



PyRain

워드

구본아 문동규 이혜진 홍순기

SUPER
www.super.com



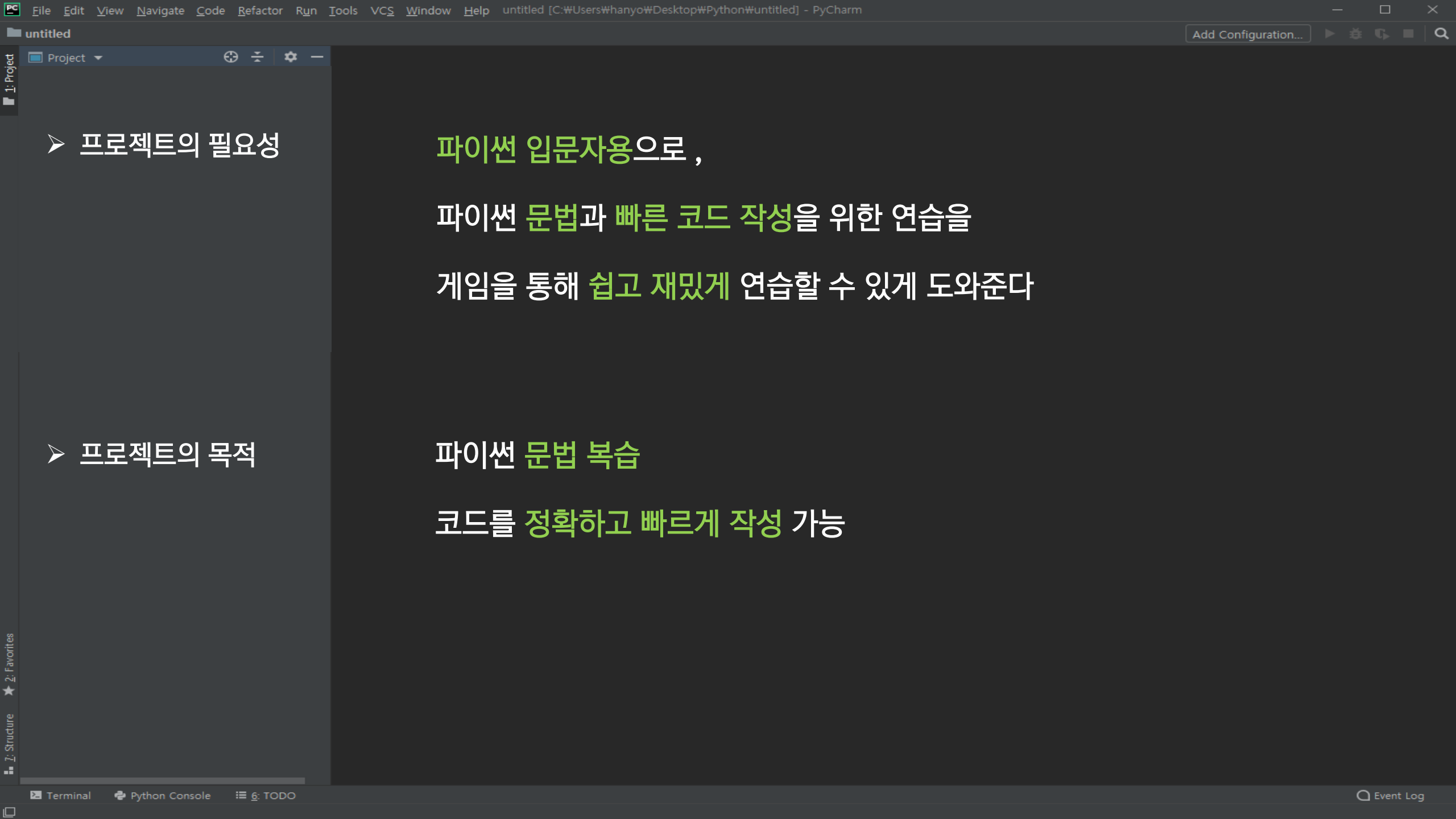
Welcome to PyCharm



목차

- + 프로젝트 제안 필요성 및 목적
 - 📁 프로젝트 구현 범위
 - 📁 프로젝트의 장점과 개선점(기존, 향후)
- + 과제 세부 내용
 - 📁 업무 분담 및 추진 일정
 - 📁 프로젝트 코드 분석 및 결과

⚙️ Configure ▼ Get Help ▼



➤ 프로젝트의 필요성

파이썬 입문자용으로 ,

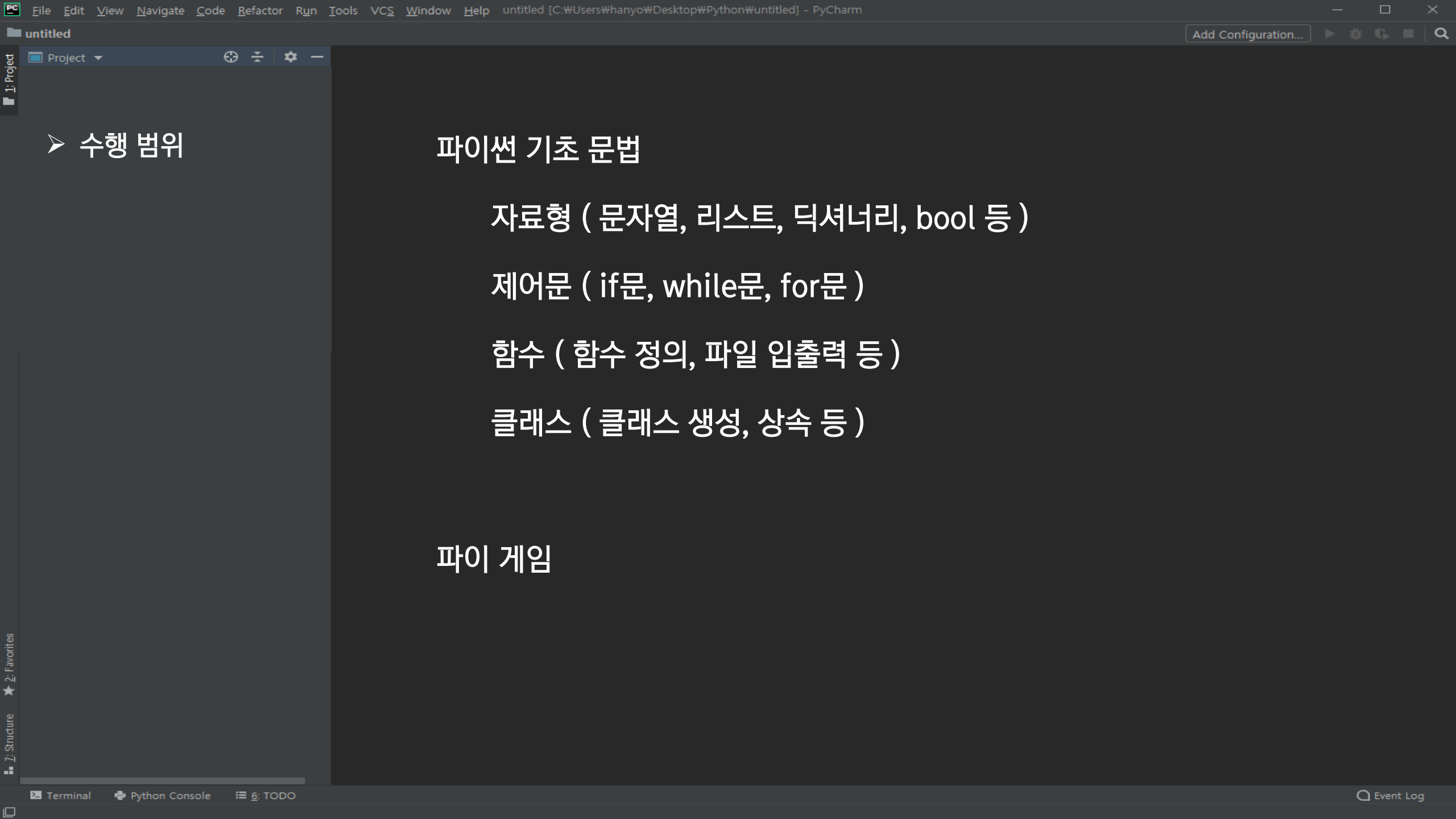
파이썬 문법과 빠른 코드 작성을 위한 연습을

게임을 통해 쉽고 재밌게 연습할 수 있게 도와준다

➤ 프로젝트의 목적

파이썬 문법 복습

코드를 정확하고 빠르게 작성 가능



➤ 수행 범위

파이썬 기초 문법

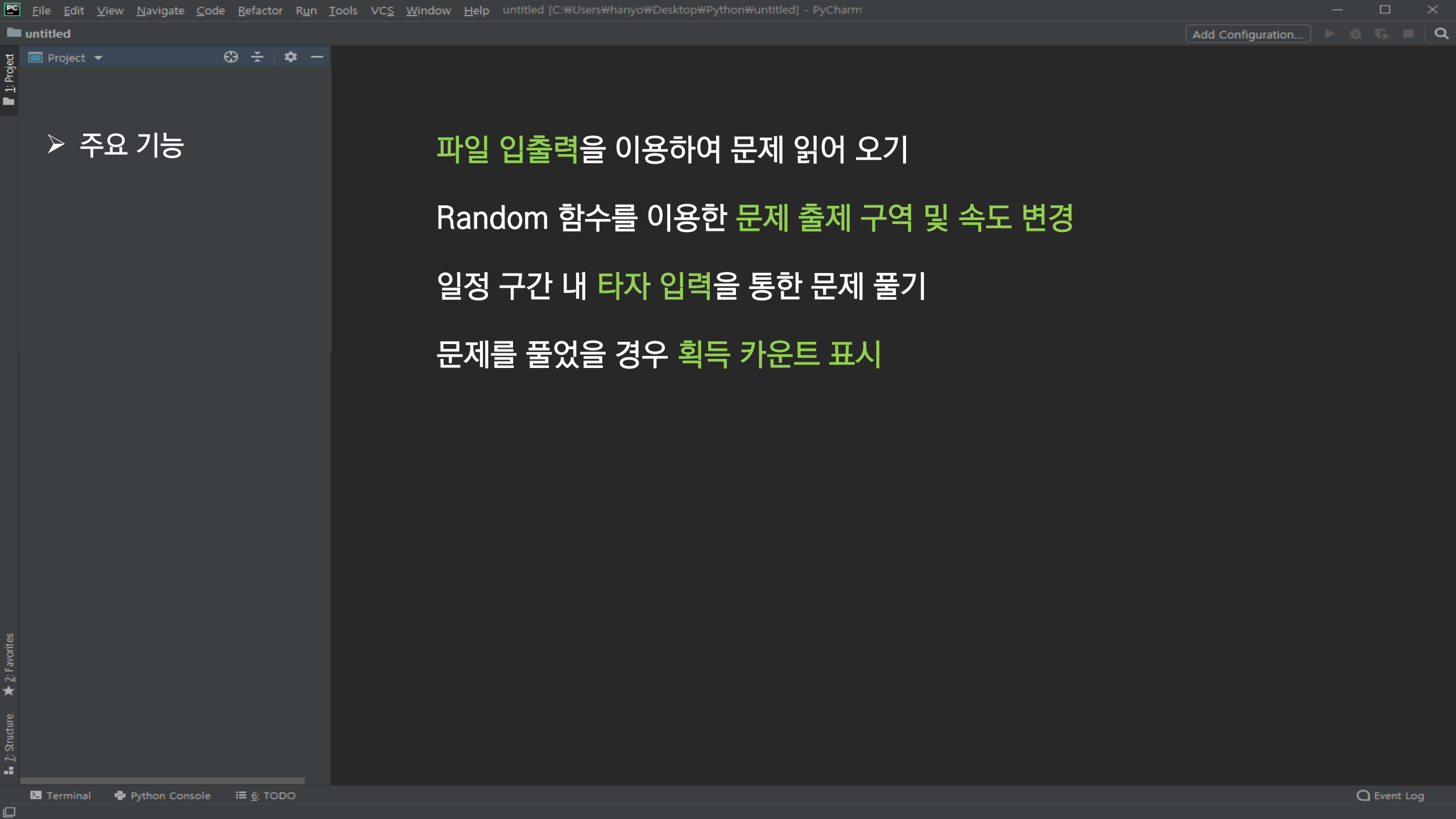
자료형 (문자열, 리스트, 딕셔너리, bool 등)

제어문 (if문, while문, for문)

함수 (함수 정의, 파일 입출력 등)

클래스 (클래스 생성, 상속 등)

파이 게임



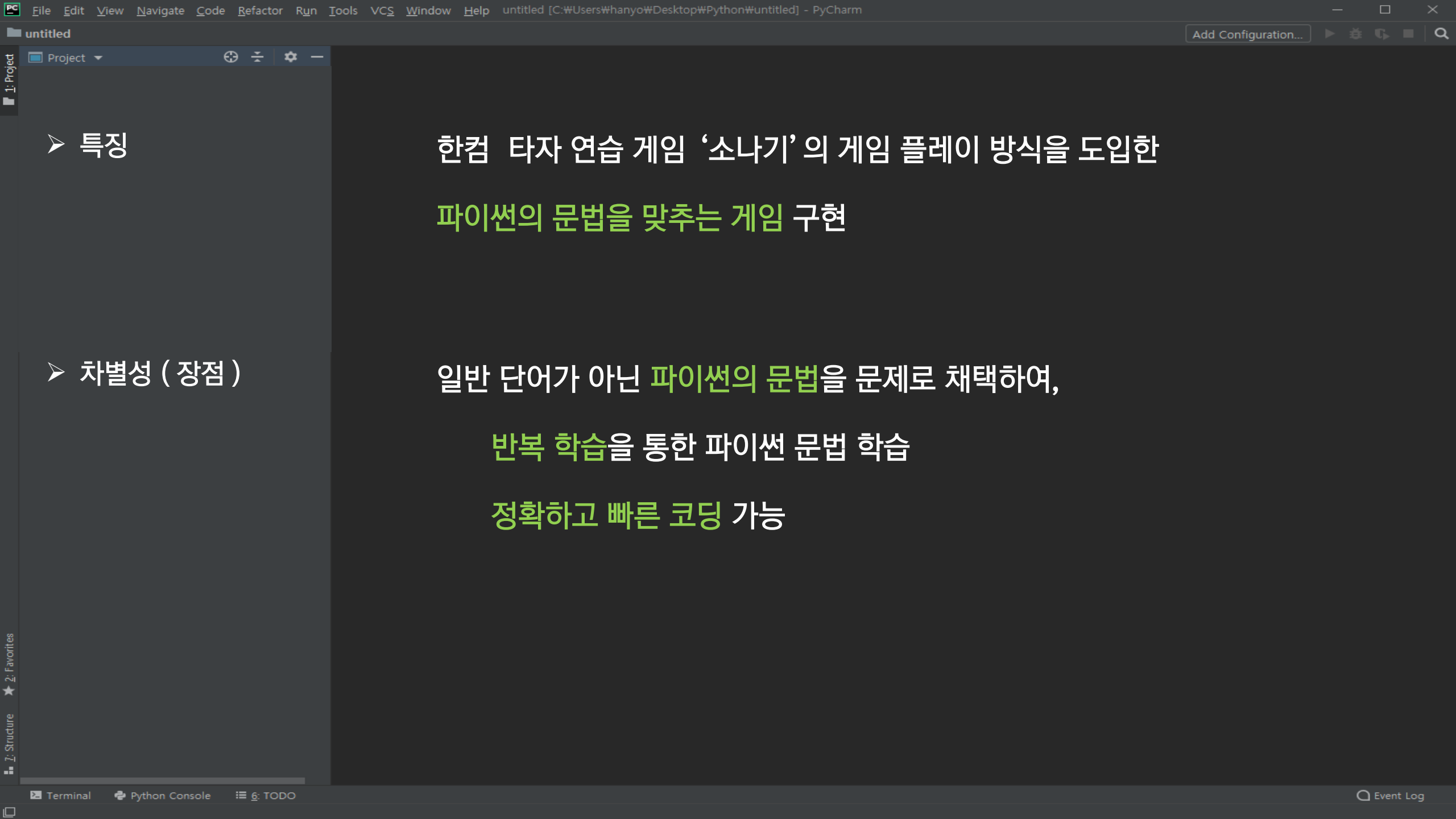
➤ 주요 기능

파일 입출력을 이용하여 문제 읽어 오기

Random 함수를 이용한 문제 출제 구역 및 속도 변경

일정 구간 내 타자 입력을 통한 문제 풀기

문제를 풀었을 경우 획득 카운트 표시



➤ 특징

한컴 타자 연습 게임 ‘소나기’의 게임 플레이 방식을 도입한

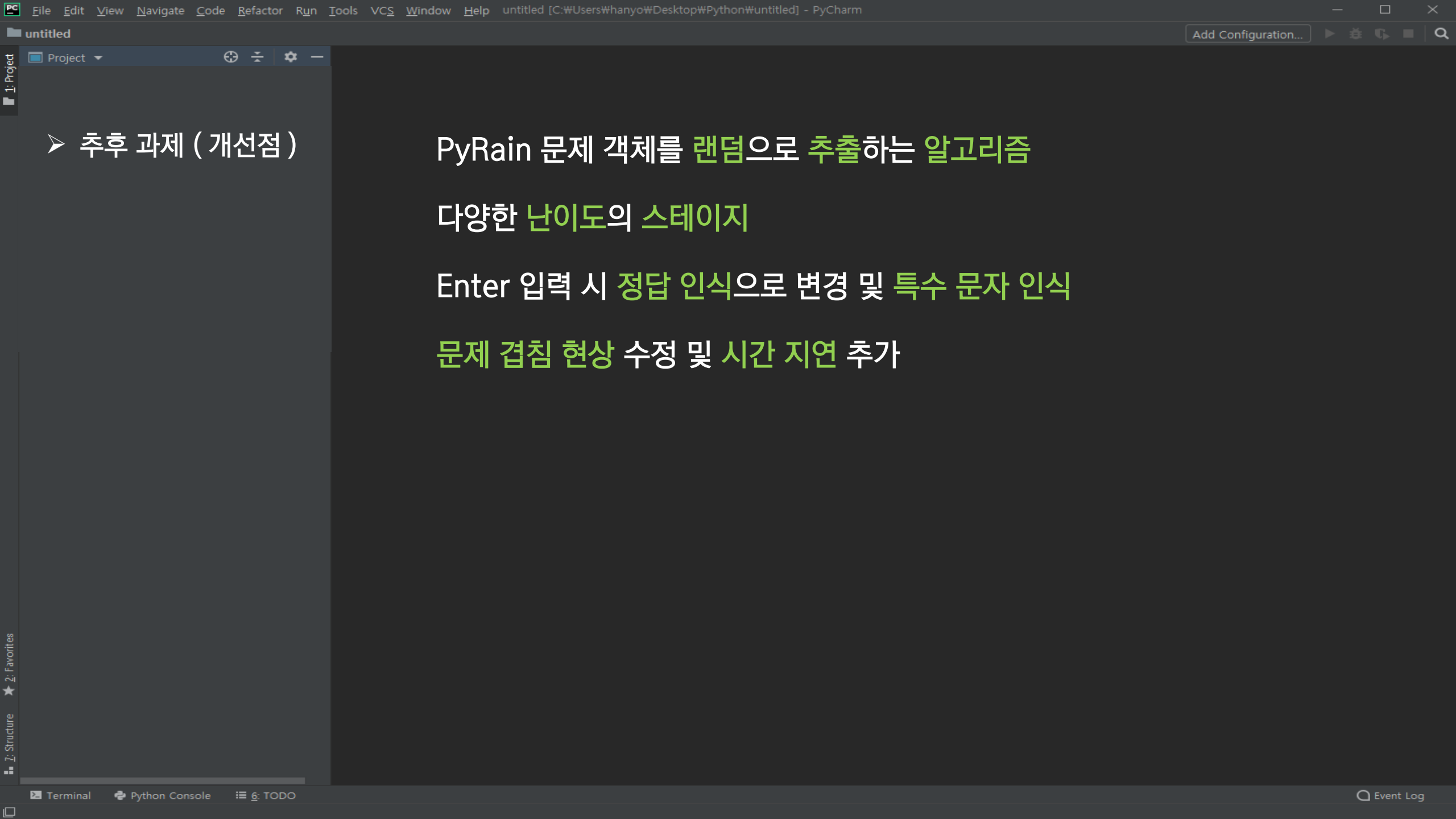
파이썬의 문법을 맞추는 게임 구현

➤ 차별성 (장점)

일반 단어가 아닌 파이썬의 문법을 문제로 채택하여,

반복 학습을 통한 파이썬 문법 학습

정확하고 빠른 코딩 가능



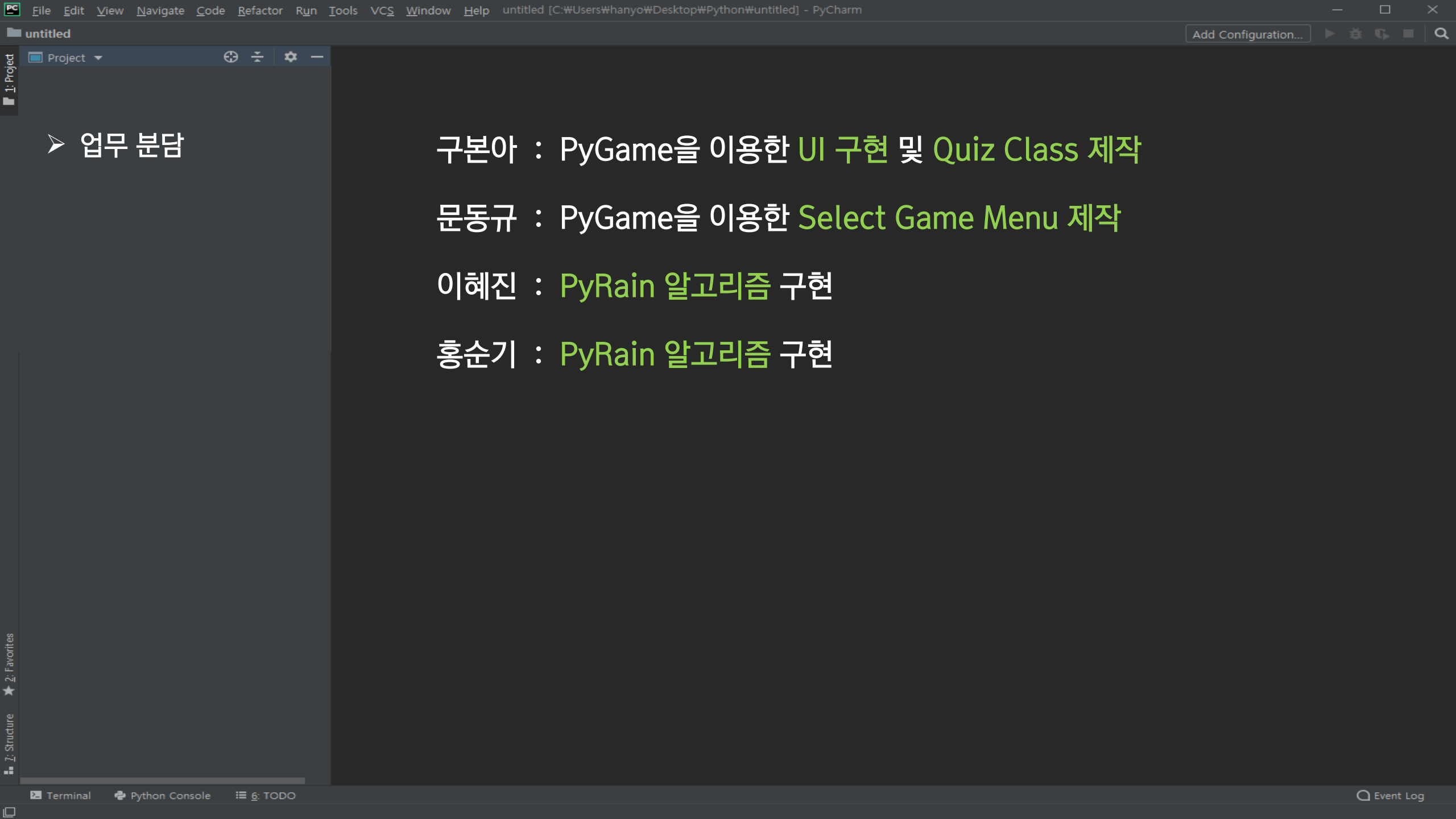
➤ 추후 과제 (개선점)

PyRain 문제 객체를 랜덤으로 추출하는 알고리즘

다양한 난이도의 스테이지

Enter 입력 시 정답 인식으로 변경 및 특수 문자 인식

문제 겹침 현상 수정 및 시간 지연 추가



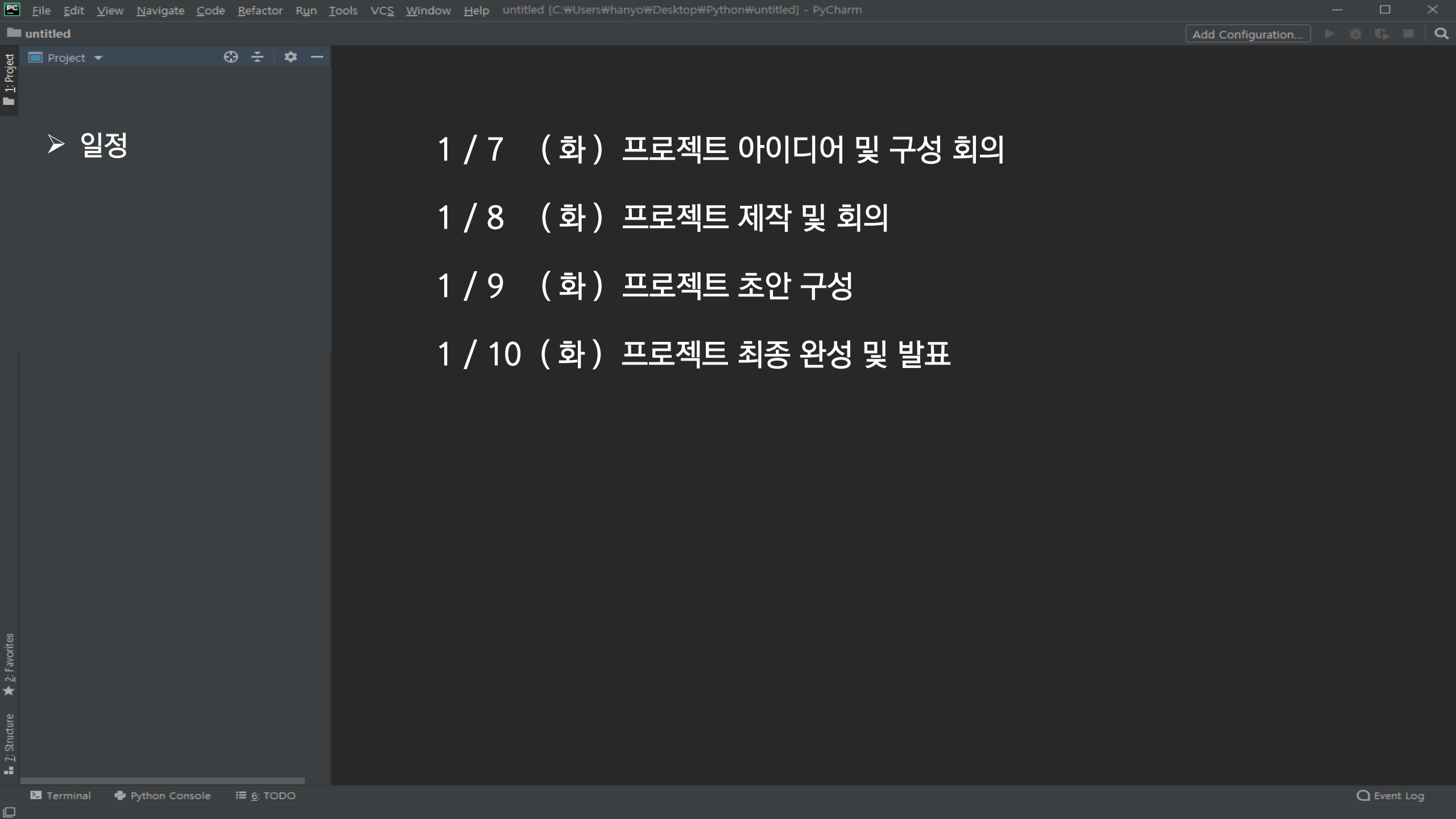
➤ 업무 분담

구본아 : PyGame을 이용한 UI 구현 및 Quiz Class 제작

문동규 : PyGame을 이용한 Select Game Menu 제작

이혜진 : PyRain 알고리즘 구현

홍순기 : PyRain 알고리즘 구현



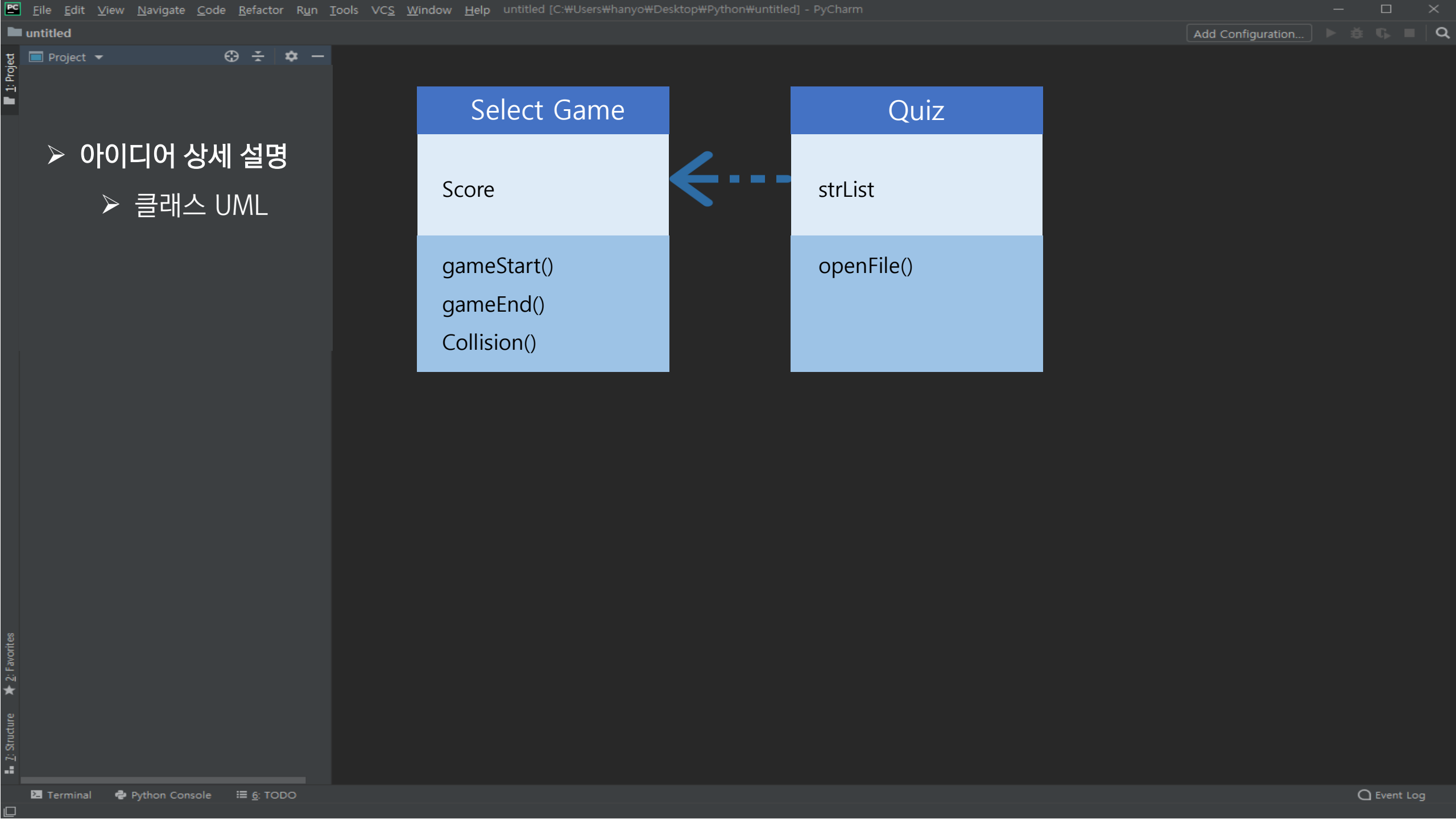
➤ 일정

1 / 7 (화) 프로젝트 아이디어 및 구성 회의

1 / 8 (화) 프로젝트 제작 및 회의

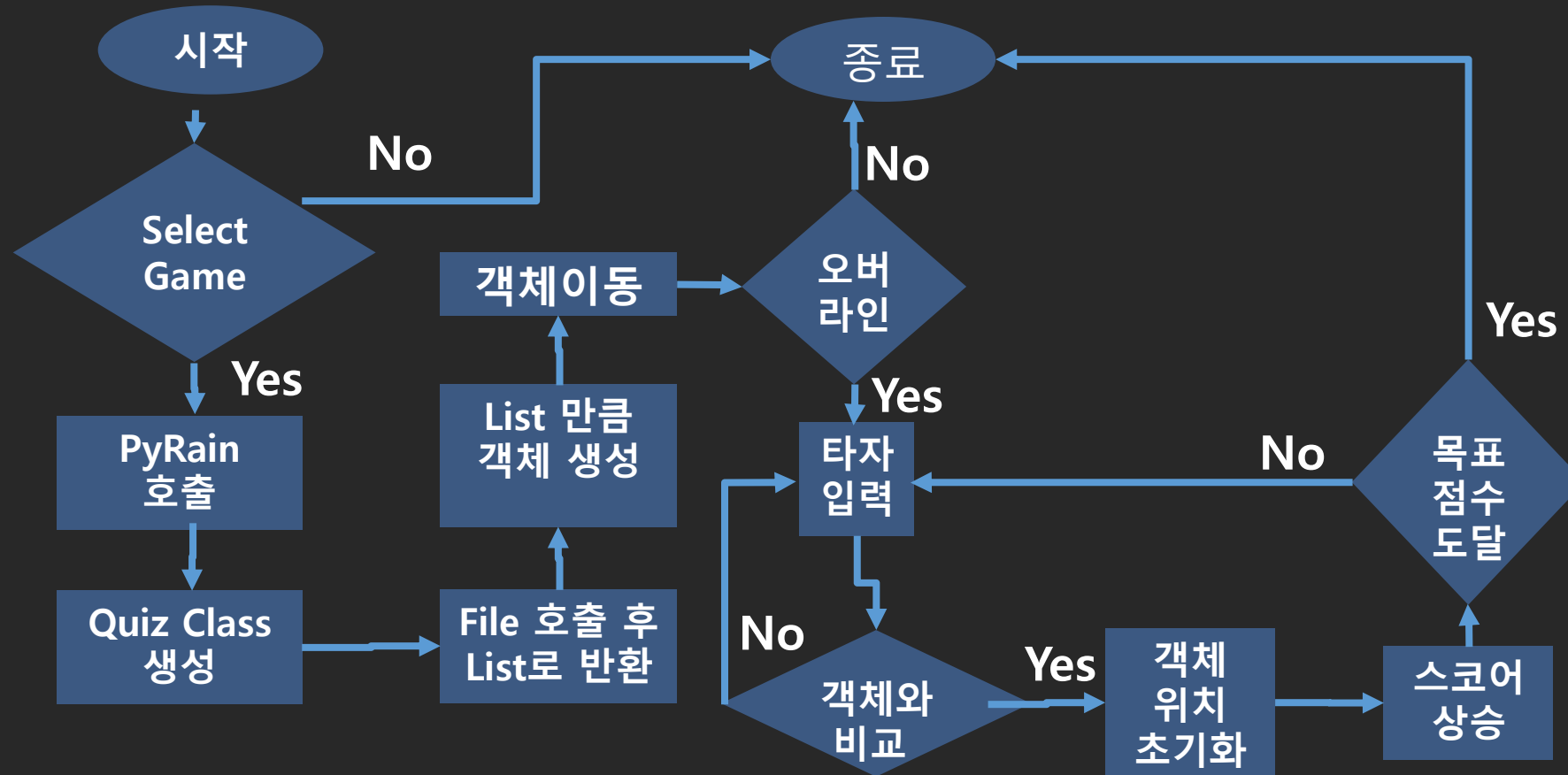
1 / 9 (화) 프로젝트 초안 구성

1 / 10 (화) 프로젝트 최종 완성 및 발표



➤ 아이디어 상세 설명

➤ 순서도



1: Project

Project

2: Favorites

3: Structure

4: IDE

5: Terminal

6: Python Console

7: TODO

8: Event Log

9: AI Assistant

10: Search

11: Recent Files

12: Recent Projects

13: Recent Locations

14: Recent Views

15: Recent Settings

16: Recent Snippets

17: Recent Templates

18: Recent Macros

19: Recent Scripts

20: Recent Plugins

21: Recent Extensions

22: Recent Languages

23: Recent Frameworks

24: Recent Libraries

25: Recent Packages

26: Recent Modules

27: Recent Classes

28: Recent Functions

29: Recent Variables

30: Recent Constants

31: Recent Comments

32: Recent Errors

33: Recent Warnings

34: Recent Messages

35: Recent Debugging

36: Recent Profiling

37: Recent Benchmarking

38: Recent Performance

39: Recent Security

40: Recent Compliance

41: Recent Auditing

42: Recent Monitoring

43: Recent Logging

44: Recent Tracing

45: Recent Telemetry

46: Recent Analytics

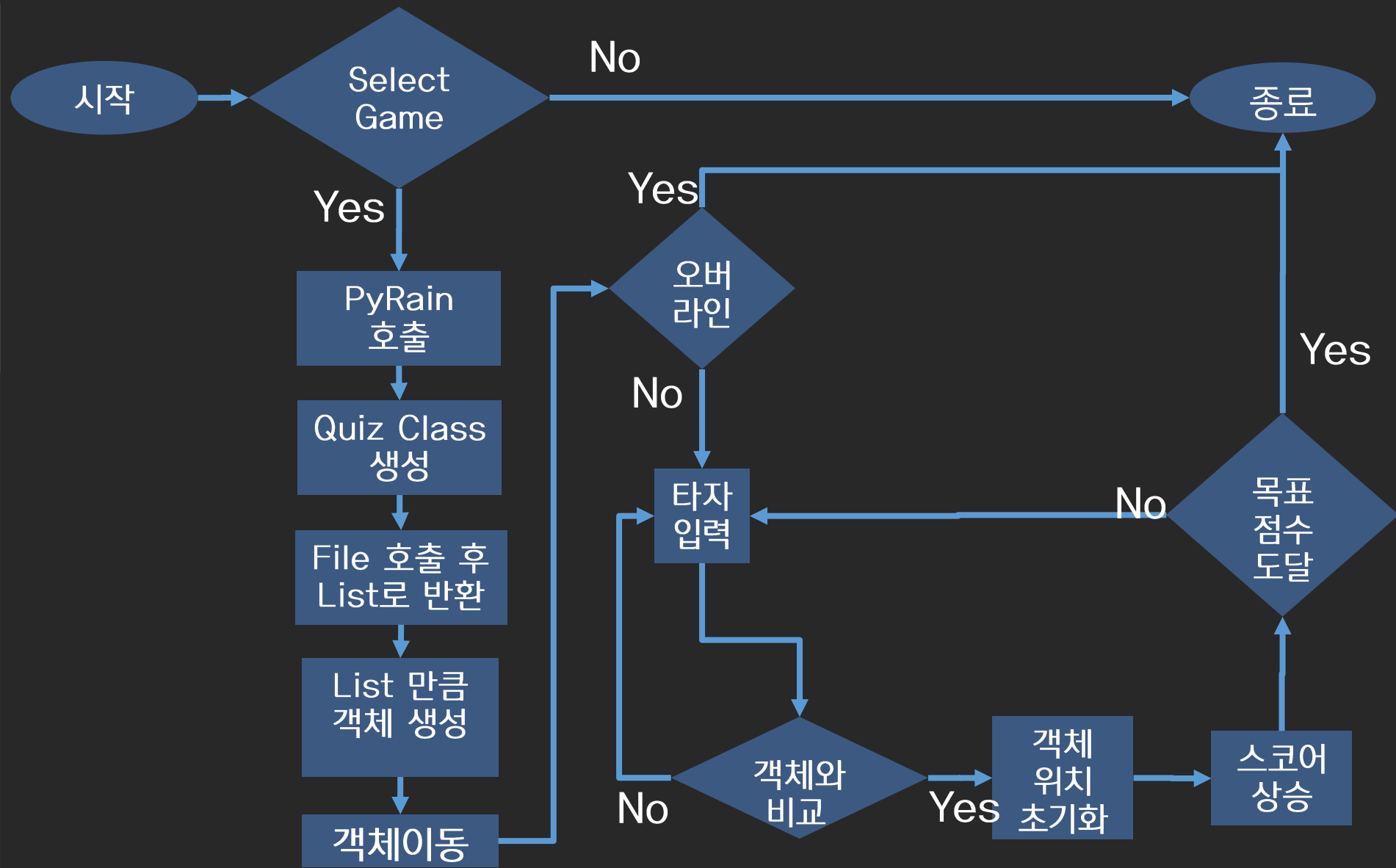
47: Recent Reporting

48: Recent Documentation

49: Recent Help

50: Recent About

- 아이디어 상세 설명
- 순서도



```
def draw_button(button, screen):
    """사각형 버튼 및 텍스트표면 생성"""
    pygame.draw.rect(screen, button['color'], button['rect'])
    screen.blit(button['text'], button['text rect'])

def create_button(x, y, w, h, text, callback):
    """버튼과 관련된 데이터가 포함된 딕셔너리
    rect, text surface 및 text rect, color
    값 반환
    """
    # 문자 설정 변수
    text_surf = font.render(text, True, black)
    # 사각 버튼 설정 변수
    button_rect = pygame.Rect(x, y, w, h)
    # 텍스트 위치 설정 변수
    text_rect = text_surf.get_rect(center=button_rect.center)
    # 딕셔너리로 값 묶기
    button = {
        'rect': button_rect,
        'text': text_surf,
        'text rect': text_rect,
        'color': gray,
        'callback': callback,
    }
    return button
```

```
def game1():
    import PyRain

    # 종료버튼으로 누르면 화면닫힘
    def quit_game():
        nonlocal done
        done = True

    # 버튼 설정
    button1 = create_button(250, 100, 250, 80, 'PyRain', game1)
    button2 = create_button(250, 400, 250, 80, 'Exit', quit_game)

    # 리스트를 이용하여 전체 버튼 받기
    button_list = [button1, button2]
```



```
# quiz 생성 클래스
```

```
class QuizClass:
```

```
    def quizMethod(self):
```

```
        f = open('gametext.txt', 'r')
```

```
        a = f.read()
```

```
        b = a.split(' ')
```

```
        return b
```

```
_quiz = QuizClass()
```

```
a = _quiz.quizMethod()
```

```
# quiz의 위치 및 속도 설정 변수
```

```
for i in range(speed_of_quiz):
```

```
    quizX.append(random.randint(0, 730))
```

```
    quizY.append(random.randint(-100, -50))
```

```
quizY_change.append(0.3)
```

```
# 충돌 시
```

```
collision = isCollision(tt)
```

```
# 초기 위치로 되돌리기
```

```
if collision:
```

```
    score_value += 10
```

```
    inputStr = ''
```

```
    quizX[i] = random.randint(0, 736)
```

```
    quizY[i] = random.randint(-150, -100)
```

```
# Game Over \
```

```
if quizY[i] > 550:
```

```
    for j in range(speed_of_quiz):
```

```
        quizY[j] = 2000
```

```
    x = True
```

```
    game_over_text()
```

```
    break
```

```
# Game Clear \
```

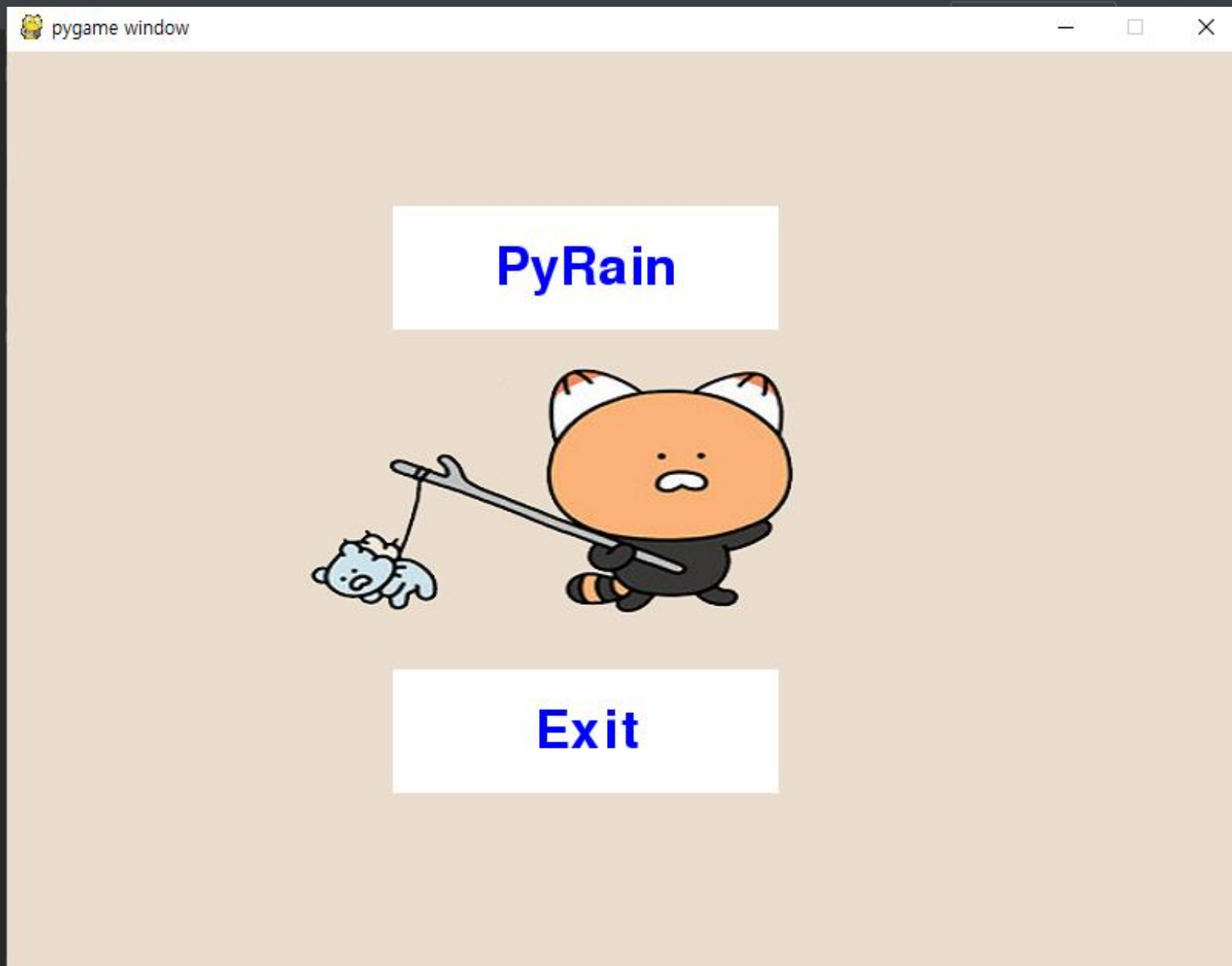
```
if score_value == 100:
```

```
    x = True
```

```
    game_clear_text()
```

```
    break
```

➤ 실행 결과



untitled

Project

1: Project

2: Favorites

3: Structure

Terminal

Python Console

6: TODO

➤ 실행 결과

