KAIST

PYTHON TUTORING #3

School of Computing, KAIST & 대덕고등학교

INTRO

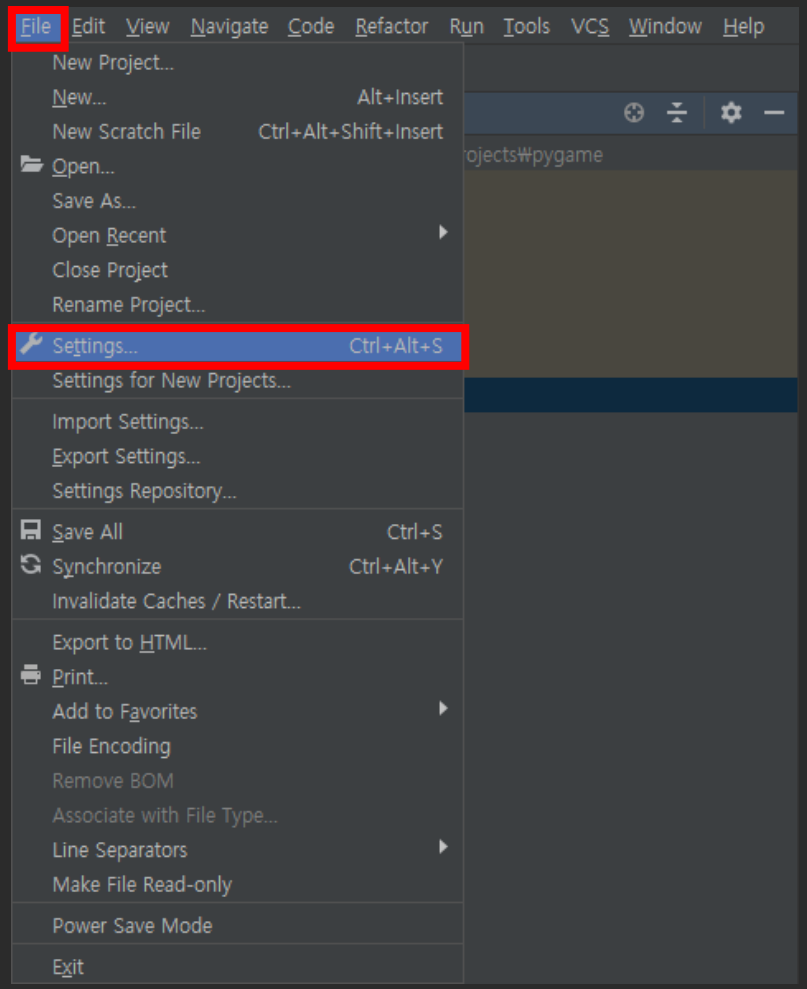
① Pygame 라이브러리 설치

② 강의에 필요한 이미지 파일 다운로드

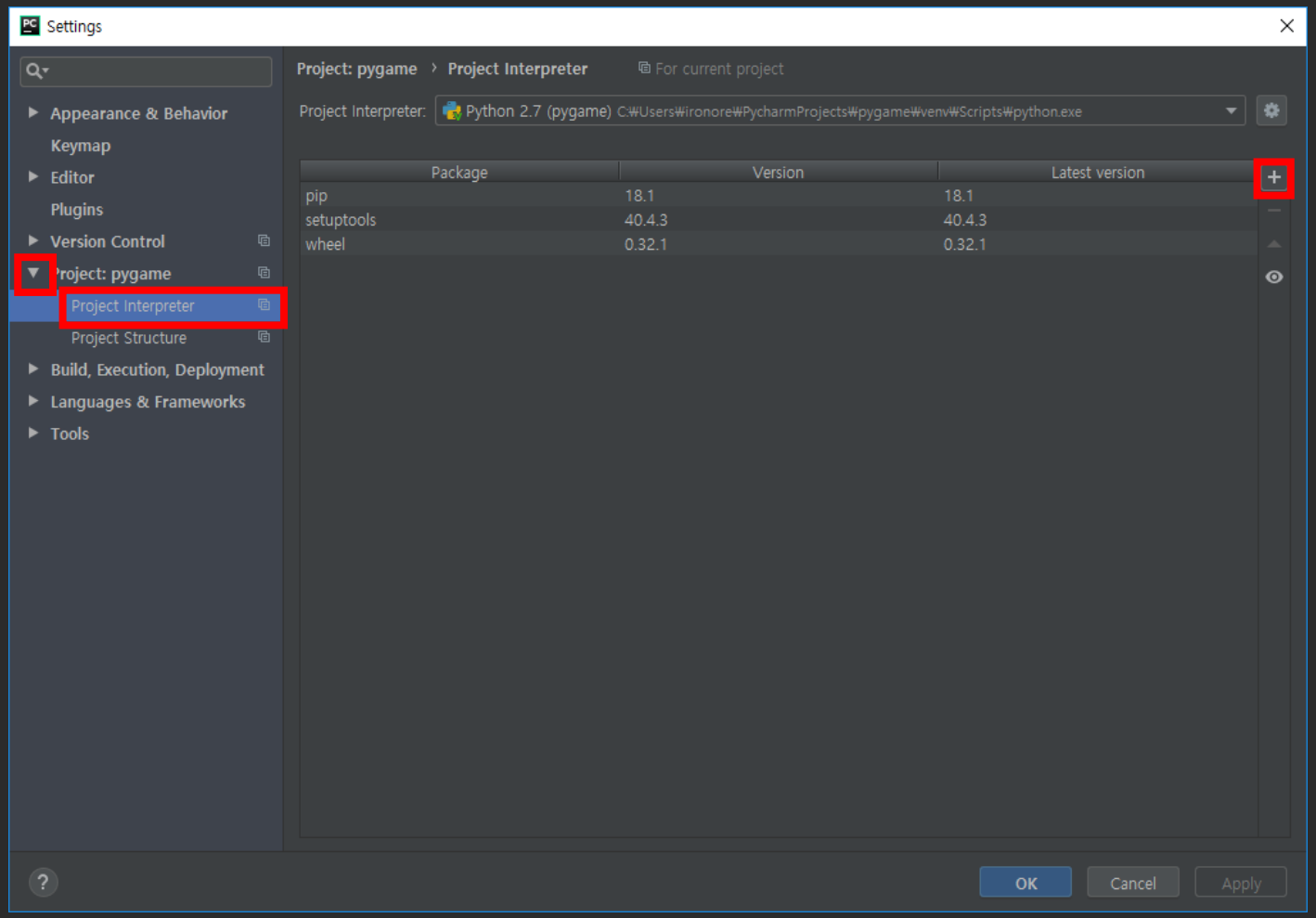
③ 저번 수업 REVIEW

④ GALAGA 구현하기

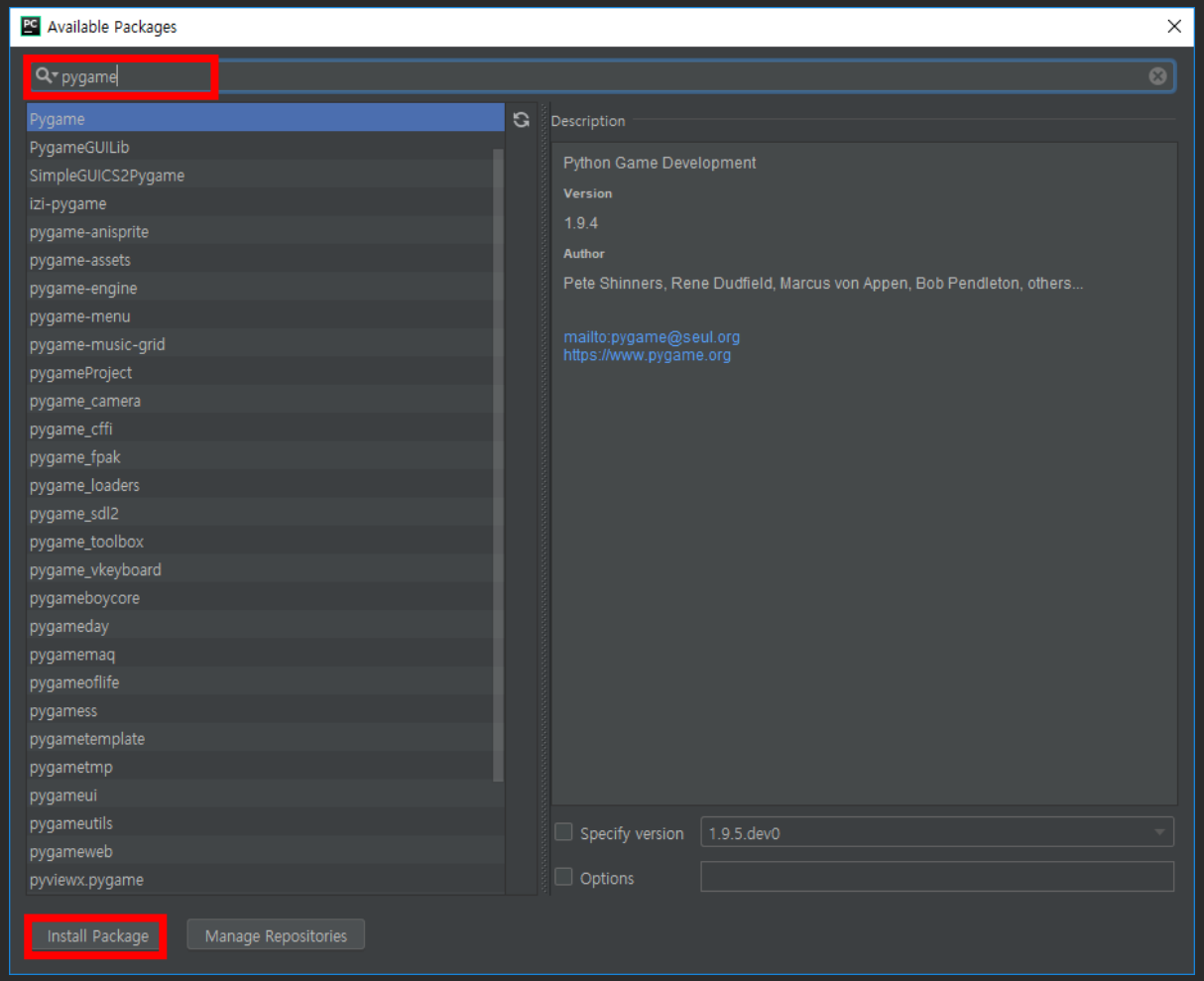
Pygame 라이브러리 설치



Pygame 라이브러리 설치

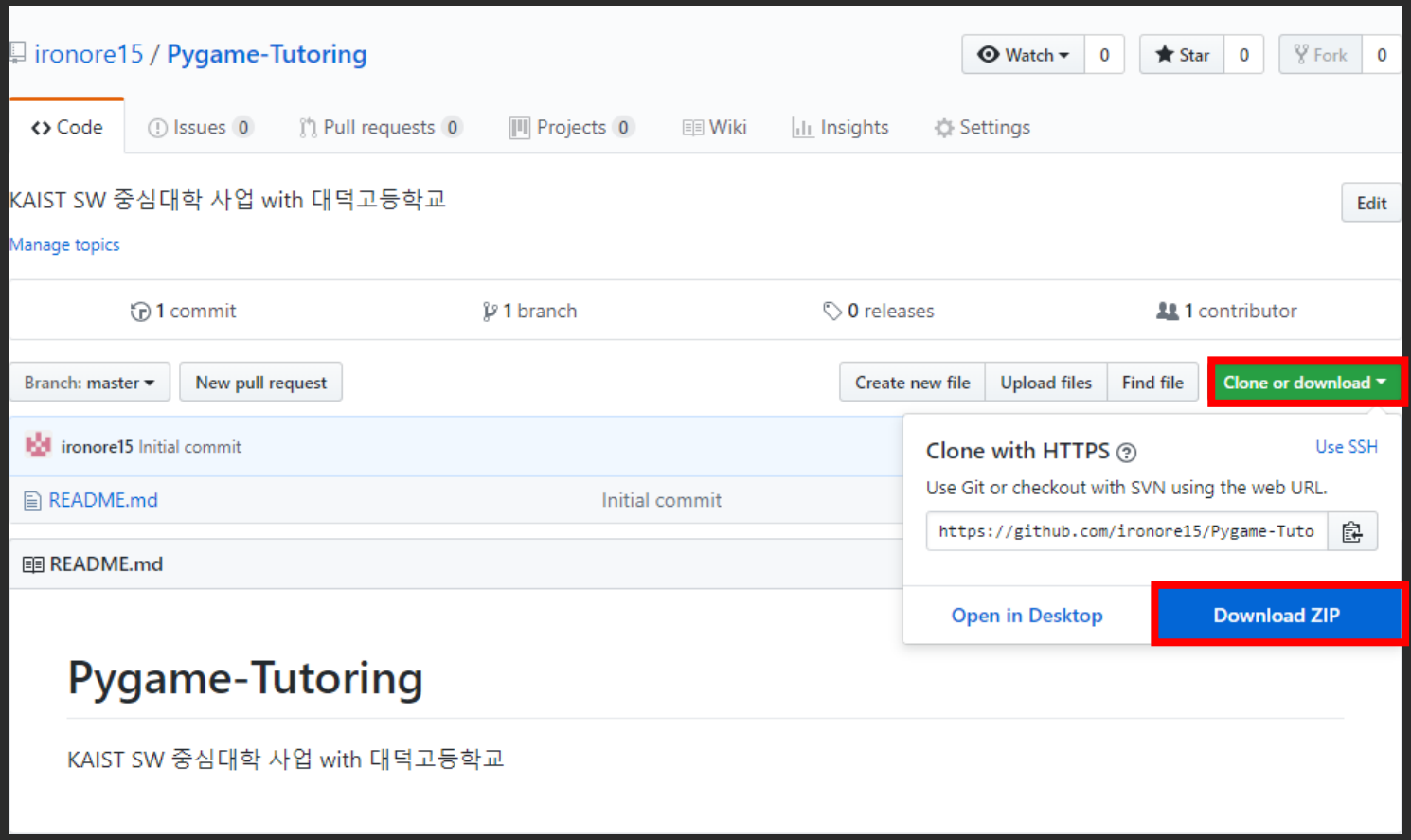


Pygame 라이브러리 설치

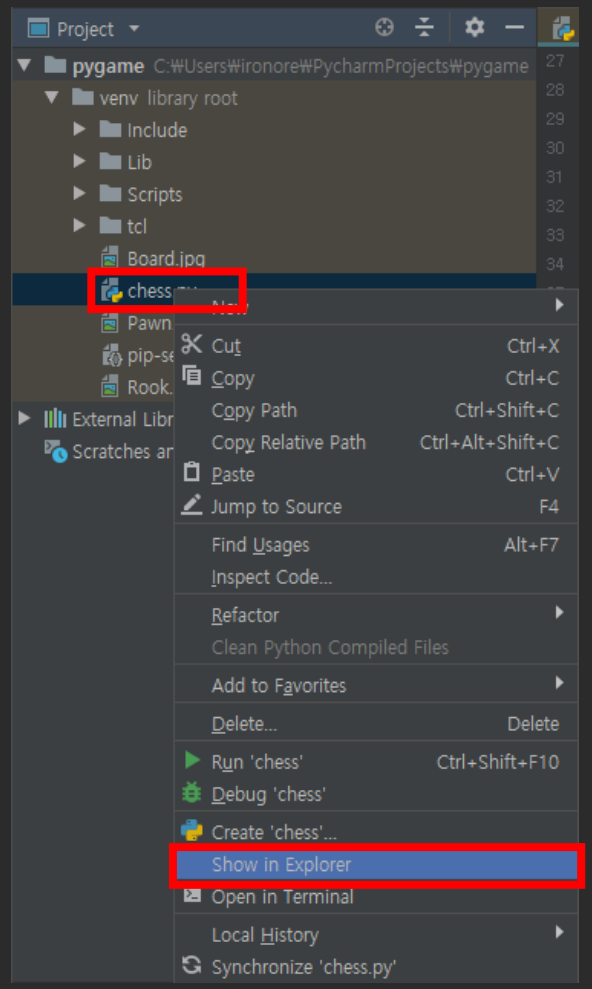


이미지 파일 다운로드

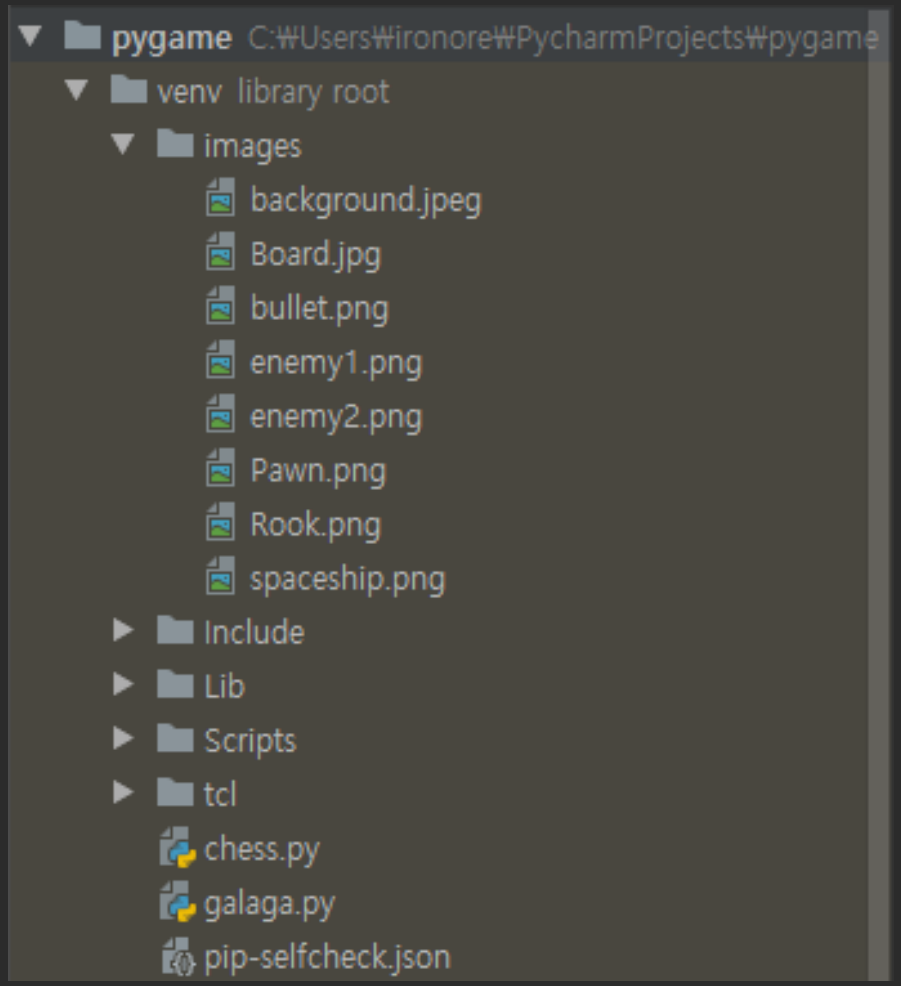
https://github.com/ironore15/Pygame-Tutoring



이미지 파일 다운로드


Project
pygame C:₩Users₩ironore₩PycharmProjects₩pygame
venv library root
Include
Lib
Scripts
tcl
Board.jpg
Pawn
External Libr
Scratches ar
Cut Ctrl+X
Copy Ctrl+C
Copy Path Ctrl+Shift+C
Copy Relative Path Ctrl+Alt+Shift+C
Paste Ctrl+V
Jump to Source F4
Find Usages Alt+F7
Inspect Code...
Refactor
Clean Python Compiled Files
Add to Favorites
Delete... Delete
Run 'chess' Ctrl+Shift+F10
Debug 'chess'
Create 'chess'...
Show in Explorer
Open in Terminal
Local History
Synchronize 'chess.py'

이미지 파일 다운로드


pygame C:₩Users₩ironore₩PycharmProjects₩pygame
venv library root
images
background.jpeg
Board.jpg
bullet.png
enemy1.png
enemy2.png
Pawn.png
Rook.png
spaceship.png
Include
Lib
Scripts
tcl
chess.py
galaga.py
pip-selfcheck.json

Make a screen

```
1  import pygame
2
3  pygame.init()
4
5  width = 400
6  height = 600
7  width, height = 400, 600
8
9  size = (width, height)
10
11 screen = pygame.display.set_mode(size)
12
```

Load images

```
(...)

back = pygame.image.load('images\\background.jpeg').convert()
back = pygame.transform.scale(back, size)

spaceship = pygame.image.load('images\\spaceship.png')
ship_rect = spaceship.get_rect()
ship_rect.center = (200, 550)

bullet = pygame.image.load('images\\bullet.png')


```

Keep screen alive

```
(...)

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

    screen.blit(background, (0, 0))
    screen.blit(spaceship, ship_rect)
    pygame.display.flip()


```

Move with keyboard (1)

```
1  ship_dx, ship_dy, speed = 0, 0, 2
2  while True:
3      for event in pygame.event.get():
4          elif event.type == pygame.KEYDOWN:
5              if event.key == pygame.K_LEFT:
6                  ship_dx -= speed
7              elif event.key == pygame.K_RIGHT:
8                  ship_dx += speed
9              elif event.key == pygame.K_UP:
10                 ship_dy -= speed
11             elif event.key == pygame.K_DOWN:
12                 ship_dy += speed
```

Move with keyboard (2)

```
while True:
    for event in pygame.event.get():
        elif event.type == pygame.KEYUP:
            if event.key == pygame.K_LEFT:
                ship_dx += speed
            elif event.key == pygame.K_RIGHT:
                ship_dx -= speed
            elif event.key == pygame.K_UP:
                ship_dy += speed
            elif event.key == pygame.K_DOWN:
                ship_dy -= speed

```

Move with keyboard (3)

```
while True:
    for event in pygame.event.get():
        elif event.type == pygame.KEYDOWN:
            (...)
        elif event.type == pygame.KEYUP:
            (...)

    ship_rect.move_ip(ship_dx, ship_dy)

    screen.blit(background, (0, 0))
    screen.blit(spaceship, ship_rect)
    pygame.display.flip()
```

Make function

```
1  def drawScreen():
2      screen.blit(background, (0, 0))
3      screen.blit(spaceship, ship_rect)
4      pygame.display.flip()
5
6  while True:
7      (...)
8
9      ship_rect.move_ip(ship_dx, ship_dy)
10     drawScreen()
11
12
```

Control FPS

```
1  def drawScreen():
2      screen.blit(background, (0, 0))
3      screen.blit(spaceship, ship_rect)
4      pygame.display.flip()
5      clock.tick(60)
6
7
8
9  screen = pygame.display.set_mode(size)
10 clock = pygame.time.Clock()
11
12
```

Learn list (1)

```
1  a = [ ]
2  b = [1, 2, 3]
3  c = [Life', 'is', 'too', 'short']
4  d = [1, 2, 'Life', 'is']
5  e = [1, 2, ['Life', 'is']]
6
7  print(b)
8  print(b[0])
9  print(b[0] + b[2])
10
11 b[2] = 5
12 print(b[2])
```

Learn list (2)

```
1  b.append(6)
2  print(b)
3
4  b.append(10)
5  print(b)
6
7  b.remove(5)
8  print(b)
9
10 for i in b:
11     print(i)
12
```

Create bullets (1)

```
1  def createBullet():
2      bullet_rect = bullet.get_rect()
3      bullet_rect.center = ship_rect.midtop
4      bullet_list.append(bullet_rect)
5
6  def drawScreen():
7      screen.blit(background, (0, 0))
8      screen.blit(spaceship, ship_rect)
9      for bullet_rect in bullet_list:
10         screen.blit(bullet, bullet_rect)
11     pygame.display.flip()
12     clock.tick(60)
```

Create bullets (2)

```
    elif event.type == pygame.KEYDOWN:
        if event.key == pygame.K_LEFT:
            ship_dx -= speed
        elif event.key == pygame.K_RIGHT:
            ship_dx += speed
        elif event.key == pygame.K_UP:
            ship_dy -= speed
        elif event.key == pygame.K_DOWN:
            ship_dy += speed
        elif event.key == pygame.K_SPACE:
            createBullet()

```

Move bullets

```
1  def moveBullets():
2      for bullet_rect in bullet_list:
3          bullet_rect.move_ip(0, -5)
4
5  while True:
6      (...)
7
8      ship_rect.move_ip(ship_dx, ship_dy)
9      moveBullets()
10     drawScreen()
11
12
```

Remove bullets

```
1  def removeBullet():
2      for bullet_rect in bullet_list:
3          if bullet_rect.bottom < 0:
4              bullet_list.remove(bullet_rect)
5              return
6
7  while True:
8      (...)
9      ship_rect.move_ip(ship_dx, ship_dy)
10     moveBullets()
11     removeBullet()
12     drawScreen()
```

Create enemy (1)

```
1  import random
2
3
4  def createEnemy():
5      enemy_rect.midtop = (random.randint(0, width), 0)
6
7
8  enemy = pygame.image.load('images\\enemy1.png')
9  enemy_rect = enemy.get_rect()
10 createEnemy()
11
12 score = 0
```

Create enemy (2)

```
1
2  def drawScreen():
3      screen.blit(background, (0, 0))
4      screen.blit(spaceship, ship_rect)
5      for bullet_rect in bullet_list:
6          screen.blit(bullet, bullet_rect)
7      screen.blit(enemy, enemy_rect)
8      pygame.display.flip()
9      clock.tick(60)
10
11
12
```

Move enemy (1)

```
1  def moveEnemy():
2      enemy_rect.move_ip(0, 3)
3
4  while True:
5      (...)
6
7      ship_rect.move_ip(ship_dx, ship_dy)
8      moveEnemy()
9      moveBullets()
10     removeBullet()
11     drawScreen()
12
```

Move enemy (2)

```
1
2  def drawScreen():
3      screen.blit(background, (0, 0))
4      screen.blit(spaceship, ship_rect)
5      for bullet_rect in bullet_list:
6          screen.blit(bullet, bullet_rect)
7
8      screen.blit(enemy, enemy_rect)
9
10     pygame.display.flip()
11     clock.tick(60)
12
```

Remove enemy (1)

```
1  def removeEnemy():
2      global score
3      for bullet_rect in bullet_list:
4          if enemy_rect.colliderect(bullet_rect):
5              score += 800
6              bullet_list.remove(bullet_rect)
7              createEnemy()
8              return
9
10     if enemy_rect.top > height:
11         score -= 200
12         createEnemy()
```

Remove enemy (2)

```
1
2  while True:
3      (...)
4
5      ship_rect.move_ip(ship_dx, ship_dy)
6      moveEnemy()
7      moveBullets()
8      removeEnemy()
9      removeBullet()
10     drawScreen()
11
12
```

Check collision

```
1
2  def checkCrash():
3      if ship_rect.colliderect(enemy_rect):
4          gameOver()
5
6  while True:
7      (...)
8      moveBullets()
9      checkCrash()
10     removeEnemy()
11     (...)
12
```

Game over

```
1  def gameOver():
2      text = gamefont.render("GAME OVER!", True, (255, 0, 0))
3      text_rect = text.get_rect()
4      text_rect.center = (width / 2, height / 2)
5      screen.blit(background, (0, 0))
6      screen.blit(text, text_rect)
7      pygame.display.flip()
8      pygame.time.wait(2000)
9      exit(0)
10 screen = pygame.display.set_mode(size)
11 gamefont = pygame.font.SysFont(None, 80)
12 scorefont = pygame.font.SysFont(None, 20)
```

Show score

```
1  def showScore():
2      text = 'Score: ' + str(score)
3      text = scorefont.render(text, True, (255, 255, 0))
4      text_rect = text.get_rect()
5      text_rect.center = (width / 2, 30)
6      screen.blit(text, text_rect)
7
8  def drawScreen():
9      (...)
10     showScore()
11     pygame.display.flip()
12     clock.tick(60)
```