



이수안 컴퓨터 연구소

suan computer laboratory

파이썬으로 슈팅 게임 만들기



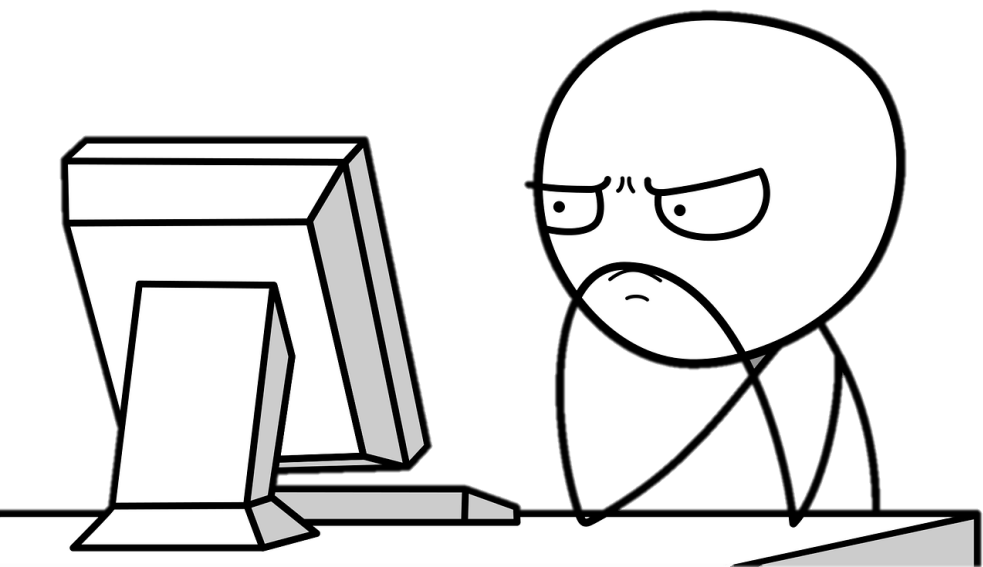
이수안



목차

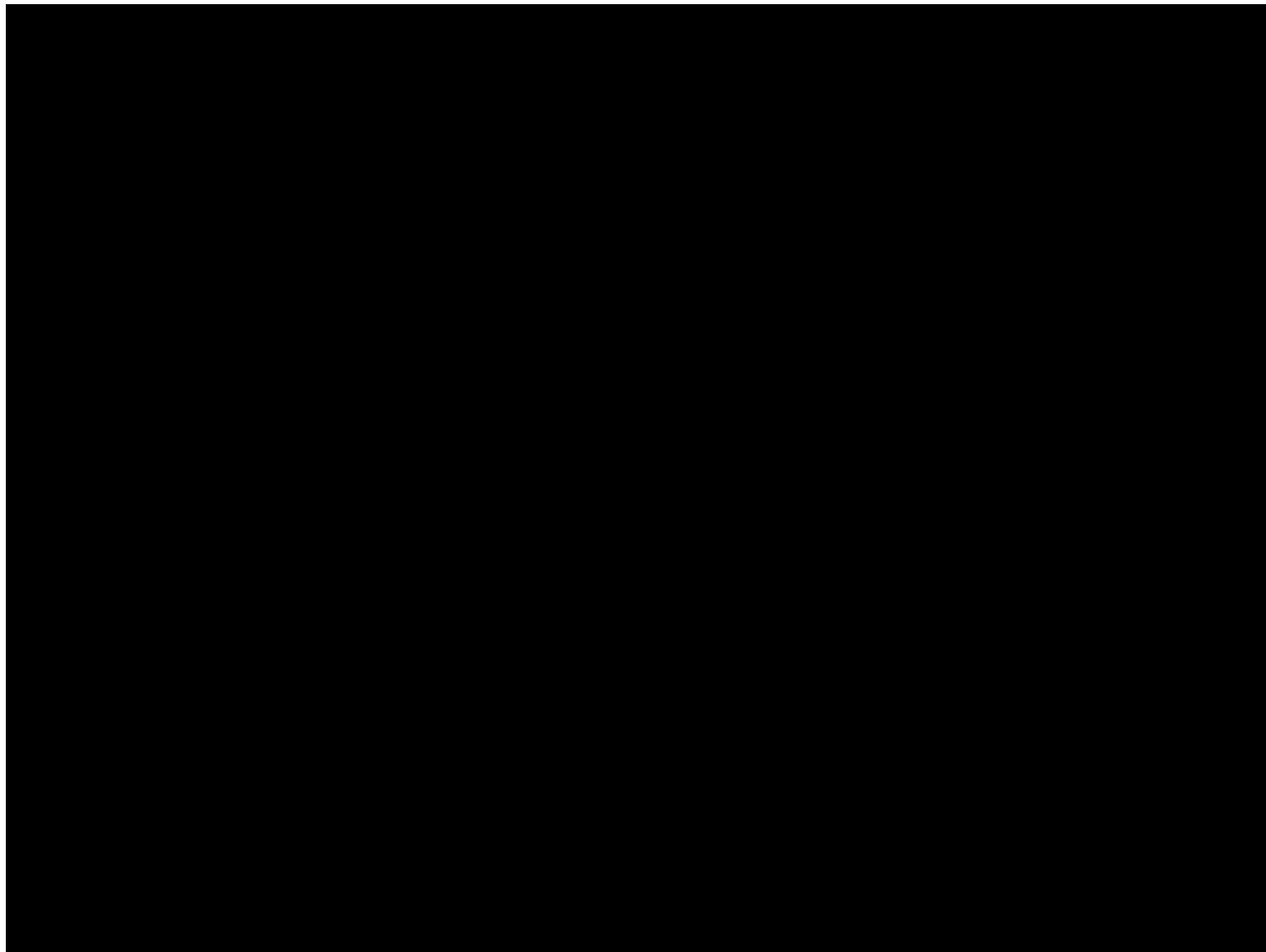
1. 슈팅 게임
2. Python 설치
3. pygame 설치
4. 슈팅 게임 만들기
 1. 게임 리소스 다운로드
 2. 게임 화면 구성
 3. 배경 그림 넣기
 4. 전투기 넣기
 5. 전투기 움직이기
 6. 미사일 발사하기
 7. 랜덤한 운석 떨어지기
 8. 미사일로 운석 파괴하기
 9. 파괴한 운석 수와 놓친 운석 수 표시하기
 10. 운석 맞추면 운석 속도 증가하기
 11. 전투기가 운석과 충돌하거나 운석을 놓치면 게임 오버
 12. 게임 사운드 넣기

1. 슈팅 게임

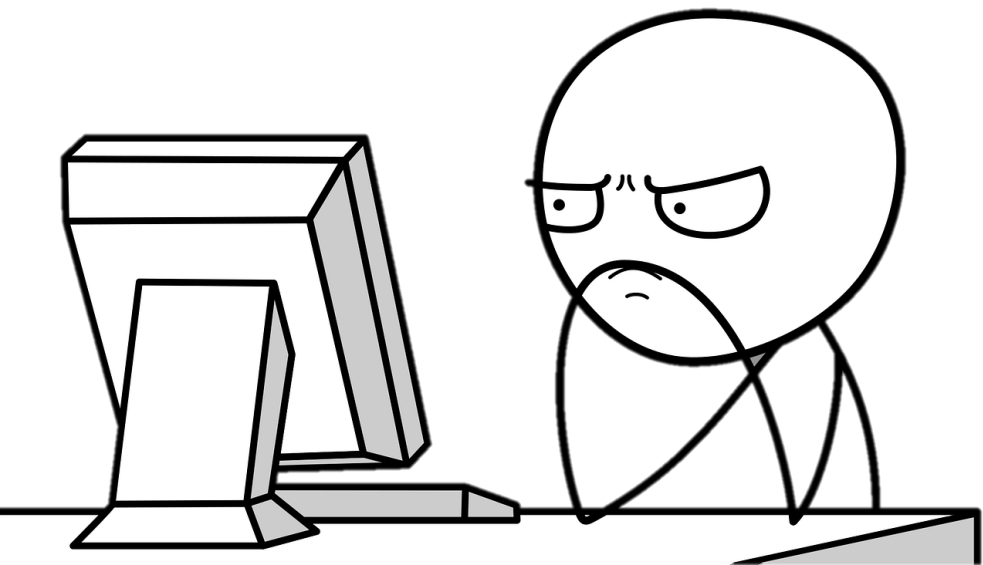


슈팅 게임

- ShooTing Game (약칭 : STG)
- 적의 공격을 피하며 무기를 쏘는 게임의 총칭
- 간단한 게임의 구조와 조작성 덕분에 게임 역사의 초창기부터 존재하였던 장르
- 프로그래밍적으로도 중력과 같은 복잡한 물리 엔진이 필요하지 않기 때문에 게임 제작 입문으로 자주 사용



2. Python 설치



Python 다운로드

The screenshot shows the Python.org homepage. At the top, there's a navigation bar with links: Python, PSF, Docs, PyPI, Jobs, and Community. Below this is the Python logo and a search bar. A secondary navigation bar contains links: About, Downloads, Documentation, Community, Success Stories, News, and Events. The 'Downloads' link is highlighted, and a dropdown menu is open, listing options: All releases, Source code, Windows, Mac OS X, Other Platforms, License, and Alternative Implementations. The 'Windows' option is selected, leading to a 'Download for Windows' section. This section features a button for 'Python 3.7.1' and a note stating that Python 3.5+ cannot be used on Windows XP or earlier. It also mentions that Python can be used on many operating systems and environments, with a link to 'View the full list of downloads.' At the bottom of the page, a large blue banner contains the text: 'Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)'.

Python

PSF

Docs

PyPI

Jobs

Community

python™

Search

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

All releases

Source code

Windows

Mac OS X

Other Platforms

License

Alternative Implementations

Download for Windows

Python 3.7.1

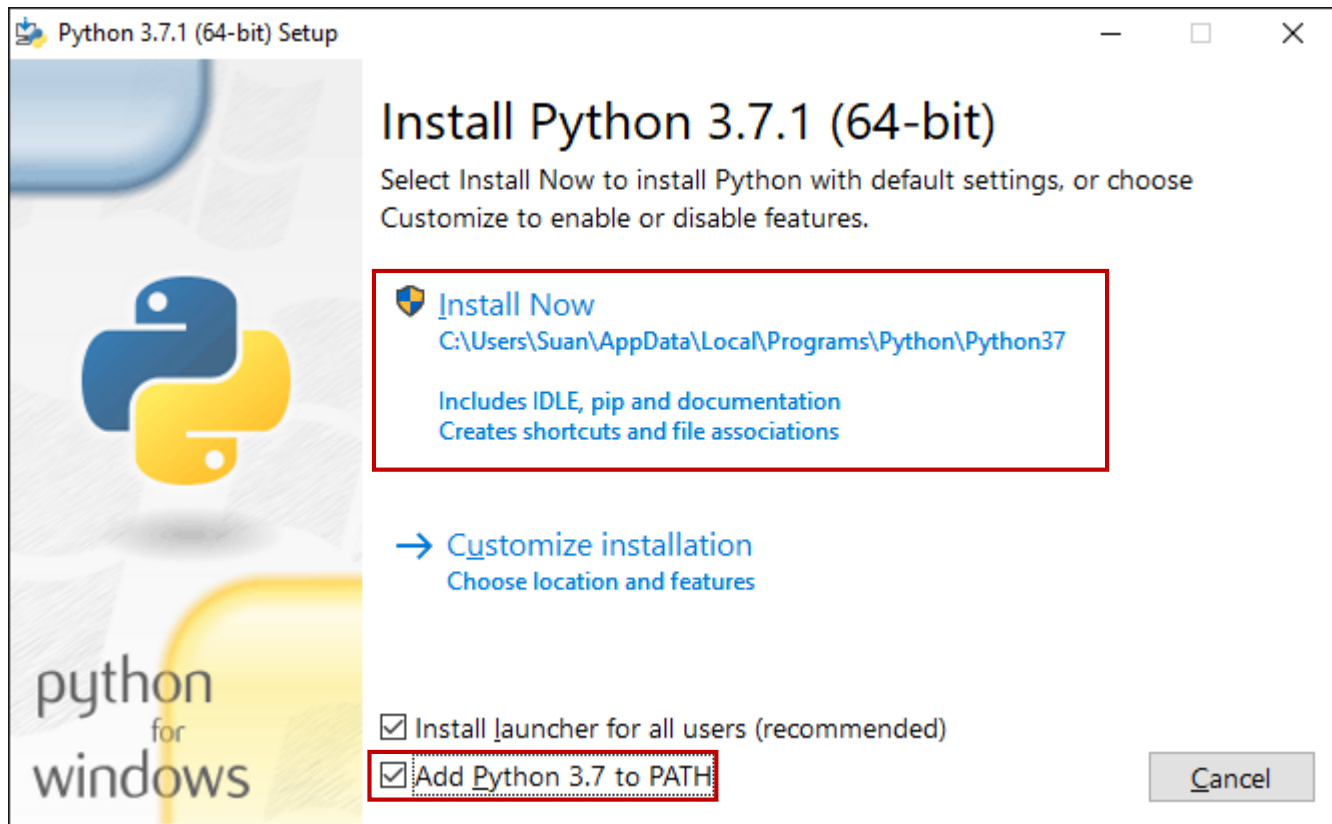
Note that Python 3.5+ cannot be used on Windows XP or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.

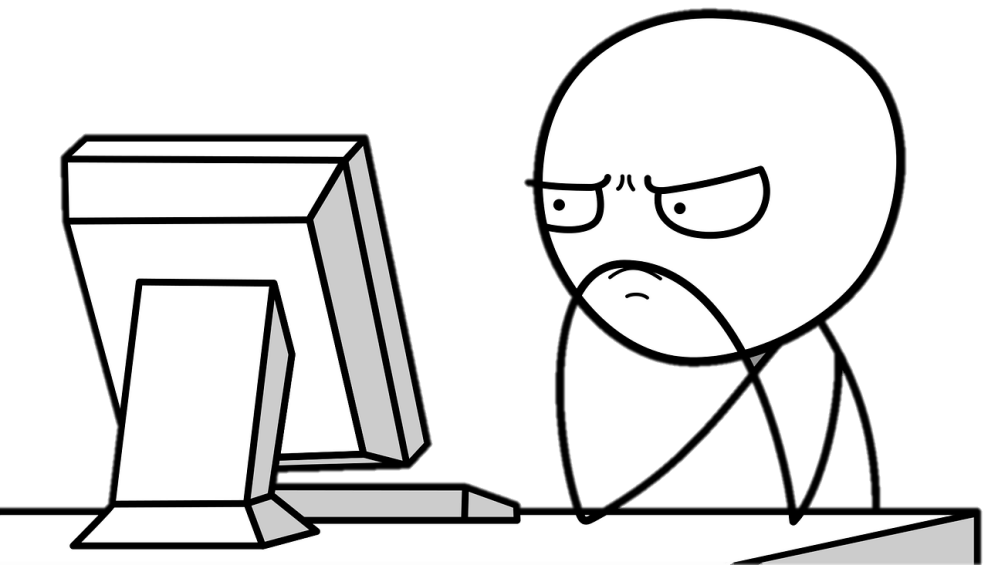
[View the full list of downloads.](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

Python 설치



3. pygame 설치



pygame


- pygame은 SDL라이브러리 위에 구축되어 게임과 같은 멀티미디어 어플리케이션을 만들기 위한 오픈 소스 파이썬 프로그래밍 라이브러리



- Silliness built in.
- Does not require OpenGL.
- Multi core CPUs can be used easily.
- Uses optimized C, and Assembly code for core functions.
- Comes with many Operating systems.
- Truly portable.
- It's Simple and easy to use.
- Does not require a GUI to use all functions.
- Small amount of code.
- It's not the best game library.

파이썬에 pygame 라이브러리 추가

- Command Prompt 열기
 - [시작] - [실행] - cmd.exe
- pygame 라이브러리 추가 명령어
 - pip install pygame



```

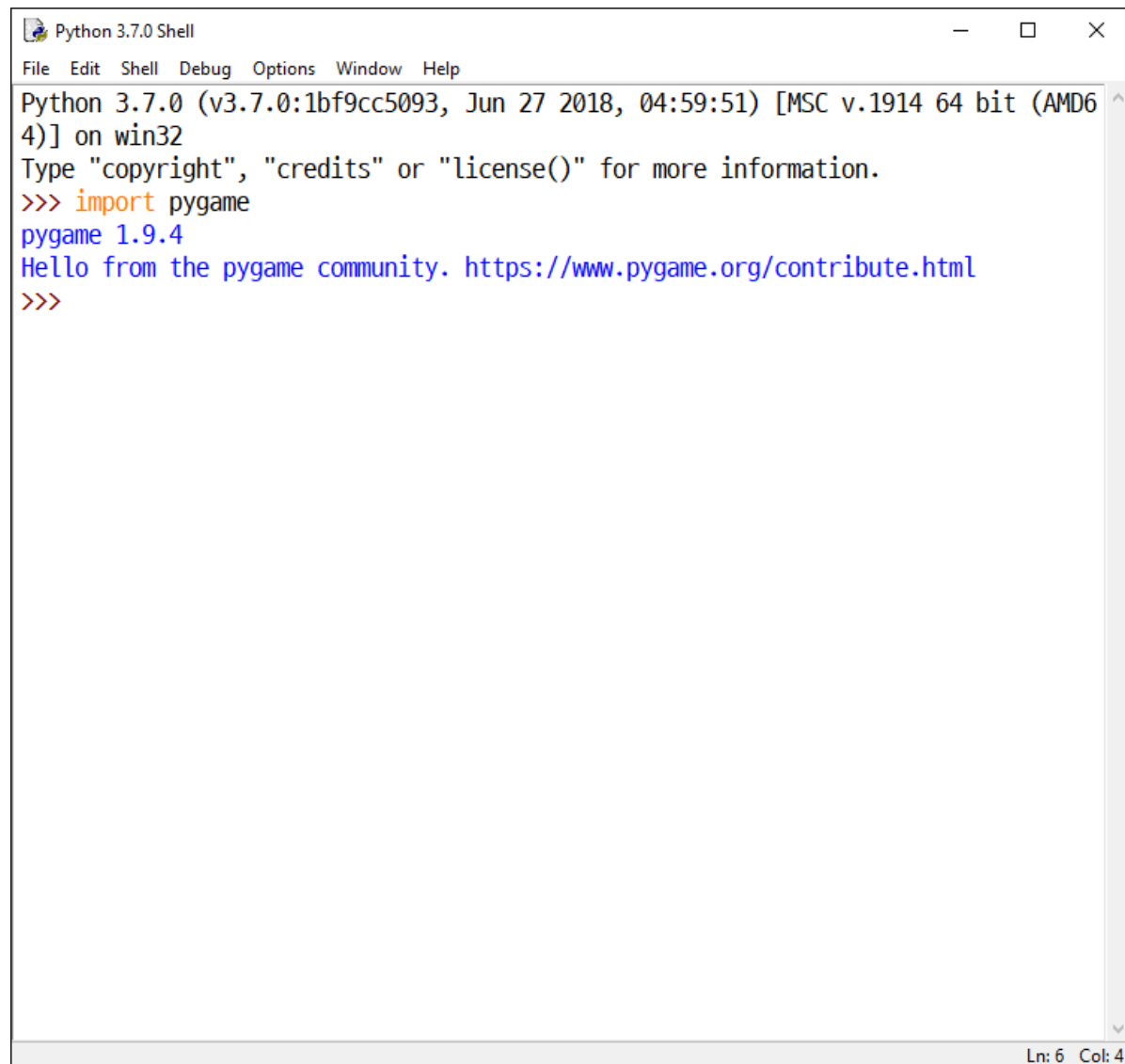
Select C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Suan>pip install pygame
Collecting pygame
  Downloading https://files.pythonhosted.org/packages/3e/f5/feabd88a2856ec86166a897b62bfad828bfe7a94a27cbd7ebf07fd670399/pygame-1.9.4-cp37-cp37m-win_amd64.whl (4.2MB)
    100% |████████████████████████████████████████| 4.2MB 2.5MB/s
Installing collected packages: pygame
Successfully installed pygame-1.9.4

C:\Users\Suan>
```

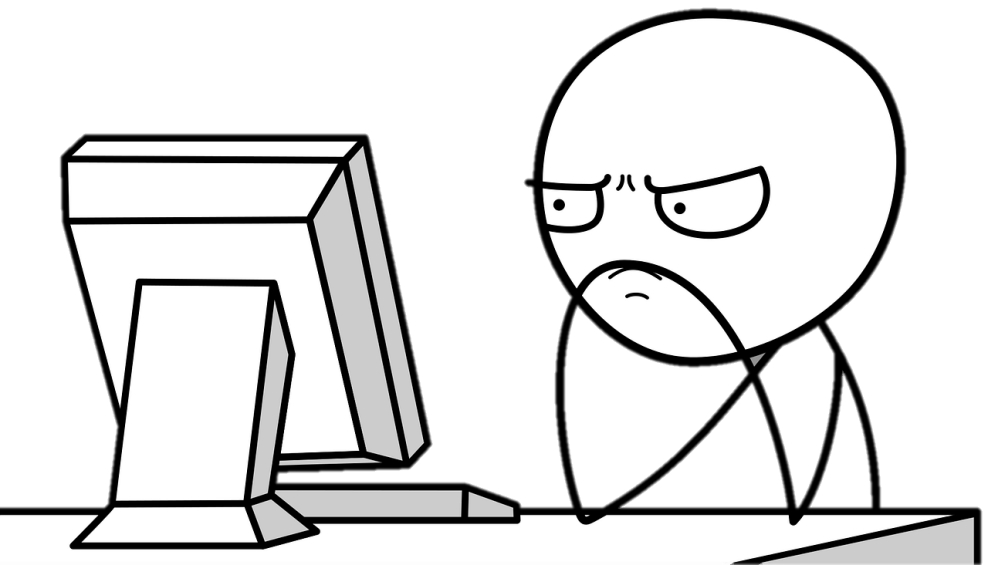
파이썬에서 pygame 설치 확인

- Python Shell에서 명령어를 통해 pygame 설치 확인
 - `import pygame`



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import pygame
pygame 1.9.4
Hello from the pygame community. https://www.pygame.org/contribute.html
>>>
```

4. 슈팅 게임 만들기



PyShooting: 지구를 지켜라



- 파이썬을 이용한 슈팅 게임
- 떨어지는 운석을 파괴하여 지구를 지켜라!

게임 리소스



explosion.png



missile.png



fighter.png



rock01.png



rock02.png



rock03.png



rock04.png



rock05.png



rock06.png



rock07.png



rock08.png



rock09.png



rock10.png



rock11.png



rock12.png



rock13.png



rock14.png



rock15.png



rock16.png



rock17.png



rock18.png



explosion01.wav



explosion02.wav



explosion03.wav



explosion04.wav



gameover.wav



missile.wav



music.wav



NanumGothic.ttf



rock19.png



rock20.png



rock21.png



rock22.png



rock23.png



rock24.png



rock25.png



rock26.png



rock27.png



rock28.png



rock29.png



rock30.png

사운드 출처: freesound.org, youtube audio library

이미지 출처: openclipart.org

이미지 출처: openclipart.org

게임 화면 구성

```
import pygame
import sys
from time import sleep

BLACK = (0, 0, 0)
padWidth = 480    # 게임화면의 가로크기
padHeight = 640   # 게임화면의 세로크기

def initGame():
    global gamePad, clock
    pygame.init()
    gamePad = pygame.display.set_mode((padWidth, padHeight))
    pygame.display.set_caption('PyShooting')    # 게임 이름
    clock = pygame.time.Clock()

def runGame():
    global gamepad, clock

    onGame = False
    while not onGame:
        for event in pygame.event.get():
            if event.type in [pygame.QUIT]:    # 게임 프로그램 종료
                pygame.quit()
                sys.exit()

        gamePad.fill(BLACK)    # 게임 화면 (검은색)

        pygame.display.update() # 게임화면을 다시그림

        clock.tick(60)         # 게임화면의 초당 프레임수를 60으로 설정

    pygame.quit() # pygame 종료

initGame()
runGame()
```



배경 그림 넣기

```
# 게임에 등장하는 객체를 드로잉
```

```
def drawObject(obj, x, y):  
    global gamePad  
    gamePad.blit(obj, (x, y))
```

```
def initGame():
```

```
    global gamePad, clock, background
```

```
    pygame.init()
```

```
    gamePad = pygame.display.set_mode((padWidth, padHeight))
```

```
    pygame.display.set_caption('PyShooting') # 게임 이름
```

```
    background = pygame.image.load('background.png') # 배경 그림
```

```
    clock = pygame.time.Clock()
```

```
def runGame():
```

```
    global gamepad, clock, background
```

```
    onGame = False
```

```
    while not onGame:
```

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

```
            if event.type in [pygame.QUIT]: # 게임 프로그램 종료
```

```
                pygame.quit()
```

```
                sys.exit()
```

```
        drawObject(background, 0, 0) # 배경 화면 그리기
```

```
        pygame.display.update() # 게임화면을 다시그림
```

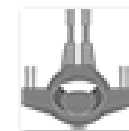


전투기 넣기 ①

```
def initGame():  
    global gamePad, clock, background, fighter  
    pygame.init()  
    gamePad = pygame.display.set_mode((padWidth, padHeight))  
    pygame.display.set_caption('PyShooting')           # 게임 이름  
    background = pygame.image.load('background.png')   # 배경 그림  
    fighter = pygame.image.load('fighter.png')         # 전투기 그림  
    clock = pygame.time.Clock()
```

```
def runGame():  
    global gamepad, clock, background, fighter
```

```
# 전투기 크기  
fighterSize = fighter.get_rect().size  
fighterWidth = fighterSize[0]  
fighterHeight = fighterSize[1]  
  
# 전투기 초기 위치 (x,y)  
x = padWidth * 0.45  
y = padHeight * 0.9  
fighterX = 0
```



fighter.png

전투기 넣기 ②

```
onGame = False
while not onGame:
    for event in pygame.event.get():
        if event.type in [pygame.QUIT]:    # 게임 프로그램 종료
            pygame.quit()
            sys.exit()

    drawObject(background, 0, 0)    # 배경 화면 그리기

    drawObject(fighter, x, y)    # 비행기를 게임 화면의 (x,y) 좌표에 그림
```



전투기 움직이기

```
onGame = False
while not onGame:
    for event in pygame.event.get():
        if event.type in [pygame.QUIT]:      # 게임 프로그램 종료
            pygame.quit()
            sys.exit()

        if event.type in [pygame.KEYDOWN]:
            if event.key == pygame.K_LEFT:    # 전투기 왼쪽으로 이동
                fighterX -= 5

            elif event.key == pygame.K_RIGHT: # 전투기 오른쪽으로 이동
                fighterX += 5

        if event.type in [pygame.KEYUP]:     # 방향키를 떼면 전투기 멈춤
            if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
                fighterX = 0

    drawObject(background, 0, 0)             # 배경 화면 그리기

    # 전투기 위치 재조정
    x += fighterX
    if x < 0:
        x = 0
    elif x > padWidth - fighterWidth:
        x = padWidth - fighterWidth
```



미사일 발사하기 ①

```
def initGame():  
    global gamePad, clock, background, fighter, missile  
    pygame.init()  
    gamePad = pygame.display.set_mode((padWidth, padHeight))  
    pygame.display.set_caption('PyShooting')           # 게임 이름  
    background = pygame.image.load('background.png')   # 배경 그림  
    fighter = pygame.image.load('fighter.png')        # 전투기 그림  
    missile = pygame.image.load('missile.png')         # 미사일 그림  
    clock = pygame.time.Clock()
```



```
def runGame():  
    global gamepad, clock, background, fighter, missile
```

```
# 무기 좌표 리스트  
missileXY = []
```

미사일 발사하기 ②

```
onGame = False
while not onGame:
    for event in pygame.event.get():
        if event.type in [pygame.QUIT]:      # 게임 프로그램 종료
            pygame.quit()
            sys.exit()

        if event.type in [pygame.KEYDOWN]:
            if event.key == pygame.K_LEFT:    # 전투기 왼쪽으로 이동
                fighterX -= 5

            elif event.key == pygame.K_RIGHT: # 전투기 오른쪽으로 이동
                fighterX += 5

            elif event.key == pygame.K_SPACE: # 미사일 발사
                missileX = x + fighterWidth/2
                missileY = y - fighterHeight
                missileXY.append([missileX, missileY])
```



미사일 발사하기 ③

`drawObject(fighter, x, y)` # 비행기를 게임 화면의 (x,y) 좌표에 그림

```
# 미사일 발사 화면에 그리기
if len(missileXY) != 0:
    for i, bxy in enumerate(missileXY): # 미사일 요소에 대해 반복함
        bxy[1] -= 10 # 총알의 y좌표 -10 (위로 이동)
        missileXY[i][1] = bxy[1]

        if bxy[1] <= 0: # 미사일이 화면 밖을 벗어나면
            try:
                missileXY.remove(bxy) # 미사일 제거
            except:
                pass

if len(missileXY) != 0:
    for bx, by in missileXY:
        drawObject(missile, bx, by)
```

`pygame.display.update()` # 게임화면을 다시그림



랜덤한 운석 떨어지기 ①

```
import pygame
import sys
import random
from time import sleep
```

```
padWidth = 480      # 게임화면의 가로크기
padHeight = 640     # 게임화면의 세로크기
```

```
rockImage = ['rock01.png', 'rock02.png', 'rock03.png', 'rock04.png', 'rock05.png', \
             'rock06.png', 'rock07.png', 'rock08.png', 'rock09.png', 'rock10.png', \
             'rock11.png', 'rock12.png', 'rock13.png', 'rock14.png', 'rock15.png', \
             'rock16.png', 'rock17.png', 'rock18.png', 'rock19.png', 'rock20.png', \
             'rock21.png', 'rock22.png', 'rock23.png', 'rock24.png', 'rock25.png', \
             'rock26.png', 'rock27.png', 'rock28.png', 'rock29.png', 'rock30.png' ]
```


랜덤한 운석 떨어지기 ②

무기 좌표 리스트

missileXY = []

운석 랜덤 생성

rock = pygame.image.load(random.choice(rockImage))

rockSize = rock.get_rect().size # 운석 크기

rockWidth = rockSize[0]

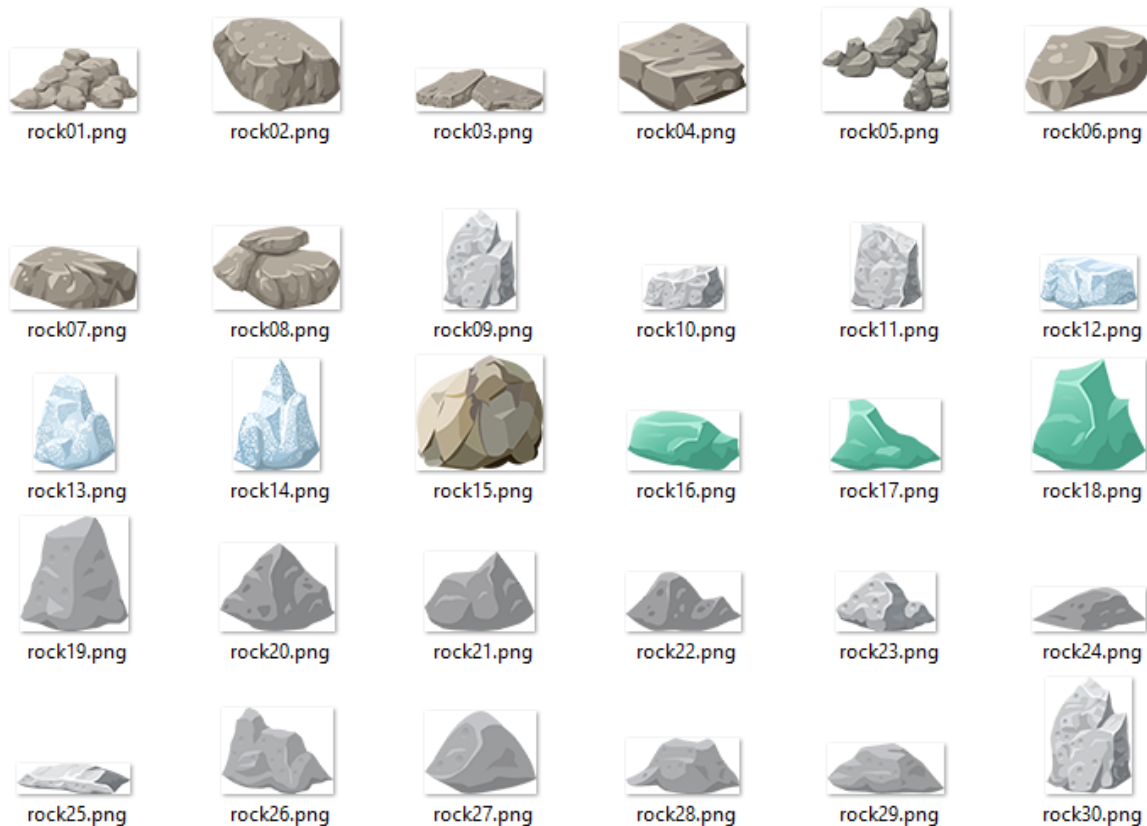
rockHeight = rockSize[1]

운석 초기 위치 설정

rockX = random.randrange(0, padWidth - rockWidth)

rockY = 0

rockSpeed = 2



랜덤한 운석 떨어지기 ③

```
if len(missileXY) != 0:  
    for bx, by in missileXY:  
        drawObject(missile, bx, by)
```

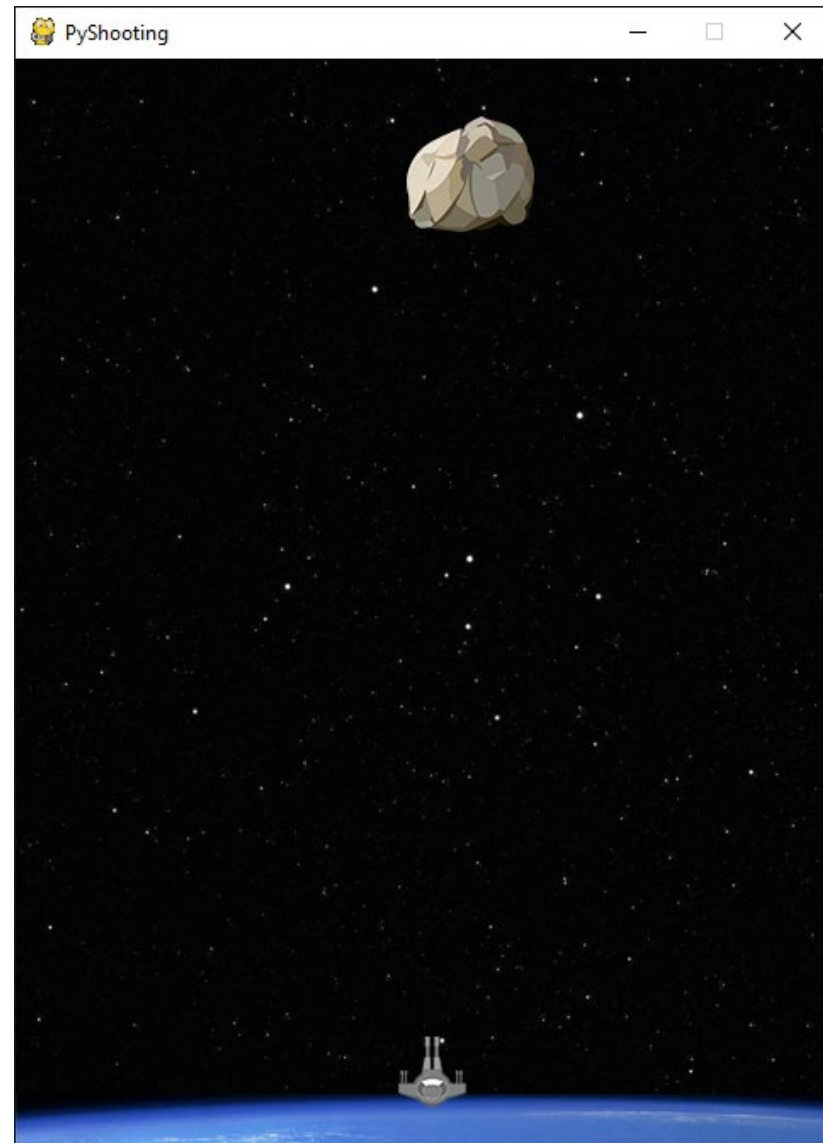
```
rockY += rockSpeed # 운석 아래로 움직임
```

```
# 운석이 지구로 떨어진 경우
```

```
if rockY > padHeight:  
    # 새로운 운석 (랜덤)  
    rock = pygame.image.load(random.choice(rockImage))  
    rockSize = rock.get_rect().size  
    rockWidth = rockSize[0]  
    rockHeight = rockSize[1]  
    rockX = random.randrange(0, padWidth - rockWidth)  
    rockY = 0
```

```
drawObject(rock, rockX, rockY) # 운석 그리기
```

```
pygame.display.update() # 게임화면을 다시그림
```



미사일로 운석 파괴하기 ①

```
def initGame():  
    global gamePad, clock, background, fighter, missile, explosion  
    pygame.init()  
    gamePad = pygame.display.set_mode((padWidth, padHeight))  
    pygame.display.set_caption('PyShooting') # 게임 이름  
    background = pygame.image.load('background.png') # 배경 그림  
    fighter = pygame.image.load('fighter.png') # 전투기 그림  
    missile = pygame.image.load('missile.png') # 미사일 그림  
    explosion = pygame.image.load('explosion.png') # 폭발 그림  
    clock = pygame.time.Clock()
```



```
def runGame():  
    global gamepad, clock, background, fighter, missile, explosion
```

```
# 전투기 미사일에 운석이 맞았을 경우 True  
isShot = False  
shotCount = 0  
rockPassed = 0
```

```
onGame = False  
while not onGame:
```

미사일로 운석 파괴하기 ②

```
# 미사일 발사 화면에 그리기
if len(missileXY) != 0:
    for i, bxy in enumerate(missileXY): # 미사일 요소에 대해 반복함
        bxy[1] -= 10 # 총알의 y좌표 -10 (위로 이동)
        missileXY[i][1] = bxy[1]
```

```
# 미사일이 운석을 맞추었을 경우
if bxy[1] < rockY:
    if bxy[0] > rockX and bxy[0] < rockX + rockWidth:
        missileXY.remove(bxy)
        isShot = True
        shotCount += 1
```

```
if bxy[1] <= 0: # 미사일이 화면 밖을 벗어나면
    try:
        missileXY.remove(bxy) # 미사일 제거
    except:
        pass
```



미사일로 운석 파괴하기 ③

```
# 운석을 맞춘 경우
if isShot:
    # 운석 폭발
    drawObject(explosion, rockX, rockY) # 운석 폭발 그리기

    # 새로운 운석 (랜덤)
    rock = pygame.image.load(random.choice(rockImage))
    rockSize = rock.get_rect().size
    rockWidth = rockSize[0]
    rockHeight = rockSize[1]
    rockX = random.randrange(0, padWidth - rockWidth)
    rockY = 0
    isShot = False

drawObject(rock, rockX, rockY) # 운석 그리기
```



파괴한 운석 수와 놓친 운석 수 표시하기 ①

```
# 운석을 맞춘 개수 계산
def writeScore(count):
    global gamePad
    font = pygame.font.Font('NanumGothic.ttf', 20)
    text = font.render('파괴한 운석 수:' + str(count), True, (255,255,255))
    gamePad.blit(text,(10,0))

# 운석이 화면 아래로 통과한 개수
def writePassed(count):
    global gamePad
    font = pygame.font.Font('NanumGothic.ttf', 20)
    text = font.render('놓친 운석 :' + str(count), True, (255,0,0))
    gamePad.blit(text, (360,0))
```



파괴한 운석 수와 놓친 운석 수 표시하기 ②

```
# 운석 맞춘 점수 표시  
writeScore(shotCount)
```

```
rockY += rockSpeed # 운석 아래로 움직임
```

```
# 운석이 지구로 떨어진 경우
```

```
if rockY > padHeight:
```

```
    # 새로운 운석 (랜덤)
```

```
    rock = pygame.image.load(random.choice(rockImage))
```

```
    rockSize = rock.get_rect().size
```

```
    rockWidth = rockSize[0]
```

```
    rockHeight = rockSize[1]
```

```
    rockX = random.randrange(0, padWidth - rockWidth)
```

```
    rockY = 0
```

```
    rockPassed += 1
```

```
# 놓친 운석 수 표시  
writePassed(rockPassed)
```

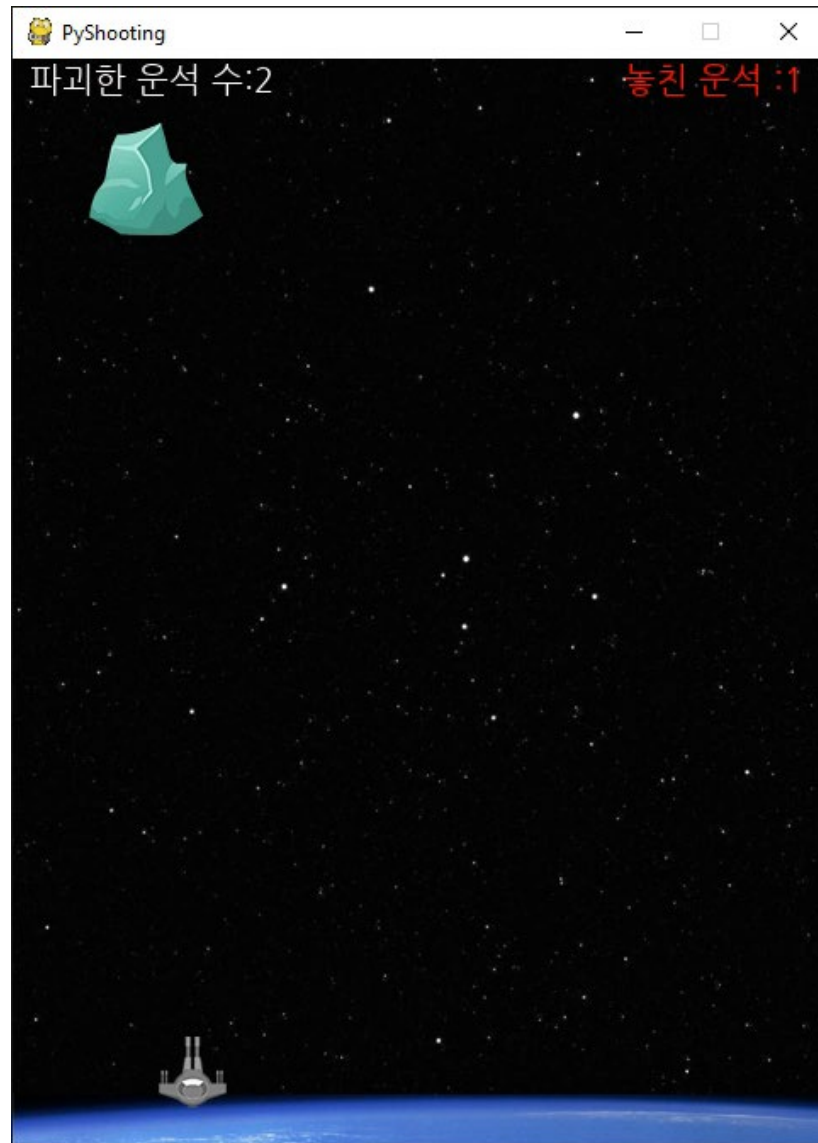


운석 맞추면 운석 속도 증가하기

```
# 운석을 맞춘 경우
if isShot:
    # 운석 폭발
    drawObject(explosion, rockX, rockY) # 운석 폭발 그리기

    # 새로운 운석 (랜덤)
    rock = pygame.image.load(random.choice(rockImage))
    rockSize = rock.get_rect().size
    rockWidth = rockSize[0]
    rockHeight = rockSize[1]
    rockX = random.randrange(0, padWidth - rockWidth)
    rockY = 0
    isShot = False
```

```
#운석 맞추면 속도 증가
rockSpeed += 0.02
if rockSpeed >= 10:
    rockSpeed = 10
```



전투기가 운석과 충돌하거나 운석을 놓치면 게임 오버 ①

게임 메시지 출력

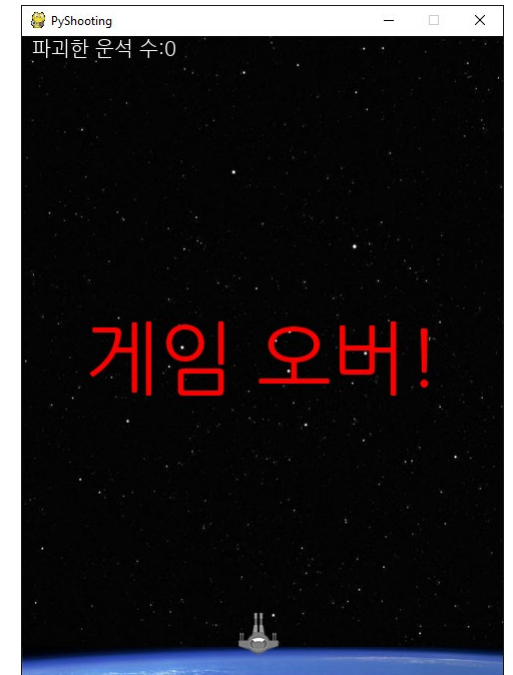
```
def writeMessage(text):  
    global gamePad  
    textfont = pygame.font.Font('NanumGothic.ttf', 80)  
    text = textfont.render(text, True, (255,0,0))  
    textpos = text.get_rect()  
    textpos.center = (padWidth/2, padHeight/2)  
    gamePad.blit(text, textpos)  
    pygame.display.update()  
    sleep(2)  
    runGame()
```

전투기가 운석과 충돌했을 때 메시지 출력

```
def crash():  
    global gamePad  
    writeMessage('전투기 파괴!')
```

게임 오버 메시지 보이기

```
def gameOver():  
    global gamePad  
    writeMessage('게임 오버!')
```



전투기가 운석과 충돌하거나 운석을 놓치면 게임 오버 ②

```
# 전투기 위치 재조정
```

```
x += fighterX
```

```
if x < 0:
```

```
    x = 0
```

```
elif x > padWidth - fighterWidth:
```

```
    x = padWidth - fighterWidth
```

```
# 전투기가 운석과 충돌했는지 체크
```

```
if y < rockY + rockHeight:
```

```
    if(rockX > x and rockX < x + fighterWidth) or \
```

```
        (rockX + rockWidth > x and rockX + rockWidth < x + fighterWidth)
```

```
        crash()
```

```
drawObject(fighter, x, y) # 비행기를 게임 화면의 (x,y) 좌표에 그림
```

```
if rockPassed == 3: #운석 3개 놓치면 게임오버  
    gameOver()
```

```
# 놓친 운석 수 표시
```

```
writePassed(rockPassed)
```



게임 사운드 넣기 ①

```
padWidth = 480      # 게임화면의 가로크기
padHeight = 640     # 게임화면의 세로크기
rockImage = ['rock01.png', 'rock02.png', 'rock03.png', 'rock04.png', 'rock05.png', \
             'rock06.png', 'rock07.png', 'rock08.png', 'rock09.png', 'rock10.png', \
             'rock11.png', 'rock12.png', 'rock13.png', 'rock14.png', 'rock15.png', \
             'rock16.png', 'rock17.png', 'rock18.png', 'rock19.png', 'rock20.png', \
             'rock21.png', 'rock22.png', 'rock23.png', 'rock24.png', 'rock25.png', \
             'rock26.png', 'rock27.png', 'rock28.png', 'rock29.png', 'rock30.png' ]
explosionSound = ['explosion01.wav', 'explosion02.wav', 'explosion03.wav', 'explosion04.wav' ]
```

게임 사운드 넣기 ②

```
def initGame():
    global gamePad, clock, background, fighter, missile, explosion, missileSound, gameOverSound
    pygame.init()
    gamePad = pygame.display.set_mode((padWidth, padHeight))
    pygame.display.set_caption('PyShooting') # 게임 이름
    background = pygame.image.load('background.png') # 배경 그림
    fighter = pygame.image.load('fighter.png') # 전투기 그림
    missile = pygame.image.load('missile.png') # 미사일 그림
    explosion = pygame.image.load('explosion.png') # 폭발 그림
    pygame.mixer.music.load('music.wav') # 배경 음악
    pygame.mixer.music.play(-1) # 배경 음악 재생
    missileSound = pygame.mixer.Sound('missile.wav') # 미사일 사운드
    gameOverSound = pygame.mixer.Sound('gameover.wav') # 게임 오버 사운드
    clock = pygame.time.Clock()
```

```
def runGame():
    global gamepad, clock, background, fighter, missile, explosion, missileSound

    elif event.key == pygame.K_SPACE: # 미사일 발사
        missileSound.play() # 미사일 사운드 재생
        missileX = x + fighterWidth/2
        missileY = y - fighterHeight
        missileXY.append([missileX, missileY])
```

게임 사운드 넣기 ③

게임 메시지 출력

```
def writeMessage(text):  
    global gamePad, gameOverSound  
    textfont = pygame.font.Font('NanumGothic.ttf', 80)  
    text = textfont.render(text, True, (255,0,0))  
    textpos = text.get_rect()  
    textpos.center = (padWidth/2, padHeight/2)  
    gamePad.blit(text, textpos)  
    pygame.display.update()  
    pygame.mixer.music.stop() # 배경 음악 정지  
    gameOverSound.play() # 게임 오버 사운드 재생  
    sleep(2)  
    pygame.mixer.music.play(-1) # 배경 음악 재생  
    runGame()
```

운석 랜덤 생성

```
rock = pygame.image.load(random.choice(rockImage))  
rockSize = rock.get_rect().size # 운석 크기  
rockWidth = rockSize[0]  
rockHeight = rockSize[1]  
destroySound = pygame.mixer.Sound(random.choice(explosionSound))
```

운석을 맞춘 경우

```
if isShot:  
    # 운석 폭발  
    drawObject(explosion, rockX, rockY) # 운석 폭발 그리기  
    destroySound.play() # 운석 폭발 사운드 재생  
  
    # 새로운 운석 (랜덤)  
    rock = pygame.image.load(random.choice(rockImage))  
    rockSize = rock.get_rect().size  
    rockWidth = rockSize[0]  
    rockHeight = rockSize[1]  
    rockX = random.randrange(0, padWidth - rockWidth)  
    rockY = 0  
    destroySound = pygame.mixer.Sound(random.choice(explosionSound))  
    isShot = False
```

PyShooting 게임 완성



