

# 프로젝트 결과 발표

201565 이성훈 · 223324 류현승 · 223344 이가현

# 목차

— 팀원 정보

— 개발 배경

— 깃허브 커밋 히스토리

— 역할 분배

— 프로젝트 결과물

# 팀원 정보

201565 이성훈 (2학년)

223324 류현승 (2학년)

223344 이가현 (2학년)

# 개발 배경

- 홈트(홈트레이닝)를 많이 함
- 세트 · 횟수를 잊을 때가 많음
- 세트 당 쉬는 시간 설정이 불편함

# 깃허브 커밋 히스토리

Merge pull request #8 from hooniverse/readme-1	26 11 2023 19:08	이성훈 <121534585+hooniver	e1f3434
readme 로고 오류 수정	26 11 2023 19:07	이성훈 <121534585+hooniver	cd2884c
Merge pull request #7 from leeghy/interface	26 11 2023 19:06	이가현 <115767150+leeghy@	69fdbbc1
Merge branch 'main' into interface	26 11 2023 19:06	이가현 <115767150+leeghy@	e5724df
인터페이스 클래스 변경	26 11 2023 19:02	egaahhh <cndl03@jnu.ac.kr>	18b650b
README 로고 추가	26 11 2023 19:01	hooniverse <sung010326@gn	74d6f21
Merge pull request #6 from hyuns22/exercise	26 11 2023 18:51	hyuns22 <115512607+hyuns2	05201de
time_count함수	26 11 2023 18:50	Ryu <yhs9910@gmail.com>	59f22b5
speake함수	26 11 2023 18:39	Ryu <yhs9910@gmail.com>	80c60c5
Exercise클래스 생성자	26 11 2023 18:38	Ryu <yhs9910@gmail.com>	7011d81
Merge pull request #5 from hyuns22/main	26 11 2023 16:03	hyuns22 <115512607+hyuns2	5d93c2c
main.py 조안	26 11 2023 16:00	Ryu <yhs9910@gmail.com>	89fa3c8
Merge pull request #4 from leeghy/interface	26 11 2023 5:41	이가현 <115767150+leeghy@	e7827d1
로고	26 11 2023 5:31	egaahhh <cndl03@jnu.ac.kr>	8f57b52
Merge branch 'interface' of https://github.com/leeghy/SelfFitnessManager into interface	26 11 2023 5:27	egaahhh <cndl03@jnu.ac.kr>	27e1dff
title 수정	26 11 2023 5:24	egaahhh <cndl03@jnu.ac.kr>	383b1ae
Merge pull request #3 from leeghy/interface	26 11 2023 5:15	이가현 <115767150+leeghy@	1671600
interface test	26 11 2023 5:13	egaahhh <cndl03@jnu.ac.kr>	2326c9d
Merge pull request #2 from hooniverse/write_readme	25 11 2023 21:38	이성훈 <121534585+hooniver	752a83c
README.md 내용 추가	25 11 2023 21:32	hooniverse <sung010326@gn	58f0a1c
Merge pull request #1 from hooniverse/write_readme	25 11 2023 21:25	이성훈 <121534585+hooniver	68c2343
readme 내용 추가	24 11 2023 16:14	hooniverse <sung010326@gn	0bb79a5
Initial commit	23 11 2023 16:57	이성훈 <121534585+hooniver	6ff6bf2

프로젝트 총 기간  
11월 23일 ~ 12월 17일  
108 Commits



hooniverse Merge branch 'main' of <https://github.com/hooniverse/SelfFitnessManager>

470cc9a · 3 minutes ago



108 Commits

# 역할 분배

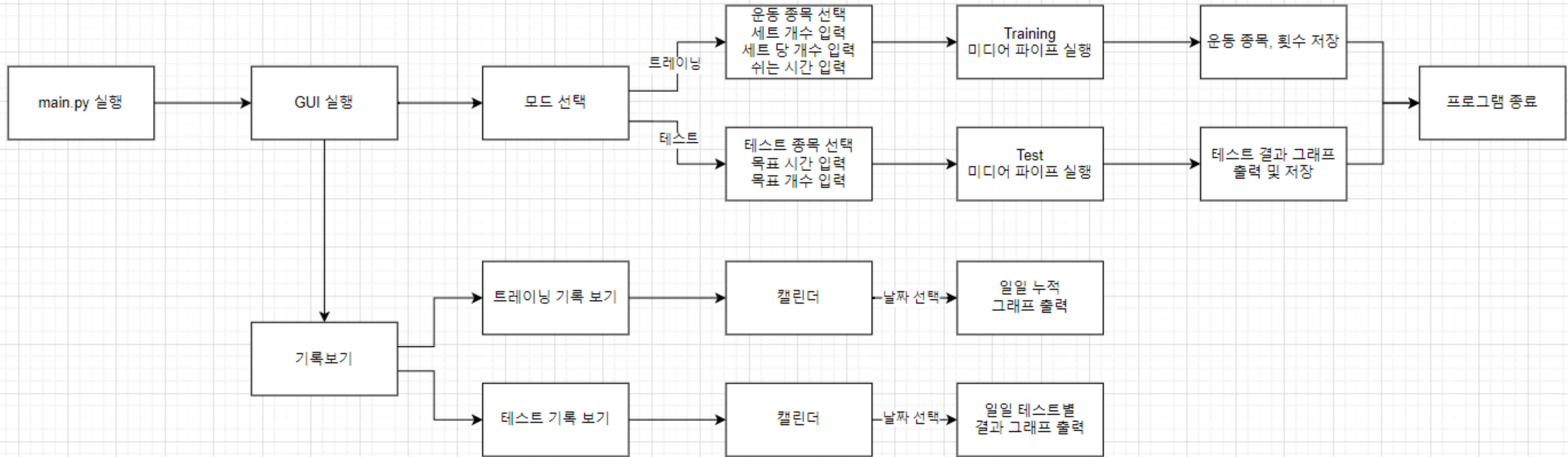
이가현 : GUI, 테스트 모드

류현승 : main.py, 트레이닝 모드, speak 함수, push-up, squat

이성훈 : 기록 및 그래프 관리, Git 관리, pull-up

# 프로젝트 결과물

# 화면 흐름도





# Main.py

```
interface = Interface.Interface() ##인터페이스 객체 생성
#
user_input = interface.run() ##인터페이스 run함수 -- return으로 리스트
# [training, 종목, set, set당 개수, 쉬는시간] / [test, 종목, 목표시간, 목표개수]

if user_input['mode'] == 'training':#training인 경우
    ⚡ exercise = Training(user_input['type'], user_input['set_count'], user_input['reps_per_set'], user_input['break_time'])
elif user_input['mode'] == 'test':
    exercise = Test(user_input['type'], user_input['goal_time'], user_input['goal_number'])##(목표시간, 목표개수)


exercise.run() #미디어파이프 실행

interface.end() #운동이 종료되었다는 메시지
```

mode가 training이면 Training 객체 생성 / test면 Test 객체 생성

# 트레이닝 모드

모드 선택



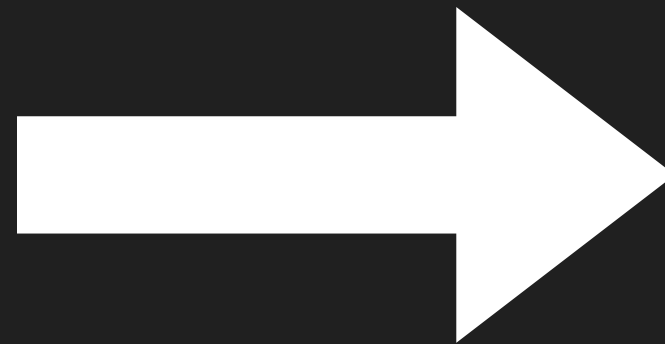
Self Fitness Manager

모드선택 ☒ 트레이닝 ☐ 테스트

기록보기

확인

트레이닝 모드  
선택 시



training\_page

운동 종목 선택 

Push-Up  
Pull-Up  
Squat

세트 개수(자연수 입력)

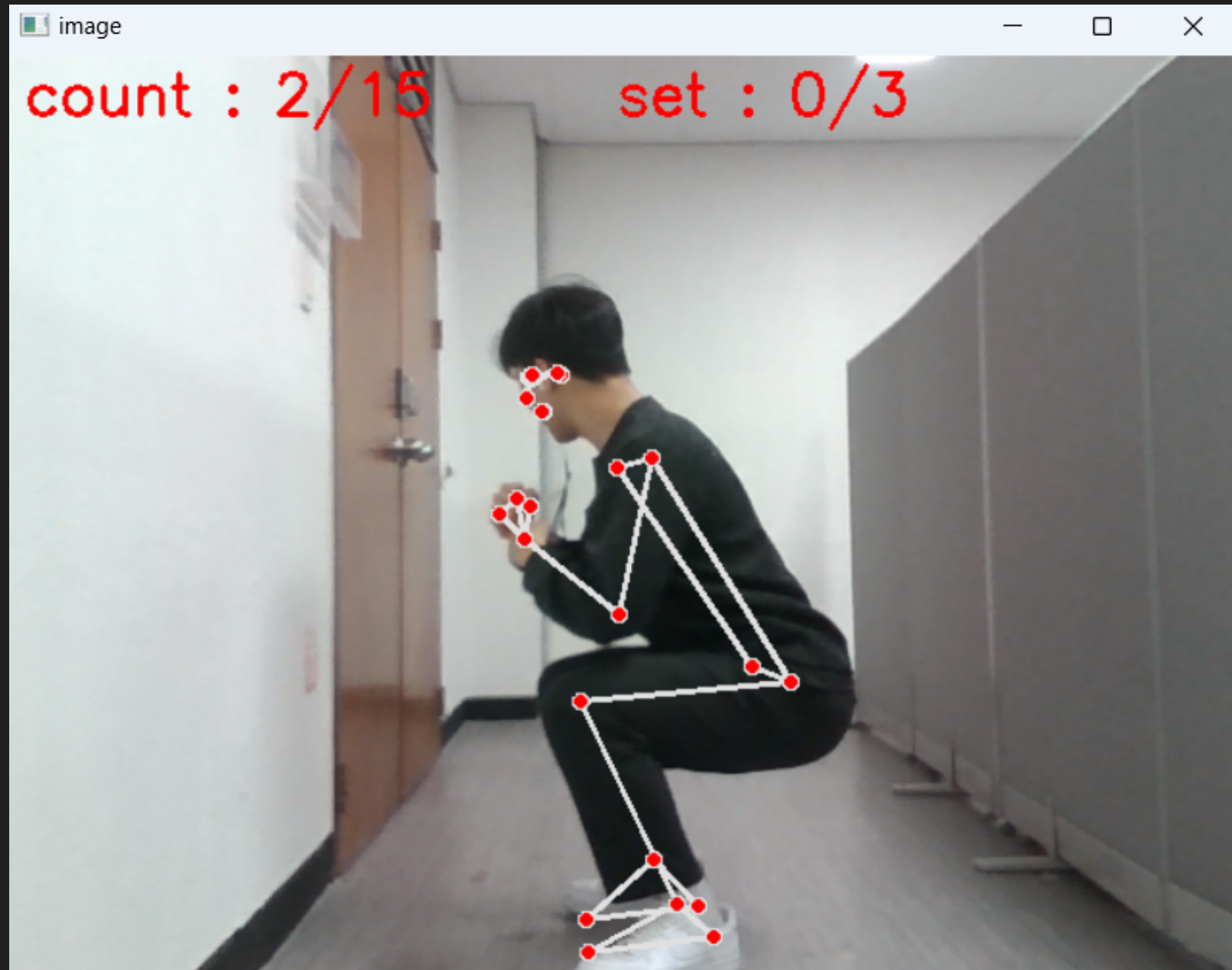
세트 당 개수(자연수 입력)

쉬는 시간(초 단위 입력)

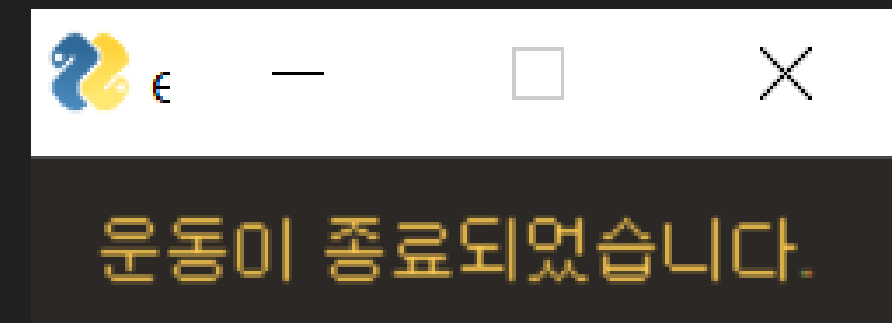
이전으로

확인

# 트레이닝 모드

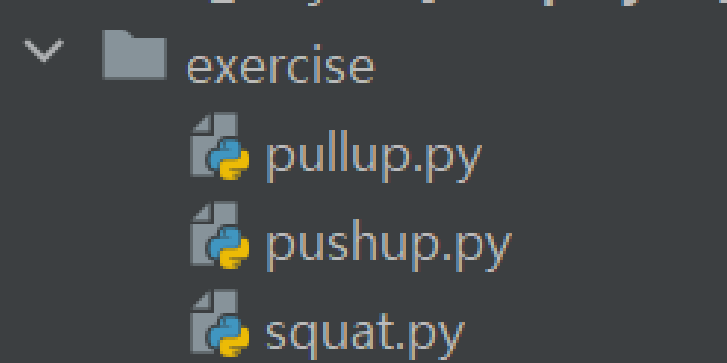
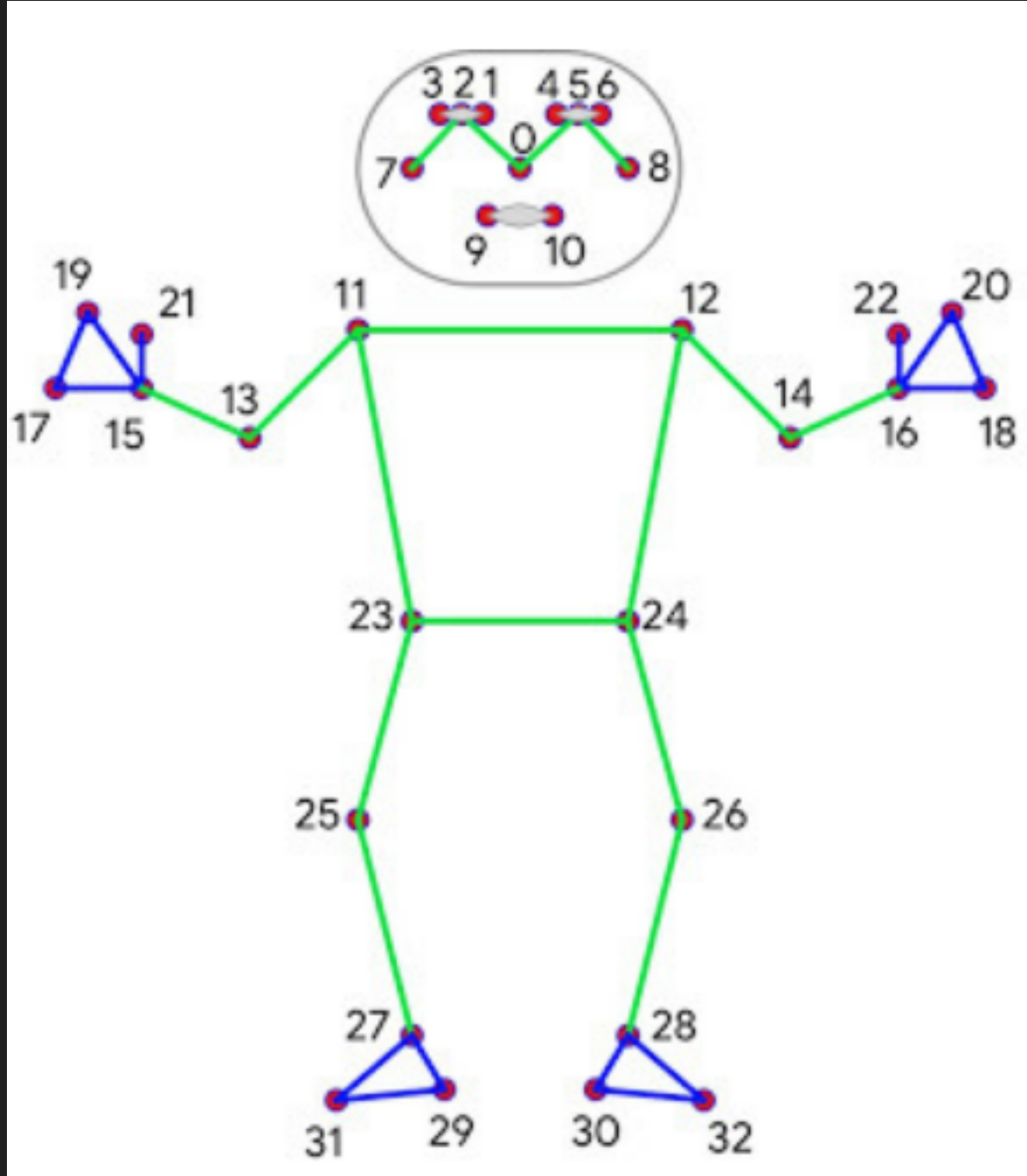


세트 완료



운동 완료

# squat, push-up, pull-up








클래스로 작성하여  
운동종목에 맞는 객체 생성

```
class Pushup:
    Ryu
    def __init__(self):
        return
    hooniverse
    def __str__(self):
        return "Push-Up"
    Ryu
    def countUp(self, count, status, pose_landmarks):
        if pose_landmarks.landmark[12].y < pose_landmarks.landmark[14].y and status:
            count += 1
            status = False
        elif pose_landmarks.landmark[12].y > pose_landmarks.landmark[14].y and not (status):
            status = True
        return count, status
```

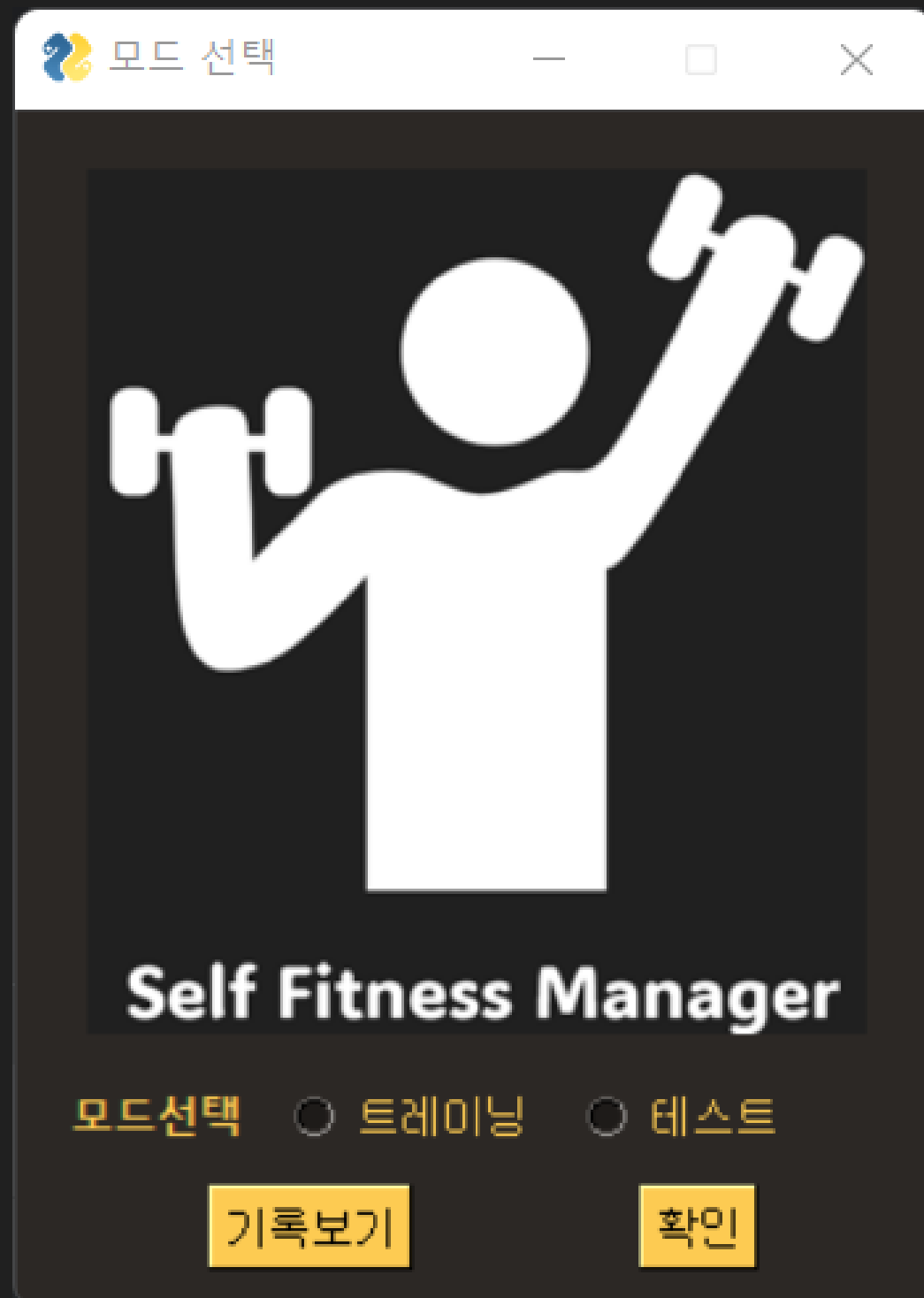
# speak 함수

```
def speak(text):  
    tts = gTTS(text=text, lang='en')  
  
    filename = './mp3_file/'+text+'.mp3'  
    if not (os.path.exists(filename)):  
        tts.save(filename)  
    playsound(filename)  
  
    return True
```

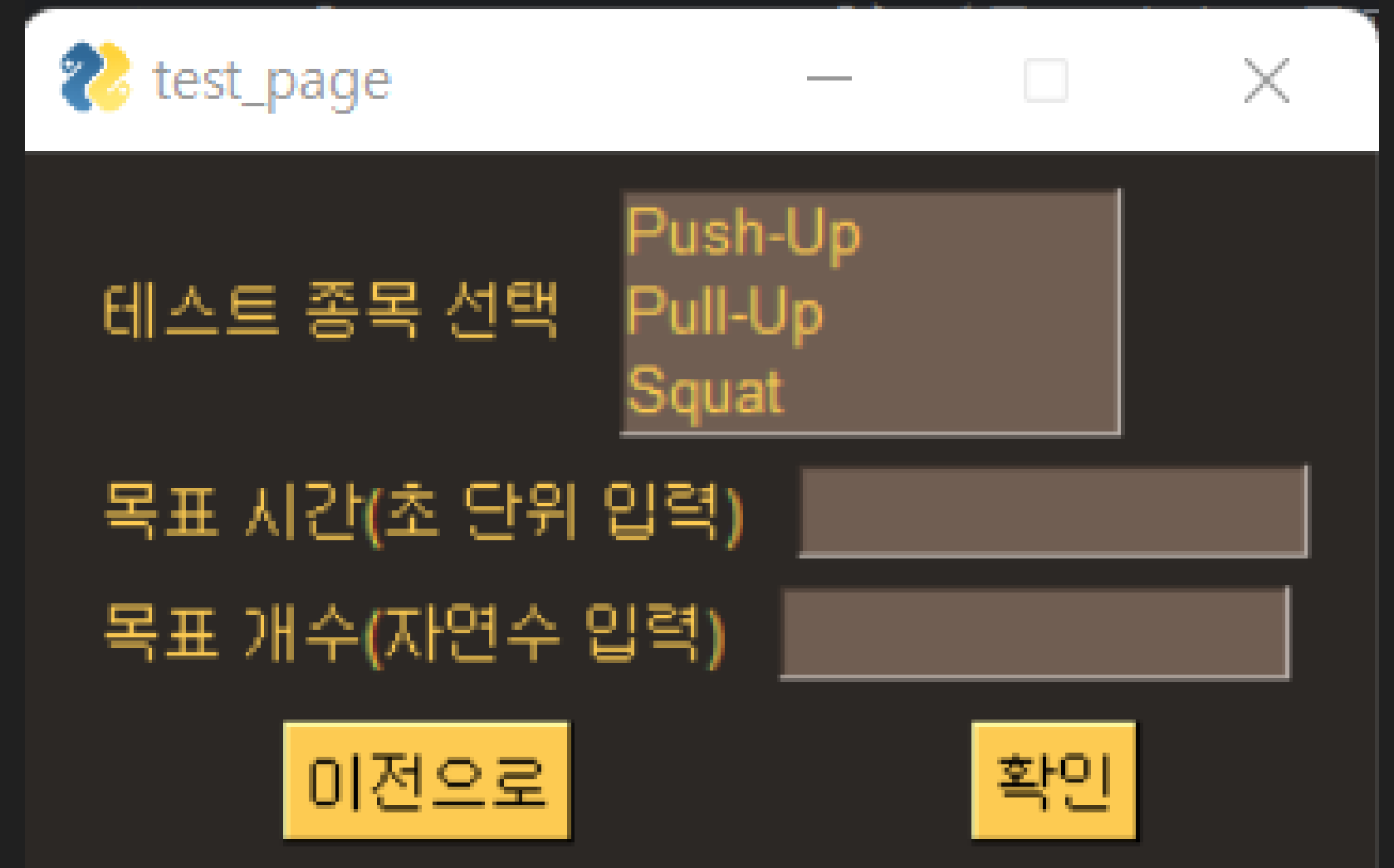
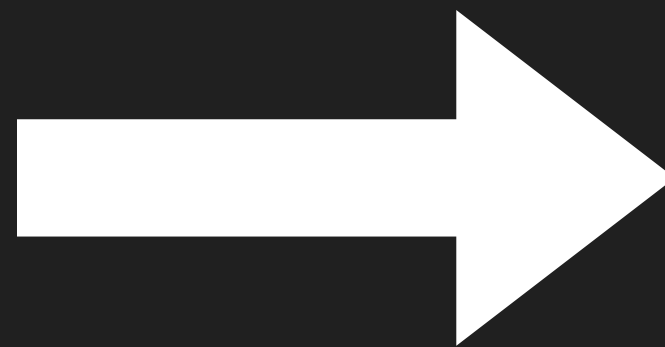
-  exercise complete!.mp3
-  set complete!.mp3
-  set start!.mp3
-  test complete!.mp3
-  test failed!.mp3

gTTS를 이용하여 text를 mp3파일로 저장  
playsound로 재생

# 테스트 모드

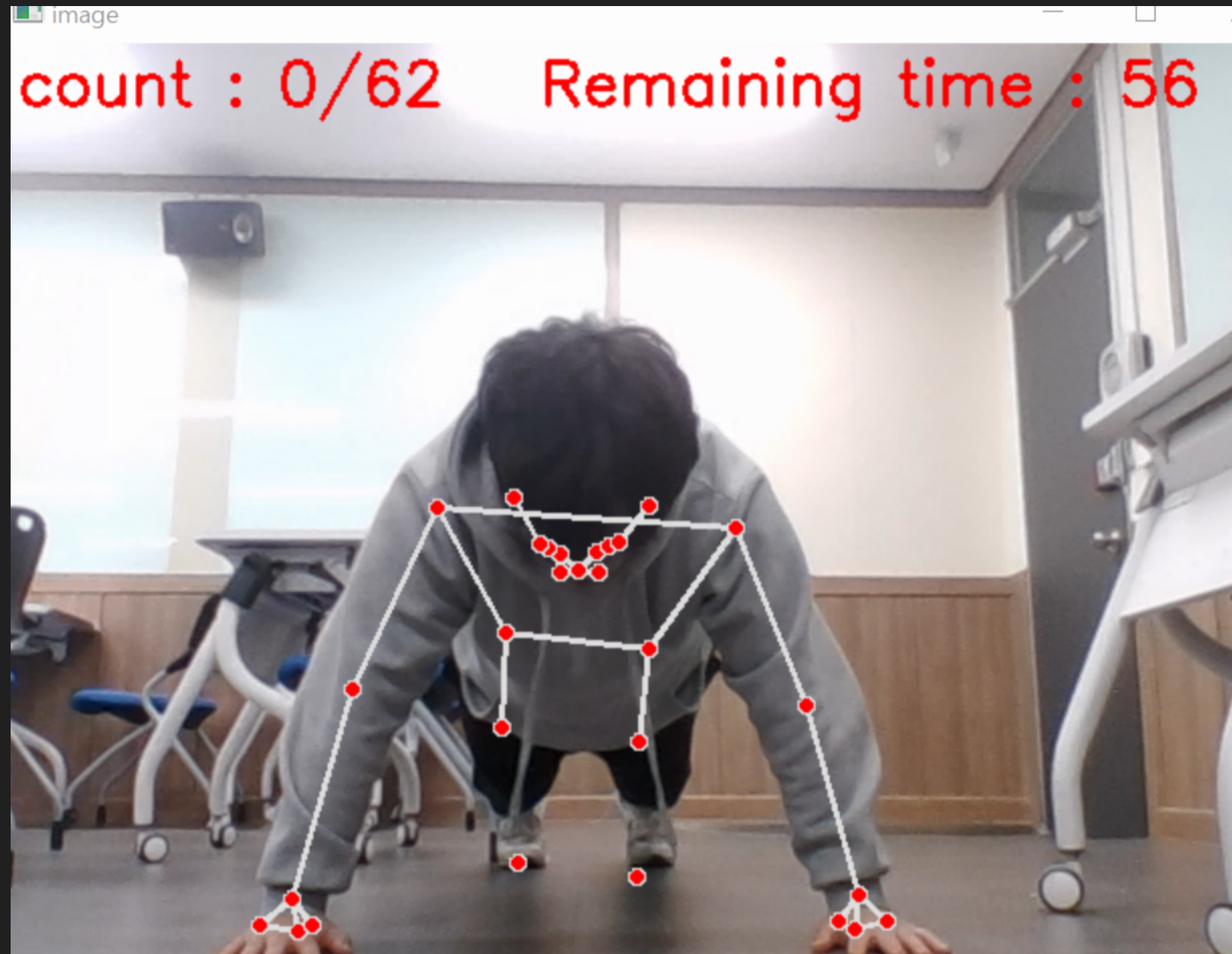


테스트 모드  
선택 시





# 테스트 모드



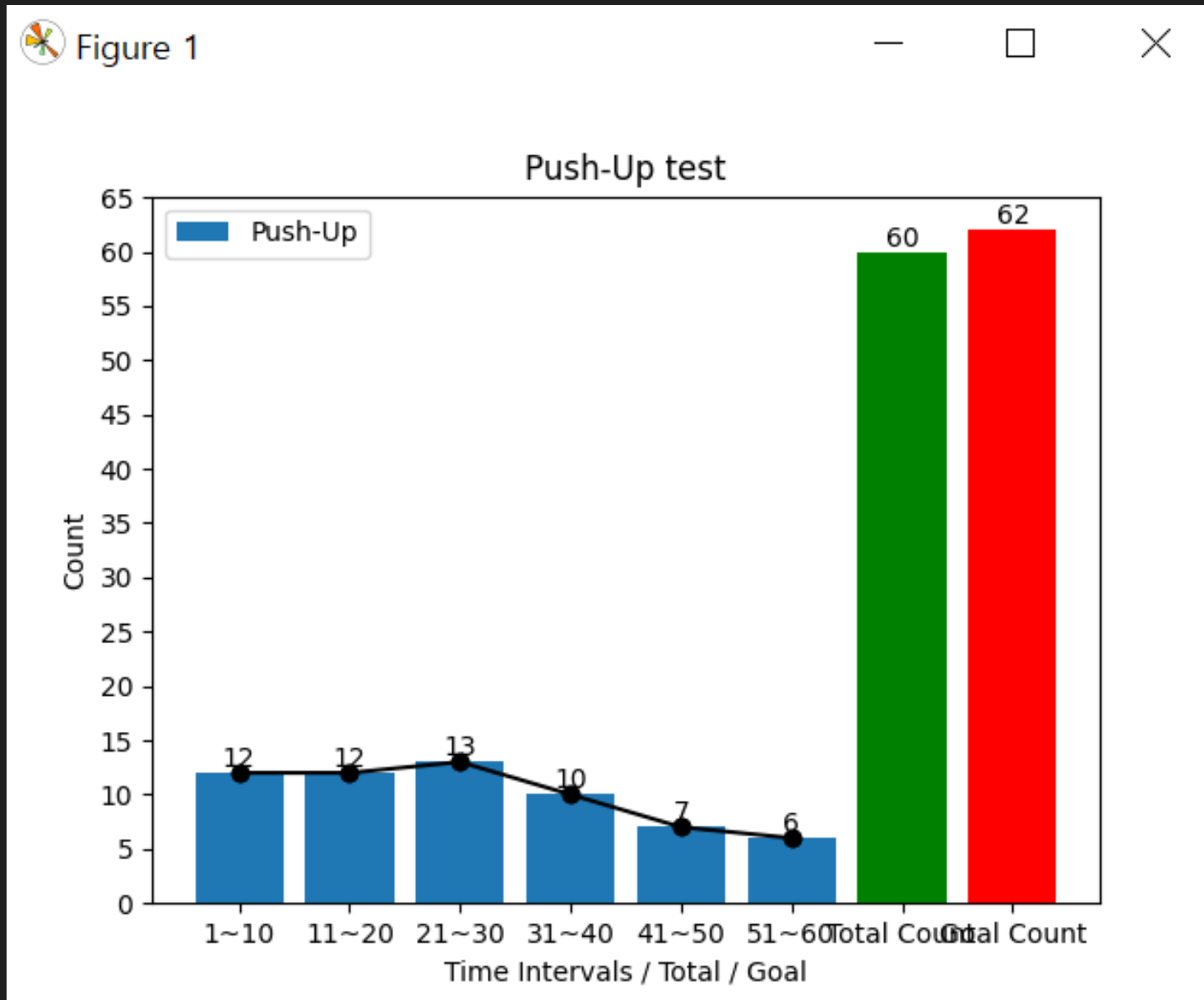
```
start_time = time.time()
```

```
elapsed_time = time.time() - start_time
```

```
remaining_time = max(0, int(int(self.goal_time) - elapsed_time))
```

remaining time 설정 코드

# 테스트 모드 결과



10초당 횟수를 카운트  
-> 몇 초대에 페이스가 떨어지는지 분석

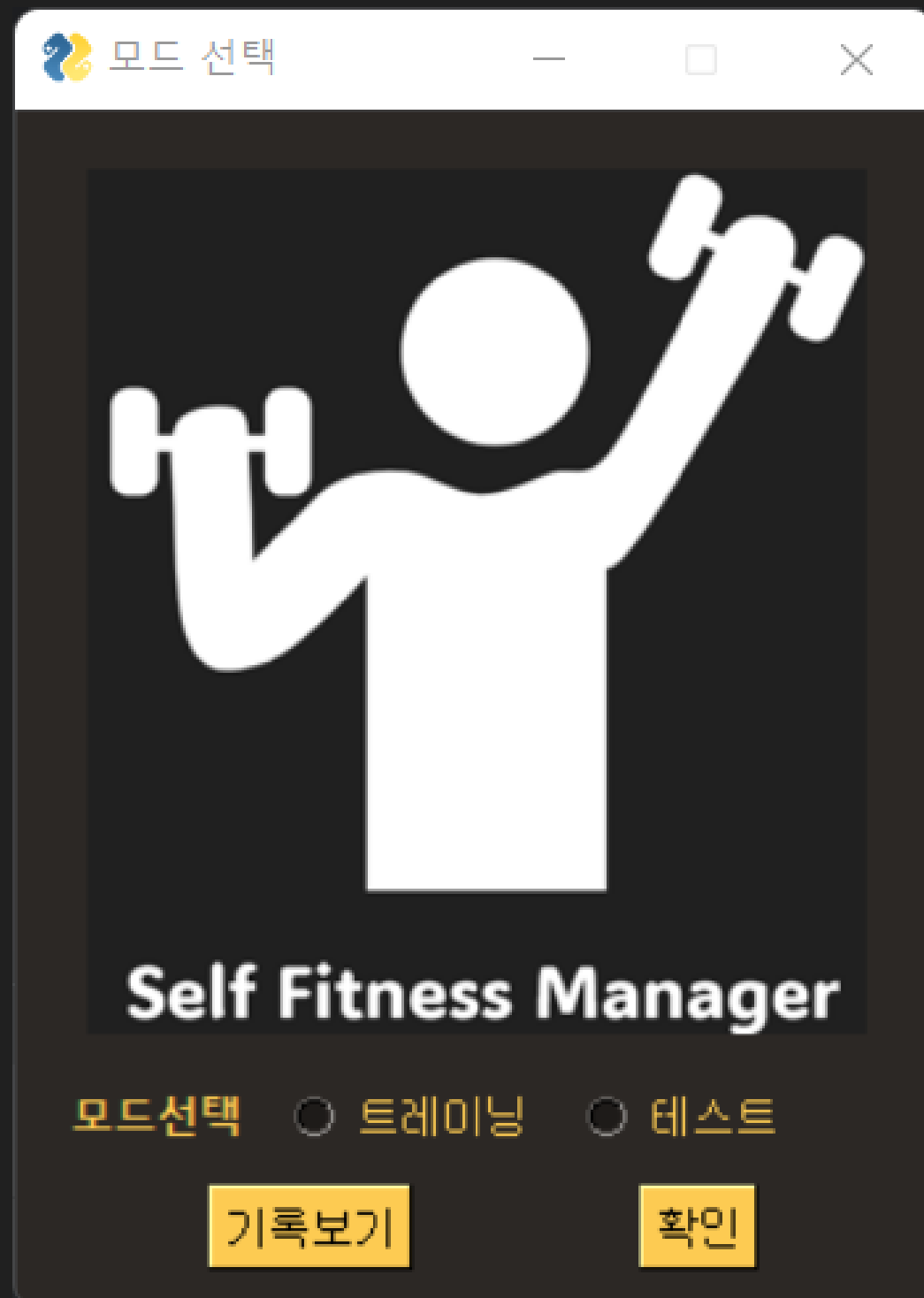


# 기록 보기 - 운동 기록 저장

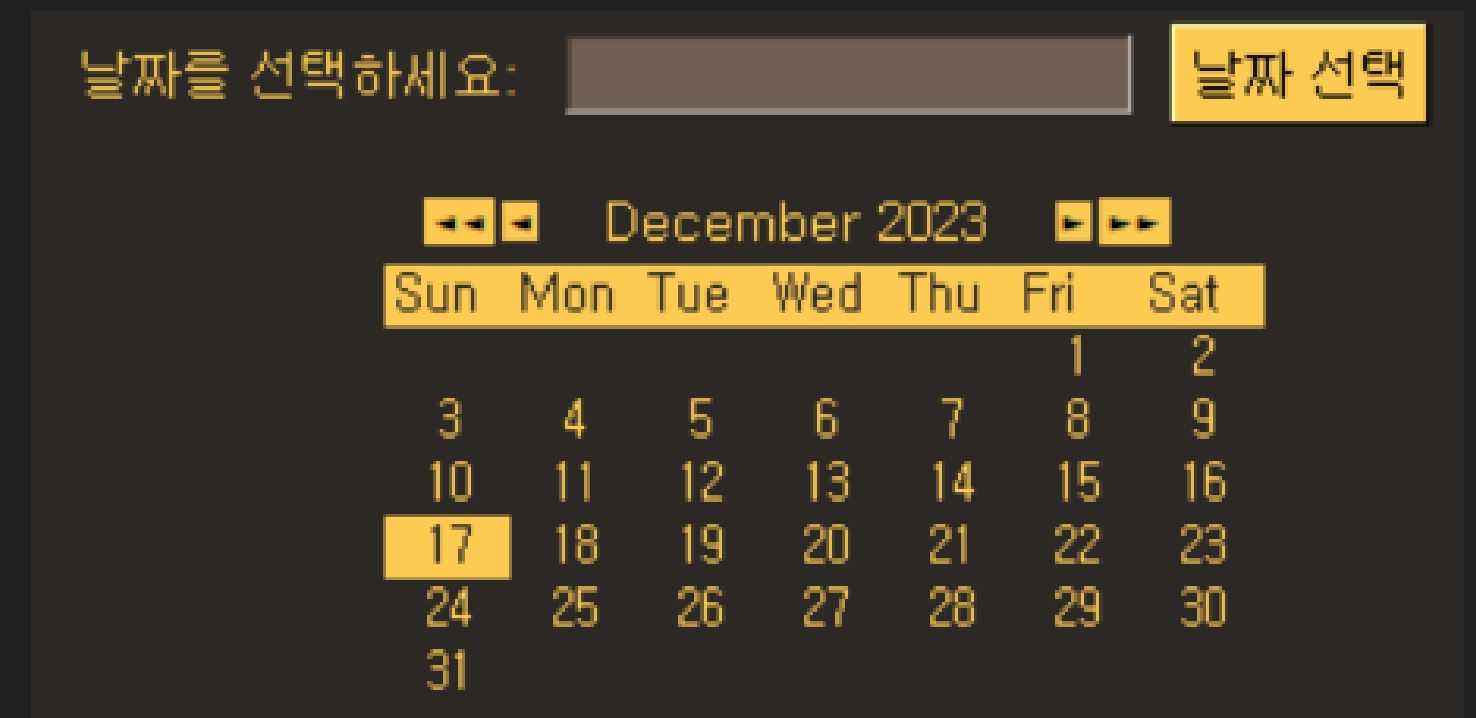
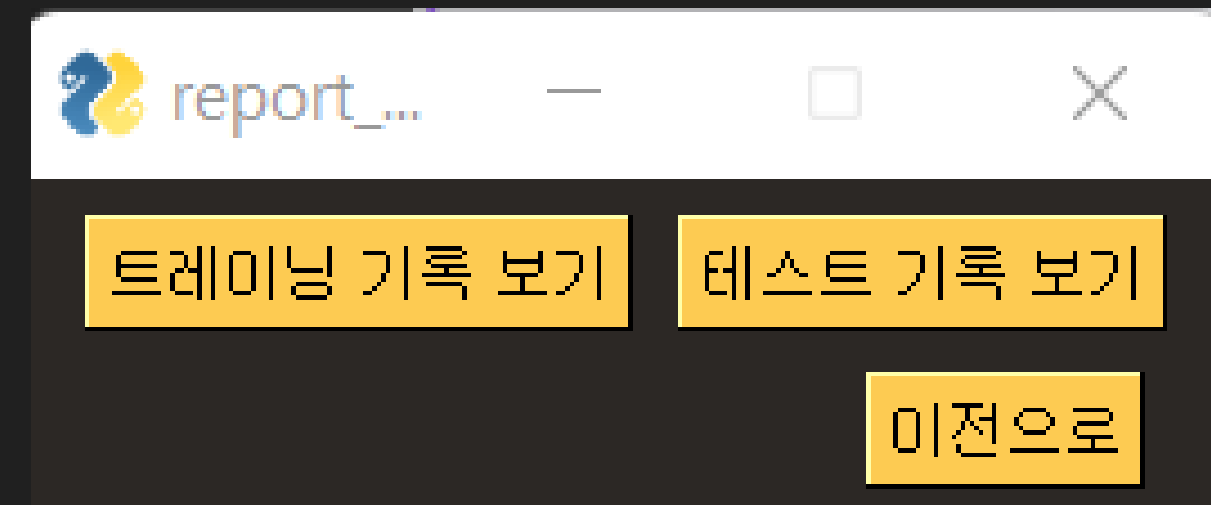
1	Date	exercise	Count
2			
3	2023-12-03 16:25	Pull-Up	11
4	2023-12-03 16:31	Push-Up	40
5			
6	2023-12-03 16:31	Squat	13
7	2023-12-03 16:25	Push-Up	1
8	2023-12-03 16:31	Push-Up	11
9			
10	2023-12-04 16:25	Pull-Up	11
11	2023-12-04 16:31	Pull-Up	24
12			
13	2023-12-05 16:25	Push-Up	1
14	2023-12-06 16:31	Push-Up	111
15	2023-12-06 18:13	Pull-Up	33
16			
17			
18	2023-12-08 15:13	Pull-Up	3
19	2023-12-08 15:14	Push-Up	12
20	2023-12-08 15:16	Pull-Up	13
21	2023-12-08 15:21	Push-Up	2

```
4 import datetime
5 import csv
6 import pandas as pd
  4 usages  👤 hooniverse
7 def write_csv(current_exercise, total_count):
8
9     csv_file_path = "exercise_record.csv"
10    current_time = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
11    try:
12        df = pd.read_csv(csv_file_path)
13    except FileNotFoundError:
14        with open(csv_file_path, mode='w', newline='') as file:
15            writer = csv.writer(file)
16            writer.writerow(['Date', 'Current Exercise', 'Total Count'])
17            df = pd.DataFrame(columns=['Date', 'Current Exercise', 'Total Count'])
18
19    with open(csv_file_path, mode='a', newline='') as file:
20        writer = csv.writer(file)
21        writer.writerow([current_time, current_exercise, total_count])
```

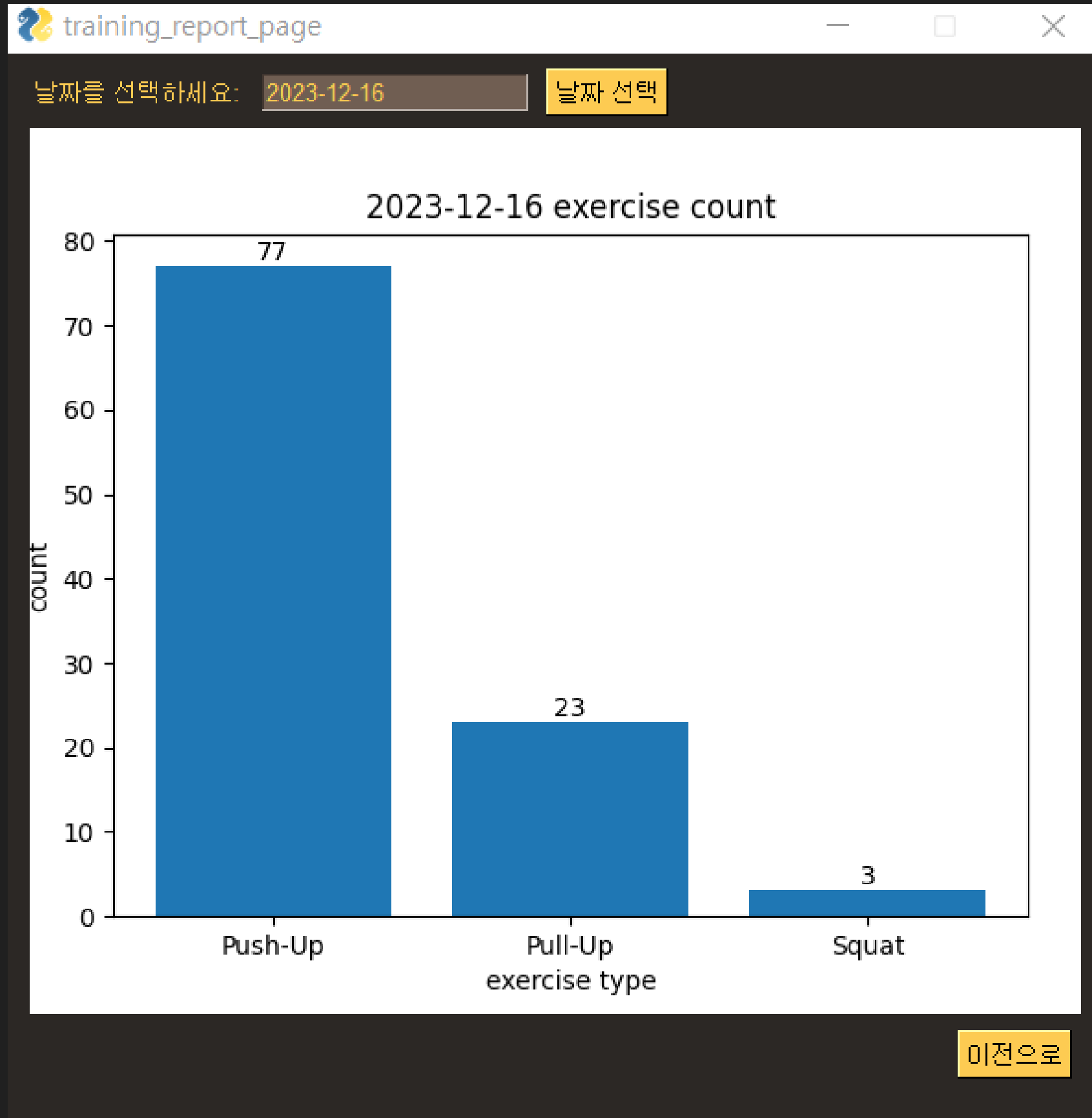
# 기록 보기 - 트레이닝



트레이닝  
기록보기  
선택 시



# 기록 보기 - 트레이닝



날짜 선택시, 당일에 실시한 운동 별  
횟수를 누적하여 그래프 출력

# 기록 보기 - 트레이닝

날짜를 선택하세요: 2023-12-02

날짜 선택



운동을 안 한 날입니다.

이전으로

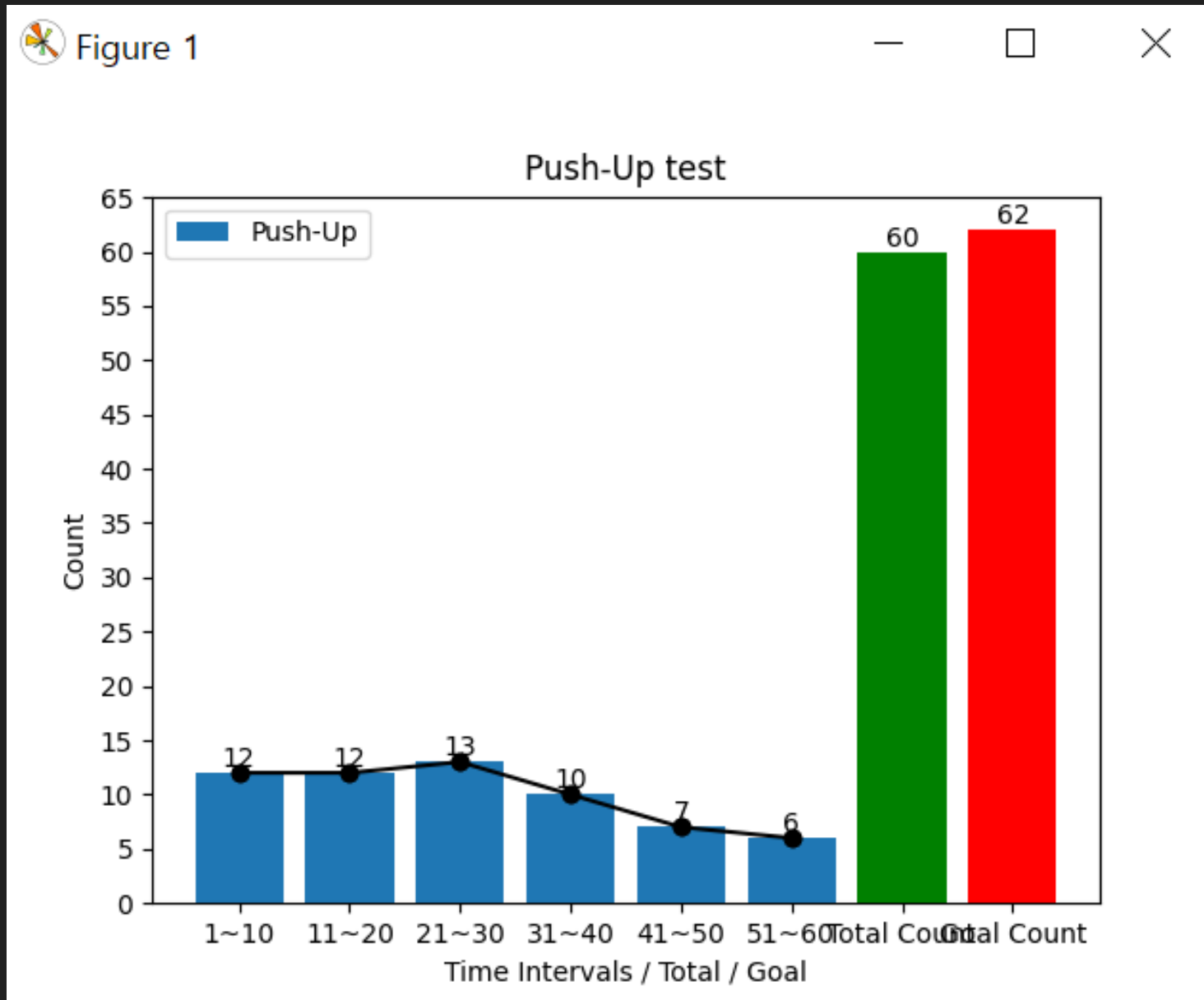
```
if os.path.exists(path):  
    window["output"].update(filename=path)  
else:  
    # 대체할 경로 또는 이미지 설정  
    alternative_path = "do_not_exercise.png"  
    window["output"].update(filename=alternative_path)
```

운동 기록이 없는 날 선택시,  
그래프가 아닌 다른 사진 출력

# 기록보기 - 테스트

```
6 import os
7 import datetime
8 import matplotlib.pyplot as plt
9
10 usage  👤 hooniverse
11
12 def test_graph(time_intervals_labels, count_list, exercise, count, goal_number):
13
14     save_directory = "graph_pictures/test_graph"
15     if not os.path.exists(save_directory):
16         os.makedirs(save_directory)
17
18     current_date = datetime.datetime.now().strftime("%Y-%m-%d")
19     file_name = f"{current_date}_{exercise}_test.png"
20     file_path = os.path.join(save_directory, file_name)
21
22     # 시간대별 운동 횟수를 그래프로 나타냅니다.
23     bars = plt.bar(time_intervals_labels, count_list, label=f"{exercise}")
24
25     # 각 막대의 맨 위 가운데에 점 찍기
26     for bar in bars:
27         height = bar.get_height()
28         plt.plot(*args: bar.get_x() + bar.get_width() / 2, height, 'ko')
29         plt.text(bar.get_x() + bar.get_width() / 2, height, str(int(height)), ha='center', va='bottom')
30
31     # 그래프의 점들을 선으로 이어주기
32     plt.plot(*args: time_intervals_labels, count_list, marker='o', color='black')
33
34     # 총 카운트를 그래프에 추가합니다.
35     plt.bar(x: ["Total Count", "Goal Count"],
36            height: [int(count), int(goal_number)],
37            color=['g', 'r'])
38
39     # 각 Total Count와 Goal Count에도 숫자 표시
40     for i, value in enumerate([int(count), int(goal_number)]):
41         plt.text(i + len(count_list), value, str(value), ha='center', va='bottom')
```

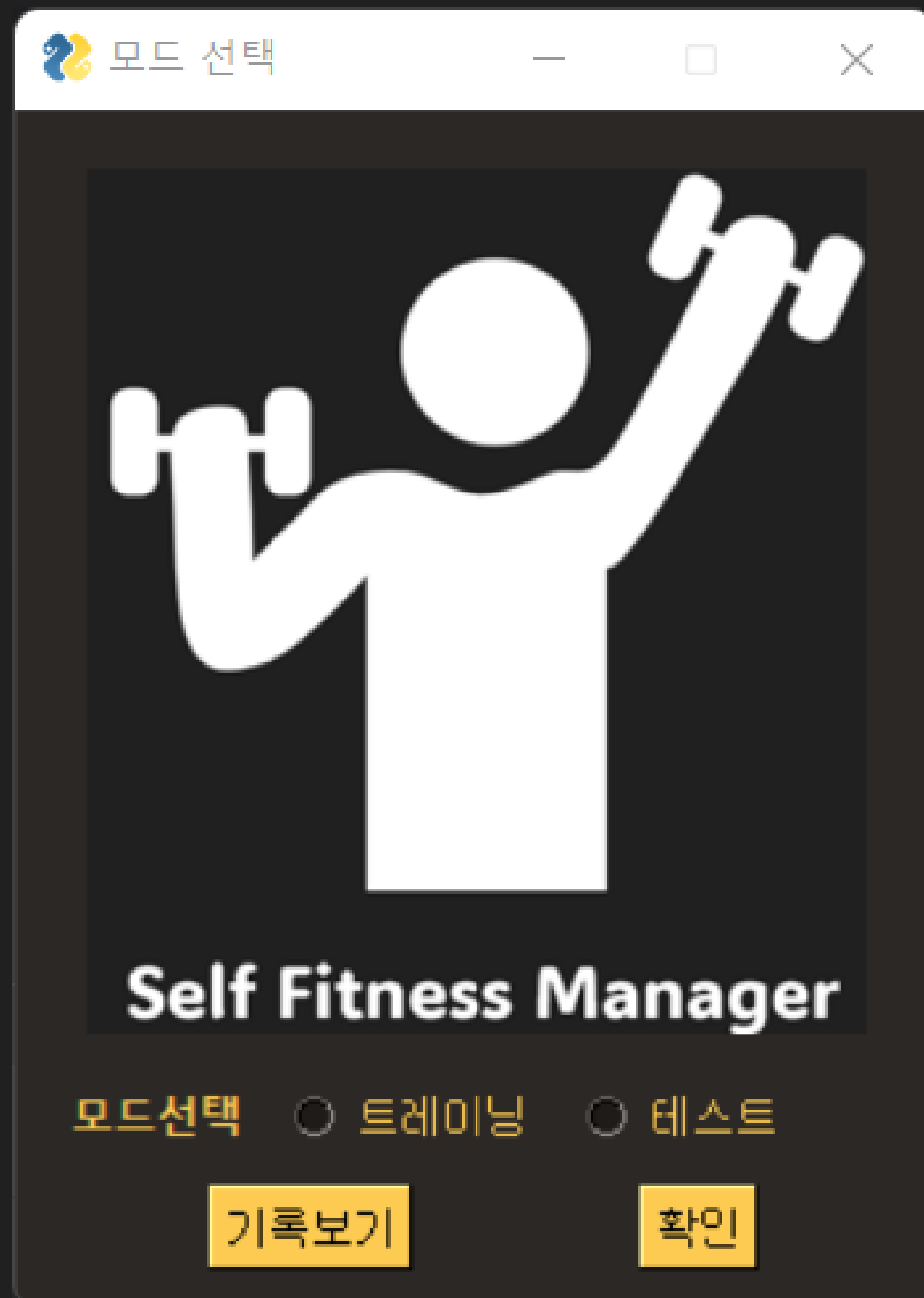
# 테스트 모드 결과



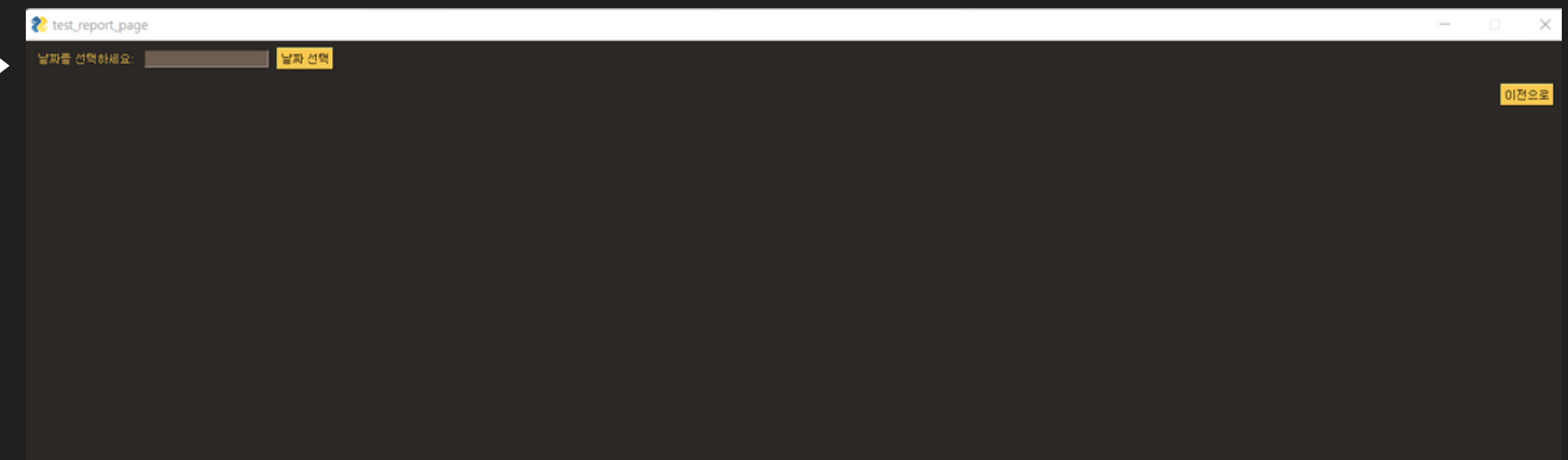
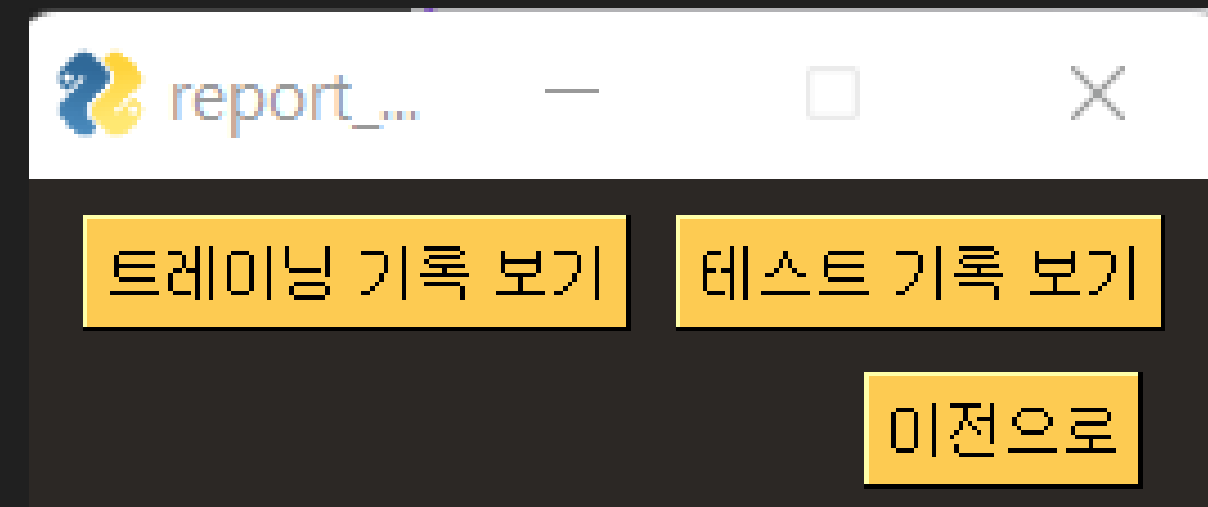
10초 간격으로 수행 횟수를 그래프화

자신의 페이스를 분석하고  
테스트 통과 전략을 세울 수 있음

# 기록 보기 - 테스트



테스트  
기록보기  
선택 시



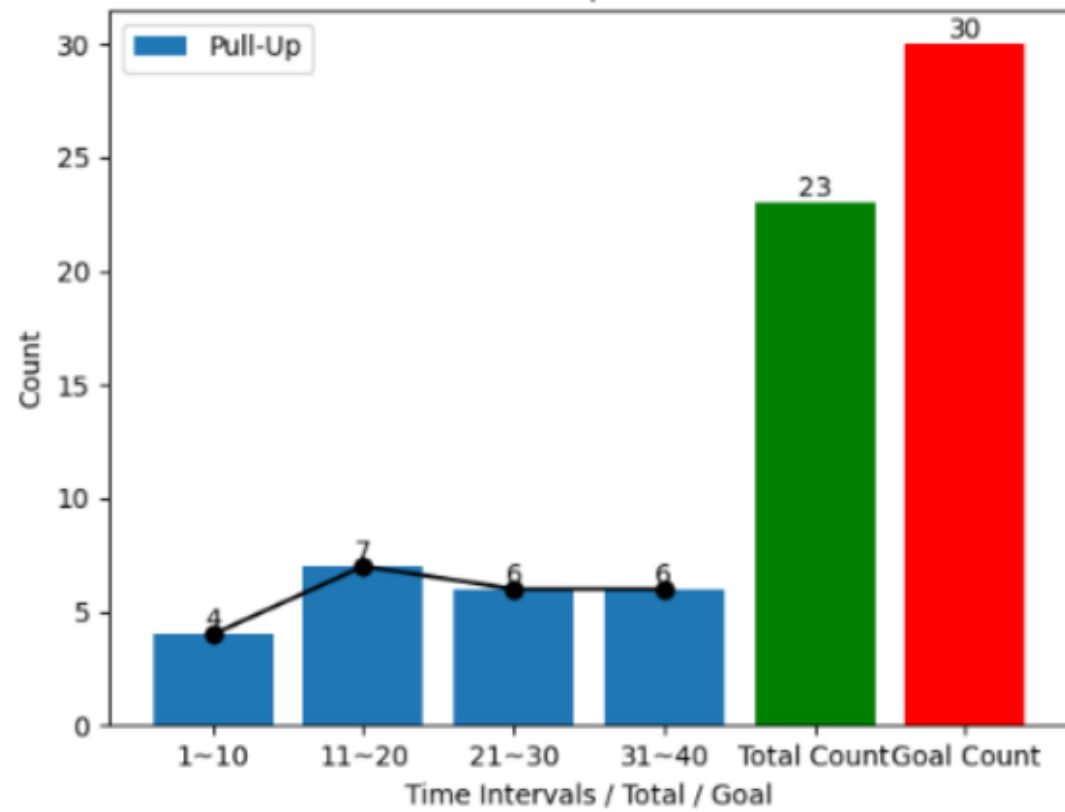
# 기록보기 - 테스트

test\_report\_page

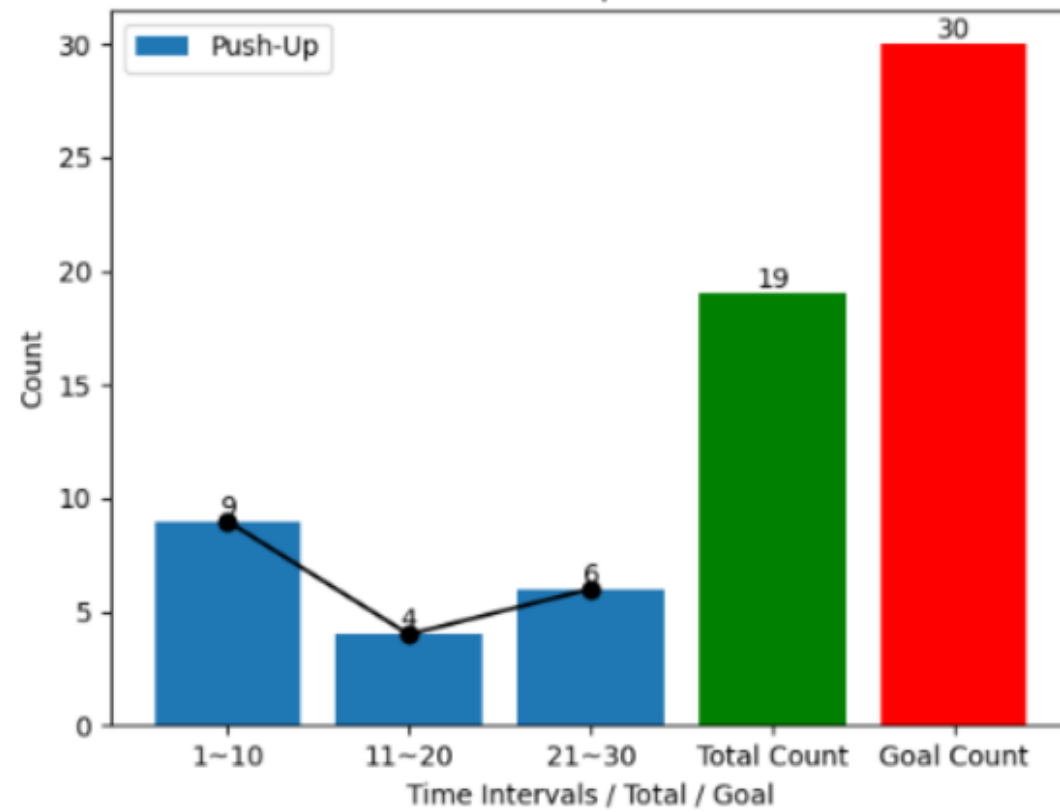
날짜를 선택하세요: 2023-12-16

날짜 선택

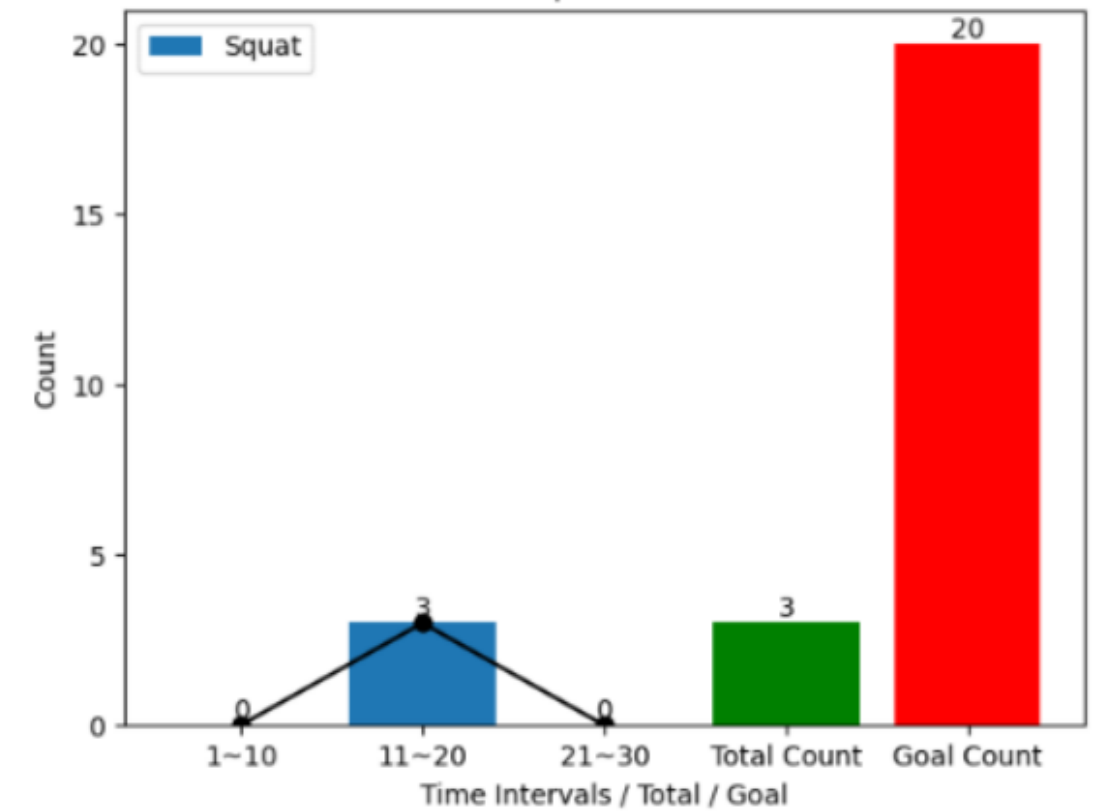
Pull-Up test



Push-Up test



Squat test



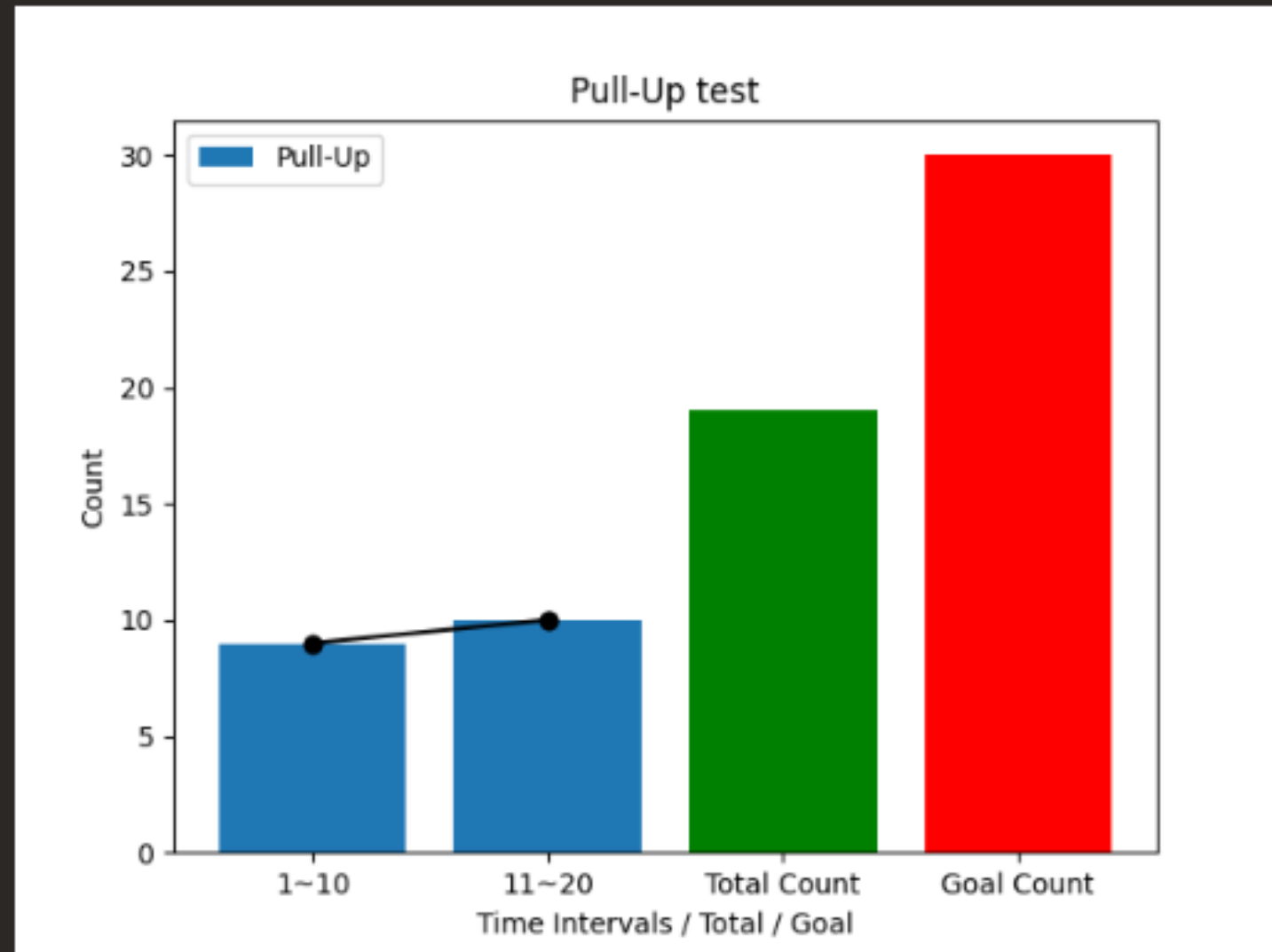
이전으로



# 기록보기 - 테스트

날짜를 선택하세요: 2023-12-12

날짜 선택



```
9 def merge_images(current_date):
10
11     image1_path = f'graph_pictures/test_graph/{current_date}_Pull-Up_test.png'
12     image2_path = f'graph_pictures/test_graph/{current_date}_Push-Up_test.png'
13     image3_path = f'graph_pictures/test_graph/{current_date}_Squat_test.png'
14
15     output_path = f'graph_pictures/test_graph/{current_date}_Merged_Image.png'
16
17     images = [] # 이미지를 저장할 리스트
18
19     # 이미지 파일이 존재하는지 확인하고 리스트에 추가
20     if os.path.exists(image1_path):
21         images.append(cv2.imread(image1_path))
22     if os.path.exists(image2_path):
23         images.append(cv2.imread(image2_path))
24     if os.path.exists(image3_path):
25         images.append(cv2.imread(image3_path))
26
27
28     if not images:
29         print("이미지 파일이 존재하지 않습니다.")
30         return
```

들어주셔서 감사합니다.