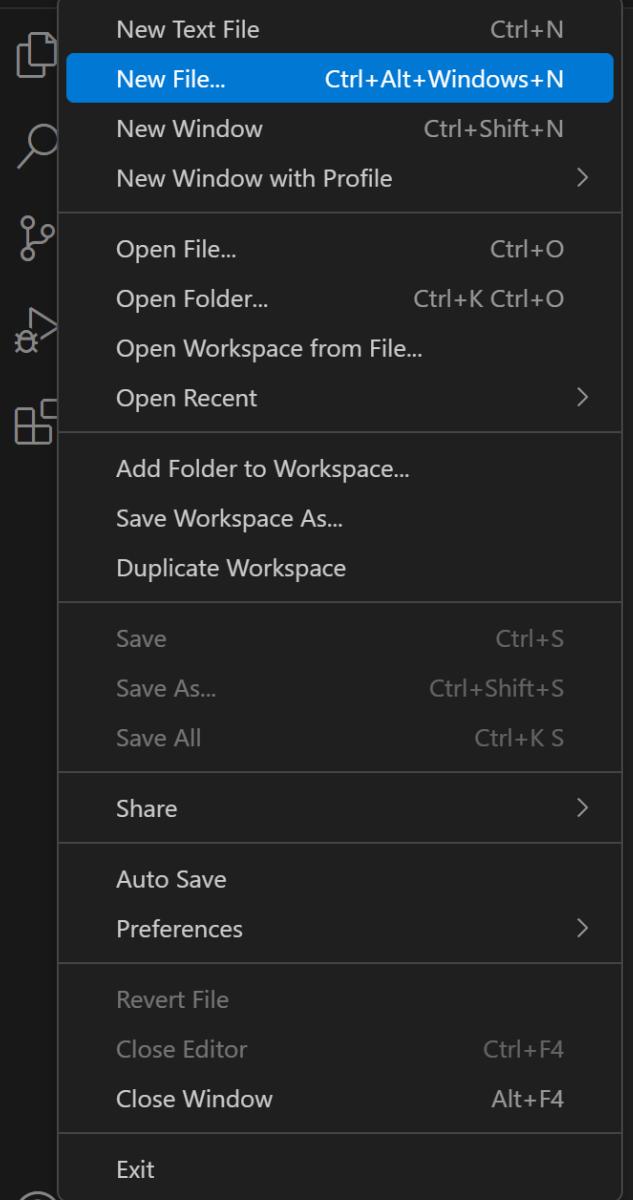


2D Game (1)

2025-09-10



Show All Commands **Ctrl + Shift + P**

Open File **Ctrl + O**

Open Folder **Ctrl + K Ctrl + O**

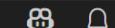
Open Recent **Ctrl + R**

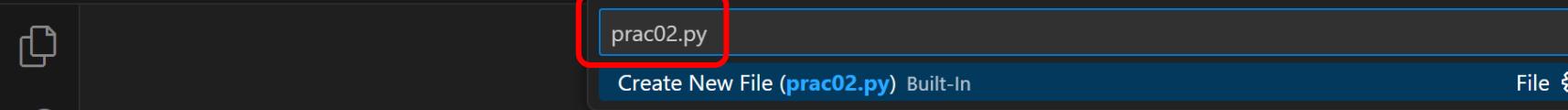
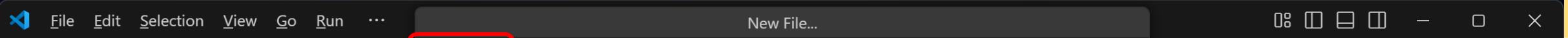
Open Chat **Ctrl + Alt + I**



Restricted Mode

0 ▲ 0





Create New File (prac02.py) Built-In

File



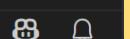
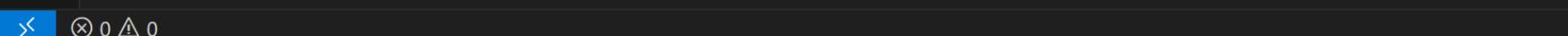
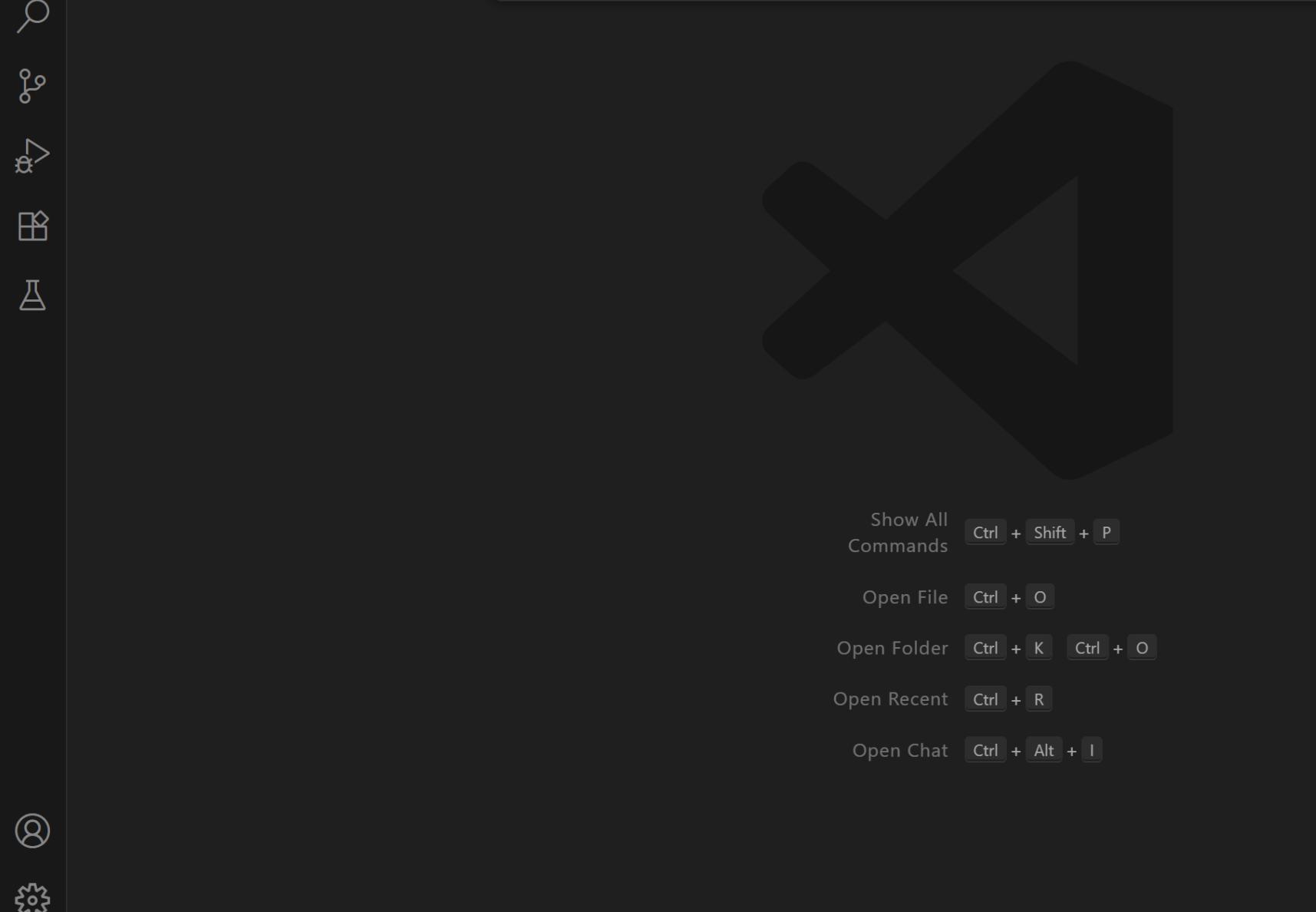
Show All
Commands Ctrl + Shift + P

Open File Ctrl + O

Open Folder Ctrl + K Ctrl + O

Open Recent Ctrl + R

Open Chat Ctrl + Alt + I



← → ▼ ↑



> 바탕 화면 > Commit

▼ C

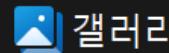
Commit 검색



구성 새 폴더



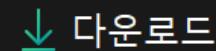
홈



갤러리



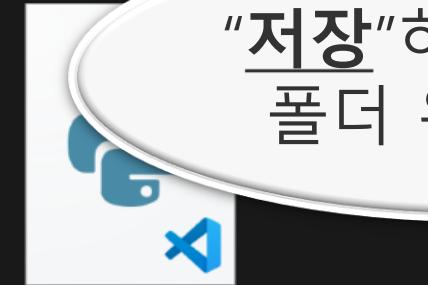
바탕 화면



다운로드



Documents



“저장”하기 원하는
폴더 위치 지정

prac01.py

school.png

파일 이름(N): prac02.py



파일 형식(I): All Files (*.*)



▲ 폴더 숨기기



Create File

취소



prac02.py X



C: > Users > sunje > Desktop > Commit > prac02.py



D ▾ □ ...

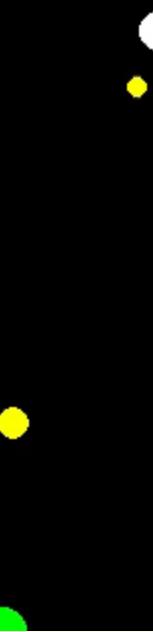
1 *Generate code (Ctrl+I), or select a Language (Ctrl+K M). Start typing to dismiss or don't show this again.*

LV 1

EXP 0/5

TIME 2s

HP 5



게임 제작 (1)

1. 화면 구성 및 캐릭터 이동

- **화면 구성** – 게임 창 설정 후 게임 화면의 크기와 배경색을 설정
- **캐릭터 생성 및 이동** – 간단한 캐릭터를 화면에 표시하고, 키보드 입력을 받아 캐릭터를 상하좌우로 이동

2. 맵 생성 및 충돌 처리

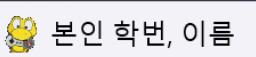
3. 적 생성 및 전투 시스템

4. 아이템 생성 및 게임 완성

The screenshot shows a Python code editor interface with a dark theme. On the left, there's a vertical toolbar with icons for file operations, search, and other development tools. The main area displays a Python script named 'prac02.py'.

```
prac01.py prac02.py ●  
C: > Users > sunje > Desktop > Commit > prac02.py > ...  
1 # Created by 임현우  
2  
3 import pygame  
4  
5 pygame.init() # pygame 기능 전부 초기화(화면/입력/사운드 등 준비)  
6  
7 # 게임 초기화 및 기본 설정  
8 screen = pygame.display.set_mode((800, 600)) # 800x600 크기의 게임 창 만들기  
9 pygame.display.set_caption("본인 학번, 이름")  
10 clock = pygame.time.Clock() # FPS(초당 프레임) 제어용 시계  
11  
12 # 플레이어 관련 변수  
13 player_pos = pygame.Vector2(400, 300) # 플레이어 위치(화면 중앙에서 시작)  
14 player_color = (255, 255, 255) # 플레이어 색  
15  
16 # 메인 루프, 모든 함수를 실행 시키는 부분  
17 while True:  
18     """  
19         창 닫기(X 버튼) 등 OS 이벤트를 처리.  
20         QUIT 이벤트가 오면 pygame을 정리하고 프로그램 종료.  
21     """  
22     for event in pygame.event.get():  
23         if event.type == pygame.QUIT:  
24             pygame.quit()  
25             exit()  
26  
27     screen.fill((0, 0, 0)) # 배경 색상  
28  
29     pygame.draw.circle(screen, player_color, player_pos, 10) # 플레이어 생성  
30  
31     pygame.display.flip() # 더블버퍼링 --> 실제 화면으로 전환  
32  
33     clock.tick(60) # 1초에 60프레임으로 고정
```

The code uses Pygame to create a simple window and draw a circle at the center. It handles the quit event and updates the screen 60 times per second. A red box highlights the code area, and a large red arrow points from the top right towards the bottom right corner of the code editor.



본인 학번, 이름





prac01.py

prac02.py ●

C: > Users > sunje > Desktop > Commit > prac02.py > ...

```
10     clock = pygame.time.Clock()          # FPS(초당 프레임) 제어용 시계
11
12     # 플레이어 관련 변수
13     player_pos = pygame.Vector2(400, 300)    # 플레이어 위치(화면 중앙에서 시작)
14     player_color = (255, 255, 255)           # 플레이어 색
15
16     # 이벤트 처리 함수 (창 닫기 등)
17     def handle_events():
18         """
19             창 닫기(X 버튼) 등 OS 이벤트를 처리.
20             QUIT 이벤트가 오면 pygame을 정리하고 프로그램 종료.
21         """
22
23         for event in pygame.event.get():
24             if event.type == pygame.QUIT:
25                 pygame.quit()
26                 exit()
27
28     # 그리기 함수 (배경/플레이어)
29     def draw_scene():
30         """
31             화면을 모두 그린 뒤 마지막에 flip()으로 실제 모니터에 표시.
32             - 플레이어를 각각 원(circle)으로 간단히 표현
33         """
34
35         screen.fill((0, 0, 0))                  # 배경 색상
36         pygame.draw.circle(screen, player_color, player_pos, 10)    # 플레이어 생성
37         pygame.display.flip()                  # 더블버퍼링 --> 실제 화면으로 전환
38
39     # 메인 루프, 모든 함수를 실행 시키는 부분
40     while True:
41         handle_events()        # 창 닫기 등 OS 이벤트 처리
42         draw_scene()           # 화면 그리기 + flip
43         clock.tick(60)         # 1초에 60프레임으로 고정
```

“함수”(function)
로 표현



⊗ 0 △ 0

연습 문제 (1)

- 키보드 입력을 받아 캐릭터를 상하좌우로 이동시키기

1

prac01.py

prac02.py

```
C: > Users > sunje > Desktop > Commit > prac02.py > player_speed
1 # Created by 임현우
2
3 import pygame
4
5 pygame.init()                                     # pygame 기능 전부 초기화(화면/입력/사운드 등 준비)
6
7 # 게임 초기화 및 기본 설정
8 screen = pygame.display.set_mode((800, 600))      # 800x600 크기의 게임 창 만들기
9 pygame.display.set_caption("본인 학번, 이름")
10 clock = pygame.time.Clock()                      # FPS(초당 프레임) 제어용 시계
11
12 # 플레이어 관련 변수
13 player_pos = pygame.Vector2(400, 300)            # 플레이어 위치(화면 중앙에서 시작)
14 player_color = (255, 255, 255)                   # 플레이어 색
15 player_speed = 3                                 # 플레이어 이동 속도
16
17 # 이벤트 처리 함수 (창 닫기 등)
18 def handle_events():
19     """
20         창 닫기(X 버튼) 등 OS 이벤트를 처리.
21         QUIT 이벤트가 오면 pygame을 정리하고 프로그램 종료.
22     """
23     for event in pygame.event.get():
24         if event.type == pygame.QUIT:
25             pygame.quit()
26             exit()
27
28 # 그리기 함수 (배경/플레이어)
29 def draw_scene():
30     """
31         화면을 모두 그린 뒤 마지막에 flip()으로 실제 모니터에 표시.
32         - 플레이어를 각각 원(circle)으로 간단히 표현
33     """
34     screen.fill((0, 0, 0))                          # 배경 색상
```





prac01.py

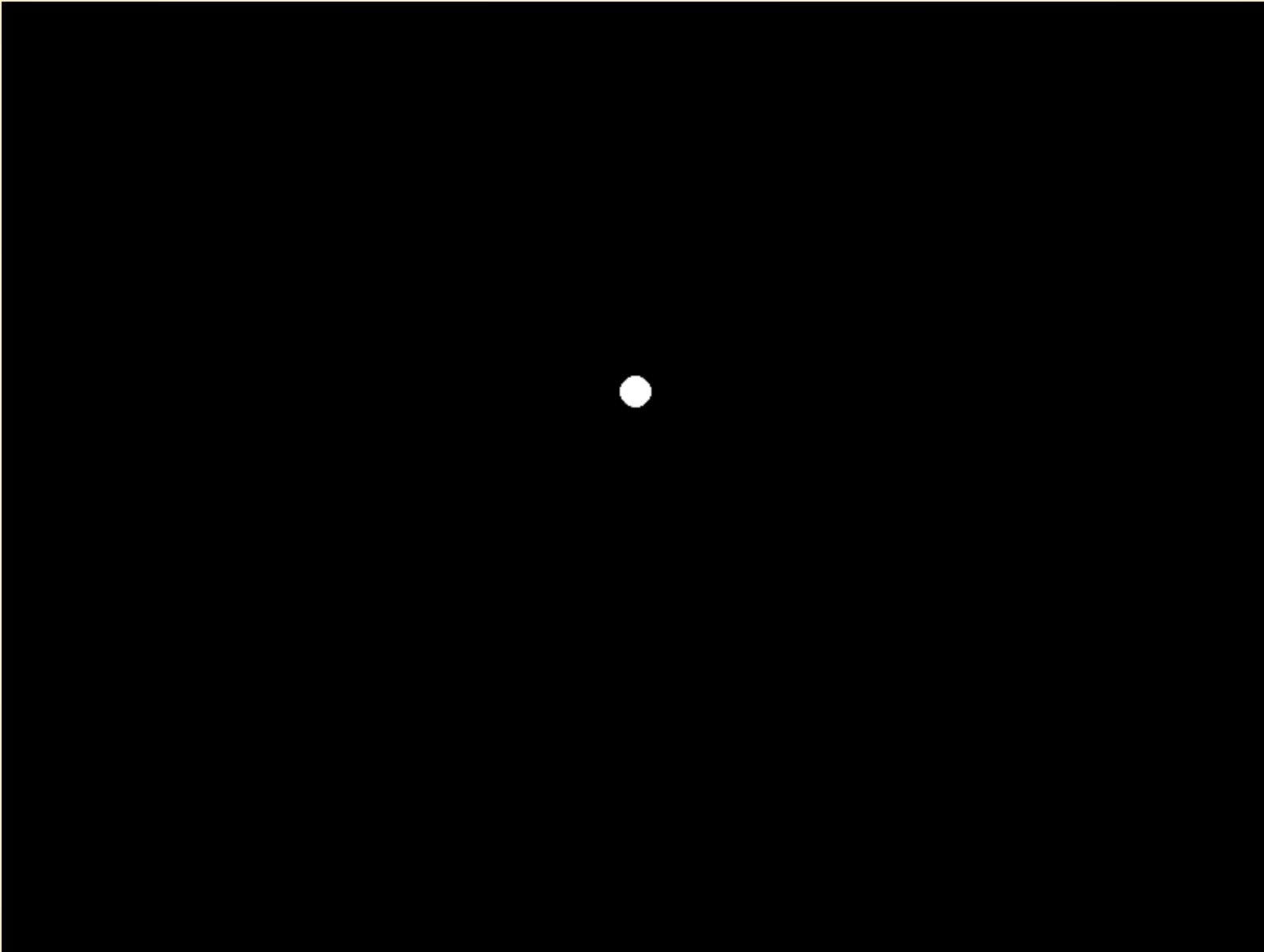
prac02.py ●

C: > Users > sunje > Desktop > Commit > prac02.py > ...

```
22     """
23     for event in pygame.event.get():
24         if event.type == pygame.QUIT:
25             pygame.quit()
26             exit()
27
28     # 그리기 함수 (배경/플레이어)
29     def draw_scene():
30         """
31             화면을 모두 그린 뒤 마지막에 flip()으로 실제 모니터에 표시.
32             - 플레이어를 각각 원(circle)으로 간단히 표현
33
34             screen.fill((0, 0, 0))                      # 배경 색상
35             pygame.draw.circle(screen, player_color, player_pos, 10)    # 플레이어 생성
36             pygame.display.flip()                      # 더블버퍼링 --> 실제 화면으로 전환
37
38     # 메인 루프, 모든 함수를 실행 시키는 부분
39     while True:
40         handle_events()                         # 창 닫기 등 OS 이벤트 처리
41
42         keys = pygame.key.get_pressed()        # 키보드 눌림 상태 얻기
43         """
44
45         키보드 입력(WASD)에 따라 플레이어 위치를 바꿈.
46         좌표는 Vector2라서 .x, .y 로 접근해 더하고 빼면 됨.
47
48         if keys[pygame.K_w]: player_pos.y -= player_speed
49         if keys[pygame.K_s]: player_pos.y += player_speed
50         if keys[pygame.K_a]: player_pos.x -= player_speed
51         if keys[pygame.K_d]: player_pos.x += player_speed
52
53         draw_scene()                          # 화면 그리기 + flip
54         clock.tick(60)                      # 1초에 60프레임으로 고정
```



⊗ 0 △ 0



The screenshot shows a code editor interface with a dark theme. On the left, there are several icons: a file icon, a search icon, a refresh icon, a file list icon, a zoom icon, and a gear icon. The main area displays a Python script named `prac02.py`. The code implements a simple game loop using Pygame. It handles events, checks for key presses, moves the player based on those keys, and then draws the scene and updates the display.

```
File Edit Selection View Go Run ... ⏪ ⏫ ⏭ Search 09:00 00:00 - X  
prac01.py prac02.py ●  
C: > Users > sunje > Desktop > Commit > prac02.py > ...  
24     if event.type == pygame.QUIT:  
25         pygame.quit()  
26         exit()  
27  
28 # 입력 처리 함수(플레이어 이동)  
29 def move_player(keys):  
30     """  
31     키보드 입력(WASD)에 따라 플레이어 위치를 바꿈.  
32     좌표는 Vector2라서 .x, .y로 접근해 더하고 빼면 됨.  
33     """  
34     if keys[pygame.K_w]: player_pos.y -= player_speed  
35     if keys[pygame.K_s]: player_pos.y += player_speed  
36     if keys[pygame.K_a]: player_pos.x -= player_speed  
37     if keys[pygame.K_d]: player_pos.x += player_speed  
38  
39 # 그리기 함수 (배경/플레이어)  
40 def draw_scene():  
41     """  
42     화면을 모두 그린 뒤 마지막에 flip()으로 실제 모니터에 표시.  
43     - 플레이어를 각각 원(circle)으로 간단히 표현  
44     """  
45     screen.fill((0, 0, 0)) # 배경 색상  
46     pygame.draw.circle(screen, player_color, player_pos, 10) # 플레이어 생성  
47     pygame.display.flip() # 더블бу퍼링 --> 실제 화면으로 전환  
48  
49 # 메인 루프, 모든 함수를 실행 시키는 부분  
50 while True:  
51     handle_events() # 창 닫기 등 OS 이벤트 처리  
52  
53     keys = pygame.key.get_pressed() # 키보드 눌림 상태 얻기  
54     move_player(keys) # 입력에 따라 플레이어 이동  
55  
56     draw_scene() # 화면 그리기 + flip  
57     clock.tick(60) # 1초에 60프레임으로 고정  
Ln 54, Col 16 Spaces: 4 UTF-8 CRLF {} Python 3.13.7
```

게임 제작 (2)

1. 화면 구성 및 캐릭터 이동
2. 맵 생성 및 충돌 처리
- 3. 적 생성 및 전투 시스템**
 - 적 캐릭터 생성 – 간단한 AI를 가진 적 캐릭터를 생성하고 맵에 배치
 - 전투 시스템 구현 – 플레이어와 적 간의 기본적인 전투 메커니즘을 구현 (예: 공격력, 체력 등)
4. 아이템 생성 및 게임 완성

연습 문제 (2)

- 일정 시간 간격마다 적 생성하기

File Edit Selection View Go Run ... ⏪ ⏩ Search

C: > Users > sunje > Desktop > Commit > prac02.py > [e] enemies

```
1 # Created by 임현우
2
3 import pygame, random
4
5 pygame.init()                                     # pygame 기능 전부 초기화(화면/입력/사운드 등 준비)
6
7 # 게임 초기화 및 기본 설정
8 screen = pygame.display.set_mode((800, 600))      # 800x600 크기의 게임 창 만들기
9 pygame.display.set_caption("본인 학번, 이름")
10 clock = pygame.time.Clock()                      # FPS(초당 프레임) 제어용 시계
11
12 # 플레이어 관련 변수
13 player_pos = pygame.Vector2(400, 300)           # 플레이어 위치(화면 중앙에서 시작)
14 player_color = (255, 255, 255)                   # 플레이어 색
15 player_speed = 3                                # 플레이어 이동 속도
16
17 # 적 관련 변수
18 enemies = []                                     # 적 리스트
19 enemy_spawn_delay = 1000                         # 적 생성 간격 (마지막 적 생성 후 enemy_spawn_delay 만큼 기다려야 다시 생성)
20 last_spawn_time = 0                             # 마지막 적 생성 시각
21
22 # 이벤트 처리 함수 (창 닫기 등)
23 def handle_events():
24     """
25         창 닫기(X 버튼) 등 OS 이벤트를 처리.
26         QUIT 이벤트가 오면 pygame을 정리하고 프로그램 종료.
27     """
28     for event in pygame.event.get():
29         if event.type == pygame.QUIT:
30             pygame.quit()
31             exit()
32
33 # 입력 처리 함수(플레이어 이동)
34 def move_player(keys):
```

Ln 18, Col 8 Spaces: 4 UTF-8 CRLF {} Python 3.13.7

File Edit Selection View Go Run ... ⏪ ⏩ Search ⏴ ⏵

C: > Users > sunje > Desktop > Commit > prac02.py > move_enemies

```
39     if keys[pygame.K_w]: player_pos.y -= player_speed
40     if keys[pygame.K_s]: player_pos.y += player_speed
41     if keys[pygame.K_a]: player_pos.x -= player_speed
42     if keys[pygame.K_d]: player_pos.x += player_speed
43
44 # 적 생성 함수
45 def spawn_enemy(now):
46     """
47     일정 간격(enemy_spawn_delay)마다 적 1마리 생성.
48     - 적은 랜덤 위치에서 등장함
49     """
50
51     global last_spawn_time
52     if now - last_spawn_time > enemy_spawn_delay:
53         # 초록색 적
54         color = (0, 255, 0)
55
56         # 랜덤 위치에서 적이 생성되도록 하는 코드
57         pos = pygame.Vector2(random.randint(0, 800), random.choice([-20, 620]))
58         # x 좌표 = random.randint(0, 800): 0~800 사이 랜덤한 위치
59         # y 좌표 = random.choice([-20, 620]): -20이나 620 중 선택
60         # -20과 620은 각각 화면의 맨 위와 맨 아래에서 스폰하는 것을 뜻함
61
62         enemies.append({"pos": pos, "color": color})      # 위에서 정해진 좌표(pos)를 이용하여 적을 생성함
63                                     # 적이 나온 시간을 기록 후, 다음 적이 생성되기 까지 기다림
64
65 # 이동 함수(적)
66 def move_enemies():
67     # (플레이어 위치 - 적 위치) --> 플레이어를 바라보는 방향 벡터
68     # normalize()로 길이를 1로 맞춘 뒤 * 2 해서 속도 적용
69     for enemy in enemies:
70         enemy["pos"] += (player_pos - enemy["pos"]).normalize() * 2    # * 2에서 숫자 2를 바꾸면 적의 속도를 바꿀 수 있음
71
72 # 그리기 함수 (배경/플레이어/적)
73 def draw_scene():
```

Ln 68, Col 25 Spaces: 4 UTF-8 CRLF {} Python ⏴ 3.13.7

1

3

17

1

八

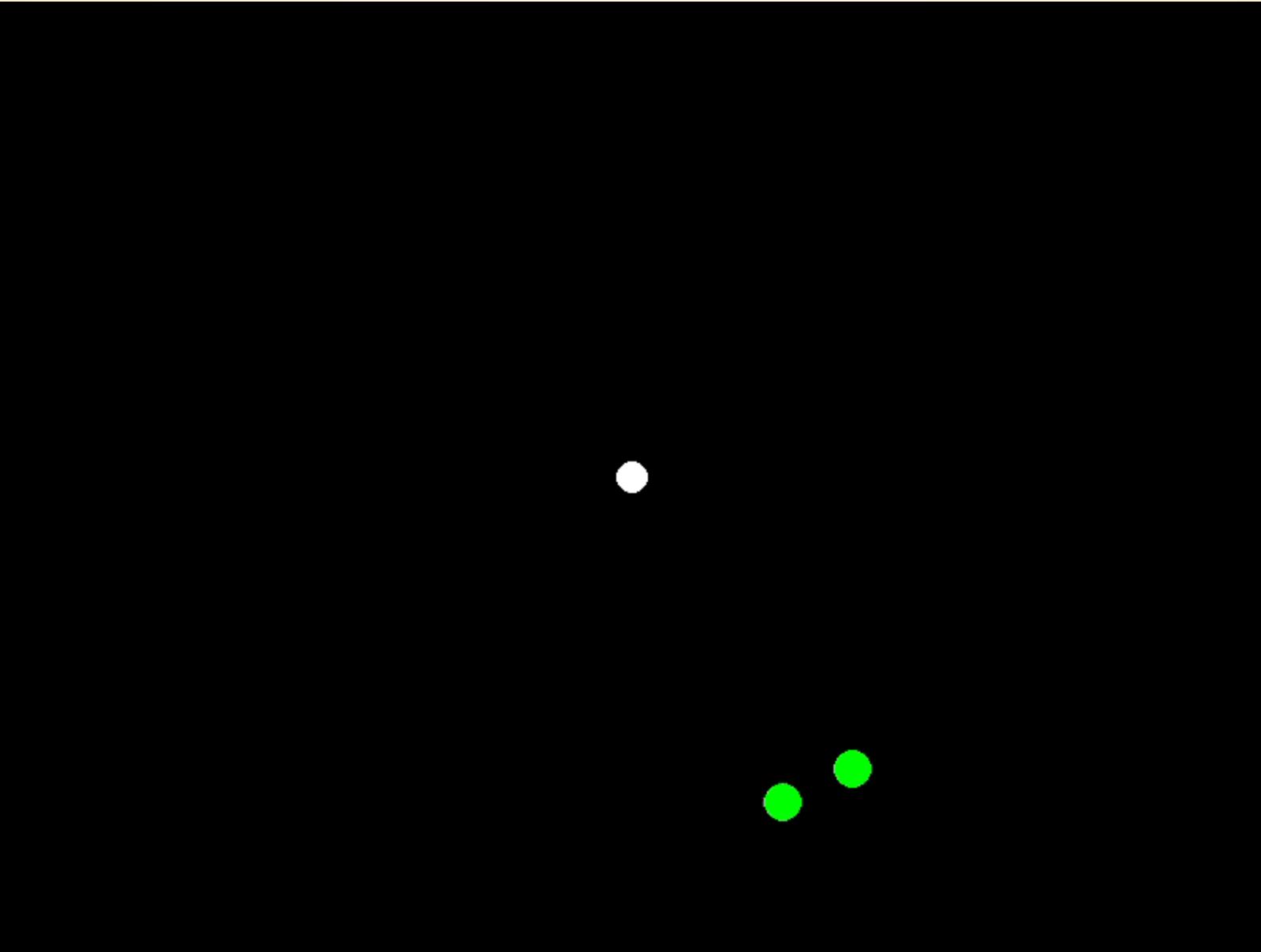
8

5

>

prac02.py





연습 문제 (3)

- 가장 가까운 적에게 자동으로 총알 발사하기

prac01.py prac02.py X

C: > Users > sunje > Desktop > Commit > prac02.py > [?] bullets

```
1 # Created by 임현우
2
3 import pygame, random
4
5 pygame.init()                                     # pygame 기능 전부 초기화(화면/입력/사운드 등 준비)
6
7 # 게임 초기화 및 기본 설정
8 screen = pygame.display.set_mode((800, 600))      # 800x600 크기의 게임 창 만들기
9 pygame.display.set_caption("본인 학번, 이름")
10 clock = pygame.time.Clock()                      # FPS(초당 프레임) 제어용 시계
11
12 # 플레이어 관련 변수
13 player_pos = pygame.Vector2(400, 300)            # 플레이어 위치(화면 중앙에서 시작)
14 player_color = (255, 255, 255)                   # 플레이어 색
15 player_speed = 3                                 # 플레이어 이동 속도
16
17 # 적 관련 변수
18 enemies = []                                     # 적 리스트
19 enemy_spawn_delay = 1000                         # 적 생성 간격 (마지막 적 생성 후 enemy_spawn_delay 만큼 기다려야 다시 생성)
20 last_spawn_time = 0                             # 마지막 적 생성 시작
21
22 # 총알 관련 변수
23 bullets = []                                     # 총알 리스트(총알을 발사하는 것 역시 객체이기 때문에)
24 shoot_delay = 1000                               # 총알 발사 간격 (마지막 발사 시작 이후 shoot_delay만큼 기다려야 다시 발사가 됨)
25 last_shot_time = 0                            # 마지막 발사 시작
26
27 # 이벤트 처리 함수 (창 닫기 등)
28 def handle_events():
29     """
30         창 닫기(X 버튼) 등 OS 이벤트를 처리.
31         QUIT 이벤트가 오면 pygame을 정리하고 프로그램 종료.
32     """
33     for event in pygame.event.get():
34         if event.type == pygame.QUIT:
```

prac01.py prac02.py X

C: > Users > sunje > Desktop > Commit > prac02.py > move_bullets

```
72     # normalize()로 길이를 1로 맞춘 뒤 * 2 해서 속도 적용
73     for enemy in enemies:
74         enemy["pos"] += (player_pos - enemy["pos"]).normalize() * 2      # * 2에서 숫자 2를 바꾸면 적의 속도를 바꿀 수 있음
75
76 # 플레이어와 적 사이 거리를 계산해주는 함수
77 def get_distance(enemy):
78     # enemy["pos"] : 적의 위치
79     # player_pos.distance_to(적 위치) --> 플레이어와 적 사이 거리(숫자) 반환
80     return player_pos.distance_to(enemy["pos"])
81
82 # 총알 생성 함수
83 def shoot_bullet(now):
84     global last_shot_time
85     if now - last_shot_time > shoot_delay:
86         if enemies:
87             target = min(enemies, key=get_distance)      # 가장 가까운 적 찾기
88             direction = (target["pos"] - player_pos).normalize()
89         else:
90             direction = pygame.Vector2(1, 0)            # 적이 없는 경우
91         bullets.append([player_pos.copy(), direction])
92         last_shot_time = now
93
94 # 이동 함수(총알)
95 def move_bullets():
96     # bullet[0] = 총알 위치, bullet[1] = 총알 방향
97     # 총알 위치를 방향 * 3 만큼 이동 (총알 속도는 3)
98     for bullet in bullets:
99         bullet[0] += bullet[1] * 3
100
101 # 그리기 함수 (배경/플레이어/적/총알)
102 def draw_scene():
103     """
104     화면을 모두 그린 뒤 마지막에 flip()으로 실제 모니터에 표시.
105     - 플레이어/적/총알을 각각 원(circle)으로 간단히 표현
106 
```



prac01.py

prac02.py X

▶ ▾ ⌂ ⌂

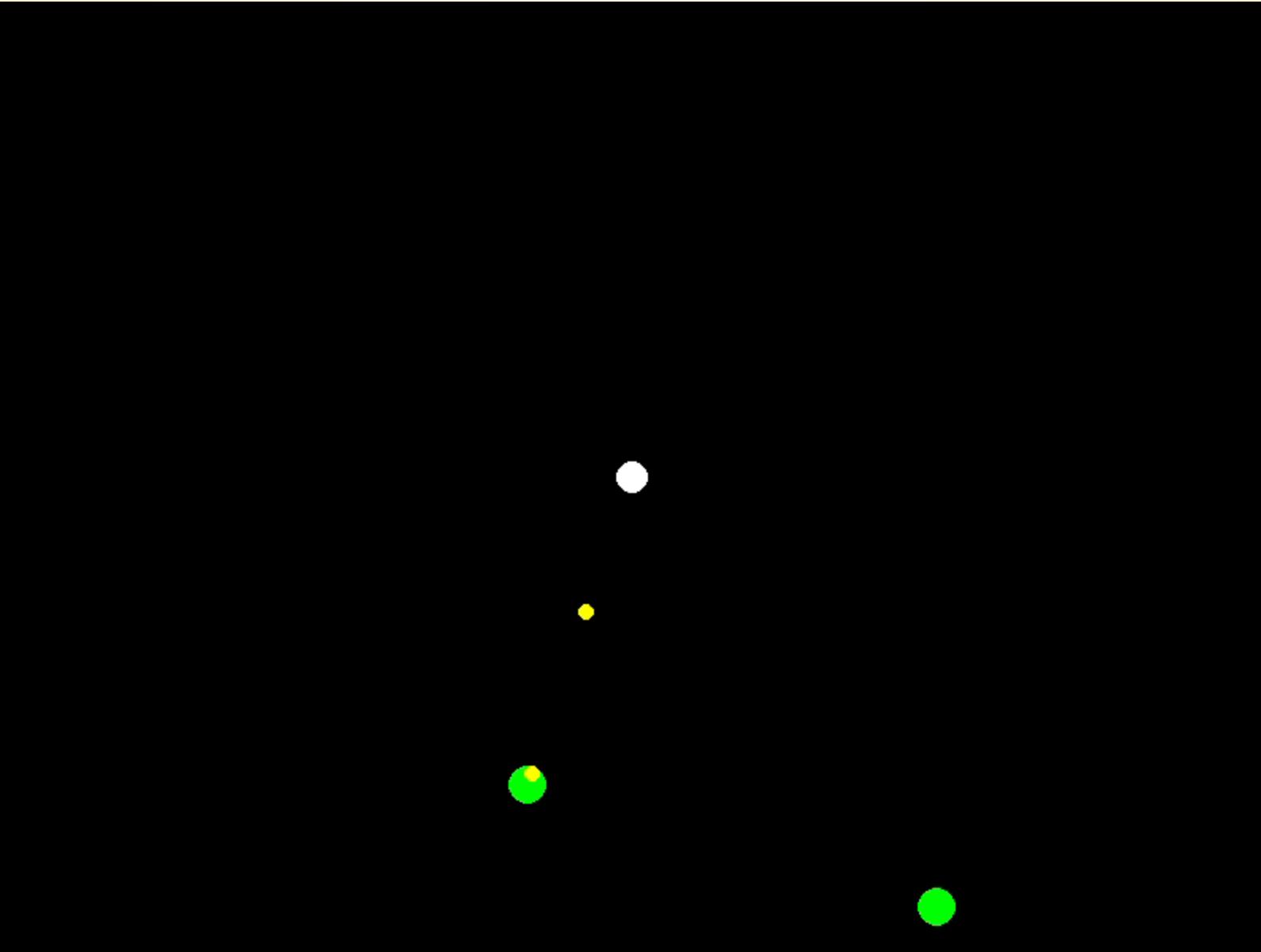


C: > Users > sunje > Desktop > Commit > prac02.py > draw_scene

```
100
101     # 그리기 함수 (배경/플레이어/적/총알)
102 def draw_scene():
103     """
104         화면을 모두 그린 뒤 마지막에 flip()으로 실제 모니터에 표시.
105         - 플레이어/적/총알을 각각 원(circle)으로 간단히 표현
106     """
107     screen.fill((0, 0, 0))                                # 배경 색상
108     pygame.draw.circle(screen, player_color, player_pos, 10)    # 플레이어 생성
109
110     for enemy in enemies:
111         pygame.draw.circle(screen, enemy["color"], enemy["pos"], 12)    # 적 객체
112     for bullet in bullets:
113         pygame.draw.circle(screen, (255, 255, 0), bullet[0], 5)        # 총알 객체
114
115     pygame.display.flip()                                    # 더블버퍼링 --> 실제 화면으로 전환
116
117     # 메인 루프, 모든 함수를 실행 시키는 부분
118     while True:
119         handle_events()                                     # 창 닫기 등 OS 이벤트 처리
120
121         keys = pygame.key.get_pressed()                  # 키보드 눌림 상태 얻기
122         move_player(keys)                             # 입력에 따라 플레이어 이동
123
124         now = pygame.time.get_ticks()                  # 현재 시각 (tick)
125         spawn_enemy(now)                            # 적 생성
126         move_enemies()                           # 적 이동
127         shoot_bullet(now)                          # 자동 사격(가까운 적 방향)
128         move_bullets()                           # 총알 이동
129
130         draw_scene()                                # 화면 그리기 + flip
131         clock.tick(60)                            # 1초에 60프레임으로 고정
```

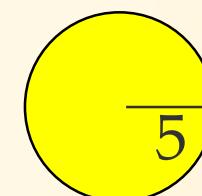
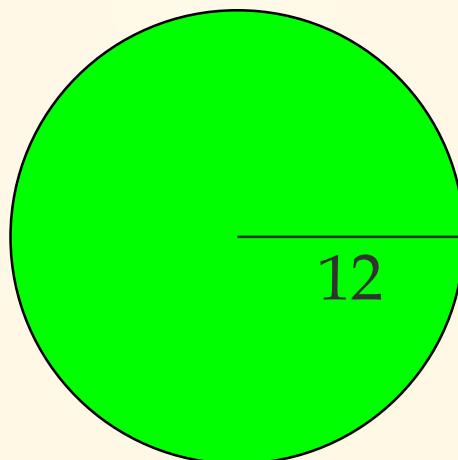
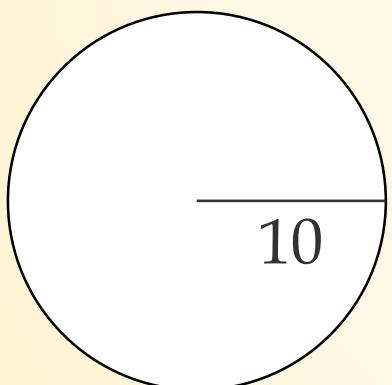


0% ⌂ ⌂ - ⌂ ⌂



연습 문제 (4)

- 충돌 처리 하기 (HP 깎기)
 - 플레이어 <-> 적
 - 적 <-> 총알





prac01.py

prac02.py X

▶ ▾ ⌂ ⌂

C: > Users > sunje > Desktop > Commit > prac02.py > spawn_enemy > hp





















































































































































































































































































































































3

1

A

6

1

>

prac01.py

prac02.py

C: > Users > sunje > Desktop > Commit >  prac02.py >  check_collisions

```
98 # 총알 위치를 방향 * 3 만큼 이동 (총알 속도는 3)
99 for bullet in bullets:
100     bullet[0] += bullet[1] * 3
101
102 # 오브젝트 충돌 처리 함수 (총알 <-> 적 | 적 <-> 플레이어)
103 def check_collisions():
104     """
105     1) 총알-적: 거리가 15 미만이면 피격 → hp 감소, 0 이하 시 경험치 오브 생성
106     2) 적-플레이어: 거리가 20 미만이면 플레이어 피격 → hp 감소, 0 이면 게임오버
107     """
108     global player_hp
109
110     # 총알-적 충돌 처리
111     for bullet in bullets:
112         for enemy in enemies:
113             if bullet[0].distance_to(enemy["pos"]) < 15:      # 총알-적 충돌 판정
114                 enemy["hp"] -= 1                                # 적 체력 1만큼 감소
115                 bullets.remove(bullet)                         # 총알 제거
116                 if enemy["hp"] <= 0:                           # 적이 죽으면
117                     enemies.remove(enemy)                      # 적 제거
118                     break                                    # 한 총알은 하나의 적만 맞춤
119
120     # 적-플레이어 충돌 처리
121     for enemy in enemies:
122         if enemy["pos"].distance_to(player_pos) < 20:    # 적-플레이어 충돌
123             enemies.remove(enemy)                          # 적 제거 (경험치 오브 생성
124             player_hp -= 1                                # 플레이어 체력 1만큼 감소
125             if player_hp <= 0:                            # 체력이 0 이라면
126                 print("GAME OVER")                       # GAME OVER 출력
127                 pygame.quit()                           # 게임 종료
128                 exit()                                 # 프로그램 종료
129
130     # 그리기 함수 (배경/플레이어/적/총알)
131     def draw_scene():
```

Ln 124, Col 22 Spaces: 4 UTF-8 CRLF {} Python 🐍 3.13.7 🔔



prac01.py

prac02.py X

▷ ▾ □ ...



```
129  
130     # 그리기 함수 (배경/플레이어/적/총알)  
131 def draw_scene():  
132     """  
133         화면을 모두 그린 뒤 마지막에 flip()으로 실제 모니터에 표시.  
134         - 플레이어/적/총알을 각각 원(circle)으로 간단히 표현  
135     """  
136     screen.fill((0, 0, 0))                                # 배경 색상  
137     pygame.draw.circle(screen, player_color, player_pos, 10)    # 플레이어 생성  
138  
139     for enemy in enemies:  
140         pygame.draw.circle(screen, enemy["color"], enemy["pos"], 12)    # 적 객체  
141     for bullet in bullets:  
142         pygame.draw.circle(screen, (255, 255, 0), bullet[0], 5)        # 총알 객체  
143  
144     pygame.display.flip()                                    # 더블버퍼링 --> 실제 화면으로 전환  
145  
146     # 메인 루프, 모든 함수를 실행 시키는 부분  
147     while True:  
148         handle_events()                                     # 창 닫기 등 OS 이벤트 처리  
149  
150         keys = pygame.key.get_pressed()                  # 키보드 눌림 상태 얻기  
151         move_player(keys)                               # 입력에 따라 플레이어 이동  
152  
153         now = pygame.time.get_ticks()                  # 현재 시각 (tick)  
154         spawn_enemy(now)                             # 적 생성  
155         move_enemies()                            # 적 이동  
156         shoot_bullet(now)                           # 자동 사격(가까운 적 방향)  
157         move_bullets()                            # 총알 이동  
158         check_collisions()                         # 충돌 처리  
159  
160         draw_scene()                                # 화면 그리기 + flip  
161         clock.tick(60)                                # 1초에 60프레임으로 고정
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\sunje> & C:/Users/sunje/AppData/Local/Programs/Python/Python313/python.exe c:/Users/sunje/Desktop/Commit/prac02.py
pygame 2.6.1 (SDL 2.28.4, Python 3.13.7)
```

Hello from the pygame community. <https://www.pygame.org/contribute.html>

GAME OVER

```
PS C:\Users\sunje>
```

연습 문제 (5)

- HP 표시하기



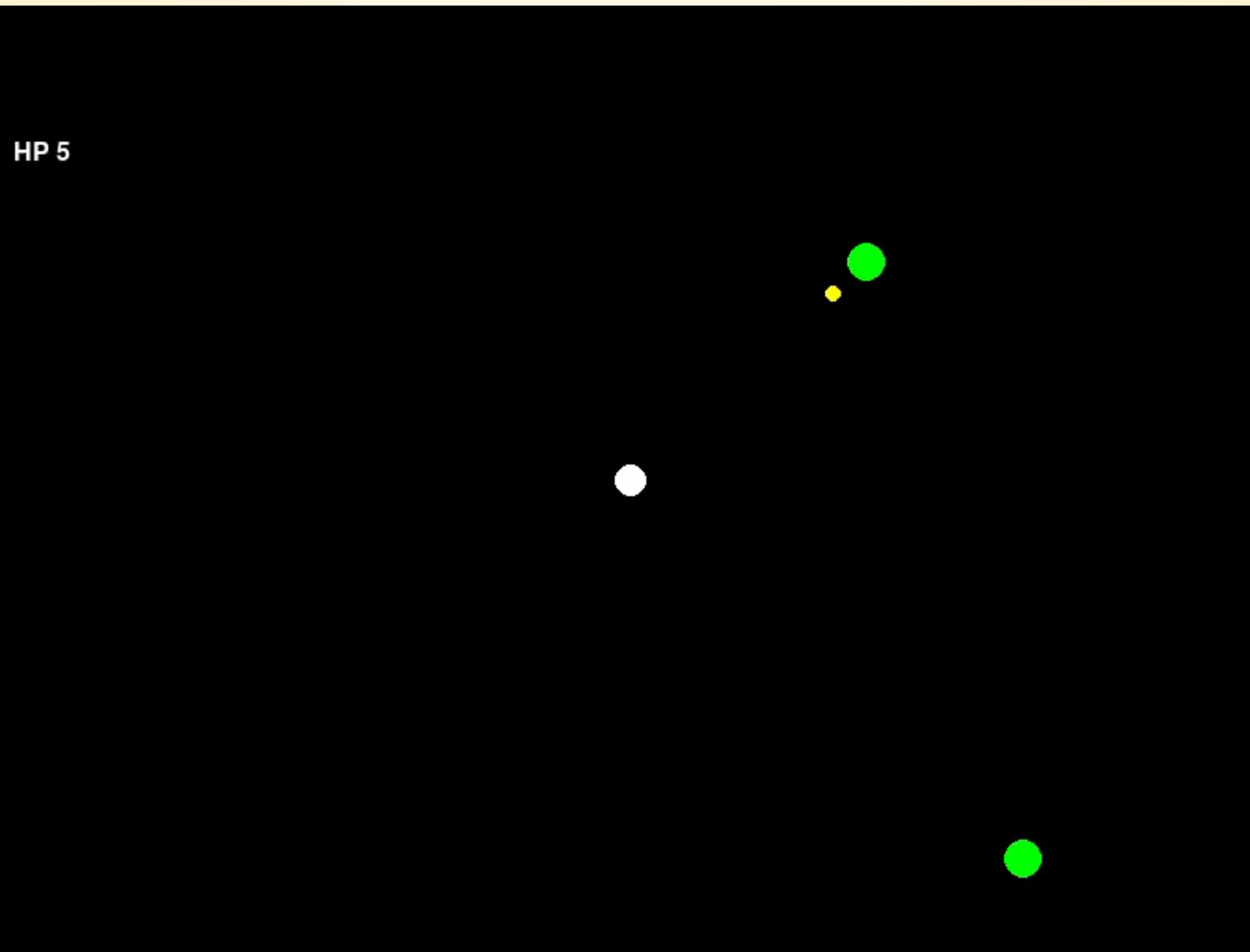
prac01.py

 prac02.py

C: > Users > sunje > Desktop > Commit >  prac02.py >  draw_scenes

```
130
131 # 그리기 함수 (배경/플레이어/적/총알/UI)
132 def draw_scene():
133     """
134     화면을 모두 그린 뒤 마지막에 flip()으로 실제 모니터에 표시.
135     - 플레이어/적/총알을 각각 원(circle)으로 간단히 표현
136     - 텍스트는 font.render로 Surface 만든 뒤 blit으로 불임
137     """
138     screen.fill((0, 0, 0)) # 배경 색상
139     pygame.draw.circle(screen, player_color, player_pos, 10) # 플레이어 생성
140
141     for enemy in enemies:
142         pygame.draw.circle(screen, enemy["color"], enemy["pos"], 12) # 적 객체
143     for bullet in bullets:
144         pygame.draw.circle(screen, (255, 255, 0), bullet[0], 5) # 총알 객체
145
146     # UI를 그리는 부분(font.render(표시될 글씨, 안티앨리어싱 여부(True), 색상), (x,y)글씨 위치 좌표)
147     screen.blit([font.render("HP " + str(player_hp), True, (255,255,255)), (10, 85)])
148
149     pygame.display.flip() # 더블버퍼링 --> 실제 화면으로 전환
150
151 # 메인 루프, 모든 함수를 실행 시키는 부분
152 while True:
153     handle_events() # 창 닫기 등 OS 이벤트 처리
154
155     keys = pygame.key.get_pressed() # 키보드 눌림 상태 얻기
156     move_player(keys) # 입력에 따라 플레이어 이동
157
158     now = pygame.time.get_ticks() # 현재 시각 (tick)
159     spawn_enemy(now) # 적 생성
160     move_enemies() # 적 이동
161     shoot_bullet(now) # 자동 사격(가까운 적 방향)
162     move_bullets() # 총알 이동
163     check_collisions() # 충돌 처리
```





게임 제작 (3)

1. 화면 구성 및 캐릭터 이동
2. 맵 생성 및 충돌 처리
3. 적 생성 및 전투 시스템
4. 아이템 생성 및 게임 완성

- **아이템 생성 및 사용** – 게임 내에서 사용할 수 있는 아이템을 생성하고, 플레이어가 아이템을 획득하고 사용할 수 있도록 구현
- **게임 완성 및 테스트** – 게임을 테스트하고 버그를 수정하여 완성 (게임의 난이도 밸런싱으로 조정 가능)

1

prac01.py

prac02.py

C: > Users > sunje > Desktop > Commit >  prac02.py > [?] item

```
1 # Created by 임현우
2
3 import pygame, random
4
5 pygame.init()                                     # pygame 기능 전부 초기화(화면/입력/사운드 등 준비)
6
7 # 게임 초기화 및 기본 설정
8 screen = pygame.display.set_mode((800, 600))      # 800x600 크기의 게임 창 만들기
9 pygame.display.set_caption("본인 학번, 이름")
10 clock = pygame.time.Clock()                      # FPS(초당 프레임) 제어용 시계
11 font = pygame.font.SysFont(None, 24)              # 기본 글꼴(크기 24)
12
13 # 플레이어 관련 변수
14 player_pos = pygame.Vector2(400, 300)            # 플레이어 위치(화면 중앙에서 시작)
15 player_color = (255, 255, 255)                   # 플레이어 색
16 player_speed = 3                                 # 플레이어 이동 속도
17 player_hp = 5                                   # 플레이어 체력
18
19 # 적 관련 변수
20 enemies = []                                     # 적 리스트
21 enemy_spawn_delay = 1000                         # 적 생성 간격 (마지막 적 생성 후 enemy_spawn_delay 만큼 기다려야 다시 생성)
22 last_spawn_time = 0                             # 마지막 적 생성 시각
23
24 # 총알 관련 변수
25 bullets = []                                     # 총알 리스트(총알을 발사하는 것 역시 객체이기 때문에)
26 shoot_delay = 1000                               # 총알 발사 간격 (마지막 발사 이후 shoot_delay만큼 기다려야 다시 발사가 됨)
27 last_shot_time = 0                            # 마지막 발사 시각
28
29 # 경험치 오브젝트 관련 변수
30 items = []                                       # 아이템 리스트(적을 잡은 후 나오는 경험치 오브젝트들도 여러개 이기 때문)
31
32 # 이벤트 처리 함수 (창 닫기 등)
33 def handle_events():
34     """
```

8

10

3

3

1

prac01.py

prac02.py

▷ ⌂ ⏷ ...

C: > Users > sunje > Desktop > Commit > prac02.py > check_collision

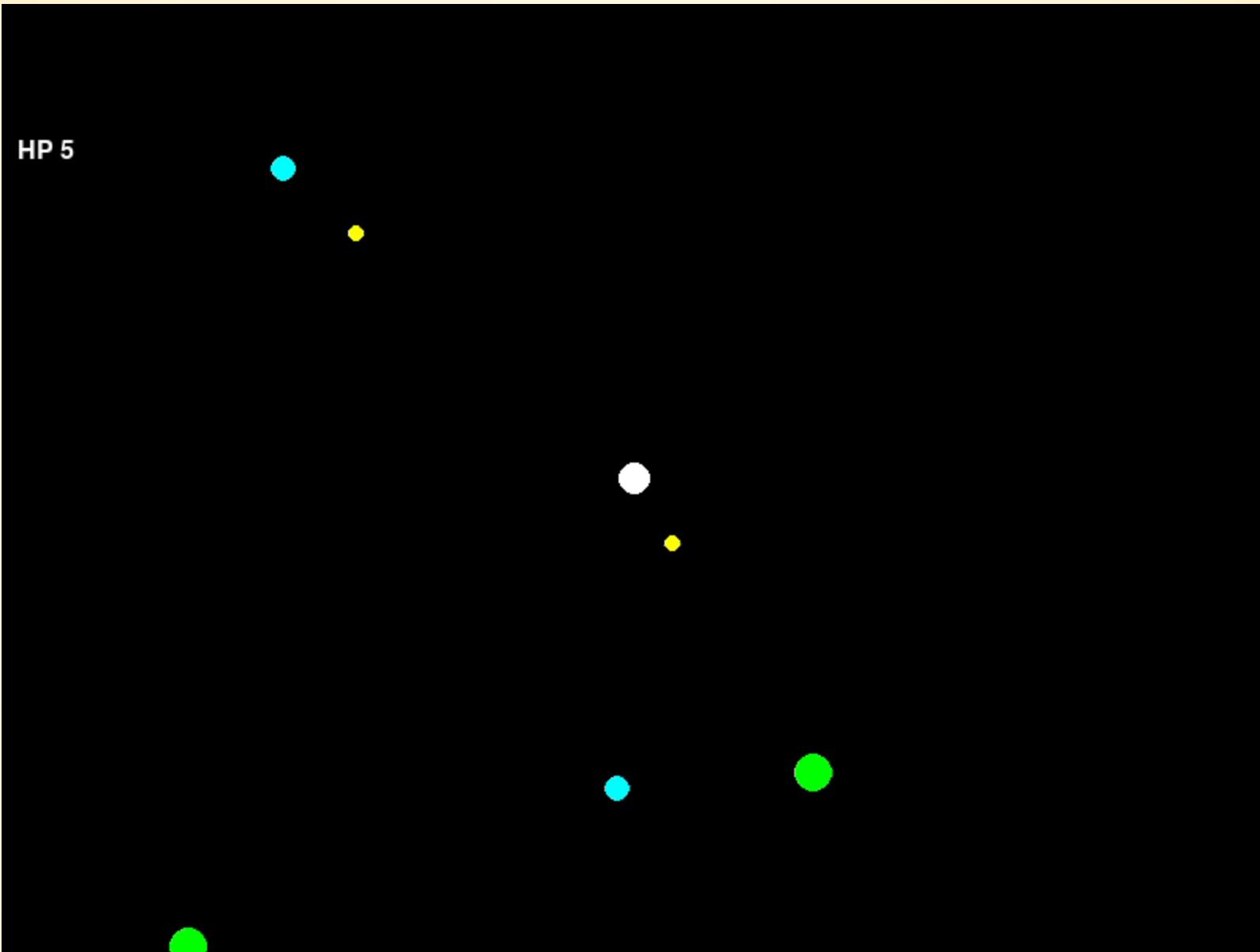
```
105
106 # 오브젝트 충돌 처리 함수 (총알 <-> 적 | 적 <-> 플레이어)
107 def check_collisions():
108     """
109     1) 총알-적: 거리가 15 미만이면 피격 → hp 감소, 0 이하 시 경험치 오브 생성
110     2) 적-플레이어: 거리가 20 미만이면 플레이어 피격 → hp 감소, 0 이면 게임오버
111     """
112     global player_hp
113
114     # 총알-적 충돌 처리
115     for bullet in bullets:
116         for enemy in enemies:
117             if bullet[0].distance_to(enemy["pos"]) < 15:      # 총알-적 충돌 판정
118                 enemy["hp"] -= 1                                # 적 체력 1만큼 감소
119                 bullets.remove(bullet)                      # 총알 제거
120                 if enemy["hp"] <= 0:                         # 적이 죽으면
121                     items.append({                           # 경험치 오브 생성 위치 = 적의 위치 값
122                         "pos": enemy["pos"].copy(),           # 경험치 오브 생성 위치 = 적의 위치 값
123                         "value": enemy["exp"]})            # 경험치 값
124
125                     enemies.remove(enemy)          # 적 제거
126                     break                   # 한 총알은 하나의 적만 맞춤
127
128     # 적-플레이어 충돌 처리
129     for enemy in enemies:
130         if enemy["pos"].distance_to(player_pos) < 20:    # 적-플레이어 충돌
131             enemies.remove(enemy)                        # 적 제거 (경험치 오브 생성 X)
132             player_hp -= 1                            # 플레이어 체력 1만큼 감소
133             if player_hp <= 0:                         # 체력이 0 이라면
134                 print("GAME OVER")                    # GAME OVER 출력
135                 pygame.quit()                      # 게임 종료
136                 exit()                          # 프로그램 종료
137
138     # 그리기 함수 (배경/플레이어/적/총알/UI)
```

File Edit Selection View Go Run ... ⏪ ⏩ Search

C: > Users > sunje > Desktop > Commit > prac02.py > draw_scene

```
137
138     # 그리기 함수 (배경/플레이어/적/총알/UI)
139     def draw_scene():
140         """
141             화면을 모두 그린 뒤 마지막에 flip()으로 실제 모니터에 표시.
142             - 플레이어/적/총알을 각각 원(circle)으로 간단히 표현
143             - 텍스트는 font.render로 Surface 만든 뒤 blit 으로 붙임
144         """
145         screen.fill((0, 0, 0))                                # 배경 색상
146         pygame.draw.circle(screen, player_color, player_pos, 10)    # 플레이어 생성
147
148         for enemy in enemies:
149             pygame.draw.circle(screen, enemy["color"], enemy["pos"], 12)    # 적 객체
150         for bullet in bullets:
151             pygame.draw.circle(screen, (255, 255, 0), bullet[0], 5)        # 총알 객체
152         for item in items:
153             pygame.draw.circle(screen, (0, 255, 255), item["pos"], 8)      # 경험치 객체
154
155         # UI를 그리는 부분(font.render(표시될 글씨, 안티앨리어싱 여부(True), 색상), (x,y)글씨 위치 좌표)
156         screen.blit(font.render("HP " + str(player_hp), True, (255,255,255)), (10, 85))
157
158         pygame.display.flip()                                     # 더블버퍼링 --> 실제 화면으로 전환
159
160     # 메인 루프, 모든 함수를 실행 시키는 부분
161     while True:
162         handle_events()                                      # 창 닫기 등 OS 이벤트 처리
163
164         keys = pygame.key.get_pressed()                      # 키보드 눌림 상태 얻기
165         move_player(keys)                                  # 입력에 따라 플레이어 이동
166
167         now = pygame.time.get_ticks()                      # 현재 시각 (tick)
168         spawn_enemy(now)                                 # 적 생성
169         move_enemies()                                  # 적 이동
170         shoot_bullet(now)                               # 자동 사격(가까운 적 방향)
```

x ⊗ 0 △ 0 Ln 152, Col 22 Spaces: 4 UTF-8 CRLF {} Python 3.13.7



연습 문제 (6)

- 게임 난이도
 - 경험치(아이템)를 모아 레벨업
 - 레벨업 할 때마다 총알 발사 딜레이를 줄임

File Edit Selection View Go Run ... ⏪ ⏩ Search

C: > Users > sunje > Desktop > Commit > prac02.py > check_collisions

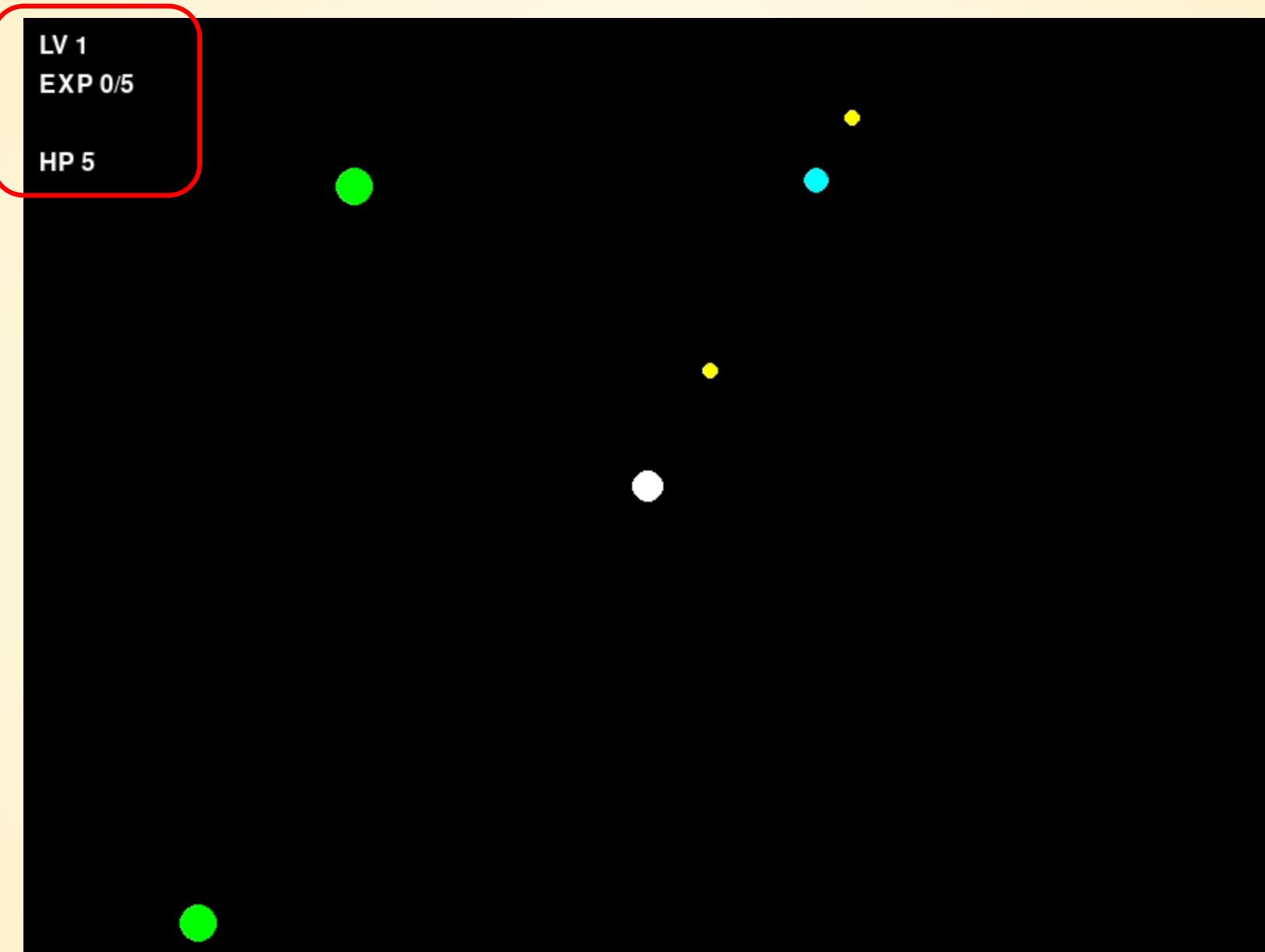
```
129
130     # 오브젝트 충돌 처리 함수 (총알 <-> 적 | 적 <-> 플레이어 | 아이템 <-> 플레이어)
131     def check_collisions():
132         """
133             1) 총알-적: 거리가 15 미만이면 피격 → hp 감소, 0 이하 시 경험치 오브 생성
134             2) 적-플레이어: 거리가 20 미만이면 플레이어 피격 → hp 감소, 0 이면 게임오버
135             3) 아이템-플레이어: 거리가 15 미만이면 경험치 획득 → 필요 경험치 달성 시 레벨업
136         """
137         global player_hp, current_exp
138
139         # 총알-적 충돌 처리
140         for bullet in bullets:
141             for enemy in enemies:
142                 if bullet[0].distance_to(enemy["pos"]) < 15:          # 총알-적 충돌 판정
143                     enemy["hp"] -= 1                                    # 적 체력 1만큼 감소
144                     bullets.remove(bullet)                            # 총알 제거
145                     if enemy["hp"] <= 0:                                # 적이 죽으면
146                         items.append({                                     # 경험치 오브 생성 위치 = 적의 위치 값을 복사해 경험치 오브 위치에 넣어줌
147                             "pos": enemy["pos"].copy(),                  # 경험치 값
148                             "value": enemy["exp"]}
149                         )
150                         enemies.remove(enemy)                      # 적 제거
151                         break                                         # 한 총알은 하나의 적만 맞춤
152
153         # 적-플레이어 충돌 처리
154         for enemy in enemies:
155             if enemy["pos"].distance_to(player_pos) < 20:      # 적-플레이어 충돌
156                 enemies.remove(enemy)                          # 적 제거 (경험치 오브 생성 X)
157                 player_hp -= 1                                # 플레이어 체력 1만큼 감소
158                 if player_hp <= 0:                            # 체력이 0 이라면
159                     print("GAME OVER")
160                     pygame.quit()
161                     exit()
162
```

Ln 137, Col 34 Spaces: 4 UTF-8 CRLF {} Python 3.13.7

File Edit Selection View Go Run ... ⏪ ⏩ ⏴ ⏵ Search

C: > Users > sunje > Desktop > Commit > prac02.py > check_collisions

```
162
163     # 아이템-플레이어 충돌 처리
164     for item in items:
165         if item["pos"].distance_to(player_pos) < 15:          # 아이템-플레이어 충돌
166             current_exp += item["value"]                         # 경험치 증가
167             items.remove(item)                                 # 경험치 오브젝트 제거
168             if current_exp >= needed_exp:                   # 필요 경험치 도달
169                 level_up()                                  # 레벨업 함수 실행
170
171     # 그리기 함수 (배경/플레이어/적/총알/UI)
172     def draw_scene():
173         """
174             화면을 모두 그린 뒤 마지막에 flip()으로 실제 모니터에 표시.
175             - 플레이어/적/총알을 각각 원(circle)으로 간단히 표현
176             - 텍스트는 font.render로 Surface 만든 뒤 blit 으로 붙임
177         """
178         screen.fill((0, 0, 0))                                # 배경 색상
179         pygame.draw.circle(screen, player_color, player_pos, 10)    # 플레이어 생성
180
181         for enemy in enemies:
182             pygame.draw.circle(screen, enemy["color"], enemy["pos"], 12)    # 적 객체
183         for bullet in bullets:
184             pygame.draw.circle(screen, (255, 255, 0), bullet[0], 5)        # 총알 객체
185         for item in items:
186             pygame.draw.circle(screen, (0, 255, 255), item["pos"], 8)      # 경험치 객체
187
188         # UI를 그리는 부분(font.render(표시될 글씨, 안티앨리어싱 여부(True), 색상), (x,y)글씨 위치 좌표)
189         screen.blit(font.render("LV " + str(level), True, (255,255,255)), (10, 10))
190         screen.blit(font.render("EXP " + str(current_exp) + "/" + str(needed_exp), True, (255,255,255)), (10, 35))
191         screen.blit(font.render("HP " + str(player_hp), True, (255,255,255)), (10, 85))
192
193         pygame.display.flip()                                # 더블버퍼링 --> 실제 화면으로 전환
194
195     # 메인 루프, 모든 함수를 실행 시키는 부분
```



연습 문제 (7)

- 게임 난이도
 - 경과시간이 길어질수록 강한 적들이 등장함

File Edit Selection View Go Run ... ⏪ ⏩ Search

C: > Users > sunje > Desktop > Commit > prac02.py > ...

```
1 # Created by 임현우
2
3 import pygame, random, time
4
5 pygame.init()                                     # pygame 기능 전부 초기화(화면/입력/사운드 등 준비)
6
7 # 게임 초기화 및 기본 설정
8 screen = pygame.display.set_mode((800, 600))      # 800x600 크기의 게임 창 만들기
9 pygame.display.set_caption("본인 학번, 이름")
10 clock = pygame.time.Clock()                      # FPS(초당 프레임) 제어용 시계
11 font = pygame.font.SysFont(None, 24)              # 기본 글꼴(크기 24)
12
13 # 플레이어 관련 변수
14 player_pos = pygame.Vector2(400, 300)            # 플레이어 위치(화면 중앙에서 시작)
15 player_color = (255, 255, 255)                   # 플레이어 색
16 player_speed = 3                                 # 플레이어 이동 속도
17 player_hp = 5                                    # 플레이어 체력
18
19 # 레벨/경험치 변수
20 level = 1                                       # 시작 레벨
21 current_exp = 0                                # 누적된 경험치
22 needed_exp = 5                                # 초기 레벨업에 필요한 경험치량(다른 함수에서 값이 증가)
23
24 # 적 관련 변수
25 enemies = []                                    # 적 리스트
26 enemy_spawn_delay = 1000                        # 적 생성 간격 (마지막 적 생성 후 enemy_spawn_delay 만큼 기다려야 다시 생성)
27 last_spawn_time = 0                            # 마지막 적 생성 시각
28
29 # 총알 관련 변수
30 bullets = []                                    # 총알 리스트(총알을 발사하는 것 역시 객체이기 때문에)
31 shoot_delay = 1000                            # 총알 발사 간격 (마지막 발사 시각 이후 shoot_delay 만큼 기다려야 다시 발사가 됨)
32 last_shot_time = 0                            # 마지막 발사 시각
33
34 # 경험치 오브젝트 관련 변수
```

Ln 3, Col 28 Spaces: 4 UTF-8 CRLF {} Python 3.13.7

The screenshot shows a Python code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** Search
- Toolbar:** Includes icons for file operations like Open, Save, Find, and Run.
- Left Sidebar:** Icons for Selection, View, Go, Run, and Help.
- Current File:** prac02.py (active tab)
- Project Path:** C: > Users > sunje > Desktop > Commit > prac02.py > ...
- Code Content:**

```
34     # 경험치 오브젝트 관련 변수
35     items = []                                # 아이템 리스트(적을 잡은 후 나오는 경험치 오브젝트들도 여러개 이기 때문)
36
37     # 시간 관련(게임 경과 시간 계산용)
38     start_time = time.time()                   # 레벨업 함수
39
40     def level_up():
41         """
42             플레이어가 필요한 경험치를 채웠을 때 호출.
43             - 레벨 +1
44             - 현재 경험치 초기화
45             - 다음 레벨까지 필요한 경험치 증가
46             - 플레이어 이동 속도 증가
47             - 총알 발사 간격 감소(= 더 빨리 씀, 최소 200ms)
48
49         """
50
51         # 밖에 있는 변수들에서 게임이 경과함(레벨업 등)에 따라 조건값을 바꿔주기 위해 global 변수로 재선언
52         global level, current_exp, needed_exp, player_speed, shoot_delay
53         level += 1
54         current_exp = 0
55         needed_exp += 3
56         player_speed += 1
57         shoot_delay = max(200, shoot_delay - 150)    # 레벨업 할때마다 총알 발사 딜레이를 150씩 줄임(1000-150), 단 200 아래로 떨어지지 않도록 설정
58
59     # 이벤트 처리 함수 (창 닫기 등)
60     def handle_events():
61         """
62             창 닫기(X 버튼) 등 OS 이벤트를 처리.
63             QUIT 이벤트가 오면 pygame을 정리하고 프로그램 종료.
64         """
65         for event in pygame.event.get():
66             if event.type == pygame.QUIT:
67                 pygame.quit()
68                 exit()
```

- Right Panel:** Shows multiple tabs of code, likely from other files or tabs in the editor.
- Bottom Status Bar:** Ln 38, Col 18, Spaces: 4, UTF-8, CRLF, {}, Python, 3.13.7, ⏺

 prac01.py

 prac02.py 

▷ ⌂ ⏺ ...

C: > Users > sunje > Desktop > Commit > prac02.py > spawn_enemy > elapsed_time

```
68
69 # 입력 처리 함수(플레이어 이동)
70 def move_player(keys):
71     """
72         키보드 입력(WASD)에 따라 플레이어 위치를 바꿈.
73         좌표는 Vector2라서 .x, .y 로 접근해 더하고 빼면 됨.
74     """
75     if keys[pygame.K_w]: player_pos.y -= player_speed
76     if keys[pygame.K_s]: player_pos.y += player_speed
77     if keys[pygame.K_a]: player_pos.x -= player_speed
78     if keys[pygame.K_d]: player_pos.x += player_speed
79
80 # 적 생성 함수
81 def spawn_enemy(now, elapsed_time):
```

```
81     def spawn_enemy(now, elapsed_time)
```

82 """"
83 일정 간격(enemy_spawn_delay)마다 적 1마리 생성.
84 - 적은 랜덤 위치에서 등장함
85 - 경과시간이 길어질수록 강한 적이 등장할 확률 존재
86 """"

```
86
87     global last_spawn_time
88     if now - last_spawn_time > enemy_spawn_delay:
89         # 초록색 적
90         hp, exp, color = 1, 1, (0, 255, 0)
```

```
93     if elapsed_time > 15 and random.random()  
94         hp, exp, color = 3, 3, (255, 168, 0)
```

게임 시간 30초 이후: 20% 확률로 빨강(hp=5, e

```
if elapsed_time > 30 and random.random()
```

```
98     hp, exp, color = 5, 5, (255, 0, 0)
```

```
100     # 랜덤 위치에서 적이 생성되도록 하는 코드  
101     pos = pygame.Vector2(random.randint(0, 800), random.choice([-20, 620]))
```



File Edit Selection View Go Run ... ⏪ ⏩ ⏴ ⏵ Search

C: > Users > sunje > Desktop > Commit > prac02.py > draw_scene > elapsed_time

```
181
182     # 그리기 함수 (배경/플레이어/적/총알/UI)
183 def draw_scene(elapsed_time):
184     """
185         화면을 모두 그린 뒤 마지막에 flip()으로 실제 모니터에 표시.
186         - 플레이어/적/총알을 각각 원(circle)으로 간단히 표현
187         - 텍스트는 font.render로 Surface 만든 뒤 blit 으로 붙임
188     """
189     screen.fill((0, 0, 0))                                # 배경 색상
190     pygame.draw.circle(screen, player_color, player_pos, 10)    # 플레이어 생성
191
192     for enemy in enemies:
193         pygame.draw.circle(screen, enemy["color"], enemy["pos"], 12)    # 적 객체
194     for bullet in bullets:
195         pygame.draw.circle(screen, (255, 255, 0), bullet[0], 5)        # 총알 객체
196     for item in items:
197         pygame.draw.circle(screen, (0, 255, 255), item["pos"], 8)      # 경험치 객체
198
199     # UI를 그리는 부분(font.render(표시될 글씨, 안티앨리어싱 여부(True), 색상), (x,y)글씨 위치 좌표)
200     screen.blit(font.render("LV " + str(level), True, (255,255,255)), (10, 10))
201     screen.blit(font.render("EXP " + str(current_exp) + "/" + str(needed_exp), True, (255,255,255)), (10, 35))
202     screen.blit(font.render("TIME " + str(elapsed_time) + "s", True, (255,255,255)), (10, 60))
203     screen.blit(font.render("HP " + str(player_hp), True, (255,255,255)), (10, 85))
204
205     pygame.display.flip()                                    # 더블버퍼링 --> 실제 화면으로 전환
206
207     # 메인 루프, 모든 함수를 실행 시키는 부분
208     while True:
209         handle_events()                                  # 창 닫기 등 OS 이벤트 처리
210
211         keys = pygame.key.get_pressed()                # 키보드 눌림 상태 얻기
212         move_player(keys)                            # 입력에 따라 플레이어 이동
213
214         now = pygame.time.get_ticks()                 # 현재 시각 (tick)
```

x ⊗ 0 △ 0 Ln 183, Col 28 Spaces: 4 UTF-8 CRLF {} Python 3.13.7

2

prac01.py

 prac02.py X

▷ ⌂ ⏷ ...

LV 1

EXP 0/5

TIME 5s

HP 5

