

Pygame으로 게임만들기 5

9. 사운드 넣기

```
319 def main():
320     global FPSLOCK, DISPLAYSURF
321     global IMAGESDICT, SOUNDSDICT
322
323     # 비행기 왼쪽 초기 위치
324     airplane_y_change = 0
325     airplane_x_change = 0
326
327     # 윈도우 변경에 따른 배경크기 변경
328     BACKGROUNDWIDTH = IMAGESDICT["background"].get_width()
329
330     # 배경 초기 위치
331     background_x = 0
332     other_background_x = BACKGROUNDWIDTH
333
334     # 개별 sprite group
335     bullet_group = pygame.sprite.Group()
336     bat_group = pygame.sprite.Group()
337     boom_group = pygame.sprite.Group()
338     fireball_group = pygame.sprite.Group()
339     # 전체 sprite group
340     sprite_group = pygame.sprite.Group()
341
342     # 비행기 sprite
343     airplane = AirPlane()
344     sprite_group.add(airplane)
345
346     # 박쥐 및 fireball 최대 개수 초기화
347     bat_maximum_num = 1
348     fireball_max_num = 1
349
350     # 게임 배경음악 초기화
351     pygame.mixer.music.load(SOUNDSDICT["background"])
352     pygame.mixer.music.play(-1, 0.0)
```

9. 사운드 넣기

```
354 # game loop
355 while True:
356     # event handle
357     for event in pygame.event.get():
358         # 종료
359         if event.type == QUIT or (event.type == KEYUP and event.key == K_ESCAPE):
360             pygame.quit()
361             sys.exit()
362         if event.type == KEYDOWN:
363             if event.key == K_UP:
364                 airplane_y_change = -5
365             elif event.key == K_DOWN:
366                 airplane_y_change = 5
367             if event.key == K_RIGHT:
368                 airplane_x_change = 5
369             elif event.key == K_LEFT:
370                 airplane_x_change = -5
371             if event.key == K_LCTRL:
372                 # 총알을 추가한다.
373                 bullet_group.add(AirplaneBullet(airplane.position()))
374                 sprite_group.add(bullet_group)
375                 SOUNDSDICT["shot"].play()
376         if event.type == KEYUP:
377             if event.key == K_UP or event.key == K_DOWN:
378                 airplane_y_change = 0
379             elif event.key == K_RIGHT or event.key == K_LEFT:
380                 airplane_x_change = 0
381         # 누르고 있으면 비행기가 움직이는 부분이다.
382         # for event 내에서는 키가 떨어지거나 붙어지지만 작동하기 때문에 밖에서 처리한다.
383         airplane.change_x(airplane_x_change)
384         airplane.change_y(airplane_y_change)
385
```

9. 사운드 넣기

```
412 # 충돌을 검사한다.  
413 # 박쥐와 총알의 충돌 검사  
414 bat_collision_dict = pygame.sprite.groupcollide(bullet_group, bat_group, False, False)  
415 if bat_collision_dict:  
416     for bullet in bat_collision_dict.keys():  
417         bat_x, bat_y = bat_collision_dict[bullet][0].position()  
418         boom_group.add(Boom(bat_x, bat_y))  
419         SOUNDS_DICT["explosion"].play()  
420     sprite_group.add(boom_group)  
421     airplane.bat_catch_add()  
422     pygame.sprite.groupcollide(bullet_group, bat_group, True, True)  
423     # 박쥐, fireball의 숫자를 규칙에 따라 늘린다.  
424     if airplane.bat_catch_return() % 2 == 0:  
425         bat_maximum_num += 1  
426         BatEnemy.bat_remove_time.insert(0, time.time())  
427     if airplane.bat_catch_return() % 4 == 0:  
428         fireball_max_num += 1
```

9. 사운드 넣기

```
430 # 비행기와 박쥐, 파이어볼의 충돌 검사
431 airplane_crash_bat = pygame.sprite.spritecollide(airplane, bat_group, True)
432 airplane_crash_fire = pygame.sprite.spritecollide(airplane, fireball_group, True)
433 if airplane_crash_bat or airplane_crash_fire:
434     # group이 그냥 초기화되면, 소멸자가 작동하지 않는 것으로 보아, 객체가 남는 것으로 보인다.
435     # group을 명시적으로 비워준다.
436     if airplane_crash_bat:
437         # 박쥐와 비행기 충돌시, 리스트로 batsprite를 넘겨줘서 bat객체가 삭제되지 않는 것으로 추정.
438         del airplane_crash_bat[:]
439     sprite_group.empty()
440     bullet_group.empty()
441     bat_group.empty()
442     boom_group.empty()
443     fireball_group.empty()
444     SOUNDS_DICT['crash'].play()
445     game_over(airplane.bat_catch_return(), "Crashed!")
```

9. 사운드 넣기

```
464 def game_init():
465     xx xx xx xx xx xx
469     global FPSLOCK, DISPLAYSURF
470     global IMAGESDICT, SOUNDSdict
471     FPSLOCK = pygame.time.Clock()
472     pygame.init()
473
474     # DISPLAY Surface 설정하기
475     DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
476     pygame.display.set_caption('PyFlying')
477
478     # 이미지 받아오기
479     IMAGESDICT = {"airplane": pygame.image.load('images/plane.png'),
480                  "background": pygame.image.load('images/background.png'),
481                  "bat": pygame.image.load('images/bat.png'),
482                  "fireball1": pygame.image.load('images/fireball.png'),
483                  "fireball2": pygame.image.load('images/fireball2.png'),
484                  "bullet": pygame.image.load('images/bullet.png'),
485                  "boom": pygame.image.load('images/boom.png'),
486                  "blank": pygame.image.load('images/blank.png')}
487
488     # 음향 받아오기
489     SOUNDSdict = {"shot": pygame.mixer.Sound("sounds/shot.wav"),
490                  "explosion": pygame.mixer.Sound("sounds/explosion.wav"),
491                  "crash": pygame.mixer.Sound("sounds/airplane_crash.wav"),
492                  "background": "sounds/Birds_in_Flight.mp3"}
```

9. 사운드 넣기

- `pygame.mixer.music.load(음악 저장 위치)`를 통해 background 음악을 받아온다.
- `pygame.mixer.music.play(반복횟수, 시작위치)`
 - 반복횟수가 -1이면 무한히 반복한다.

```
# 게임 배경음악 초기화
pygame.mixer.music.load(SOUNDS_DICT["background"])
pygame.mixer.music.play(-1, 0.0)
```

9. 사운드 넣기

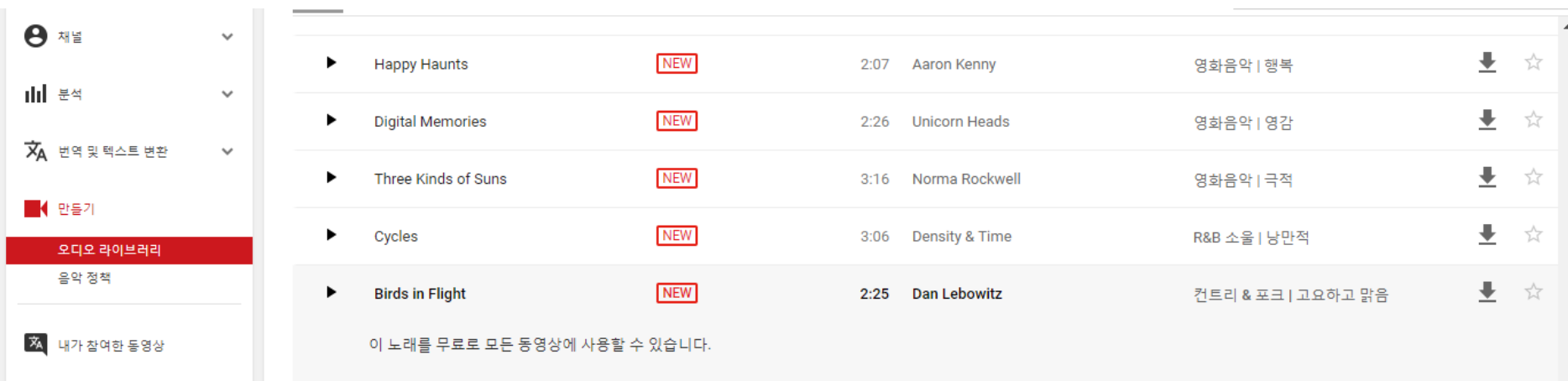
- `pygame.mixer.Sound("파일 위치")`를 통해 sound 객체를 만든다.
- `Sound.play()`를 통해서 재생한다.
- `Sound.stop()`을 통해서 멈출 수 있다.

```
SOUNDS_DICT["shot"].play()
```

```
SOUNDS_DICT = {"shot": pygame.mixer.Sound("sounds/shot.wav"),  
                "explosion": pygame.mixer.Sound("sounds/explosion.wav"),  
                "crash": pygame.mixer.Sound("sounds/airplane_crash.wav"),  
                "background": "sounds/Birds_in_Flight.mp3"}
```


9. 사운드 넣기

- 배경 음악은 무료 음악에서 받아왔다.
- 더 많은 무료 배경음악 찾기는 슬라이드 노트에 주소를 놓아두었다.



The screenshot shows a music library interface. On the left is a sidebar with navigation options: 채널 (Channels), 분석 (Analytics), 번역 및 텍스트 변환 (Translation and Text Conversion), 만들기 (Create), 오디오 라이브러리 (Audio Library), 음악 정책 (Music Policy), and 내가 참여한 동영상 (Videos I've participated in). The main area displays a list of songs, each with a play button, title, a red 'NEW' tag, duration, artist, genre, and download/favorite icons.

▶	Happy Haunts	NEW	2:07	Aaron Kenny	영화음악 행복	↓ ☆
▶	Digital Memories	NEW	2:26	Unicorn Heads	영화음악 영감	↓ ☆
▶	Three Kinds of Suns	NEW	3:16	Norma Rockwell	영화음악 극적	↓ ☆
▶	Cycles	NEW	3:06	Density & Time	R&B 소울 낭만적	↓ ☆
▶	Birds in Flight	NEW	2:25	Dan Lebowitz	컨트리 & 포크 고요하고 맑음	↓ ☆

이 노래를 무료로 모든 동영상에 사용할 수 있습니다.

9. 사운드 넣기

- 상단 아이콘 바꾸기

```
464 def game_init():
465     global FPSLOCK, DISPLAYSURF
466     global IMAGESDICT, SOUNDSdict
467     FPSLOCK = pygame.time.Clock()
468     pygame.init()
469
470     # DISPLAY Surface 설정하기
471     DISPLAYSURF = pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))
472     pygame.display.set_caption('PyFlying')
473     pygame.display.set_icon(pygame.image.load('images/icon.jpg'))
```

- icon은 32*32크기로 만들어서 넣어야 된다. .png를 사용해야 배경이 투명해진다.

10. exe파일 만들기

- 명령프롬프트(cmd)창에서 pip install cx_freeze로 설치한다.

```
C:\Users\WKJH>pip install cx_freeze
```

10. exe파일 만들기

- setup.py파일 만들기.
- exe파일을 만들 파이썬 폴더 안에 있어야 한다.
- 원래는 이렇게 만들어야 된다.

```
1 import sys
2 import os
3 from cx_Freeze import setup, Executable
4
5 os.environ['TCL_LIBRARY'] = r'C:\Users\KJH\AppData\Local\Programs\Python\Python36\tcl\tcl8.6'
6 os.environ['TK_LIBRARY'] = r'C:\Users\KJH\AppData\Local\Programs\Python\Python36\tcl\tk8.6'
7
8 build_exe_options = {
9     "packages": ["pygame", "sys", "random", "time"],
10    "excludes": ["numpy"],
11    "include_files": ["images/"]
12 }
13
14 base = None
15 if sys.platform == "win32":
16     base = "Win32GUI"
17
18 setup(
19     name="Test",
20     version="1.0",
21     description="test",
22     author="Kim Jaehyeong",
23     options=[("build_exe", build_exe_options)]
```

10. exe파일 만들기

- 쓸모 없는 것들이 많이 추가되어 packages를 삭제하고 사용한다.

```
1  import sys
2  from cx_Freeze import setup, Executable
3
4  build_exe_options = {
5      "excludes": ["numpy"],
6      "include_files": ["images/"]
7  }
8
9  base = None
10 if sys.platform == "win32":
11     base = "Win32GUI"
12
13 setup(
14     name="Test",
15     version="1.0",
16     description="test",
17     author="Kim Jaehyeong",
18     options={"build_exe": build_exe_options},
19     executables=[Executable("8_rule_add.py", base=base)]
20 )
```

10. exe파일 만들기

- build_exe_options는 만들 때 필요한 option을 정해준다. - 꽤나 많은 옵션이 있다.
- 그 아래에 있는 setup에서 사용된다.
- build_exe는 만들어지는 폴더 이름
- includes는 포함될 패키지 이름(리스트)
- exclude는 미포함될 패키지 이름(리스트)
- include_files는 포함될 파일 이름(리스트)이고, 폴더는 폴더이름/으로 포함시킬 수 있다.

```
build_exe_options = {  
    "build_exe": "test2",  
    "includes": [],  
    "excludes": ["numpy"],  
    "include_files": ["images/"]  
}
```

10. exe파일 만들기

- 폴더 이름으로 포함시키면 폴더가 그대로 복사되고, 폴더/이름.확장자로 하면, 만들어진 폴더에 파일이 그대로 복사되기 때문에 주의한다.

```
build_exe_options = {  
    "build_exe": "test2",  
    "includes": [],  
    "excludes": ["numpy"],  
    "include_files": ["images/"]  
}
```

10. exe파일 만들기

- options는 딕셔너리이어서 dict()을 통해서 만들 수 있다. 이렇게 하는게 만들기가 좀 더 편하다.

```
4     build_exe_options = {  
5         "excludes": ["numpy"],  
6         "include_files": ["images/"]  
7     }  
8     build_exe_options = dict(  
9         excludes=["numpy"],  
10        include_files=["images/"]  
11    )
```


10. exe파일 만들기

```
14 base = None
15 if sys.platform == "win32":
16     base = "Win32GUI"
```

- base는 파일을 어떤 형태로 만들지 정한다.
- None이면 명령프롬프트 형식
- pygame은 GUI임으로 Win32GUI로 정해준다.
- sys.platform은 지금 사용하는 플랫폼이 윈도우인지 확인하는 부분이다.
- 리눅스나 OSX같은 곳에서 사용하는 실행파일은 그 OS에서 실행하여 만들어야 한다.

10. exe파일 만들기

- setup부분은 파일내에 입력되는 부분(이라 추정)이다.
- name은 파일 이름
- version은 파일의 버전
- description은 파일의 설명

```
18  setup(  
19      name="Test",  
20      version="1.0",  
21      description="test",  
22      author="Kim Jaehyeong",  
23      options={"build_exe": build_exe_options},  
24      executables=[Executable("8_rule_add.py", base=base)]  
25  )
```

10. exe파일 만들기

- author은 만든 이
- options는 build_exe에 위에서 만든 옵션을 넣는다.
- executables는 어떤 파일을 실행해 py파일을 만들지 찾는다. base는 위에서 사용한 base이며, 생략하면 None이 기본값이다.

```
18 setup(  
19     name="Test",  
20     version="1.0",  
21     description="test",  
22     author="Kim Jaehyeong",  
23     options={"build_exe": build_exe_options},  
24     executables=[Executable("8_rule_add.py", base=base)]  
25 )
```

10. exe파일 만들기

- 입력해보자.

```
1  import sys
2  from cx_Freeze import setup, Executable
3
4  build_exe_options = {
5      "excludes": ["numpy"],
6      "include_files": ["images/",
7                       "sounds/"]
8  }
9
10 build_exe_options = dict(
11     excludes=["numpy"],
12     include_files=["images/",
13                  "sounds/"]
14 )
15
16 base = None
17 if sys.platform == "win32":
18     base = "Win32GUI"
19
20 setup(
21     name="airplane_game",
22     version="1.0",
23     description="airplane_game distribute",
24     author="Kim Jaehyeong",
25     options={"build_exe": build_exe_options},
26     executables=[Executable("9_add_sound.py", base=base)]
27 )
```

10. exe파일 만들기

- 명령프롬프트에서 py파일이 있는 곳으로 이동한다.
- 그 후 `python setup.py build` 라고 입력하면 자동적으로 그 폴더 내 build(옵션을 다르게 주면 준 위치)로 실행파일이 만들어진다.
- 설치 형식일 경우 `python setup.py bdist_msi`로 만든다. 설치 형식으로 만들 경우, 파일이름, 버전, 만든이 등이 만들어진 파일이름과 파일정보-자세히에서 보인다.

10. exe파일 만들기

- 만들어진 파일 내 lib부분에는 패키지가 들어가는데, 몇몇 쓸 때 없는 것들도 있다. 더 용량을 줄이고 싶으면 지워보고 실행했을 때 오류가 안나는 것을 확인해 보면서 해도 된다. (보통 이름 보면 지워도 될 게 보이긴 한다.)

10. exe파일 만들기

- 단일 exe file로 만들 때는 pyinstaller를 사용할 수 있으나, 최신 python3와 pygame조합에서는 오류가 나는 것으로 보여 포기하였다.
- pip로 설치한 뒤, pyinstaller 파일.py -f -w를 주면 된다.(-f는 하나의 실행파일로 만들기, -w는 console창을 열지 않는다.)
- 자세한 건 슬라이드 노트에 적힌 사이트를 본다.

11. 기타

- 마우스로도 이벤트를 받아 움직일 수 있다.
- `pygame.time.set_timer()`를 통해서 일정 시간이 지나면 이벤트가 발생할 수 있게 만들 수 있다.
 - <http://vod3.kocw.net/KOCW/document/2016/pusan/heoseokyeol/16.pdf>
 - <https://code.i-harness.com/ko/q/1649527>
- 카메라 뷰-실제로 보이는 부분
 - 본 책 chapter 8을 참조할 것.

11. 기타

- fps조절하기

- <http://contents2.kocw.or.kr/KOCW/document/2016/pusan/heoseokyeol/15.pdf>
- clock.get_fps()를 통해서(이 세미나에서는 FPSCLOCK으로 clock객체를 받았다.) 실제 fps를 구하고 이를 통해서 조절할 수 있다. pdf참고
- https://github.com/Mekire/pygame-delta-time/blob/master/dt_example.py
- delta를 통해서 조절할 수 있도록 잘 되어 있다.

11. 기타

- 전체 이미지의 크기 조절
 - window사이즈에 따라서 전체 이미지 크기를 조절할 수 있다.
background바꾼 것 처럼 바꾸면 된다.

11. 기타

- 다음 ppt에서도 도움을 받았다.
 - <http://cs.sch.ac.kr/lecture/Python/2017/GUI-2017/17-PythonGUI.htm>
 - <http://www.kocw.net/home/search/kemView.do?kemId=1195460>

11. 기타

- git hub에 그동안 사용한 슬라이드와 노트가 포함된 pdf가 있다.
 - <https://github.com/imn00133/pythonSeminar17>
- 그 외에 했던 숙제도 있으니 궁금하면 확인해보면 된다.