환

여러 코드들을 작성해 보면서 문제상황을 여러가지 방법으로 해결해 보려고 노력하는 점에서 오류를 해결하는 능력이나 문제해결 능력을 키울 수 있었으며 확실히 혼자 문제를 해결하는 것 보다는 여럿이 해결하는게 더 낫다는 것을 깨닫게 되었다.

### 4) 조원 이름 : 김다준

## 2) 프로젝트를 하면서 아쉬웠던 점

### 1) 조원 이름 : 문서영

몇 번에 걸쳐서 언급하지만, 학기 중이어서 충분한 시간을 투자하는데 어려움이 있었다. 그러나 우리 조는 최선을 다하여 우리가 할 수 있는 최대의 노력을 했다고 생각한다. 후회가 없기에, 아쉬움또한 없다.

### 2) 조원 이름 : 이지형

파이썬에 대한 지식이 너무 부족해서 참고자료 이해도 어렵고 처음보는 함수 사용도 너무 어려워 기획한 만큼 구현하지 못한점이 제일 아쉬웠다. 파이썬이나 다른 프로그래밍을 더 배우고 다시 프로젝트를 하는 기회를 가졌으면 좋겠다고 생각한다.

### 3) 조원 이름 : 서동환

오픈소스에 대해서 개념만 다룰 것이 아니라 오픈소스에 관련된 프로그램이나 os등을 이용하여 프로젝트를 구성할수 있는 능력을 키울 수 있었다라면면 더 좋았을것 같다

### 4) 조원 이름 : 김다준

