

기말 프로젝트 보고서

서울여자대학교 소프트웨어와 창의적 사고 6팀



서울여자대학교
SEOUL WOMEN'S UNIVERSITY

2023111398 이시연

2023111389 박송희

2023111344 이승은

2023111365 홍서연

목 차

1. 프로젝트

- 1) 프로젝트의 필요성
- 2) 프로젝트의 중요성
- 3) 프로젝트 목표 및 내용
 - 3-1) 개발 목표
 - 3-2) 개발 내용
- 4) 프로그램 흐름도
- 5) 팀원의 역할 분담

2. 프로젝트 수행 결과 및 소감

- 1) 프로젝트 수행과정
- 2) 프로젝트 수행결과
- 3) 프로젝트 수행 시 문제점
- 4) 프로젝트의 향후 개선점
- 5) 프로젝트 수행 소감

1. 프로젝트

1) 프로젝트의 필요성

- py game이라는 라이브러리를 이용하여 유년 시절에 했던 게임을 만들고자 함

2) 프로젝트의 중요성

- 아이디어를 내고 수업에서 배운 개념과 새로운 개념을 융합시켜 소프트웨어로 실현

3) 프로젝트 목표 및 내용

3-1) 개발 목표

- 수업 시간에 배운 개념과 새로운 개념을 융합하여 비행기 슈팅 게임 구현하기

3-2) 개발 내용

- 우주선에서 나오는 레이저를 이용하여 장애물인 외계인을 맞춰 없애는 게임
- 외계인을 없앨 경우 점수가 올라가고, 마지막에 보스와 싸워 이기는 방식으로 제작함

4) 프로그램 흐름도

- 파이 게임을 이용하여 코딩
- 파이썬으로 작성 가능한 게임들의 멀티미디어 표현을 위한 라이브러리 생성
- 게임 배경 1024 x 720으로 설정하기
- 캐릭터의 크기와 생성 코드 설정하기
- 외계인의 충돌 수를 표시하는 함수 생성하기
- 놓친 외계인의 수가 10마리 이상이면 게임 끝을 표시하는 함수 생성하기

- 우주선이 외계인에 3번 충돌하면 게임 끝을 표시하는 함수 생성하기
- 게임에서 승리할 때 표시되는 함수 설정하기
- 보수의 체력을 표시하는 코드 만들기
- 우주선을 움직일 수 있는 방향키 조절이 가능하도록 코딩하기

디자인 부분

- 외계인 캐릭터 등 관련 이미지와 배경 디자인 일러스트로 제작하기

5) 팀원의 역할 분담

- 박송희 2023111389: 코딩
- 이승은 2023111344: ppt 제작 & 디자인
- 이시연 2023111398: 발표 & 보고서 작성
- 홍서연 2023111365: 디자인 & 기획서 작성

2. 프로젝트 수행 결과 및 소감

1) 프로젝트 수행 과정

```
1  import pygame
2  import sys
3  import random
4  from time import sleep
```

파이썬으로 게임을 만들기 위해서는 pygame이라는 라이브러리를 이용해야 한다.

```
6  #배경 색 및 가로 세로 설정
7  BLACK = (0, 0, 0)
8  padwidth = 720
9  padheight = 1024
10
11 #각 이미지 불러오기
12 Alienimage = ['외계인 1.png', '외계인 2.png']
13 AngryAlien = ['공격 1.png', '공격 2.png']
14 Bossimage = ['보스.png', '화내는 보스.png']
15 Henchmanimage = ['외계인 3.png', '외계인 4.png']
16 Sound = ['폭발 1.mp3', '폭발 2.mp3']
17
```

먼저 게임 창을 불러오기 위해 가로, 세로, 색 코드에 전역변수를 지정한 후 게임에 필요한 이미지들의 이름을 리스트에 넣어준다.

```
18 #이미지 게임에 불러오기 함수
19 def Object(obj, x, y):
20     global gamePad
21     gamePad.blit(obj,(x,y))
```

게임판에서 각 캐릭터들의 x, y값 위치에 나타낼 수 있도록 함수를 만든다.

```
23 #게임 화면 구성
24 def initGame():
25     global gamePad, clock, background, fighter, missile, die, A_die, B_die, missilesound, Gameoversound
26     pygame.init()
27     gamePad = pygame.display.set_mode((padwidth, padheight))
28     pygame.display.set_caption('shooting_game')
29     background = pygame.image.load('background.jpg') #배경 화면
30     fighter = pygame.image.load('testfighter.png') #전투기 그림
31     missile = pygame.image.load('missile.png') #미사일 그림
32     die = pygame.image.load('explosion1.png') #외계인 죽을 때 그림
33     A_die = pygame.image.load('explosion2.png') #화난 외계인 죽을 때 그림
34     B_die = pygame.image.load('explosion3.png') #보스 죽을 때 그림
35     pygame.mixer.music.load('backgroundmusic.wav') #배경 음악
36     pygame.mixer.music.set_volume(0.5)
37     Gameoversound = pygame.mixer.Sound('crash.mp3') #게임 오버 음악
38     missilesound = pygame.mixer.Sound('missile.mp3') #미사일 음악
39     pygame.mixer.music.play(-1) #계속해서 실행
40     clock = pygame.time.Clock()
```

게임을 초기화하고 시작하는 함수이다. 이때 게임을 시작하기 위해서 필요한 전역 변수들을 설정해주고 이미지, 사운드를 불러온다.

26째 줄 - 파이게임 라이브러리 초기화

27째 줄 - 화면 구성

40째 줄 - 초당 프레임

```
476     pygame.display.update()
477     clock.tick(100)
478
479     pygame.quit()
480
481     initGame()
482     runGame()
```

화면을 업데이트하고, 초당 프레임을 100으로 설정하였다. 끝으로 초기화 한 후 다시 게임을 시작하도록 시작 함수를 불러온다.

```

42 #화면 중간에 메시지 띄우기
43 def Message(text):
44     global gamePad, Gameoversound
45     textfont = pygame.font.Font('NanumGothic.ttf',80)
46     text = textfont.render(text, True, (255,0,0))
47     text_center = text.get_rect()
48     text_center.center = ((padwidth/2),(padheight/2))
49     gamePad.blit(text,text_center)
50     pygame.display.update()
51

```

```

73 #총탄 외계인 수가 10마리이면 게임 끝을 표시하는 함수
74 def Gameover():
75     global gamePad
76     Message('--GAME OVER--')
77     pygame.mixer.music.stop()
78     Gameoversound.play()
79     sleep(10)
80     pygame.mixer.music.play(-1) #(-1)은 무한반복을 뜻함
81     runGame()
82
83 #전투기가 외계인에 3번 충돌하면 게임 끝을 표시하는 함수
84 def Crash():
85     global gamePad
86     Message('--전투기 파괴--')
87     pygame.mixer.music.stop()
88     Gameoversound.play()
89     sleep(10)
90     pygame.mixer.music.play(-1)
91     runGame()
92
93 #게임에서 승리할 때 표시되는 함수
94 def Win():
95     global gamePad
96     Message('게임 승리!!')
97     pygame.mixer.music.stop()
98     sleep(5)
99     pygame.quit()

```

먼저 Message 함수를 설정해 특정 사건이 발생 되면 화면 중간에 텍스트를 나타낼 수 있도록 한다. “pygame.display.update()” 코드를 통해 메시지를 화면에 표현할 수 있도록 업데이트한다.

Message 함수를 이용해 Gameover, Crash, Win 함수를 만든다.

Gameover 함수: 외계인을 10마리 놓치거나 보스를 잡는 시간이 오버 되면 텍스트와 함께 게임 오버 사운드가 나옴고 배경음악은 멈춘다. 그리고 10초 뒤에 다시 게임을 시작하기 위해서 runGame 함수를 불러오고, 다시 배경음악을 무한 재생한다.

Crach 함수: 외계인과 전투기가 부딪히면 텍스트와 게임오버 사운드가 나옴고 이때 배경음악은 멈춘다. 그리고 10초 뒤에 다시 게임을 시작하기 위해서 runGame 함수를 불러오고, 다시 배경음악을 무한 재생한다.

Win 함수: 보스를 잡으면 텍스트가 나온다. 그리고 게임은 자동으로 꺼진다.


```

52 #보스 시간 설정
53 total_time = 100 #100초
54 def timer(count):
55     global gamePad
56     font = pygame.font.Font('NanumGothic.ttf', 25)
57     text = font.render('남은 시간 :'+str(count)+'초',True,(255,255,255))
58     gamePad.blit(text,(520,0))
59
60 #처치한 외계인 수와 놓친 외계인 수 표시하는 함수
61 def Score(count):
62     global gamePad
63     font = pygame.font.Font('NanumGothic.ttf', 20)
64     text = font.render('점수 :' + str(count),True,(255,255,255))
65     gamePad.blit(text,(10,0))
66
67 def Passed(count):
68     global gamePad
69     font = pygame.font.Font('NanumGothic.ttf', 20)
70     text = font.render('놓친 외계인 :' + str(count),True,(255,0,0))
71     gamePad.blit(text,(10,30))
72

```

Time 함수: 보스 이벤트가 실행되면, 남은 시간을 표현해주기 위해서 위치와 색을 지정해준다.

Score: 외계인 처치 시 점수 표시를 위한 함수

Passed 함수: 놓친 외계인의 수를 표시

```

101 #보스 체력 표시 함수
102 def Health(count):
103     global gamePad
104     font = pygame.font.Font('NanumGothic.ttf', 20)
105     text = font.render('목숨 :{}'.format(str(count)),True,(255,255,255))
106     gamePad.blit(text,(200,0))

```

보스의 체력이 실시간으로 변하는 것을 표현하기 위해 format을 썼다.

```

def runGame():
    global gamePad, clock, background, fighter, missile, die, A_die, B_die, missilesound

```

실제 게임 구동을 위한 함수이다. 함수 안에 내용이 너무 길기 때문에 설명은 모두 이 함수 안의 내용이라고 생각하면 된다.

```

111     #전투기 크기
112     f_size = fighter.get_rect().size
113     f_width = f_size[0]
114     f_height = f_size[1]
115
116     #전투기 초기 위치
117     x = padwidth * 0.45
118     y = padheight * 0.9
119     fighterX = 0
120     fighterY = 0
121
122     #미사일 좌표
123     missileXY = []
124
125     #외계인 랜덤 생성 및 크기 설정
126     Alien = pygame.image.load(random.choice(Alienimage))
127     A_size = Alien.get_rect().size
128     A_width = A_size[0]
129     A_height = A_size[1]
130     ExplotionSound1 = pygame.mixer.Sound(Sound[0])
131
132     #화난 외계인 랜덤 생성 및 크기 설정
133     AAlien = pygame.image.load(random.choice(AngryAlien))
134     AA_size = AAlien.get_rect().size
135     AA_width = AA_size[0]
136     AA_height = AA_size[1]
137
138     #보스 생성 및 크기 설정
139     Boss = pygame.image.load(Bossimage[0])
140     B_size = Boss.get_rect().size
141     B_width = B_size[0]
142     B_height = B_size[1]
143
144     #화난 보스 생성 및 크기 설정
145     AngryBoss = pygame.image.load(Bossimage[1])
146     AB_size = AngryBoss.get_rect().size
147     AB_width = AB_size[0]
148     AB_height = AB_size[1]
149     ExplotionSound2 = pygame.mixer.Sound(Sound[1])
150
151     #보수 부하 생성 및 크기 설정
152     Henchman = pygame.image.load(random.choice(Henchmanimage))
153     H_size = Henchman.get_rect().size
154     H_width = H_size[0]
155     H_height = H_size[1]

```

전투기, 외계인, 미사일의 위치(좌표)와 크기를 설정한다. 또한 각 캐릭터의 너비와 높이가 앞으로 필요함으로 각 Size 리스트를 이용해주었다.


```

157     #외계인 초기 설정
158     AlienX = random.randrange(0, padwidth - A_width)
159     AlienY = 0
160     A_speed = 2
161
162     #화난 외계인 초기 설정
163     AngryAlienX = random.randrange(0, padwidth - AA_width)
164     AngryAlienY = 0
165     AA_speed = 3.5
166
167     #보스 초기 설정
168     Boss_Health = 100
169     BossX = 250
170     BossY = 250
171     B_speed = 2
172
173     #화난 보스 초기 설정
174     ABossX = 250
175     ABossY = 250
176     AB_speed = 3
177
178     #보스 부하 초기 설정
179     HenchmanX = random.randrange(0, padwidth - H_width)
180     HenchmanY = 0
181     H_speed = 4
182
183     #외계인이 미사일에 맞았을 때
184     hit = False
185     hitcount = 0
186     A_pass = 0
187
188     #화난 외계인이 미사일에 맞았을 때
189     A_hit = False
190
191     #보스 부하들이 미사일에 맞았을 때
192     H_hit = False

```

외계인과 보스 부하는 랜덤 라이브러리를 이용해 X좌표가 랜덤으로 배정된다. 단 게임판을 나가거나 밖에서 생성되지 않도록 범위를 제한한다. 캐릭터마다 스피드를 다르게 하였고, 각 외계인이 미사일에 맞았을 때의 상황에서 필요한 변수를 지정한다.

```

194     onGame = False
195     while not onGame:
196         for event in pygame.event.get():
197             if event.type in [pygame.QUIT]:
198                 pygame.quit()
199                 sys.exit()
200
201             #전투기 움직이기
202             if event.type in [pygame.KEYDOWN]:
203                 if event.key == pygame.K_LEFT:
204                     fighterX -= 3
205
206                 elif event.key == pygame.K_RIGHT:
207                     fighterX += 3
208             #추가한 요소 / 방향키 위 버튼 누르면 앞으로, 뒤 버튼 누르면 뒤로
209             elif event.key == pygame.K_UP:
210                 fighterY -= 3
211
212             elif event.key == pygame.K_DOWN:
213                 fighterY += 3
214             #ESC 누르면 화면 꺼짐
215             if event.key == pygame.K_ESCAPE:
216                 onGame = True
217             #스페이스바 누르면 미사일 발사
218             elif event.key == pygame.K_SPACE:
219                 missilesound.play()
220                 missileX = x + f_wigth/2
221                 missileY = y - f_height
222                 missileXY.append([missileX, missileY])
223
224             if event.type in [pygame.KEYUP]:
225                 if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
226                     fighterX = 0
227                 elif event.key == pygame.K_UP or event.key == pygame.K_DOWN:
228                     fighterY = 0

```

While문을 쓰는 이유는 While문 안에서의 모든 코드는 루프를 돌아가게, 즉 이해하기 쉽게 알고리즘이라고 생각하면 된다.

196째 줄 - pygame.event.get()은 다양한 이벤트를 리턴한다.

197~199 줄 - 마우스로 창을 닫으면 게임이 종료된다.

201~213 줄 - 방향키를 지정해 앞, 뒤, 왼, 오로 움직일 수 있게 했다.

214~216 줄 - ESC를 누르면 게임이 꺼진다.

217~222 줄 - 스페이스바를 누르면 미사일 소리와 전투기의 위치에서 미사일이 나가도록 변수를 지정한다. 미사일의 X, Y값을 위해 지정해 두었던(123) missileXY 리스트에 추가한다.

```

230     Object(background, 0, 0)
231     Object(fighter, x, y)
472     Object(Alien, AlienX, AlienY)
473     Object(AAlien, AngryAlienX, AngryAlienY)

```

배경과 전투기, 외계인, 화난 외계인을 게임판에 그린다.

```

233     x += fighterX
234     if x < 0:
235         x = 0
236     elif x > padwidth - f_wigth:
237         x = padwidth - f_wigth
238
239     #전투기가 앞으로 가는 구역 제한
240     y += fighterY
241     if y < 500 :
242         y = 500
243     elif y > padheight - f_height:
244         y = padheight - f_height

```

전투기가 화면 밖으로 나가기 않도록 구역을 제한한다. Y값은 보스와의 접촉을 피하기 위해 짧게 제한해주었다.

```

245
246     #전투기가 외계인과 충돌했는지 체크
247     if y < AngryAlienY - AA_height:
248         if (AngryAlienX > x and AngryAlienX < x + f_wigth) or (AngryAlienX + AA_width > x and AngryAlienX + AA_width < x + f_wigth):
249             Crash()
250
462     #전투기가 부하와 충돌했는지 체크
463     if y < HenchmanY - H_height:
464         if (HenchmanX > x and HenchmanX < x + f_wigth) or (HenchmanX + H_width > x and HenchmanX + H_width < x + f_wigth):
465             Crash()

```

좌표를 계산해서 전투기와 장애물이 부딪히면 Crash 함수를 이용해 게임을 종료 시킨다.

```

251     #미사일 화면에 그리기
252     if len(missileXY) != 0:
253         for i, xy in enumerate(missileXY):
254             xy[1] -= 10
255             missileXY[i][1] = xy[1]
256
257             #미사일이 외계인을 맞혔을 경우
258             if xy[1] < AlienY:
259                 if xy[0] > AlienX and xy[0] < AlienX + A_width:
260                     missileXY.remove(xy)
261                     hit = True
262                     hitcount += 1
263
264             if xy[1] <= 0:
265                 try:
266                     missileXY.remove(xy)
267                 except:
268                     pass
269
270             #화난 외계인을 맞혔을 경우
271             if xy[1] < AngryAlienY:
272                 if xy[0] > AngryAlienX and xy[0] < AngryAlienX + AA_width:
273                     missileXY.remove(xy)
274                     A_hit = True
275                     hitcount += 1
276
277             if xy[1] <= 0:
278                 try:
279                     missileXY.remove(xy)
280                 except:
281                     pass
282
283             #보스를 맞출경우
284             if xy[1] < BossY:
285                 if xy[0] > BossX and xy[0] < BossX + B_width:
286                     missileXY.remove(xy)
287                     hitcount += 2
288                     Boss_Health -= 2
289
290             if xy[1] <= 0:
291                 try:
292                     missileXY.remove(xy)
293                 except:
294                     pass
295
296     if len(missileXY) != 0:
297         for bx, by in missileXY:
298             Object(missile, bx, by)
299
300     #점수 표시
301     Score(hitcount)

```

```

395 #미사일이 화난 보스를 맞출 경우
396 if len(missileXY) != 0:
397     for i, xy in enumerate(missileXY):
398         xy[1] -= 10
399         missileXY[i][1] = xy[1]
400     if xy[1] < ABossY:
401         if xy[0] > ABossX and xy[0] < ABossX + AB_width:
402             missileXY.remove(xy)
403             hitcount += 4
404             Boss_Health -= 2
405
406     if xy[1] <= 0:
407         try:
408             missileXY.remove(xy)
409         except:
410             pass
411
412     if len(missileXY) != 0:
413         for bx, by in missileXY:
414             Object(missile, bx, by)
415
422     if len(missileXY) != 0:
423         for i, xy in enumerate(missileXY):
424             xy[1] -= 10
425             missileXY[i][1] = xy[1]
426
427     if xy[1] < HenchmanY:
428         if xy[0] > HenchmanX and xy[0] < HenchmanX + H_width:
429             missileXY.remove(xy)
430             H_hit = True
431             hitcount += 3
432
433     if xy[1] <= 0:
434         try:
435             missileXY.remove(xy)
436         except:
437             pass

```

251~262 줄 - missileXY 리스트에 좌표가 들어 있으면 하나씩 추출해 좌표를 갱신한다. 마사일의 속도는 10으로 하였고 미사일이 게임판을 넘어가면 리스트에서 해당 좌표를 삭제한다.

263~268 줄 - 그런 다음 미사일이 외계인에 명중했는지 If문을 통해 체크한다. 명중 시 미사일 좌표를 삭제하고 hit를 True로 바꿔준다. 외계인마다 점수를 달리하였다. 그리고 보스를 맞출 경우 피가 까이도록 하였다.

301째 줄 - Score 함수를 이용해 점수를 표시한다.


```

331     #외계인의 죽음 표시하기
332     if hit:
333         Object(die, AlienX, AlienY)
334         ExplotionSound1.play() #폭발 사운드
335
336         #새로운 외계인 랜덤 소환
337         Alien = pygame.image.load(random.choice(Alienimage))
338         A_size = Alien.get_rect().size
339         A_width = A_size[0]
340         A_height = A_size[1]
341         AlienX = random.randrange(0, padwidth - A_width)
342         AlienY = 0
343         hit = False
344
345         #외계인 3마리 잡을 때 마다 속도 증가 및 유지
346         A_speed += 0.2
347         if A_speed >= 5:
348             A_speed = 5
349
350     #화난 외계인의 죽음 표시하기
351     if A_hit:
352         Object(A_die, AngryAlienX, AngryAlienY)
353         ExplotionSound1.play() #폭발 사운드
354
355         #새로운 화난 외계인 랜덤 소환
356         AAlien = pygame.image.load(random.choice(AngryAlien))
357         AA_size = AAlien.get_rect().size
358         AA_width = AA_size[0]
359         AA_height = AA_size[1]
360         AngryAlienX = random.randrange(0, padwidth - AA_width)
361         AngryAlienY = 0
362         A_hit = False
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

미사일이 외계인, 부하를 맞췄을 때, 폭발 이미지와 죽는 사운드를 넣어준다. 그리고 새로운 외계인과 부하를 등장시킨다. 또 일반 외계인을 잡으면 일반 외계인의 속도가 증가한다.


```

304     AlienY += A_speed
305
306     if AlienY > padheight:
307         Alien = pygame.image.load(random.choice(Alienimage))
308         A_size = Alien.get_rect().size
309         A_width = A_size[0]
310         A_height = A_size[1]
311         AlienX = random.randrange(0, padwidth - A_width)
312         AlienY = 0
313         A_pass += 1
314
315     Passed(A_pass)
316
317     #화면 외계인 움직임
318     AngryAlienY += AA_speed
319     if AngryAlienY > padheight:
320         AAlien = pygame.image.load(random.choice(AngryAlien))
321         AA_size = AAlien.get_rect().size
322         AA_width = AA_size[0]
323         AA_height = AA_size[1]
324         AngryAlienX = random.randrange(0, padwidth - AA_width)
325         AngryAlienY = 0
326
327     #만약 그냥 외계인 10마리를 놓치면 게임 끝
328     if A_pass == 10:
329         Gameover()
330
331     HenchmanY += H_speed
332
333     if HenchmanY > padheight:
334         Henchman = pygame.image.load(random.choice(Henchmanimage))
335         H_size = Henchman.get_rect().size
336         H_width = H_size[0]
337         H_height = H_size[1]
338         HenchmanX = random.randrange(0, padwidth - H_width)
339         HenchmanY = 0
340
341     #타이머 설정
342     B_time = (pygame.time.get_ticks())/1000
343
344     #타이머 설정
345     B_time = (pygame.time.get_ticks())/1000
346

```

외계인의 움직임을 표현하기 위한 것이다.

만약 외계인이 화면 밖으로 나가면 다른 외계인을 랜덤 소환한다. 그리고 10마리를 화면 밖으로 놓치면 게임이 끝난다.

```

364     #타이머 설정
365     B_time = (pygame.time.get_ticks())/1000
366

```

타이머를 설정한다. get_ticks()는 초단위라서 /1000을 해준다.

```

367     #보스 등장
368     if hitcount >= 10:
369         Object(Boss, BossX, BossY)
370         Health(Boss_Health) #보스 체력 보여주기
371         timer(total_time-int(B_time)) #타이머 시작
372         AlienY = -100 #화면 위로 올려서 없애기
373         AngryAlienY = -100 #화면 위로 올려서 없애기
374
375         BossX += B_speed
376         if BossX >= padwidth - B_width:
377             B_speed = -B_speed
378
379         elif BossX <= 0:
380             B_speed = -B_speed
381             BossX = 0
382
383         ABossX += AB_speed
384         if ABossX >= padwidth - AB_width:
385             AB_speed = -AB_speed
386
387         elif ABossX <= 0:
388             AB_speed = -AB_speed
389             ABossX = 0
390
391         #시간이 0초 이하가 되면 게임 끝
392         if total_time - B_time <= 0:
393             Gameover()

```

외계인 10마리를 처치하면 보스가 등장하도록 한다. 보스를 함수로 불러오고 체력과 타이머를 불러온다. 그리고 외계인들을 화면 밖으로 보내서 보이지 않게 한다.

375~389 줄 - 보스가 양 옆으로 왔다 갔다 할 수 있도록 만들어준다.

392~393 줄 - 시간이 0초 이하가 되면 게임 종료

```

416     #보스 체력이 일정 이하로 떨어지면 발생하는 이벤트.
417     if Boss_Health <= 50:
418         BossY = -300
419         Object(Henchman, HenchmanX, HenchmanY)
420         Object(AngryBoss, ABossX, ABossY)

```

보스의 체력이 50이하가 되면 기존 보스를 화면 밖으로 보내고 화난 보스와 보스 부하를 불러온다.

```

467         if Boss_Health <= 0:
468             Object(B_die, BossX, BossY)
469             ExplotionSound2.play()
470             Win()

```

보스의 체력이 0 이하가 되면 폭발 사운드와 죽는 이미지와 함께 Win 함수를 불러온다.

2) 프로젝트 수행 결과

- 필요한 코드들을 적절히 활용하는 능력을 키우게 됨
- 오류 발생의 원인과 대안을 분석하는 해결 능력을 키움
- 팀 프로젝트를 통해 창의적 기획과 협동심을 키움

3) 프로젝트 수행 시 문제점

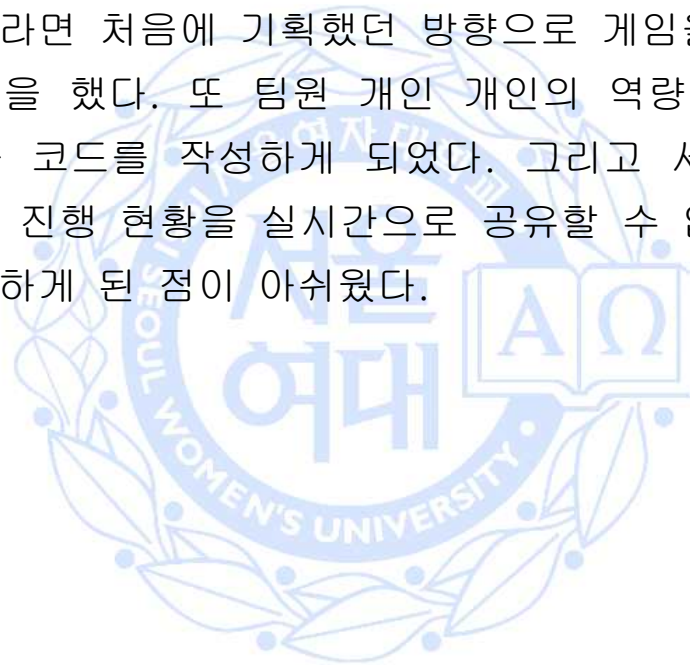
- 문제점 1: 화난 보스 이벤트가 실행되면 앞서 나왔던 외계인들을 다시 불러오려고 했다. 하지만 코드 적인 문제가 있어선지 앞서 불러왔던 방식으로 시도했지만 오류가 걸렸다.
- 문제점 2: 원래는 클래스화를 해서 코드의 길이를 줄여보려 했으나 시간 관계상 하지 못했다.
- 문제점 3: 점수와 처치 시간을 점수화해서 랭킹 시스템을 도입하려고 했으나 어떻게 시간을 점수로 바꿀까라는 고민과 랭킹 시스템의 코드 도입의 기획을 마무리하지 못해 시간 관계상 하지 못했다.
- 문제점 4: 이미지들이 PNG 형태였음에도 불구하고 투명한 부분도 미사일에 인식되었다. 투명한 부분에도 미사일이 맞아도 점수가 올라가거나 비행기가 부딪혀 실패하는 경우가 발생했다. 대체할 수 있는 방법을 찾았으나 대체하게 될 시 모든 코드를 새로 짜야했기 때문에 바꾸지 못했다.

4) 프로젝트의 향후 개선점

- 프로젝트를 진행하면서 생겼던 문제점들과 프로그램을 더 업그레이드를 시킬 수 있는 방법들을 찾아보면서 게임을 마무리할 것이다.

5) 프로젝트 수행 소감

- 파이게임이라는 개념이 처음 접하는 내용이다보니 알아가는데 시간이 필요했다. 그러다보니 고퀄리티의 게임을 만드는데 한계가 있었고, 코드를 짜면서 기획했던 부분에 차질이 생겼다. 진행하면서 생긴 문제점들은 다른 코드를 이용하거나 기획을 변경하여 대처했지만, 시간의 여유가 있었다면 처음에 기획했던 방향으로 게임을 만들 수 있었을 것이라는 생각을 했다. 또 팀원 개인 개인의 역량이 달라 코딩 실력이 좋은 친구가 코드를 작성하게 되었다. 그리고 서로 시간을 맞추기가 힘들어 개발 진행 현황을 실시간으로 공유할 수 없어 팀원들이 나중에 코드를 이해하게 된 점이 아쉬웠다.



감사합니다

서울여자대학교 소프트웨어와 창의적 사고 6팀



서울여자대학교
SEOUL WOMEN'S UNIVERSITY

2023111398 이시연

2023111389 박송희

2023111344 이승은

2023111365 홍서연