

# 13장 PYGAME을 이용한 게임 작성





# pygame을 이용한 게임 작성 . p458

- Python으로 작성 가능한 게임 등의 멀티미디어 표현을 위한 라이브러리
- SDL 기반. 오픈 소스이자 무료 도구
- 게임 개발 도구이지만 이미지 프로세스 또는 조이스틱 입력, 음악 재생 등의 기능만 떼어다 쓸 수도 있다.
- pygame을 설치하려면 명령 프롬프트에서 “pip install pygame”을 입력한다.

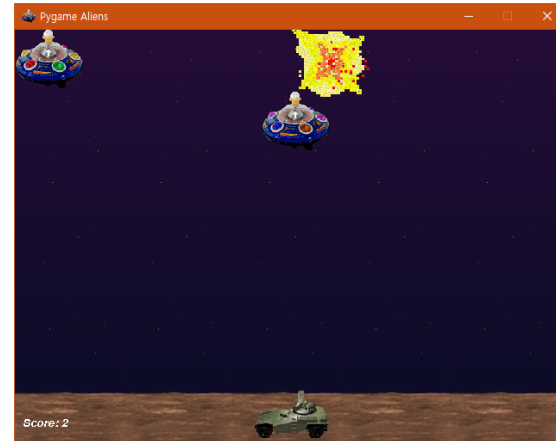
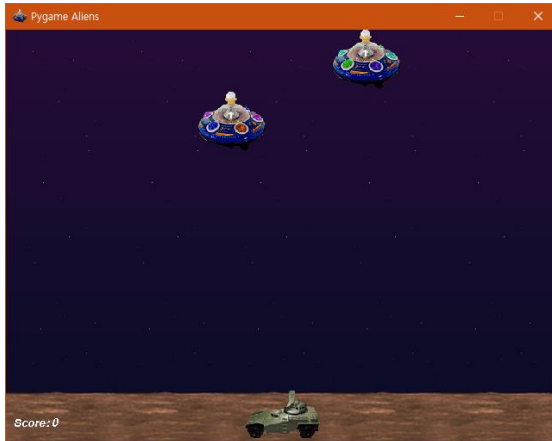
```
Anaconda Prompt
(base) C:\Users\kkim>pip install pygame
Collecting pygame
  Downloading https://files.pythonhosted.org/packages/ed/56/b63ab3724acff69f4080e54c4bc5f55d1fbdeeb19b92b70acf45e88a5908/ppygame-1.9.6-cp37-cp37m-win_amd64.whl (4.3MB)
    100% |#####| 4.3MB 5.1MB/s
Installing collected packages: pygame
Successfully installed pygame-1.9.6
```



# pygame을 이용한 게임 작성 . p458

- 설치가 올바르게 되었는지 테스트해보자

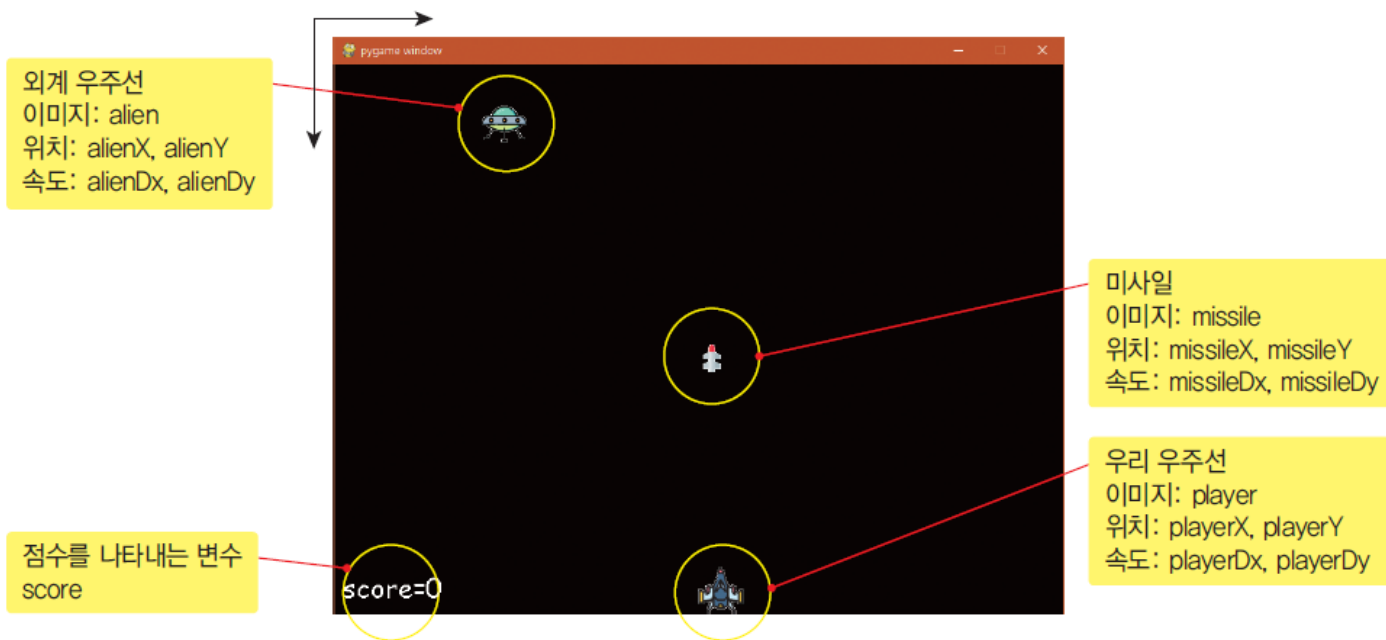
C> py -m pygame.examples.aliens





# 게임 설계. p459

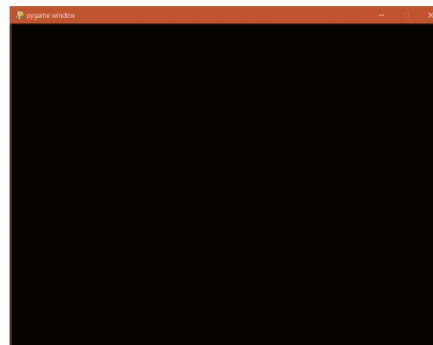
## □ 어떤 변수가 필요한가?



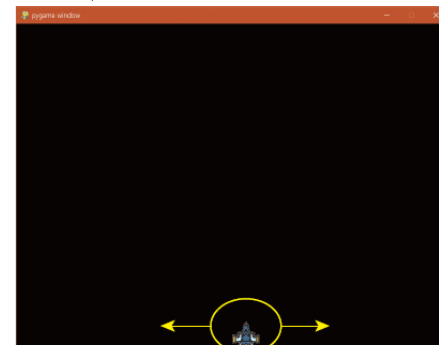


# 구현 단계. p460

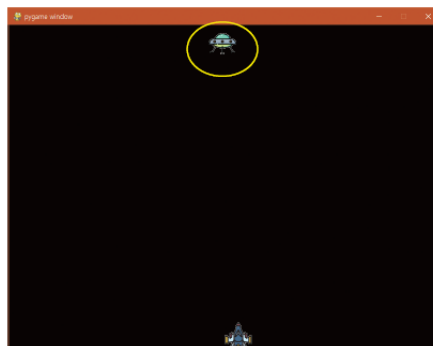
Step #1: pygame을 이용하여 빈 화면을 생성한다.



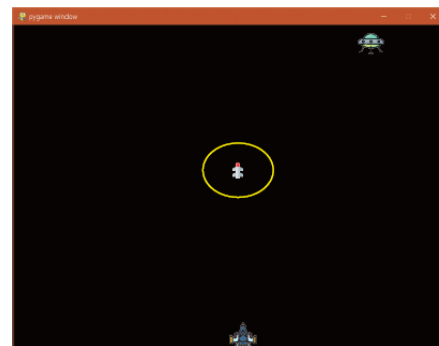
Step #2: 화면에 우리 우주선을 추가하고 방향키로 우주선을 움직인다.



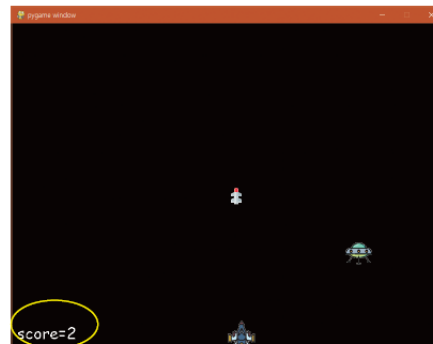
Step #3: 외계 우주선을 생성하여서 자동으로 움직이게 한다.



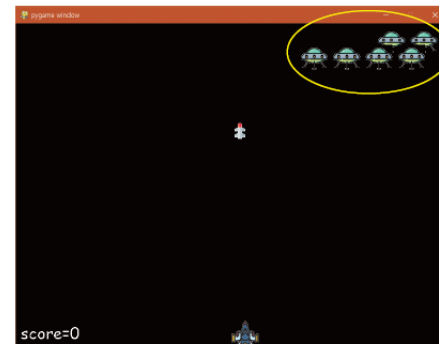
Step #4: 우리 우주선에서 미사일을 발사하도록 한다.



Step #5: 외계 우주선이 미사일에 맞으면 점수를 증가시킨다.



Step #6: 여러 개의 외계 우주선을 생성해본다.





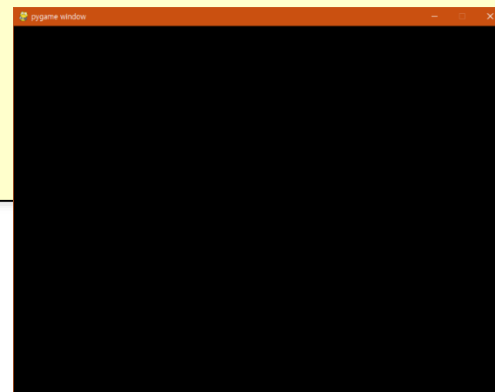
# STEP #1: 윈도우를 생성해보자. p461

```
import pygame

pygame.init()
display = pygame.display.set_mode((800, 600))    # 윈도우 생성

running = True
while running :                                # 게임루프
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

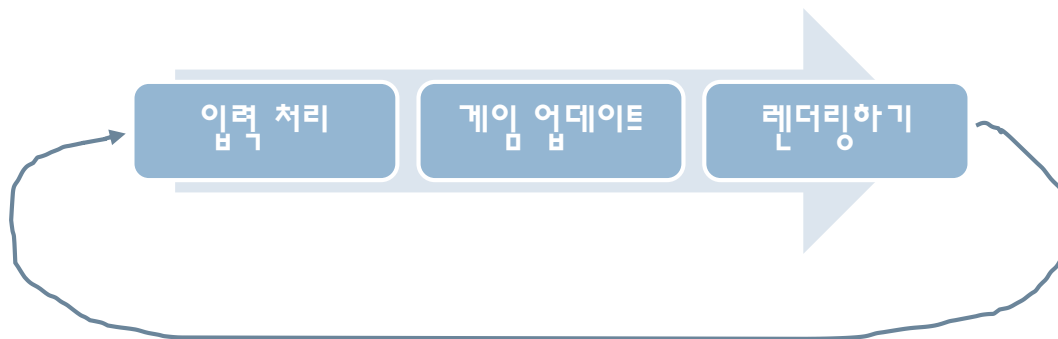
pygame.quit()
```





# 게임 루프. p462

- 게임 루프 : 모든 게임은 반복 루프를 사용하여 게임 플레이를 제어한다
  - 사용자의 입력을 처리한다.
  - 모든 게임 객체의 상태를 업데이트하고 이동시킨다.
  - 디스플레이 및 오디오 출력을 업데이트한다.
  - 게임의 속도를 조절한다.





## STEP #2: 화면에 이미지 표시하기. p462

```
import pygame

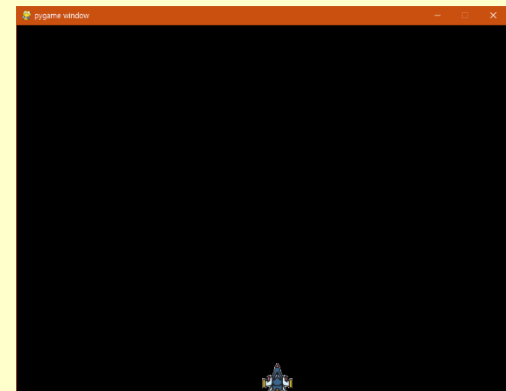
pygame.init()
display = pygame.display.set_mode((800, 600))
myfont = pygame.font.SysFont('Comic Sans MS', 30)
score = 0

player=pygame.image.load("spaceship.png")
playerX, playerY, playerDx , playerDy = 400, 550, 0, 0

running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    display.fill((0, 0, 0))
    display.blit(player, (playerX, playerY))
    pygame.display.update()

pygame.quit()
```







## STEP #2: 우주선 움직이기. p463

...

...

while running:

```
for event in pygame.event.get():  
    if event.type == pygame.QUIT:  
        running = False
```

```
    if event.type == pygame.KEYDOWN:
```

```
        if event.key == pygame.K_LEFT:
```

```
            playerDx = -0.1
```

```
        if event.key == pygame.K_RIGHT:
```

```
            playerDx = 0.1
```

```
    if event.type == pygame.KEYUP:
```

```
        if event.key == pygame.K_RIGHT or event.key == pygame.K_LEFT:
```

```
            playerDx = 0
```

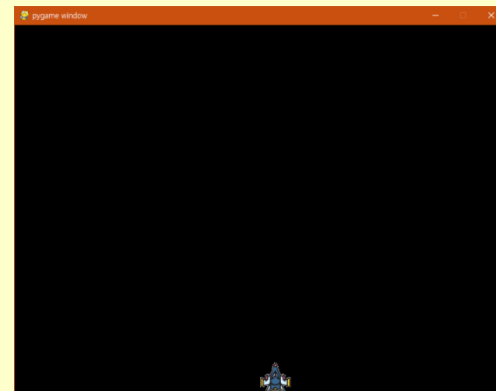
```
    playerX += playerDx
```

```
display.fill((0, 0, 0))
```

```
display.blit(player, (playerX, playerY))
```

```
pygame.display.update()
```

```
pygame.quit()
```





## STEP #3: 외계인 우주선 생성. p465

...

# (1) 외계 우주선 이미지를 읽는다.

```
alien=pygame.image.load("alien.png")
```

```
alienX, alienY, alienDx , alienDy  = 0, 10, 0.1, 0.1
```

```
running = True
```

```
while running:
```

```
    for event in pygame.event.get():
```

```
        if event.type == pygame.QUIT:
```

```
            running = False
```

```
        if event.type == pygame.KEYDOWN:
```

```
            if event.key == pygame.K_LEFT:                playerDx = -0.1
```

```
            if event.key == pygame.K_RIGHT:                playerDx = 0.1
```

```
        if event.type == pygame.KEYUP:
```

```
            if event.key == pygame.K_RIGHT or event.key == pygame.K_LEFT:
```

```
                playerDx = 0
```

```
    playerX += playerDx
```

```
    display.fill((0, 0, 0))
```

```
    display.blit(player, (playerX, playerY))
```

# (1) 외계 우주선 이미지를 화면에 표시한다.

```
    display.blit(alien, (alienX, alienY))
```

```
    pygame.display.update()
```

...

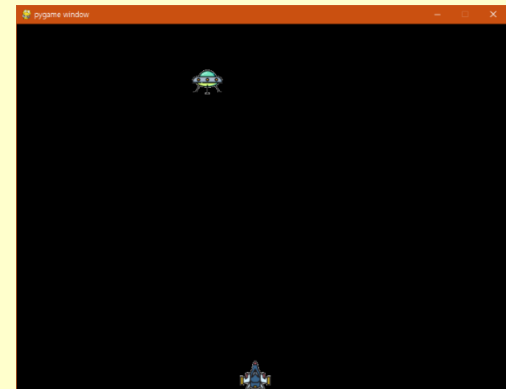


# STEP #3: 외계 우주선을 움직여보자. p466

```
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                playerDx = -0.1
            if event.key == pygame.K_RIGHT:
                playerDx = 0.1
        if event.type == pygame.KEYUP:
            if event.key == pygame.K_RIGHT or event.key == pygame.K_LEFT:
                playerDx = 0
    playerX += playerDx
    alienX += alienDx
    if alienX <= 0 or alienX > 750:
        alienDx *= -1
        alienY += 30
    display.fill((0, 0, 0))
    display.blit(player, (playerX, playerY))
    display.blit(alien, (alienX, alienY))
    pygame.display.update()

pygame.quit()
```





missileX, missileY = playerX, playerY



## STEP #4: 미사일 만들기

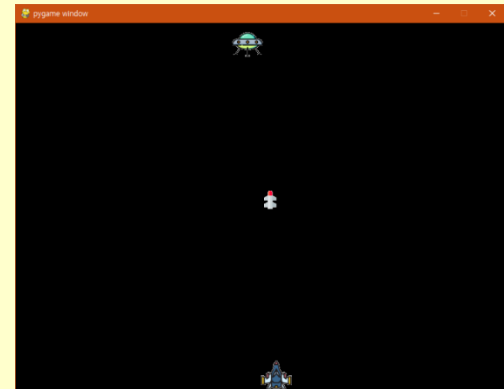
```
alienX += alienDx
if alienX <= 0 or alienX > 750:
    alienDx *= -1
    alienY += 30

if missileY <= 0:
    missileY = 1000
    missileState = "hidden"
if missileState == "fire":
    missileY -= missileDy

display.fill((0, 0, 0))
display.blit(player, (playerX, playerY))

display.blit(missile, (missileX, missileY))
display.blit(alien, (alienX, alienY))
pygame.display.update()

pygame.quit()
```





# STEP #5: 충돌 감지. P469

- `colliderect()` 함수 : 객체를 둘러싸는 사각형이 겹치는지 겹치지 않는지를 알고리즘으로 검사

```
import pygame

pygame.init()
display = pygame.display.set_mode((800, 600))

myfont = pygame.font.SysFont('Comic Sans MS', 30)
score = 0

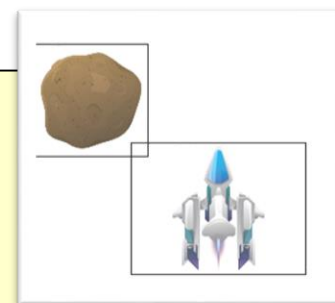
player=pygame.image.load("spaceship.png")
playerX, playerY, playerDx , playerDy = 400, 550, 0, 0

alien=pygame.image.load("alien.png")
alienX, alienY, alienDx , alienDy  = 0, 10, 0.1, 0.1

missile = pygame.image.load('missile.png')
missileX, missileY, missileDx , missileDy  = 0, 1000, 0, 0.1
missileState = "hidden"

...

...
```





# STEP #5: 충돌 감지! P469

```
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    ...

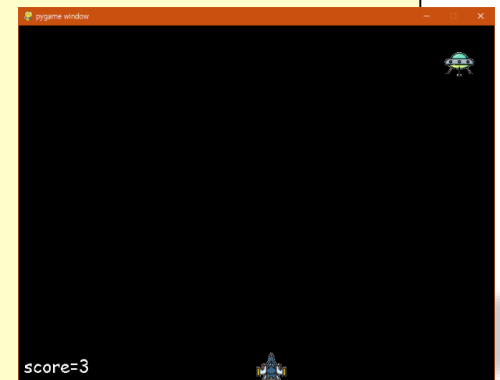
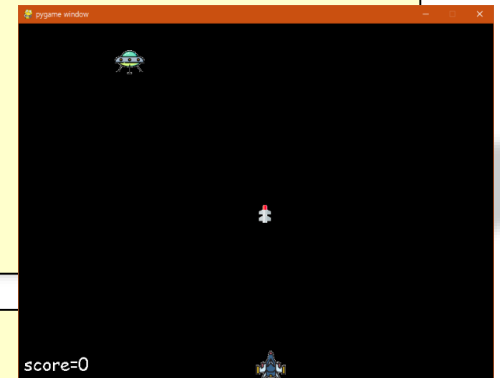
    rect1 = pygame.Rect(alien.get_rect(topleft=(alienX, alienY)))
    rect2 = pygame.Rect(missile.get_rect(topleft=(missileX, missileY)))
    if rect1.colliderect(rect2) and missileState != "hidden":
        score += 1
        alienX, alienY, alienDx, alienDy = 0, 10, 0.1, 0.1
```

```
display.fill((0, 0, 0))
display.blit(player, (playerX, playerY))
display.blit(missile, (missileX, missileY))
display.blit(alien, (alienX, alienY))

text = myfont.render(f'score={score}', False, (255, 255, 255))
display.blit(text, (10, 550))

pygame.display.update()

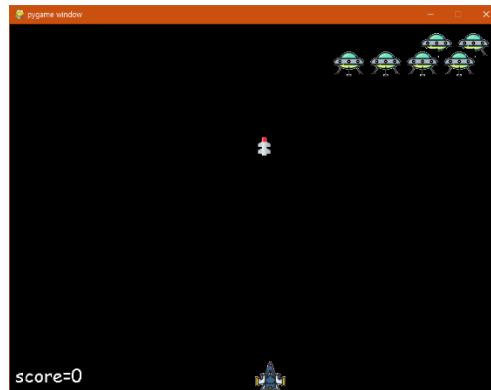
pygame.quit()
```





## STEP #6: 외계인 우주선 많이 생성하기. p472

- 우주선의 개수를 늘리려면 어떻게 하면 좋을까? 변수보다는 리스트를 사용해야 한다. 외계인 우주선의 위치와 속도를 저장하는 리스트를 4개 생성한다.
  - `alienX[ ]` : 외계인 우주선의 x좌표를 저장한다.
  - `alienY[ ]` : 외계인 우주선의 y좌표를 저장한다.
  - `alienDx[ ]` : 외계인 우주선의 x 속도를 저장한다.
  - `alienDy[ ]` : 외계인 우주선의 y 속도를 저장한다.







# STEP #6: 외계인 우주선 많이 생성하기. p472

```
import pygame

pygame.init()
display = pygame.display.set_mode((800, 600))
myfont = pygame.font.SysFont('Comic Sans MS', 30)
score = 0

player = pygame.image.load("spaceship.png")
playerX, playerY, playerDx, playerDy = 400, 550, 0, 0

alien=pygame.image.load("alien.png")
alienX = [ ]
alienY = [ ]
alienDx = [ ]
alienDy = [ ]
alienNumber = 6

for i in range(alienNumber):
    alienX.append(20+i*60)
    alienY.append(10)
    alienDx.append(0.1)
    alienDy.append(0.0)
...

```



## STEP #6: 외계인 우주선 많이 생성하기. p472

```
...  
...  
playerX += playerDx  
display.fill((0, 0, 0))  
  
for i in range(alienNumber):  
    alienX[i] += alienDx[i]  
    alienY[i] += alienDy[i]  
    if alienX[i] <= 0 or alienX[i] > 750:  
        alienDx[i] *= -1  
        alienY[i] += 30  
    rect1 = pygame.Rect(alien.get_rect(topleft=(alienX[i], alienY[i])))  
    rect2 = pygame.Rect(missile.get_rect(topleft=(missileX, missileY)))  
    if rect1.colliderect(rect2) and missileState != "hidden":  
        score += 1  
        alienX[i], alienY[i], alienDx[i], alienDy[i] = 0, 1000, 0.1, 0.0  
    display.blit(alien, (alienX[i], alienY[i]))  
...  
...
```