
anzin
See the Wonders
Presenter: Edan Chen

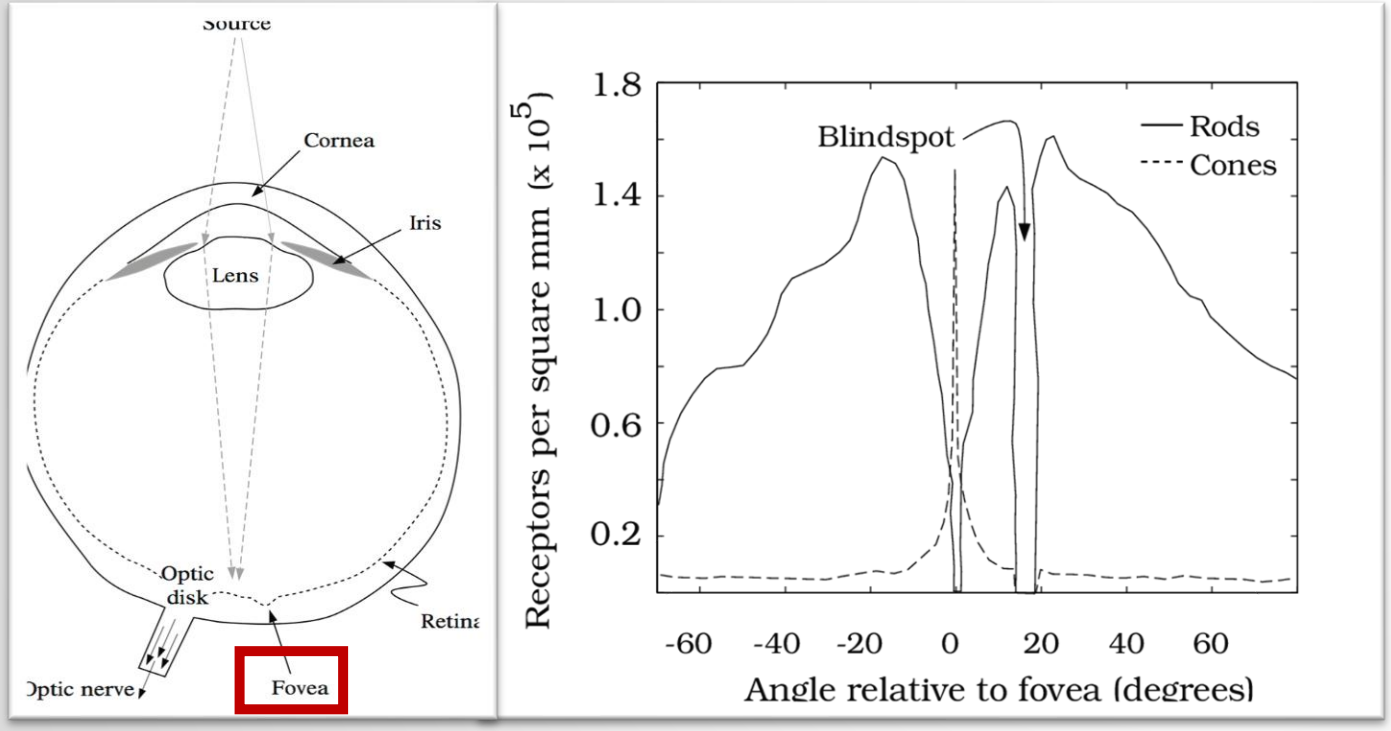
Outline

- 初探眼動追蹤
- 見臻的穿戴式眼動儀技術
- Sol SDK 的詳細介紹
- 使用 Sol SDK 實作的酷炫 Demo
- TAICHI x Ganzin Design Competition 細節

初探眼動追蹤

眼動是什麼? 從眼動當中能得到什麼寶貴的資訊?

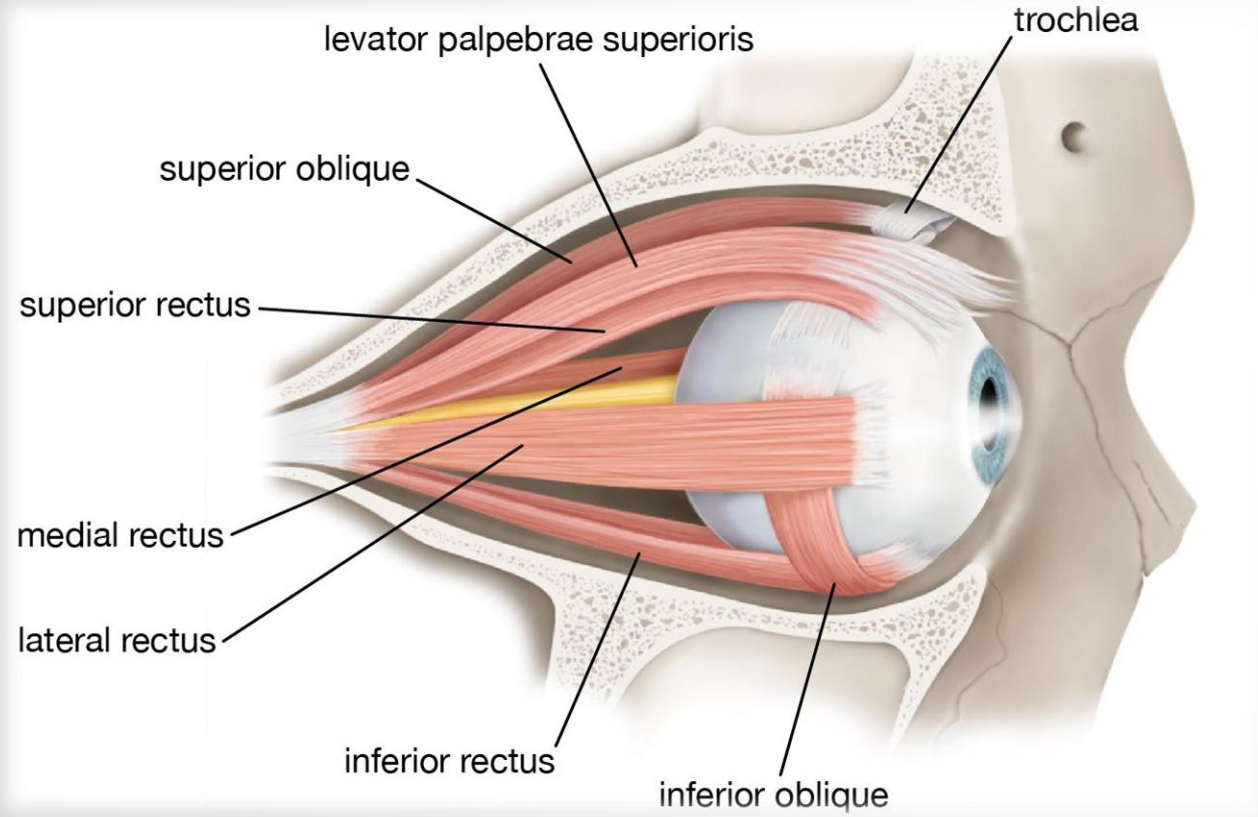
中央窩視覺 (Foveal Vision)



Wandell, B. A. (1995)

- 網膜中的**錐細胞(cones)**處理高空間頻率和顏色相關的視覺刺激。
- 錐細胞主要散布在中央窩(forvea)，因此這個**約兩度**視角範圍內的視力(visual acuity)最佳。
- 一般認知的視線(gaze)位置多指這個區域所接收處理的訊息。

眼外肌 (Extraocular Muscles)



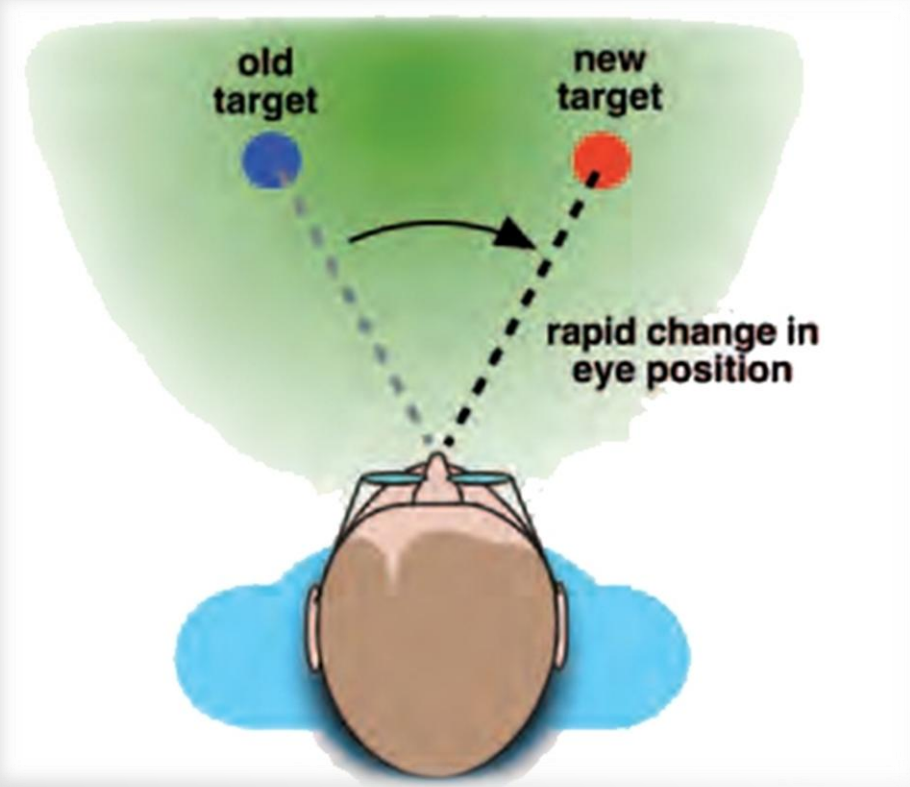
圖片來源: Britannica Inc.

- 控制眼睛的肌肉
  - 垂直方向運動
    - 上直肌(Superior Rectus)
    - 下直肌(Inferior Rectus)
  - 水平方向運動
    - 內直肌(Medial Rectus)
    - 外直肌(Lateral Rectus)
  - 內旋外旋
    - 上斜肌(Superior Oblique)
    - 下斜肌(Inferior Oblique)
  - 張眼閉眼
- 提上眼瞼肌
  (Levator Palpebrae Superioris)

凝視 (Fixation)

- 為了處理特定視覺資訊，視線停留在特定位置一段時間，讓這些資訊可以維持在**中央窩**的範圍內，獲得最清晰的視覺資訊。
- 凝視時間和次數反映處理視覺訊息的**深度**。
- 對特定凝視位置投注**多少注意力**的指標。
- 凝視時間(Fixation duration)短至**數十毫秒**(ms)，長至**1000 ms**或以上也有可能，但多分布在**200-300 ms**之間 (Holmqvist et al., 2011)。
- 凝視行為包含飄移、震顫、微跳視三種眼動事件。

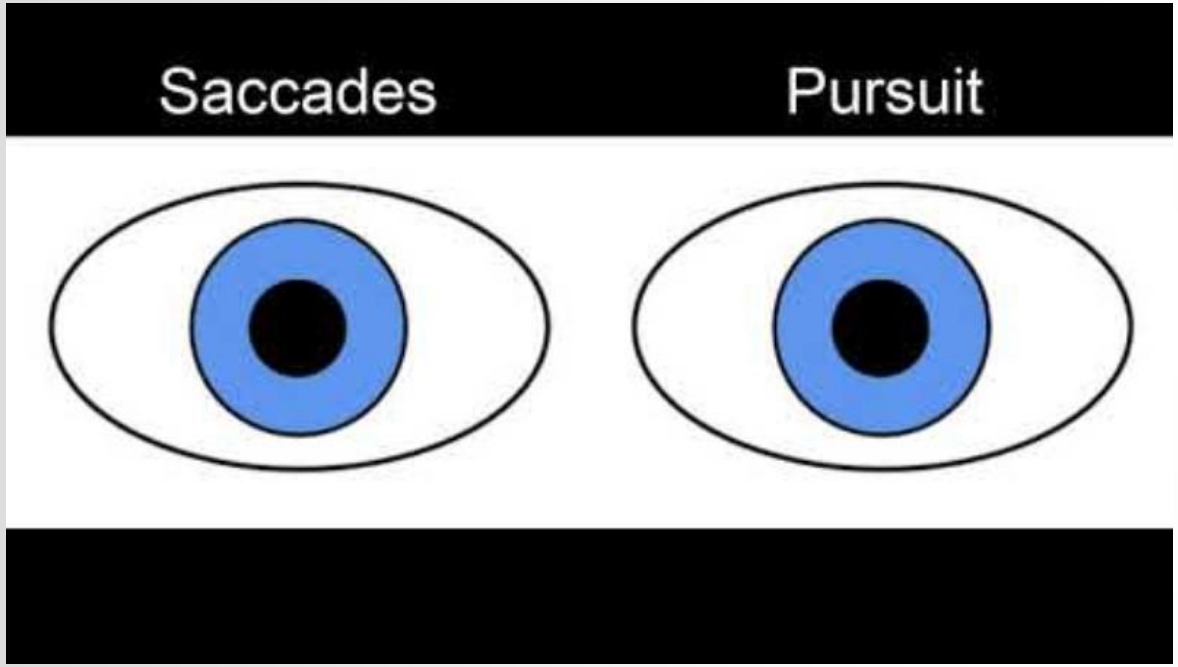
跳視 (Saccade)



圖片來源: https://www.nasafordoctors.co.za/

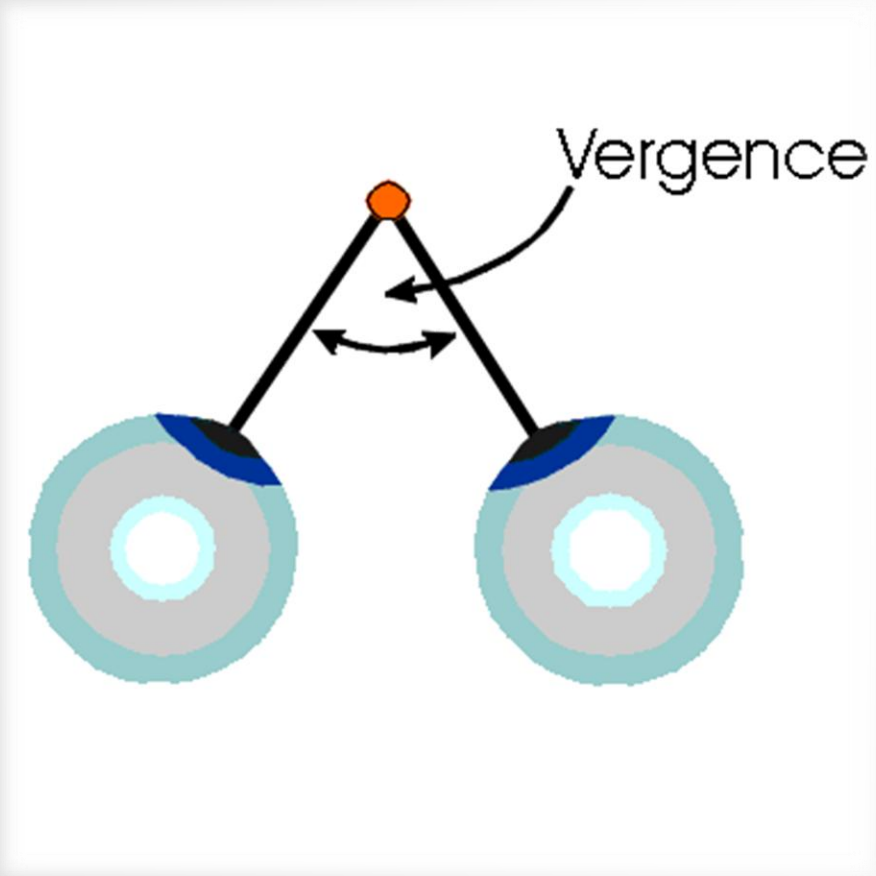
- 眼睛在**不同凝視點之間**的快速移動。
- 最快速且普遍的移動方式。
- 跳視速度最低約在**30°/s**，最高可達**800°/s** 以上 (Zigmond, et al., 1999)。
- 普遍的跳視幅度在**4-20°**之間，跳視時間(saccade duration)多數落在**30 – 80 ms**之間。
- 跳視潛伏期(saccade latency): 決定下一個凝視位置，並啟動跳視的所需時間，通常至少需要**80 ms**。
- 跳視時視覺訊息是**模糊的**(saccadic suppression)。

平滑追瞄移動 (Smooth pursuit)



- 追蹤特定**移動物體**時的眼球運動。
- 速度多在**10-30°/s**，但個體差異大，最高可能達到100 °/s以上 (Zaccara et al., 1991)。
- 當追蹤目標以超過30°/s的速度移動時，觀察者通常會用**跳視**來跟上目標(catch-up saccade, Land & Tatler, 2012)。
- 潛伏期(latency)約**100-150** ms (Bowers et al., 1983)。

輻輳運動 (Vergence)

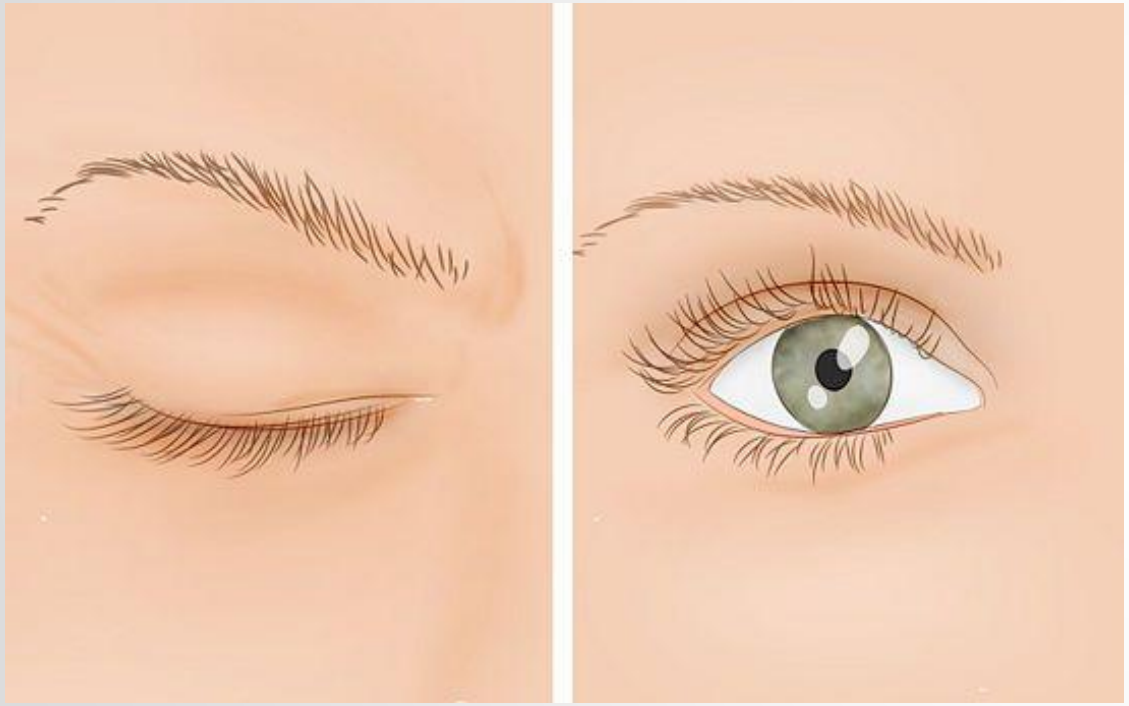


- 追蹤物體的**深度變化**時發生的眼動現象 (Giesel et al., 2019)。
  - 遠到近:
    收斂(convergence)，雙眼向鼻側移動。
  - 近到遠:
    發散(divergence) ，眼睛向顳側移動。
- 除輻輳運動外，所有的眼動行為都是雙眼往同一個方向移動。

前庭反射 (Vestibular ocular reflex, VOR)

- 當凝視特定位置時轉動頭部，眼睛會同步反向運動，使**被凝視的目標穩定地落在網膜中心位置**。
- 快速且無意識的**反射行為**。
- 眼睛轉動速度和頭動速度一致(Land & Tatler, 2012)。

眨眼 (Blink)



圖片來源: 20x20.com

- 並非眼睛運動，但是**眨眼頻率**(blink rate)是很常使用的行為指標。
- 和**多巴胺**活動水準有關(Karson, 1983)
- 普遍認為和外在**注意力、疲勞(fatigue)**程度有關(Maffei & Angrilli, 2018)。
  - 注意力上升，眨眼頻率下降。
  - 疲勞程度上升，眨眼時長(duration)和頻率上升。
- 因為**多巴胺**活動水準和**創意思考**能力(creativity)有正相關(Ashby, Isen, & Turken, 1999)，眨眼頻率也可能作為問題解決能力的指標(Chermahini & Hommel, 2010)。

瞳孔縮放 (Pupillometry)

圖片來源: Neuroscience News

- 受**自律神經系統**調節，可以當作生理激發(arousal)程度的測量指標。
- **情緒**激發程度越高，瞳孔越大(Wang et al., 2017)。
- **認知努力**(cognitive effort)程度越高，瞳孔越大(van der Wel & van Steenbergen, 2018)。
- 瞳孔大小受亮度影響大，環境亮時瞳孔變小，暗時瞳孔放大，因此研究時和瞳孔相關的認知及情緒因子時需控制刺激亮度。

常見眼動事件整理

- **凝視和跳視**是跟認知與學習關聯較高的眼動事件 (陳學志等人，2010)

眼動事件	英文對照	功能/相關認知處理
凝視	**Fixation**	**視覺注意力**
跳視	**Saccade**	**注意力轉移**
平滑追瞄移動	Smooth Pursuit	物體追蹤
輻輳運動	Vergence	深度視覺
前庭反射	Vestibular ocular reflex	彌補頭部轉動
眨眼	Blink	疲勞、注意力狀態
瞳孔縮放	Pupillometry	認知努力、情緒激發

見臻的穿戴式眼動儀技術

一探見臻眼動技術的奧秘！

ganzin

See the Wonders

Micro Eye Tracking Technology:
The most easy-to-install eye tracking solution on the market

2016

MOEA Value Creation Project

2018

Spun-off from National Taiwan University (Q1)

2021

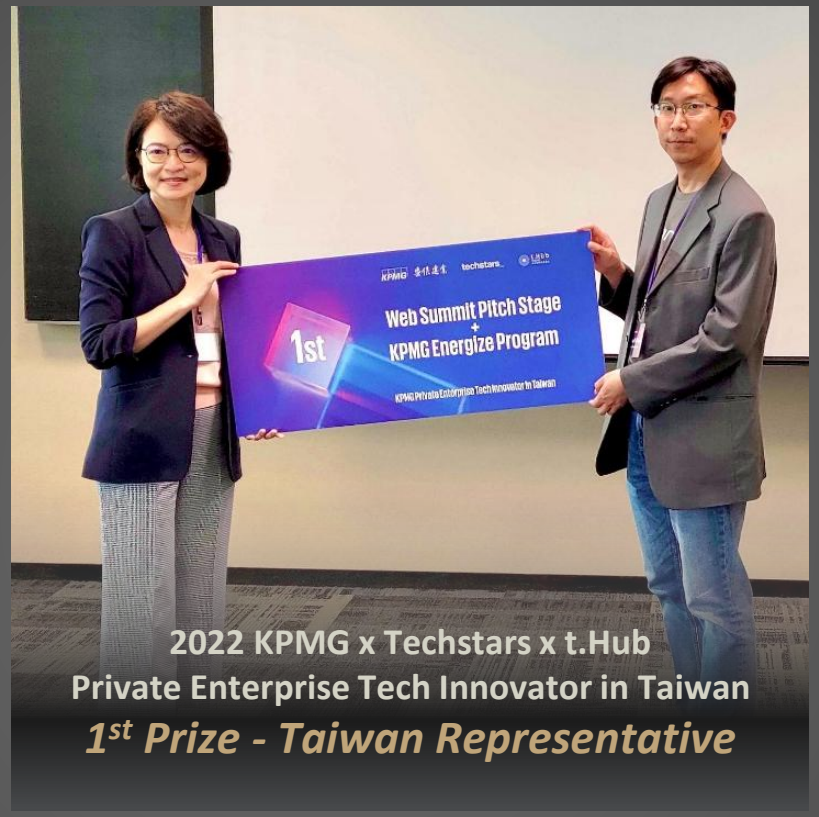
- Pre-A round fund-raising (Q4)
- Mass Production of our software solution with integration partners

2023-2024

- **2023/6** Sol official release
- Total **27** customers and deliver **36** Sol Bundles over **4** countries
- **2** Conference papers

Recent Awards

InnoVEX 2022
Taiwania Capital Innovation Award



2022 KPMG x Techstars x t.Hub
Private Enterprise Tech Innovator in Taiwan
1st Prize - Taiwan Representative

Taiwan Innotech Expo
Tech Innovation Excellence Award

2020 Neo Star
年度最具成長潛力新創

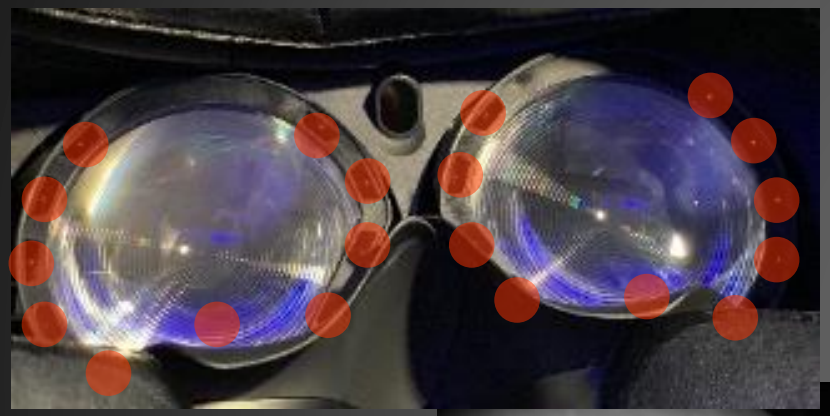
XR Golden Awards
XR 創星金點大賞

EDISON AWARDS
INNOVATIONS & INNOVATORS

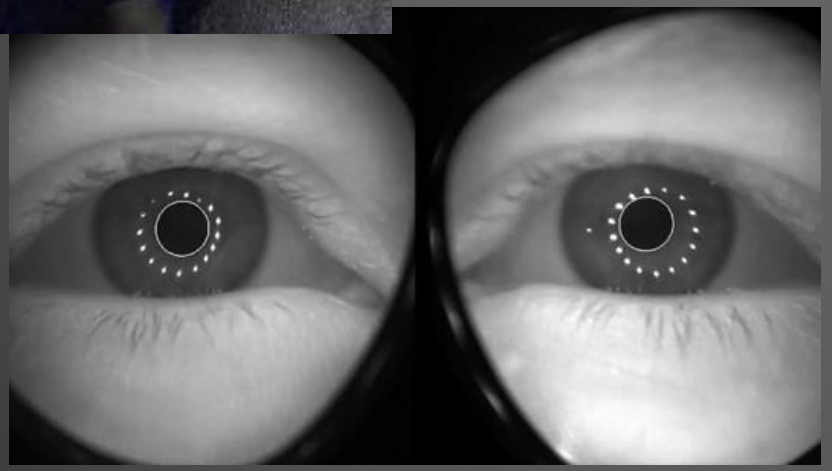
CES INNOVATION AWARDS 2020 HONOREE

Qualcomm® Innovate in Taiwan Challenge

Existing Solution – Corneal Reflection



: LEDs and sensors



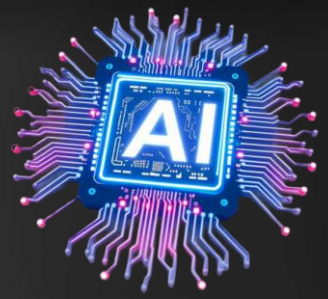
tobii

7INVENSUN

Too **Complex** and **Restricted!**

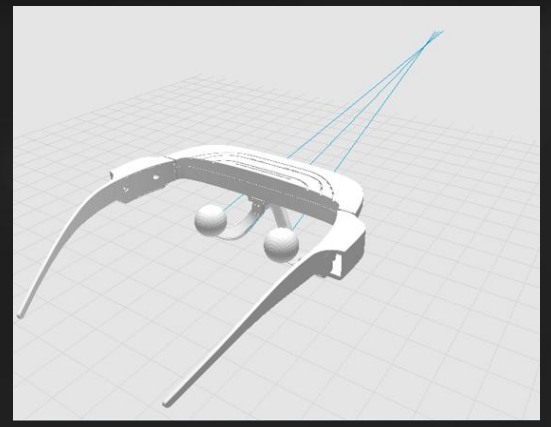
- **Low flexibility**
- **Unsuitable for compact headset**
- **High components requirements**

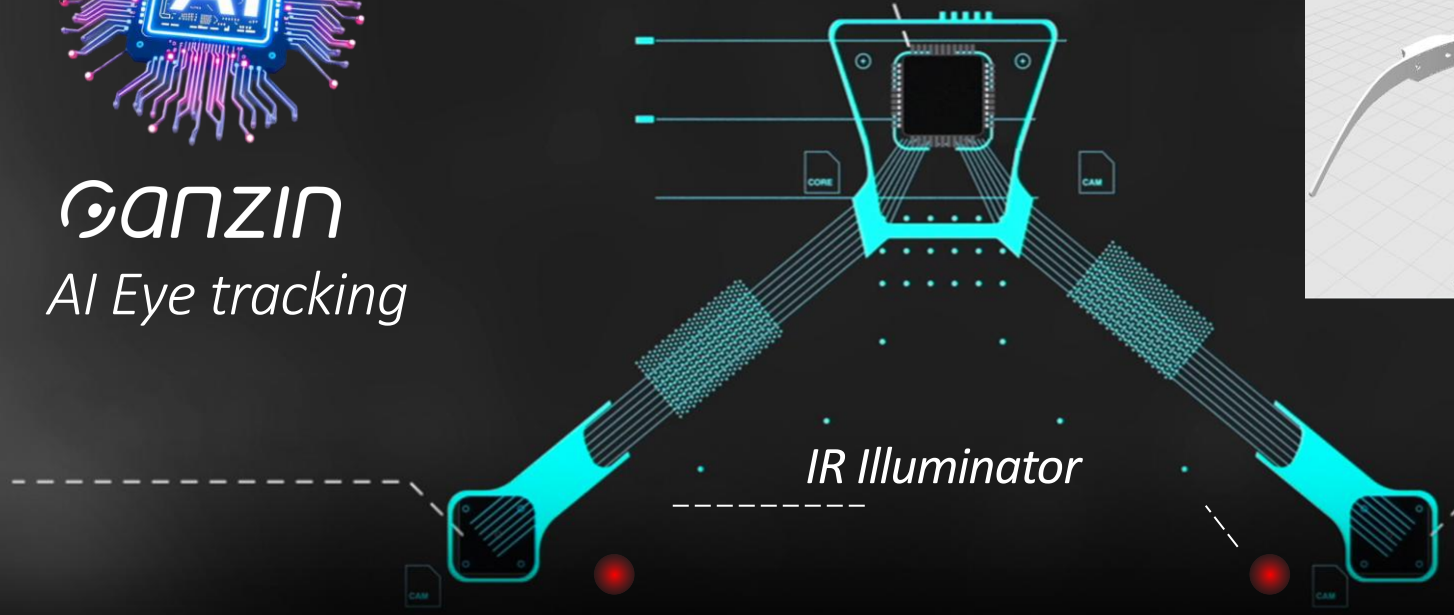
AI-Powered Eye Tracking Solution



ganzin

AI Eye tracking





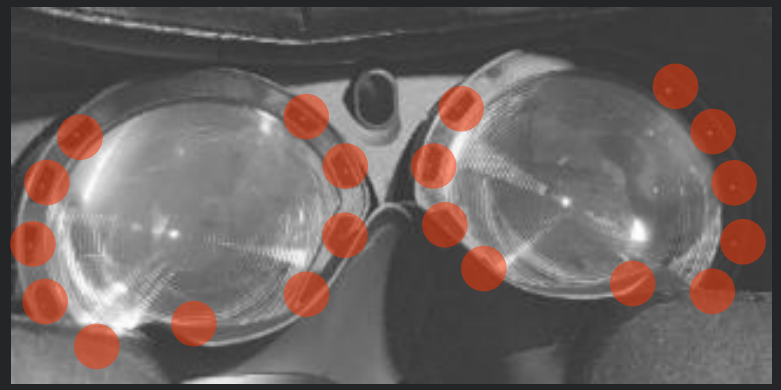
IR Illuminator

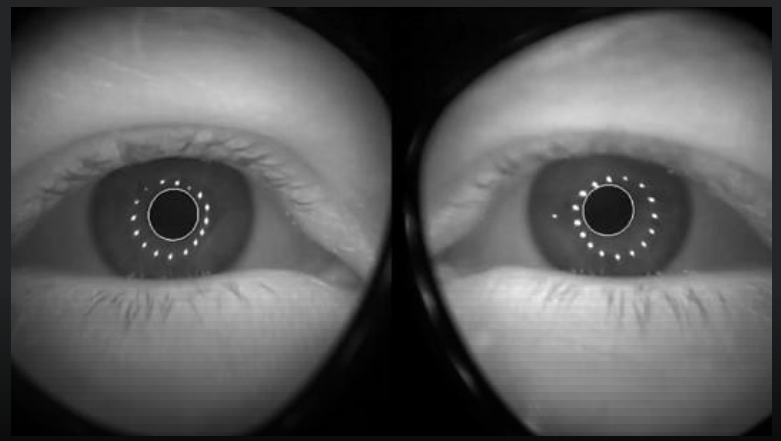
ganzin *EyeSensor*

AI-Powered Eye Tracking Solution with ganzin Eye Database

1.5°
Accurate
Gaze Detection

99%
Robust
Detection





Ganzin Aurora

● : LEDs

● : Camera Sensors

3x

Fewer
Components

2x

Lower
BOM Cost

Highly
Flexible

Ganzin Sol Glasses :
Detect Visual Attention in Natural Environments

Advanced Demo with Sol SDK

android

- Slow developing process
- Complex environment
- Steep learning curve

vs.

python

- Everyone knows how to use
- TOP programming language in the world!
- Many other packages can be integrated (PsychoPy, OpenAI API, etc.)

Fast eye tracking application development!

Python SDK is ready!

Sol Glasses

Understand human attention in real-world

Glasses & Companion	Analysis Software – Caelus
Eye Tracking Feature - **110±10 Hz** gaze sampling rate - **3D gaze, pupil size**, blink info - 1328 x 1200 wide FOV scene cam - Calibration-free / 1-pt calibration - (Optional) Perspiration lens kit **Companion Phone** - Samsung Galaxy S23 (128 GB) - Chronus	**Key Feature** - Project management - Gaze & on-scene event preview - Automatic gaze remapping **Analyzed Data** **- Heatmap** **- Gaze plot** **- AOI analysis** -Event & Raw data

Sol SDK 的詳細介紹

來了解這個可以讓眼動 real-time 運算的工具吧！

Advanced Demo with Sol SDK

android

- Slow developing process
- Complex environment
- Steep learning curve

vs.

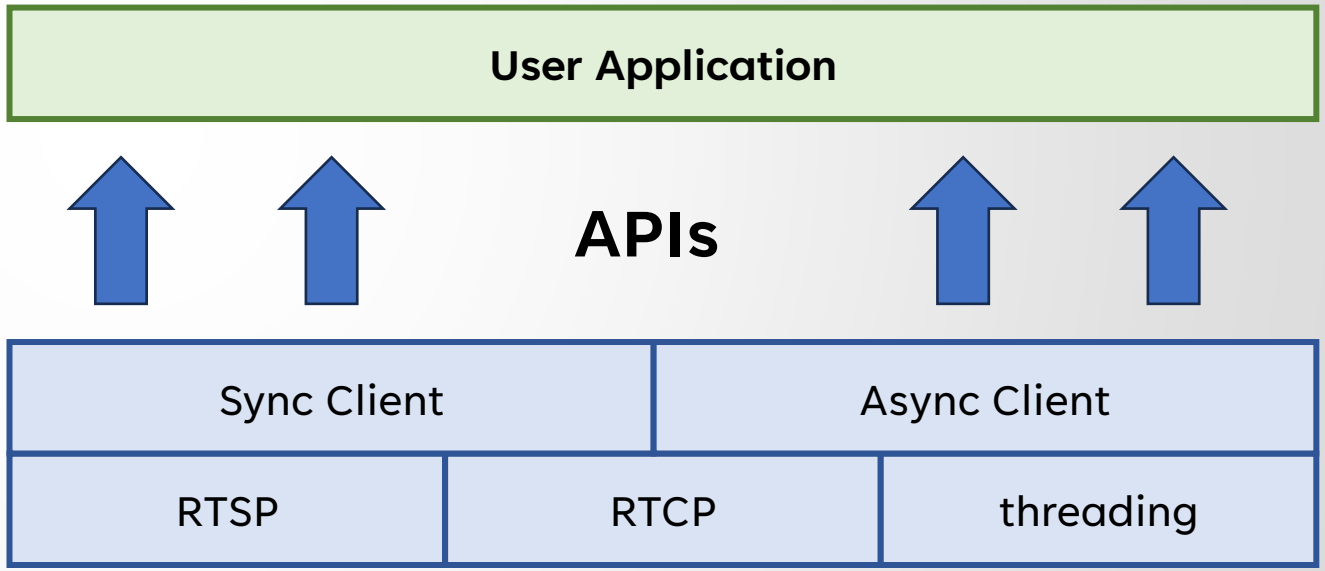
python

- Everyone knows how to use
- TOP programming language in the world!
- Many other packages can be integrated (PsychoPy, OpenAI API, etc.)

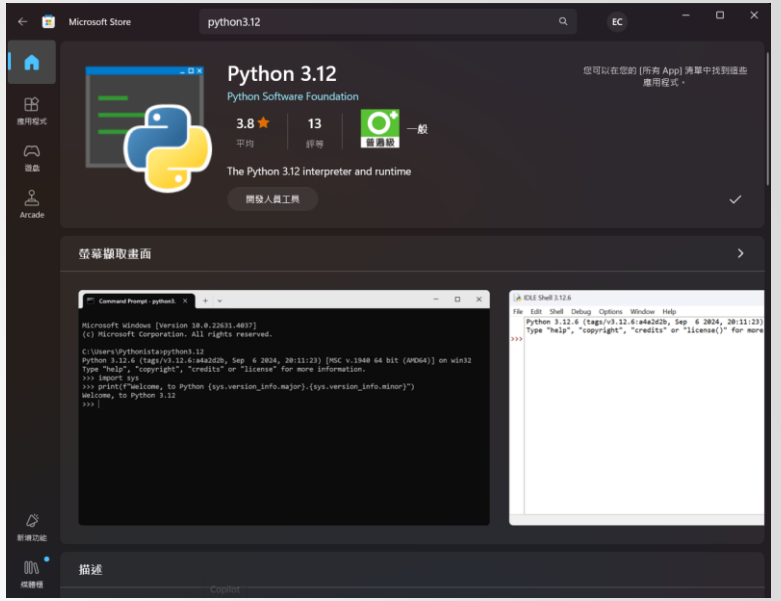
Fast eye tracking application development!

Sol SDK 架構

- Communication protocol: RTSP (Real Time Streaming Protocol), RTCP (Real-time Transport Control Protocol)
- Muti-threading: Python threading library
- Provides two kinds of clients
  - Sync Client
    - Easy to program
    - Good start point
    - Poor performance due to blocking
  - Async Client
    - More complicate
    - More real-time demonstration
- Let’s learn the Sol SDK by going through the example code!!!



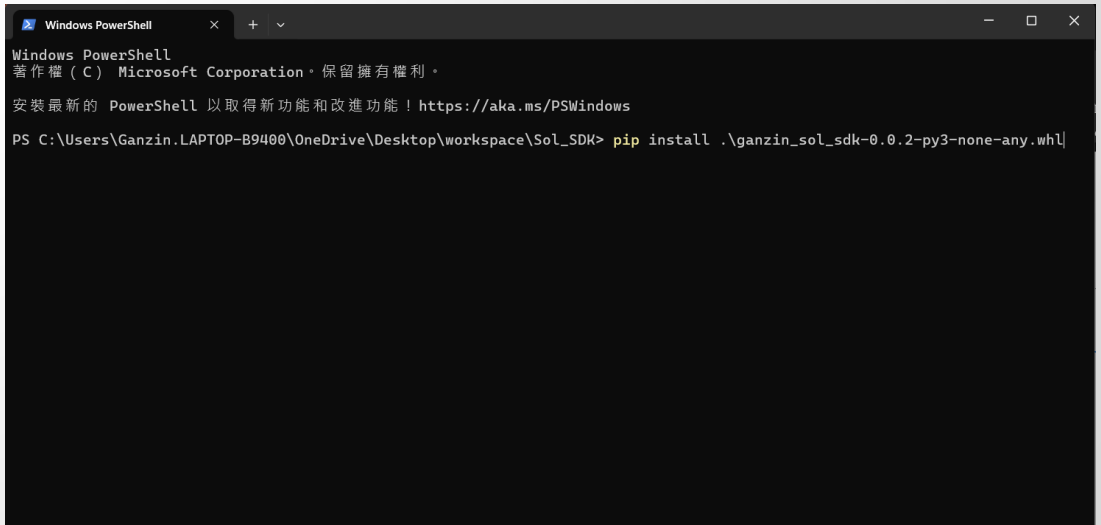
Sol SDK 安裝



- Install Python3 (suggested version: 3.12)
  - https://www.microsoft.com/store/productId/9NCVDN91XZQP?ocid=pdpshare
- Tutorial for installing Python3 with Anaconda on windows
  - https://www.youtube.com/watch?v=4DQGBQMvwZo

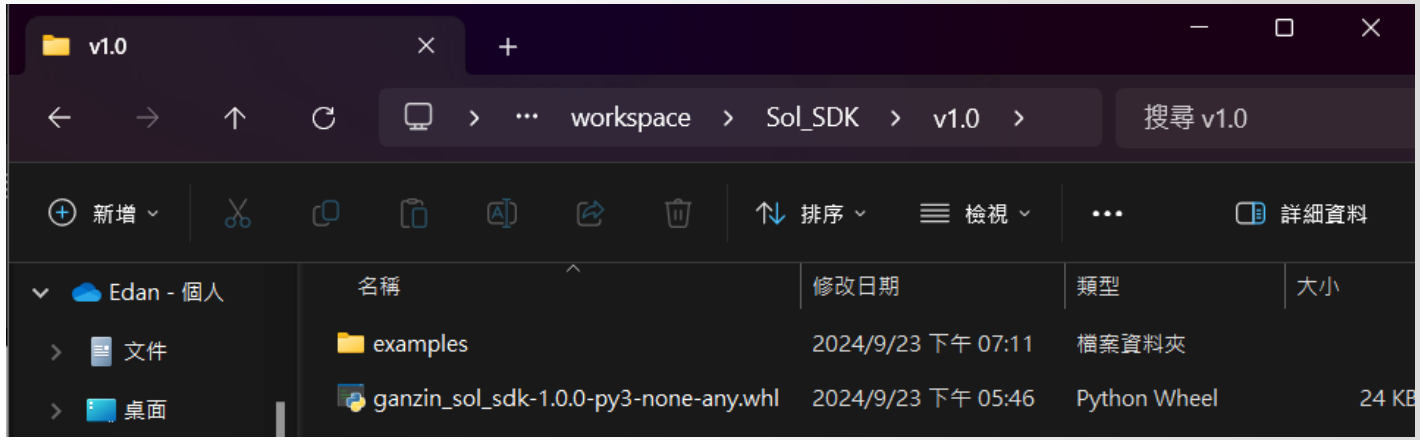
- Download Sol SDK (v1.1.1)
  - whl: https://drive.google.com/file/d/1uSti9pXNiON29pv9qIWC0QnsIf7D8d9p/view?usp=drive_link
  - examples.zip: https://drive.google.com/file/d/1UgQnnOuxsZjPXSgIPZlxF-WOsaKLD8Ym/view?usp=drive_link
  - Document: https://drive.google.com/file/d/1CWCOxSLSHWKYiONA0TkwncyMkE46B8d-/view?usp=drive_link

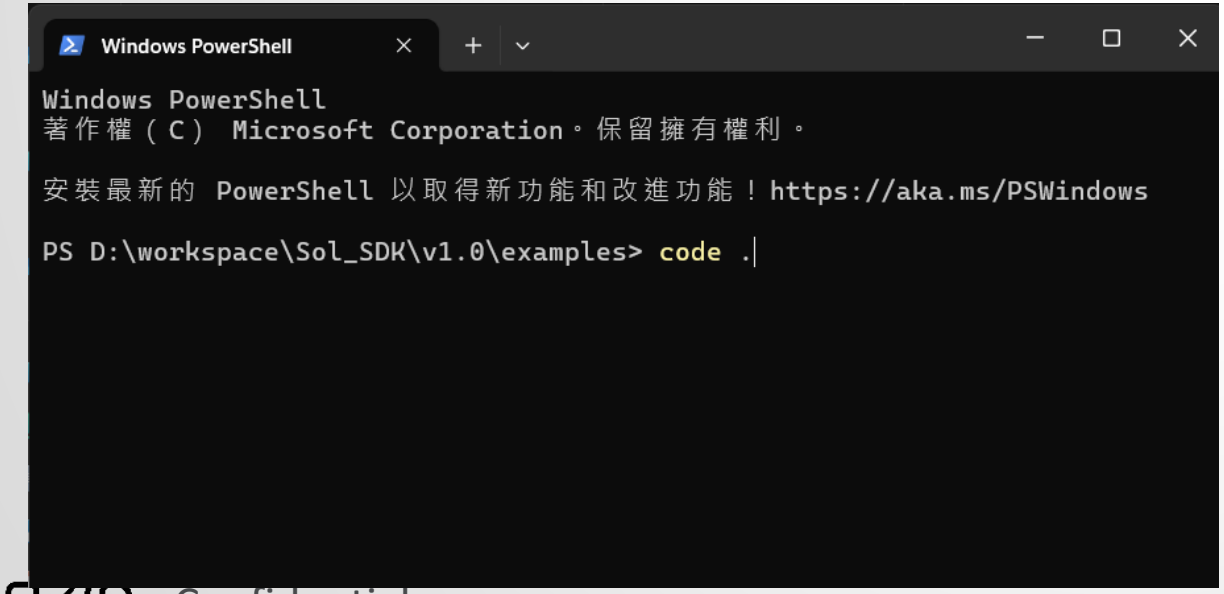
- Install SDK
  - In the Sol SDK directory
    - Right click to open the powershell
  - *$ pip install .\ganzin_sol_sdk-1.1.1-py3-none-any.whl*
- Install other packages will used later
  - *$ pip install requests opencv-python*



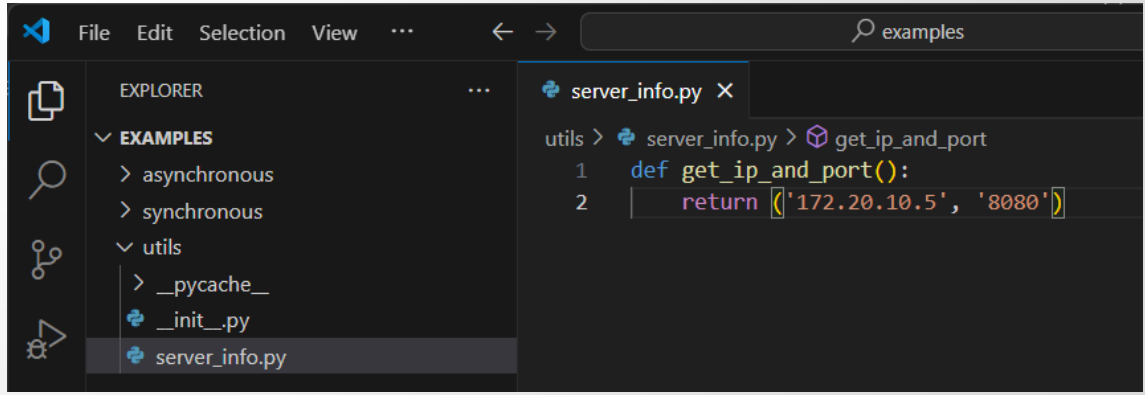
範例程式說明

- Extract examples.zip
  - Check the *examples* directory exist
- Open vscode in the examples directory
  - Go into the examples directory
  - Right click and open powershell
  - *$ code .*



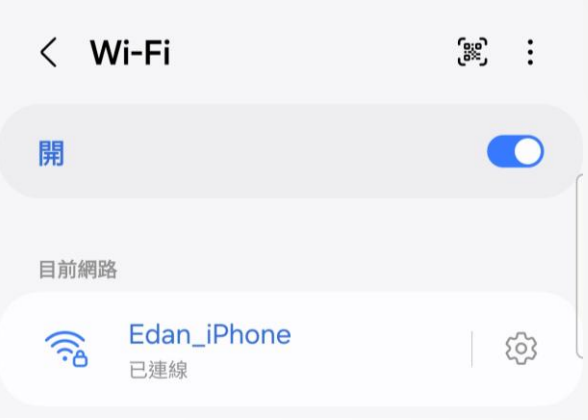


- Check the hierarchy in vscode

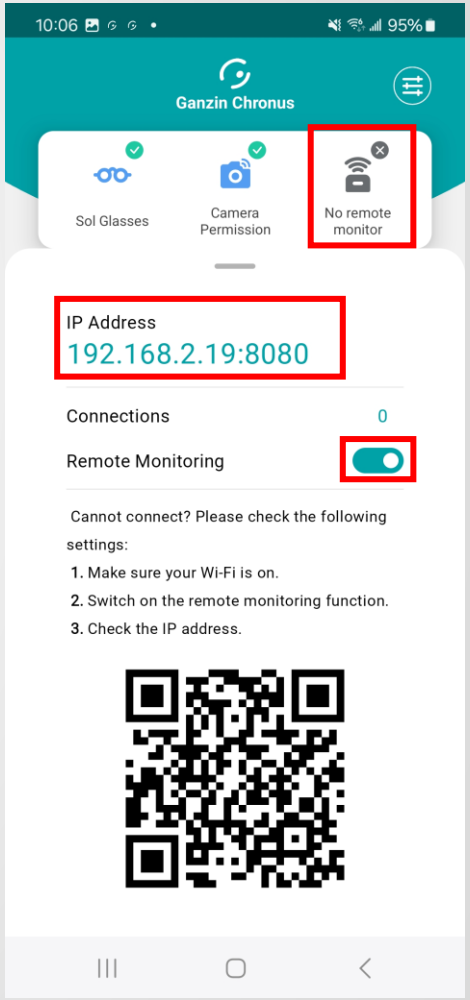


環境設定 (1/2)

- Connect Sol Glasses's phone and your PC under the same AP (e.g., your own hotspot)

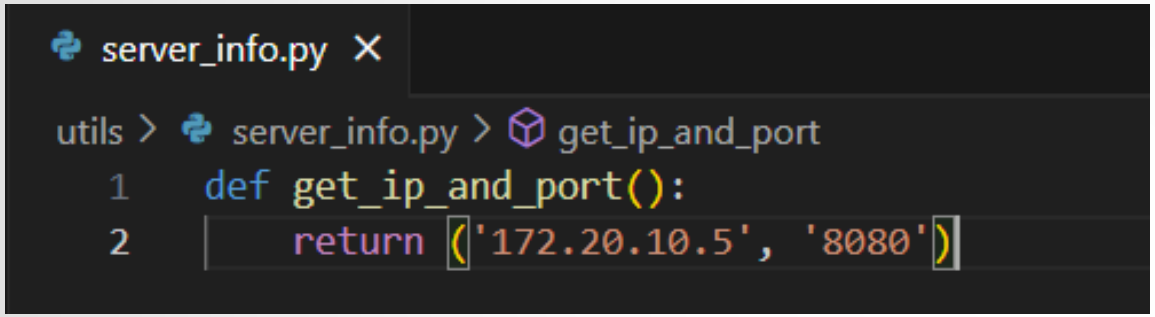


- Open the Chronus app and click the remote monitor icon
  - Get the IP and Port from the phone screen
  - The remote monitoring toggle needs to be on



環境設定 (2/2)

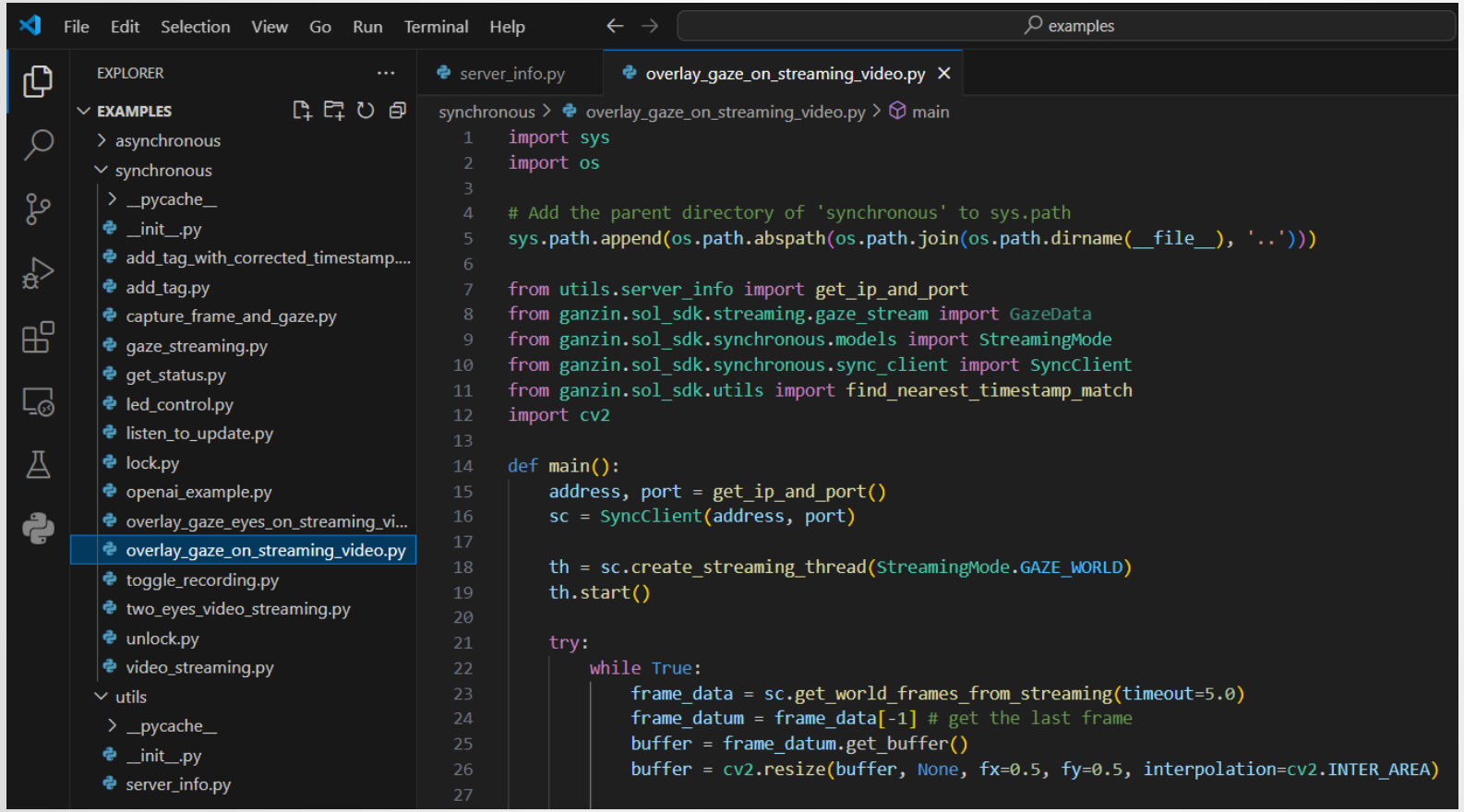
- Fill the ip and port into the *utils/server_info.py*



- Please notice that when using the Sol SDK, the Chronus app need to stay on the foreground

眼動影片範例 – Sync Client 版本

- Open the *synchronous/overlay_gaze_on_streaming_video.py* example



深入解析程式碼 (1/4)

- Import modules
  - sys, os: to make the server_info importable
  - ganzin.sol_sdk: the sdk to remote control the Sol Glasses
  - cv2: visualize the result get from the Sol SDK

```python
import sys
import os

# Add the parent directory of 'synchronous' to sys.path
sys.path.append(os.path.abspath(os.path.join(os.path.dirname(__file__), '..')))

from utils.server_info import get_ip_and_port
from ganzin.sol_sdk.streaming.gaze_stream import GazeData
from ganzin.sol_sdk.synchronous.models import StreamingMode
from ganzin.sol_sdk.synchronous.sync_client import SyncClient
from ganzin.sol_sdk.utils import find_nearest_timestamp_match
import cv2
```

深入解析程式碼 (2/4)

- Sync Client setup
  - Get the address and port from *get_ip_and_port()*
  - Create a Sync Client using the address and port
  - Create a streaming thread for the Sync Client
  - Start the thread

```
14  def main():
15      address, port = get_ip_and_port()
16      sc = SyncClient(address, port)
17
18      th = sc.create_streaming_thread(StreamingMode.GAZE_WORLD)
19      th.start()
20
```

深入解析程式碼 (3/4)

- Getting the gazes and the frames
  - Create an infinite while loop to keep getting the gazes and frames
  - Get the frame data from *get_world_frames_from_streaming()*
  - Get the gazes data from *get_gazes_from_streaming()*
  - Get the exactly gaze using the timestamp of the frame

```
try:
    while True:
        frame_data = sc.get_world_frames_from_streaming(timeout=5.0)
        frame_datum = frame_data[-1] # get the last frame
        buffer = frame_datum.get_buffer()

        gazes = sc.get_gazes_from_streaming(timeout=5.0)
        gaze = find_nearest_timestamp_match(frame_datum.get_timestamp(), gazes)
```

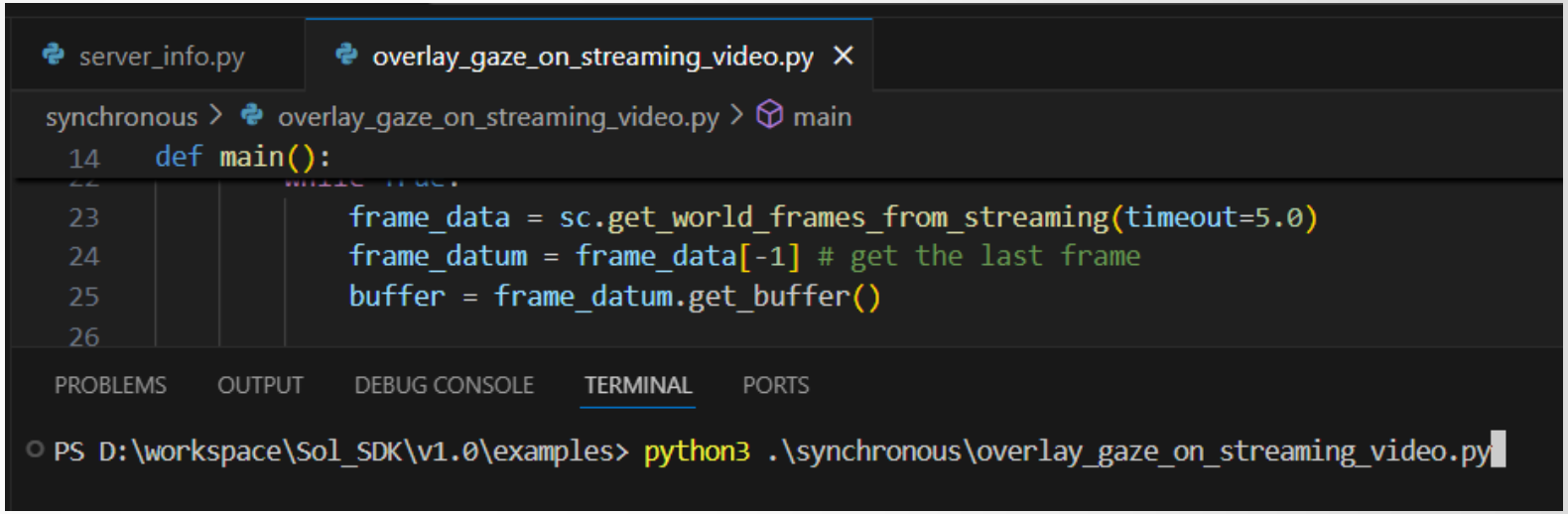
深入解析程式碼 (4/4)

- Show the result using cv2
  - Resize the buffer as the resolution of the Sol Glasses front facing camera is 1328x1200 (exceed 1080)
  - Calculate the center of the circle
  - Set the circle's radius, color, and thickness
  - Draw the circle using *cv2.circle()*
  - Show the result using *cv2.imshow()*

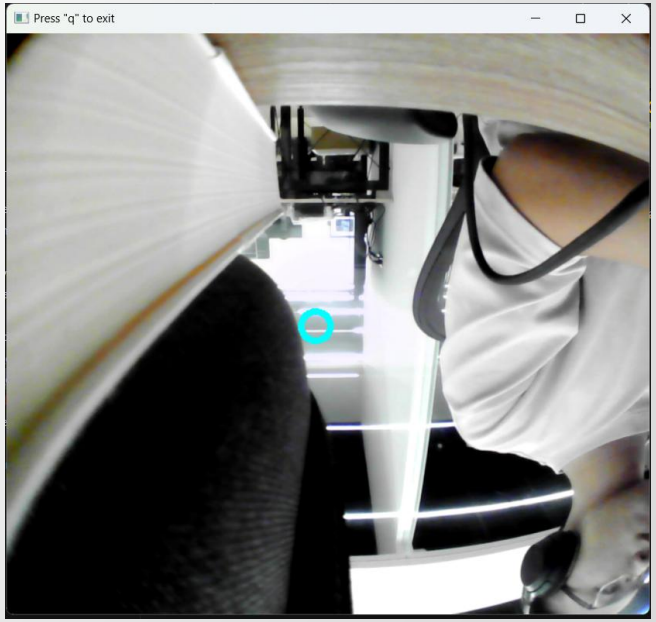
```
30          buffer = cv2.resize(buffer, None, fx=0.5, fy=0.5, interpolation=cv2.INTER_AREA)
31          center = (int(gaze.combined.gaze_2d.x/2), int(gaze.combined.gaze_2d.y/2))
32          radius = 15
33          bgr_color = (255, 255, 0)
34          thickness = 5
35          cv2.circle(buffer, center, radius, bgr_color, thickness)
36
37          cv2.imshow('Press "q" to exit', buffer)
38          if cv2.waitKey(1) & 0xFF == ord('q'):
39              break
```

執行程式

- Open terminal from the vscode

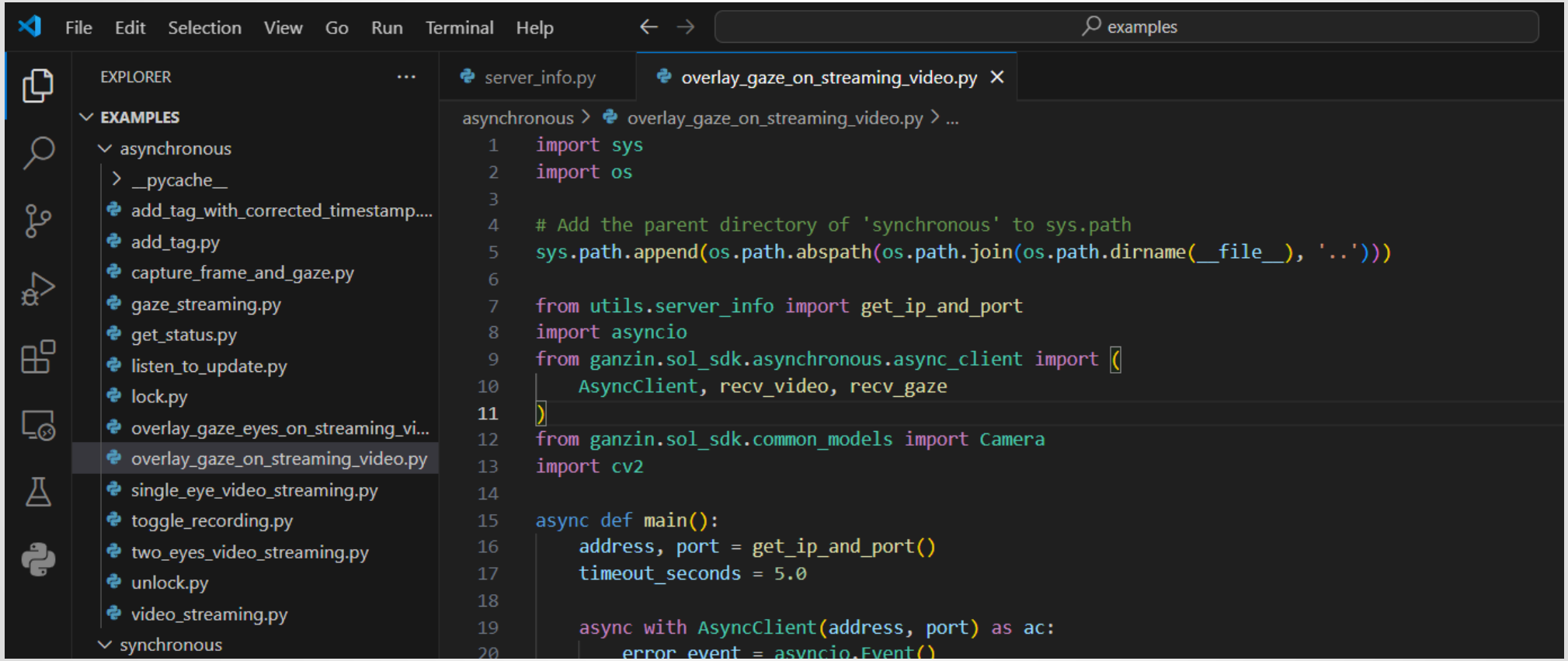


- *$ python3 .\synchronous\overlay_gaze_on_streaming_video.py*
  - Check whether the streaming window appear
  - Press ‘q’ to exit



眼動影片範例 – Async Client 版本

- Open the *asynchronous/overlay_gaze_on_streaming_video.py* example



深入解析程式碼 (1/5)

- Module import
  - asyncio: modules for asynchronous programming, enables the execution of I/O-bound tasks without blocking the execution of other tasks.
  - ganzin.sol_sdk: import the AsyncClient

```python
import sys
import os

# Add the parent directory of 'synchronous' to sys.path
sys.path.append(os.path.abspath(os.path.join(os.path.dirname(__file__), '..')))

from utils.server_info import get_ip_and_port
import asyncio
from ganzin.sol_sdk.asynchronous.async_client import (
    AsyncClient, recv_video, recv_gaze
)
from ganzin.sol_sdk.common_models import Camera
import cv2
```

深入解析程式碼 (2/5)

- Main function
  - Create two queues: frames, gazes, for putting the frame/gaze into the queue
  - Create two tasks: collect_video_task, collect_gaze_task, for putting the frame get from the AsyncClient into the queue
  - Await for the *draw_gaze_on_frame()*, draw the frame and gaze when there is gaze/frame in the queues

```python
async def main():
    address, port = get_ip_and_port()
    timeout_seconds = 5.0

    async with AsyncClient(address, port) as ac:
        error_event = asyncio.Event()

        frames = asyncio.Queue(1)
        collect_video_task = asyncio.create_task(keep_last_video_frame(ac, frames, error_event))

        gazes = asyncio.Queue()
        collect_gaze_task = asyncio.create_task(collect_gaze(ac, gazes, error_event, timeout_seconds))

        try:
            await draw_gaze_on_frame(frames, gazes, error_event, timeout_seconds)
        finally:
            collect_video_task.cancel()
            collect_gaze_task.cancel()
```

深入解析程式碼 (3/5)

- *keep_last_video_frame()*
  - async for loop for getting frame from *recv_video()*
    - This will be blocked by the IO
  - Remove the item in queue if full ASAP, the queue has only one seat
  - Put the frame into the queue ASAP
- *collect_gaze()*
  - async for loop for getting gaze from *rec_gaze()*
  - put the gaze into the queue
    - The queue can preserve multiple gazes

```python
async def keep_last_video_frame(ac: AsyncClient, queue: asyncio.Queue, error_event: asyncio.Event) -> None:
    async for frame in recv_video(ac, Camera.WORLD):
        if error_event.is_set():
            break

        if queue.full():
            queue.get_nowait()
        queue.put_nowait(frame)

async def collect_gaze(ac: AsyncClient, queue: asyncio.Queue, error_event: asyncio.Event, timeout) -> None:
    try:
        async for gaze in recv_gaze(ac):
            if error_event.is_set():
                break

            await asyncio.wait_for(queue.put(gaze), timeout=timeout)
    except Exception as e:
        error_event.set()
```

深入解析程式碼 (4/5)

- *draw_gaze_on_frame()*
  - Get the frame from the *frame_queue*
  - Get the gaze from the *gaze_queue*
  - Draw the gaze on the frame using cv2

```python
async def draw_gaze_on_frame(frame_queue, gazes, error_event: asyncio.Event, timeout):
    while not error_event.is_set():
        frame = await get_video_frame(frame_queue, timeout)
        gaze = await find_gaze_near_frame(gazes, frame.get_timestamp(), timeout)
        frame_buffer = frame.get_buffer()
        frame_buffer = cv2.resize(frame_buffer, None, fx=0.5, fy=0.5, interpolation=cv2.INTER_AREA)

        center = (int(gaze.combined.gaze_2d.x/2), int(gaze.combined.gaze_2d.y/2))
        radius = 15
        bgr_color = (255, 255, 0)
        thickness = 3
        cv2.circle(frame_buffer, center, radius, bgr_color, thickness)

        cv2.imshow('Press "q" to exit', frame_buffer)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            return
```

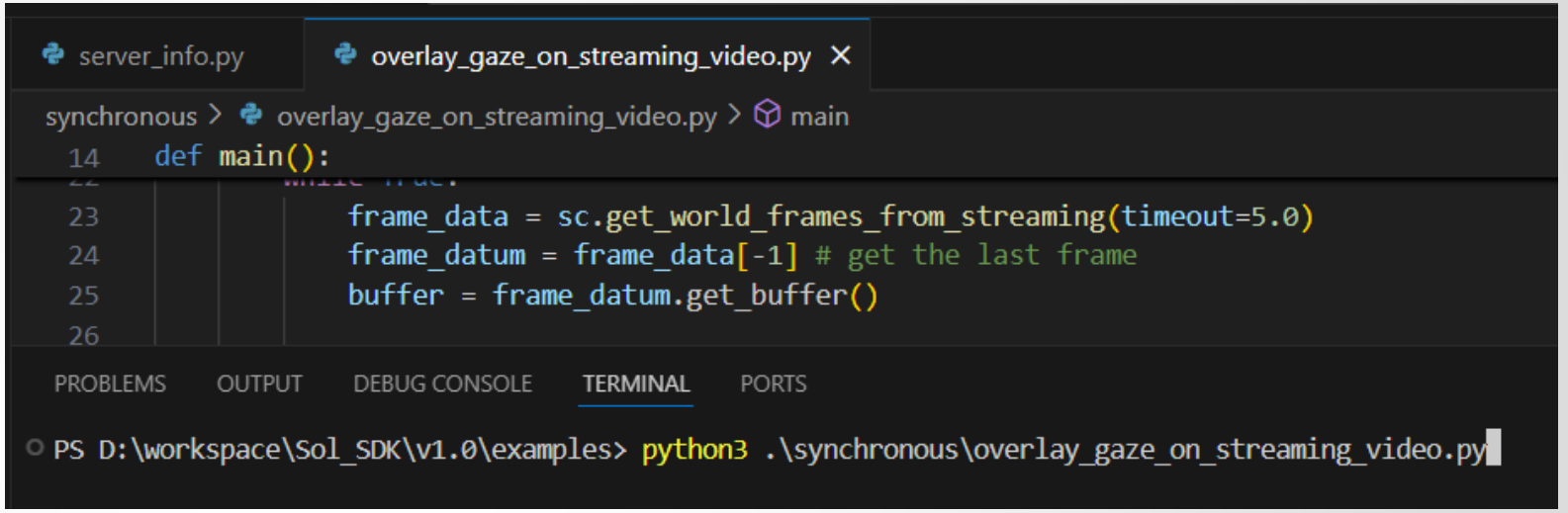
深入解析程式碼 (5/5)

- *get_video_frame()*
  - Async function for getting the frame out of the queue
- *find_gaze_near_frame()*
  - Async function for finding the most appropriate gaze in the gaze queue

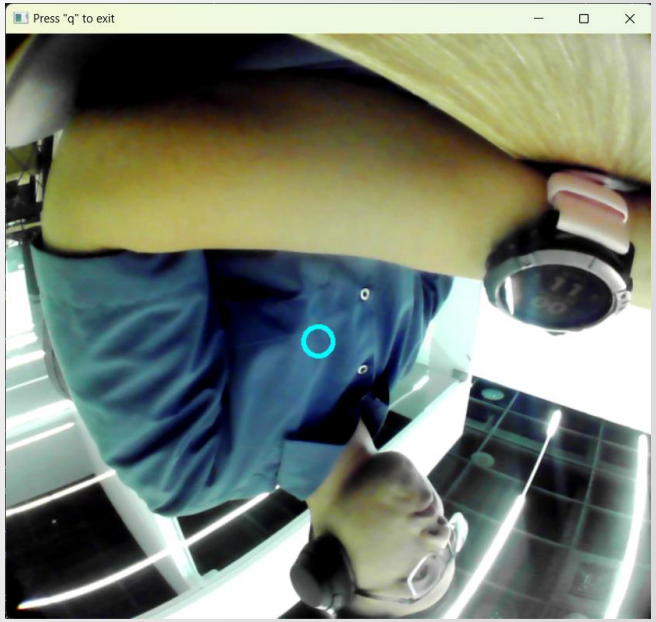
```
70  async def get_video_frame(queue, timeout):
71      return await asyncio.wait_for(queue.get(), timeout=timeout)
72
73  async def find_gaze_near_frame(queue, timestamp, timeout):
74      item = await asyncio.wait_for(queue.get(), timeout=timeout)
75      if item.get_timestamp() > timestamp:
76          return item
77
78      while True:
79          if queue.empty():
80              return item
81          else:
82              next_item = queue.get_nowait()
83              if next_item.get_timestamp() > timestamp:
84                  return next_item
85              item = next_item
```

執行程式

- Open terminal from the vscode



- *$ python3 .\asynchronous\overlay_gaze_on_streaming_video.py*
  - Check whether the streaming window appear
  - Press ‘q’ to exit
  - Same result from the synchronous counterpart



使用 Sol SDK 實作的酷炫 Demo

從別人的作品當中尋找靈感

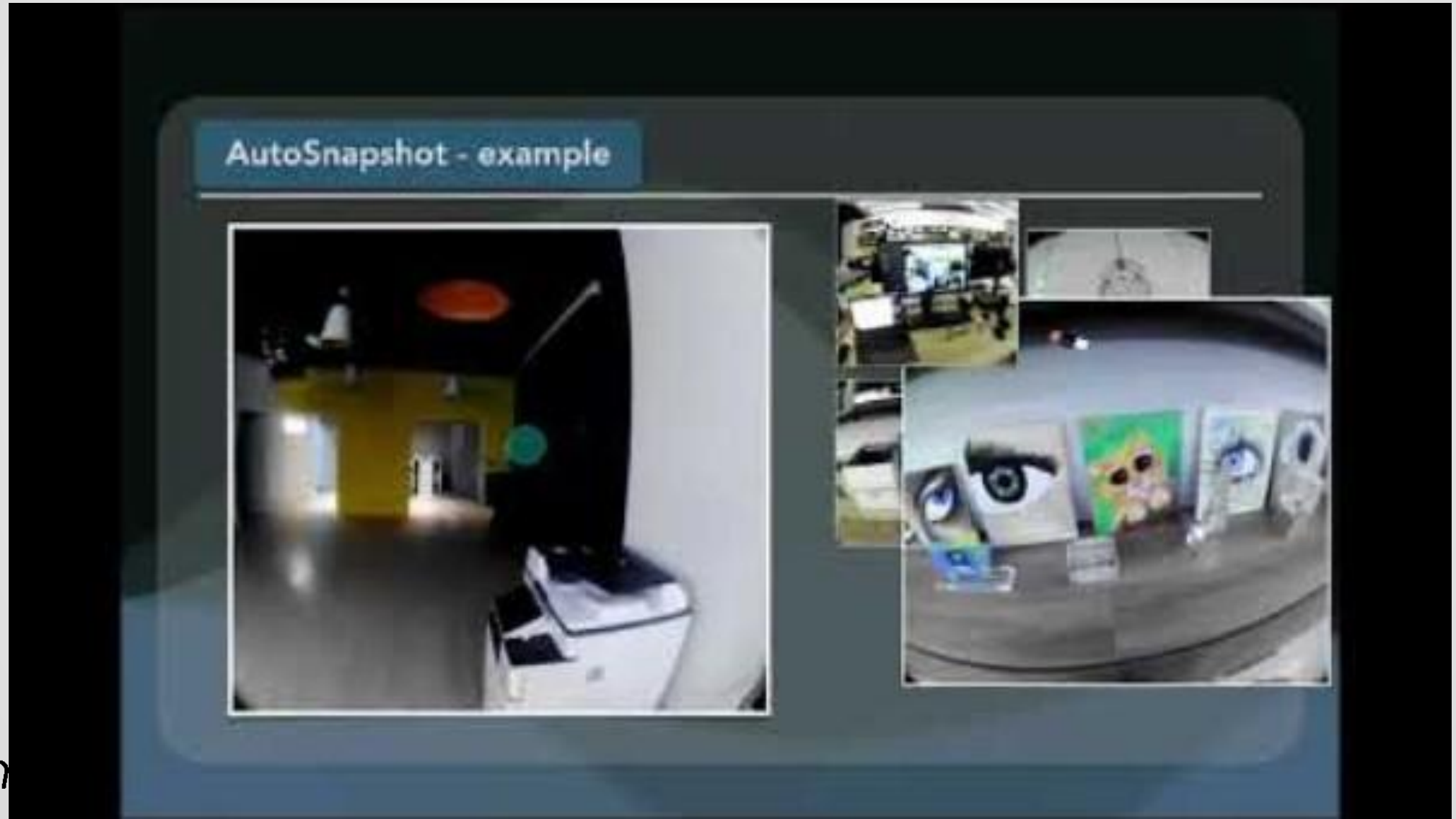
Focus Glasses

Ganzin

Smart Robot w/ Sol Glasses

Ganzin

AI Agent w/ Sol Glasses



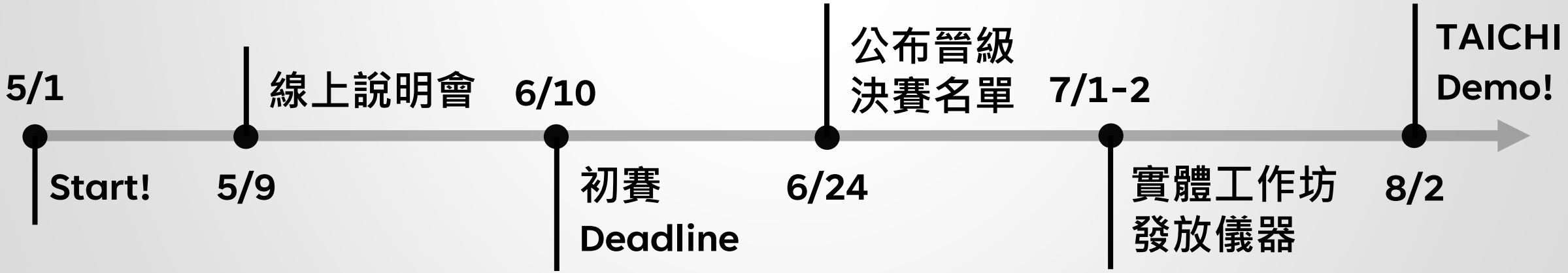
Ganzin

TAICHI x Ganzin Design Competition 細節

心動不如馬上行動！立馬了解比賽細節，報名拿取巨額獎金！

TAICHI x Ganzin Design Competition

- 目標
  - 使用穿戴式眼動儀實作一酷炫Demo
  - 可以結合其他裝置整合成一系統
- Timeline
  - 初賽賽程：5/1 – 6/24
  - 決賽賽程：7/1 – 8/2



競賽題目說明 – 主題方向

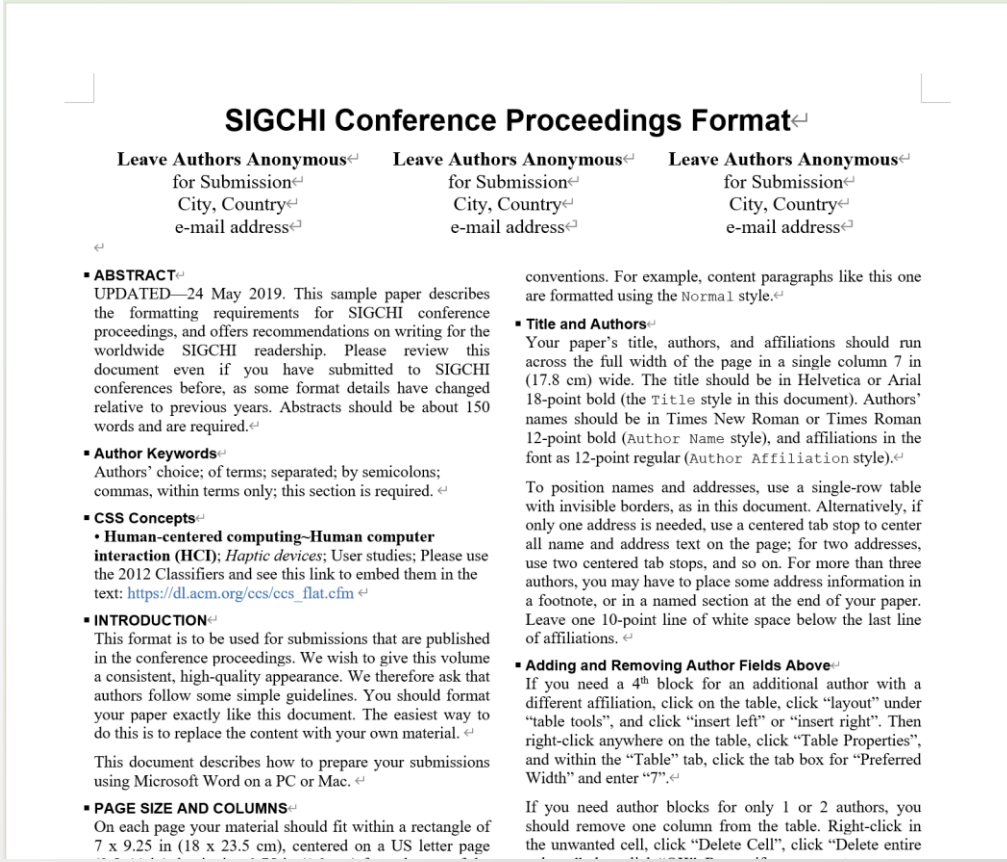
- 主題方向 (包括但不限於)
  - **人機互動介面**：
    利用眼球控制使用者介面，優化人機互動的體驗。
  - **認知監測**：
    偵測專注度、情緒狀態等，提供監測或輔助功能。
  - **娛樂/遊戲**：
    將注視以及眨眼等訊號應用於互動式遊戲或體驗中。
  - **即時數據分析**：
    利用眼動追蹤數據創建各行各業的分析工具 (例如市場研究、教育、運動等) 。

競賽題目說明 – 設計空間

- **感測面：**
  - **Sol Glasses 輸出**：能取得**眼睛影像**、**2D/3D 注視點**、**瞳孔直徑**、**眨眼**與**聲音**資訊等，可進一步實現視線手勢、表情偵測與視線控制等功能。
  - **額外感測器：** 若有其他同樣支援 Python 介面的感測器，也能結合使用。
- **輸出與反饋：**
  - **螢幕與近眼顯示**：可以在螢幕上顯示通知訊息，或使用螢幕實現進階技術，如注視點宣染技術等應用。
  - **聲音/音訊**：透過語音或音效提示，協助使用者掌握系統資訊。
  - **物聯網設備**：與 IoT 系統整合，根據眼動資料自動控制或回饋。
- **計算平台：**
  - **物聯網裝置**：一些物聯網裝置具備執行簡單任務所需的運算能力，這些設備可以在本地端執行運算。
  - **邊緣主機**：您也可以使用個人電腦（Edge Host）進行數據收集，透過區域網路傳輸並進行更複雜的運算。
  - **伺服器**：系統可以運行於雲端伺服器上，提供更高的計算能力以處理更複雜的任務。
  - **SaaS 工具**：許多雲端服務供應商提供強大的軟體工具（如 ChatGPT）。可以將這些工具與 Sol Glasses 結合，創造創新的應用程式。

初賽提案繳交說明

- Step1: 找好隊友進行組隊 (2-5 人)，如果找不到隊友，也可以加入 Discord 找隊友 channel
- Step2: 收集資訊，鎖定特定應用，選定需要之儀器、運算資源與工具。
- Step3: 規劃整體系統架構，且確立資料流，完成對應流程圖。
- Step4: 撰寫提案 (2 – 4 頁，SIGCHI 格式，點我下載)
  - Abstract: 提案簡述
  - Introduction (Background & Motivation): 說明提案之重要性
  - Implementation Details: 規劃之實作細節
    - 包含整體系統圖、資料流程圖等
    - 包含使用的感測器、運算資源與服務說明
    - 說明越**明確**越好
  - Expected Results: 預計達到的效果
  - 上述架構只是建議，只要能清楚說明 idea 都可以接受！
- Step5: 填寫表單，完成初賽報名
  - 主辦單位收到表單後，會寄信確認，如果沒收到信，請和主辦單位聯絡



SIGCHI Conference Proceedings Format

Leave Authors Anonymous
for Submission
City, Country
e-mail address

Leave Authors Anonymous
for Submission
City, Country
e-mail address

Leave Authors Anonymous
for Submission
City, Country
e-mail address

ABSTRACT

UPDATED—24 May 2019. This sample paper describes the formatting requirements for SIGCHI conference proceedings, and offers recommendations on writing for the worldwide SIGCHI readership. Please review this document even if you have submitted to SIGCHI conferences before, as some format details have changed relative to previous years. Abstracts should be about 150 words and are required.

Author Keywords

Authors' choice; of terms; separated; by semicolons; commas, within terms only; this section is required.

CSS Concepts

• **Human-centered computing~Human computer interaction (HCI)**; *Haptic devices*; User studies; Please use the 2012 Classifiers and see this link to embed them in the text: https://dl.acm.org/ccs/ccs_flat.cfm

INTRODUCTION

This format is to be used for submissions that are published in the conference proceedings. We wish to give this volume a consistent, high-quality appearance. We therefore ask that authors follow some simple guidelines. You should format your paper exactly like this document. The easiest way to do this is to replace the content with your own material.

This document describes how to prepare your submissions using Microsoft Word on a PC or Mac.

PAGE SIZE AND COLUMNS

On each page your material should fit within a rectangle of 7 x 9.25 in (18 x 23.5 cm), centered on a US letter page

conventions. For example, content paragraphs like this one are formatted using the `Normal` style.

Title and Authors

Your paper's title, authors, and affiliations should run across the full width of the page in a single column 7 in (17.8 cm) wide. The title should be in Helvetica or Arial 18-point bold (the `Title` style in this document). Authors' names should be in Times New Roman or Times Roman 12-point bold (`Author Name` style), and affiliations in the font as 12-point regular (`Author Affiliation` style).

To position names and addresses, use a single-row table with invisible borders, as in this document. Alternatively, if only one address is needed, use a centered tab stop to center all name and address text on the page; for two addresses, use two centered tab stops, and so on. For more than three authors, you may have to place some address information in a footnote, or in a named section at the end of your paper. Leave one 10-point line of white space below the last line of affiliations.

Adding and Removing Author Fields Above

If you need a 4th block for an additional author with a different affiliation, click on the table, click "layout" under "table tools", and click "insert left" or "insert right". Then right-click anywhere on the table, click "Table Properties", and within the "Table" tab, click the tab box for "Preferred Width" and enter "7".

If you need author blocks for only 1 or 2 authors, you should remove one column from the table. Right-click in the unwanted cell, click "Delete Cell", click "Delete entire

初賽評分標準說明

- **實用性 (25%)**
  - 解決方案或想法滿足目標需求或問題的程度，以及為使用者帶來的實際好處。
- **創意性 (25%)**
  - 解決方案展現的新穎性與創造力程度，即相較於現有方法，其想法或手法有多麼獨特或突破。
- **技術深度 (25%)**
  - 該解決方案所使用到的技術難度與整體系統實作難度，也包含理論基礎等等。
- **可行性 (25%)**
  - 在提出的技術下實際完成的可能性，同時參考完成系統/技術的手段以及提案當中的時程規劃。

獎項說明

- **決賽參加獎**
  - 該團隊成員可免 TAICHI 2025 報名費
- **第一名 ($20,000 NTD)**
  - 評審團根據 TAICHI 當天 Demo 後評分最高者
- **最佳創意應用獎 ($10,000 NTD)**
  - 評審團根據 TAICHI 當天 Demo 後創意評分最高者 (排除第一名)
- **最佳實現獎 ($10,000 NTD)**
  - 評審團根據 TAICHI 當天 Demo 後實作評分最高者 (排除第一名與最佳創意應用獎)
- **最佳人氣獎 (由會議參與者投票選出) ($5,000 NTD)**

Q&A

- 心動不如馬上行動！
- 馬上加入 Discord 群組，專人服務