

# Machine Learning in Unity



Danai Jedsadathitikul

Jarut Busarathid

[DcG.IT@PBRU](mailto:DcG.IT@PBRU)

เป็นการผสมผสานระหว่างการพัฒนาเกมและการใช้  
โมเดล AI เพื่อเพิ่มประสบการณ์ผู้ใช้งานและสร้างความ  
ลุ่มมัยในเกมหรือแอปพลิเคชัน

ตัวอย่างการใช้งานอาจรวมถึงการควบคุมพฤติกรรม  
ของ NPC, การสร้าง AI ที่เรียนรู้จากผู้เล่น หรือการ  
จำลองระบบซับซ้อน

# เครื่องมือและฟีเจอร์

# Unity ML-Agents Toolkit

- เป็นโอเพ่นซอร์สจาก Unity สำหรับการพัฒนาเกมที่ใช้ AI
- ใช้ Reinforcement Learning (RL) หรือพฤติกรรมจำลอง
- รองรับการเทรนโมเดลด้วย Python และ TensorFlow หรือ PyTorch
- เหมาะสำหรับ:
  - การสร้าง NPC ที่ฉลาด
  - การจำลองการตัดสินใจที่ซับซ้อน
  - การเรียนรู้ของ AI ผ่านการเล่นเกม

# ดูข้อมูลเพิ่มเติมได้ที่ Unity ML-Agents GitHub

<https://github.com/Unity-Technologies/ml-agents>

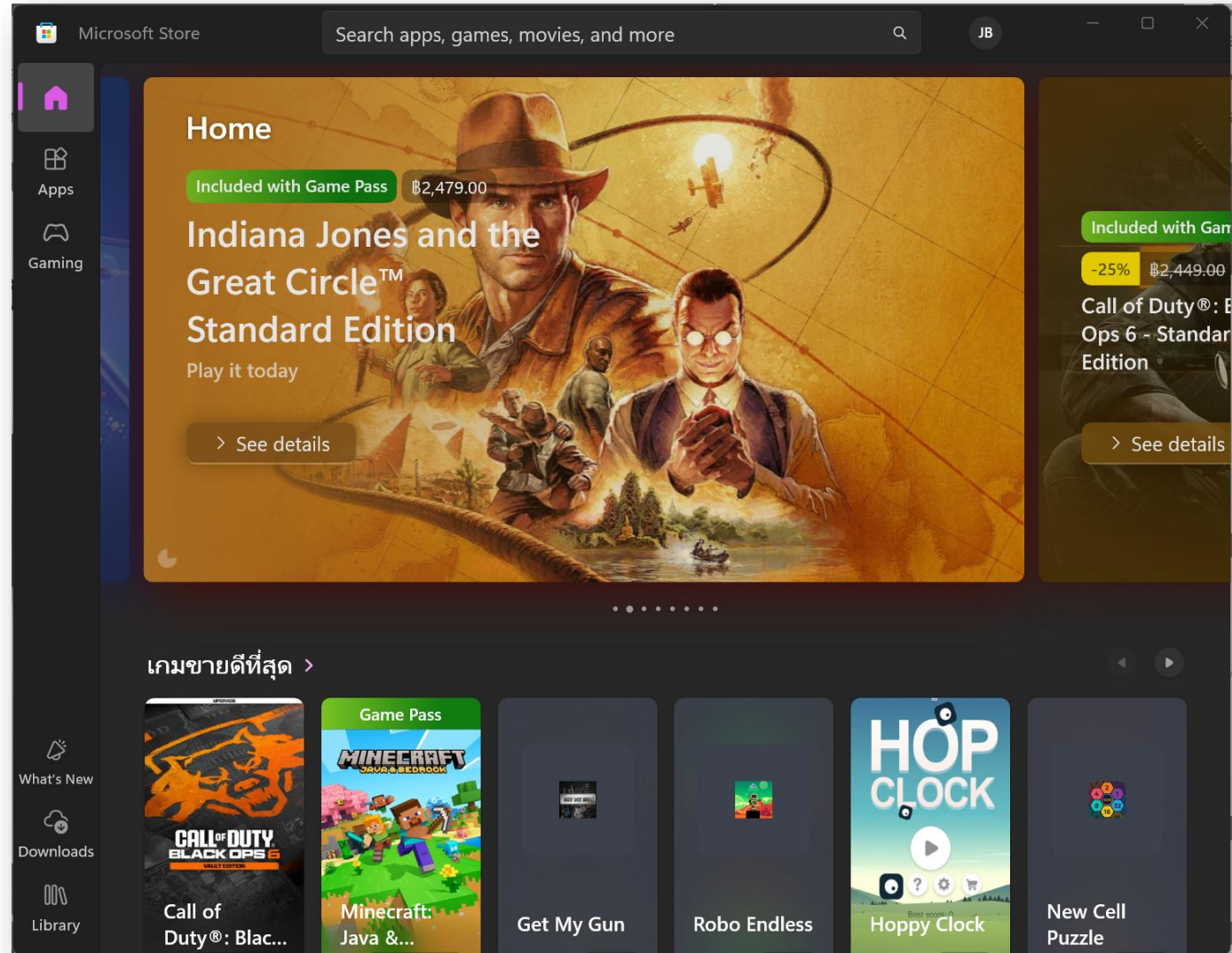
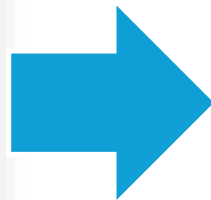
# Integrating Pre-trained Models

- ใช้โมเดล AI ที่สร้างไว้แล้ว เช่น OpenAI, Hugging Face, หรือ TensorFlow Lite
- นำเข้าโมเดลที่เทรนแล้วและใช้ใน Unity ผ่าน Barracuda ซึ่งเป็นเครื่องมือของ Unity สำหรับรับโมเดล AI

# การใช้ Unity Computer Vision

- เหมาะสำหรับเกมหรือแอปที่ต้องใช้ข้อมูลภาพ เช่น การวิเคราะห์วัตถุ หรือการติดตามการเคลื่อนไหว

# บทวนติดตั้ง Python





Microsoft Store



Apps



Gaming

Home

Included with Ga

-25% \$2,449.00

Call of Duty  
Ops 6 - S  
Edition

Save now

> See details

python



python



python 3 10



python 3 9



Countdown  
Explore deals



Python 3.11  
App



Python 3.10  
App



What's New

Downloads

Library



Python 3.7

4.0 ★ Apps Developer tools

Free

The Python 3.7 interpreter and runtime.

```
cmd - python3.7
Windows [Version 10.0.19041.450]
Microsoft Corporation. All rights reserved.

C:\Users\user>python3.7
(tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:00) [MSC v.1900 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more
>>>
Welcome to Python (sys.version_info.major).(sys.version_info.minor)
Python 3.7
```



Python 3.13

5.0 ★ Apps Developer tools

Free

The Python 3.13 interpreter and runtime

```
Command Prompt - python3.13
Microsoft Windows [Version 10.0.22100.2314]
Microsoft Corporation. All rights reserved.

C:\Users\user>python3.13
(tags/v3.13.1:tags/v3.13.1:10672451, Dec 3 2024, 19:06:28) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
Welcome to Python (sys.version_info.major).(sys.version_info.minor)
Python 3.13
```



Python 3.8

Apps Developer tools

Free

The Python 3.8 interpreter and runtime.

```
Prompt - python3.8
Windows [Version 10.0.19042.928]
Microsoft Corporation. All rights reserved.

C:\Users\user>python3.8
(tags/v3.8.10:3d8993e, May 3 2021, 11:48:05) [MSC v.1928 64 bit (AMD64)]
on win32
Type "copyright", "credits" or "license()" for more
>>>
Welcome to Python (sys.version_info.major).(sys.version_info.minor)
Python 3.8
```

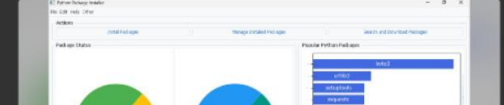


Python Package Installer

Apps Developer tools

\$174.00

The Python Package Installer streamlines Python package management with an intuitive interface, enabling easy...





# Python 3.13

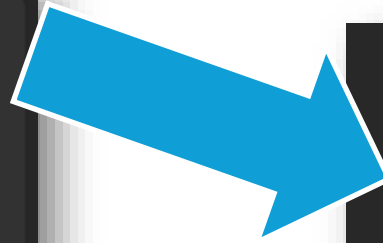
5.0 ★ Apps Developer tools

Free

The Python 3.13 interpreter and runtime

```
Command Prompt - python3.13
Microsoft Windows [Version 10.0.26100.2314]
Microsoft Corporation. All rights reserved.

C:\Users\Pythonista>python3.13
Python 3.13.1 (tags/v3.13.1:0671451, Dec 3 2024, 19:06:28) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
>import sys
>print(f"Welcome, to Python {sys.version_info.major}.{sys.version_info.minor}")
Welcome, to Python 3.13
```





# Python 3.13

Python Software Foundation

5.0 ★ 1 rating Developer tools

 3+

The Python 3.13 interpreter and runtime

Preparing to download  
Getting files ready





# Python 3.13

Python Software Foundation

5.0 ★ 1 rating Developer tools

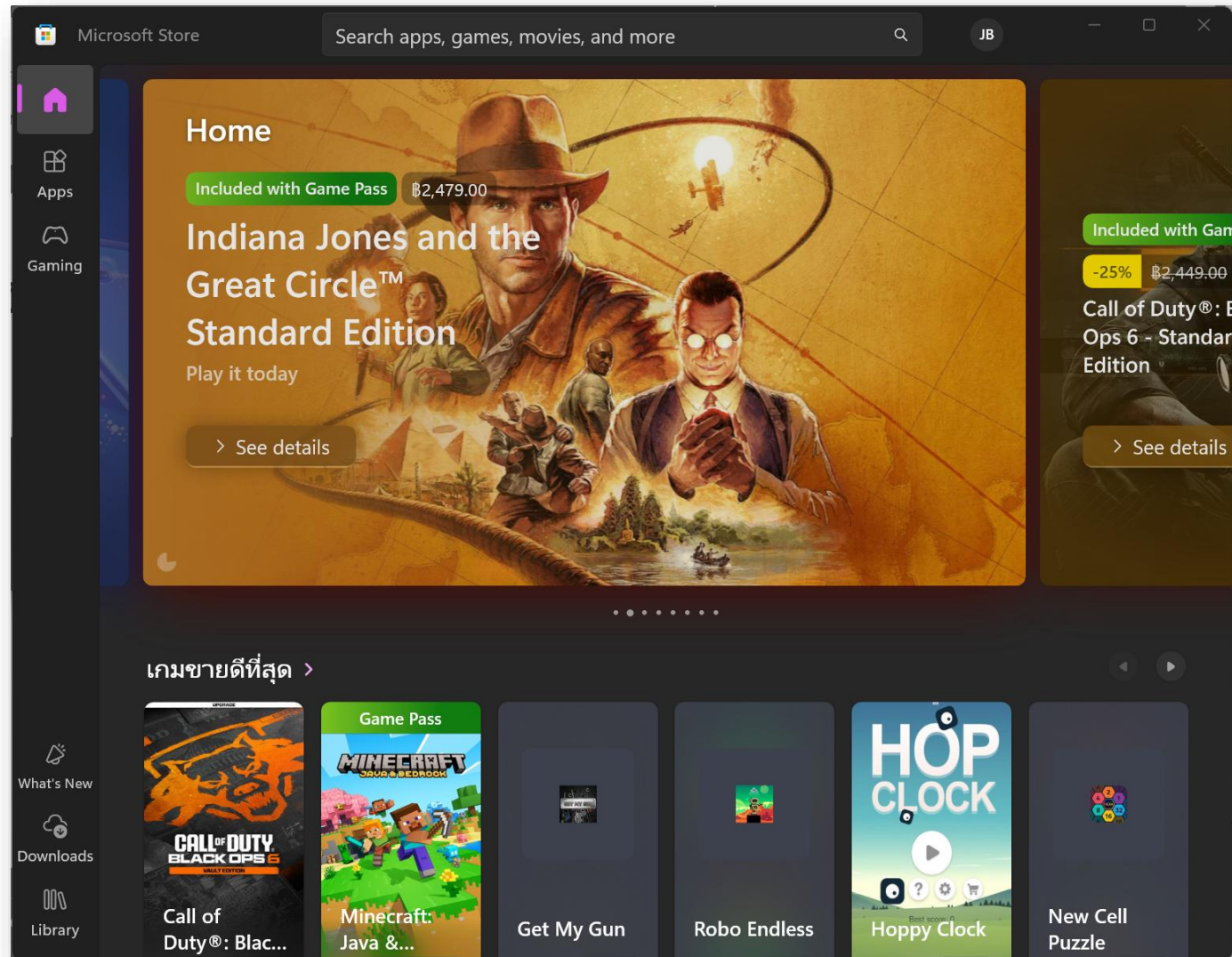
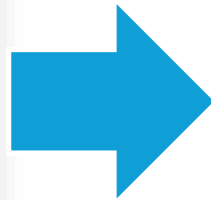
 3+

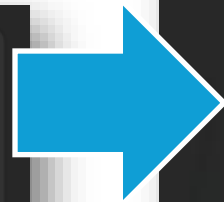
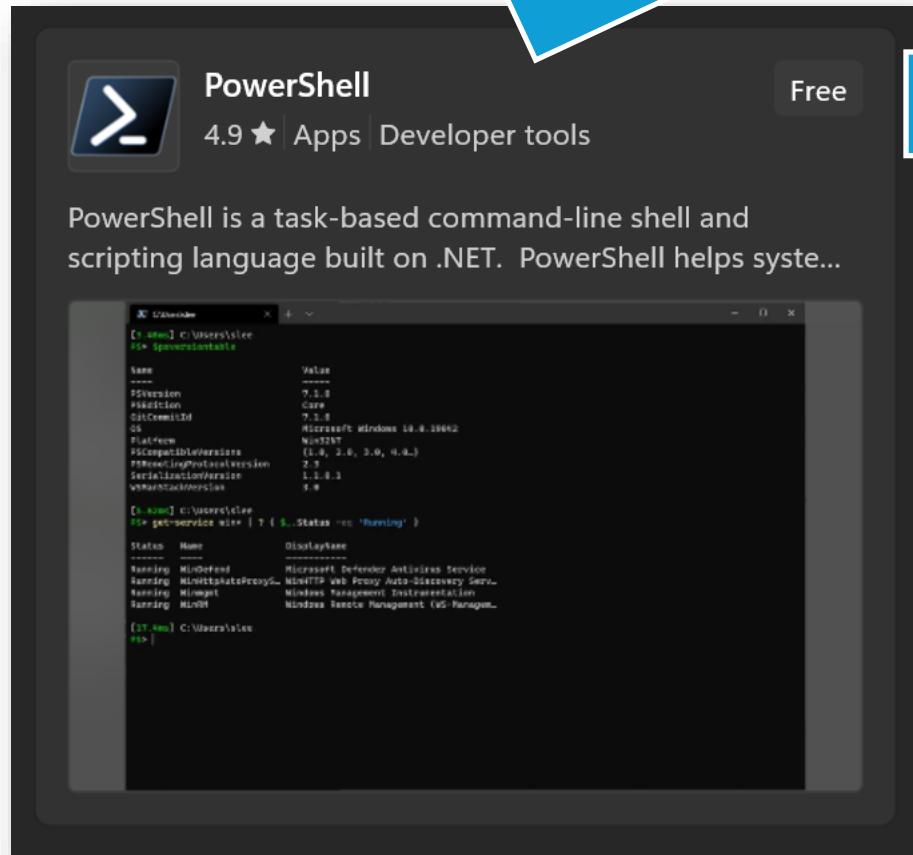
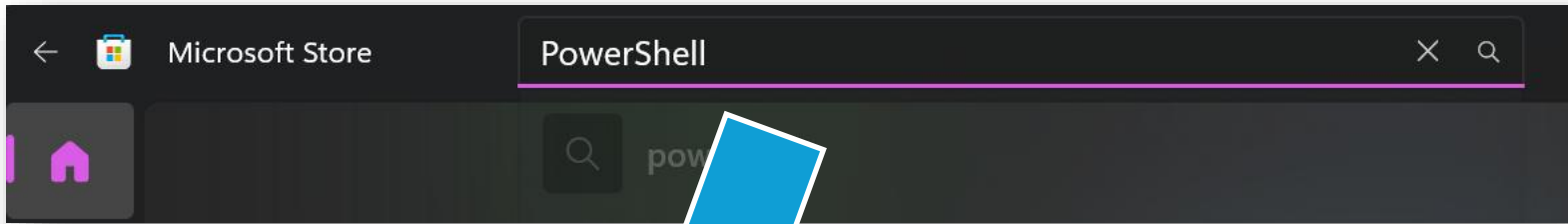
The Python 3.13 interpreter and runtime

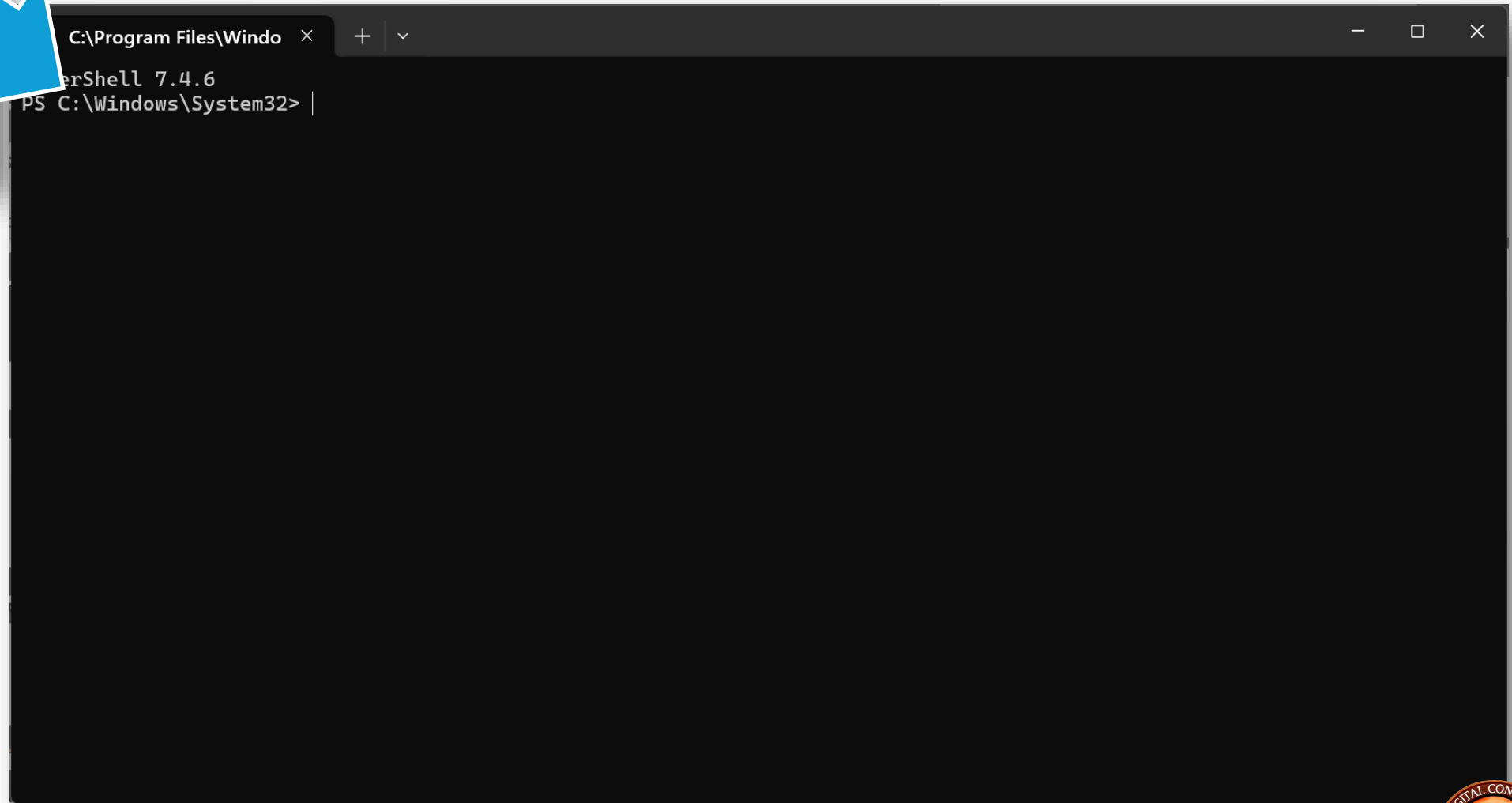
Installed



# ติดตั้ง PowerShell







# สร้างสภาพแวดล้อม plab

```
C:\Program Files\Windo × + v
PowerShell 7.4.6
PS C:\Windows\System32> cd \
PS C:\> python -m venv plab
PS C:\>
```

# เข้าใช้สภาพแวดล้อม plab

```
C:\Program Files\Windo × + v
PowerShell 7.4.6
PS C:\Windows\System32> cd \
PS C:\> cd plab
PS C:\plab> cd scripts
PS C:\plab\Scripts> .\Activate.ps1
(plab) PS C:\plab\Scripts> |
```

# ออกจากสภาพแวดล้อม plab

```
C:\Program Files\Windo × + v  
(plab) PS C:\plab\Scripts> deactivate  
PS C:\plab\Scripts>
```

# อัปเดต pip

```
C:\Program Files\Windo × + v
(plab) PS C:\plab\Scripts> python -m pip install --upgrade pip
Requirement already satisfied: pip in c:\plab\lib\site-packages (24.3.1)
(plab) PS C:\plab\Scripts> |
```

# ติดตั้งไลบรารี 1

```
C:\Program Files\Windo × + v  
(plab) PS C:\plab\Scripts> pip install --upgrade numpy pandas matplotlib pytk
```

```
C:\Program Files\Windo × + v  
(plab) PS C:\plab\Scripts> pip install --upgrade numpy pandas matplotlib pytk  
Collecting numpy  
  Downloading numpy-2.2.0-cp312-cp312-win_amd64.whl.metadata (60 kB)  
Collecting pandas  
  Downloading pandas-2.2.3-cp312-cp312-win_amd64.whl.metadata (19 kB)  
Collecting matplotlib  
  Downloading matplotlib-3.10.0-cp312-cp312-win_amd64.whl.metadata (11 kB)  
Collecting pytk  
  Downloading pytk-0.0.2.1-py2.py3-none-any.whl.metadata (2.1 kB)  
Collecting python-dateutil>=2.8.2 (from pandas)  
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)  
Collecting pytz>=2020.1 (from pandas)  
  Downloading pytz-2024.2-py2.py3-none-any.whl.metadata (22 kB)  
Collecting tzdata>=2022.7 (from pandas)  
  Downloading tzdata-2024.2-py2.py3-none-any.whl.metadata (1.4 kB)  
Collecting contourpy>=1.0.1 (from matplotlib)  
  Downloading contourpy-1.3.1-cp312-cp312-win_amd64.whl.metadata (5.4 kB)  
Collecting cyclor>=0.10 (from matplotlib)  
  Downloading cyclor-0.12.1-py3-none-any.whl.metadata (3.8 kB)  
Collecting fonttools>=4.22.0 (from matplotlib)  
  Downloading fonttools-4.55.3-cp312-cp312-win_amd64.whl.metadata (168 kB)  
Collecting kiwisolver>=1.3.1 (from matplotlib)
```

```
C:\Program Files\Windo x + v
Downloading pytz-2024.2-py2.py3-none-any.whl (508 kB)
Downloading tzdata-2024.2-py2.py3-none-any.whl (346 kB)
Downloading flyingcircus-0.1.4.1-py2.py3-none-any.whl (130 kB)
Downloading six-1.17.0-py2.py3-none-any.whl (11 kB)
Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Downloading blessed-1.20.0-py2.py3-none-any.whl (58 kB)
Downloading setuptools-75.6.0-py3-none-any.whl (1.2 MB)
1.2/1.2 MB 60.1 MB/s eta 0:00:00
Downloading setuptools_scm-8.1.0-py3-none-any.whl (43 kB)
Downloading JinJinx-1.3.0-py2.py3-none-any.whl (33 kB)
Downloading wcwidth-0.2.13-py2.py3-none-any.whl (34 kB)
Downloading ansicon-1.89.0-py2.py3-none-any.whl (63 kB)
Installing collected packages: wcwidth, pytz, appdirs, ansicon, tzdata, six, setuptools,
  pyparsing, pillow, packaging, numpy, kiwisolver, JinJinx, fonttools, cyclical, setuptools-
  scm, python-dateutil, contourpy, blessed, pandas, matplotlib, flyingcircus, pytk
Successfully installed ansicon-1.89.0 appdirs-1.4.4 blessed-1.20.0 contourpy-1.3.1 cycle
r-0.12.1 flyingcircus-0.1.4.1 fonttools-4.55.3 JinJinx-1.3.0 kiwisolver-1.4.7 matplotlib-
3.10.0 numpy-2.2.0 packaging-24.2 pandas-2.2.3 pillow-11.0.0 pyparsing-3.2.0 python-date
util-2.9.0.post0 pytk-0.0.2.1 pytz-2024.2 setuptools-75.6.0 setuptools-scm-8.1.0 six-1.1
7.0 tzdata-2024.2 wcwidth-0.2.13
(plab) PS C:\plab\Scripts>
(plab) PS C:\plab\Scripts>
```



# ติดตั้งไลบรารี 2

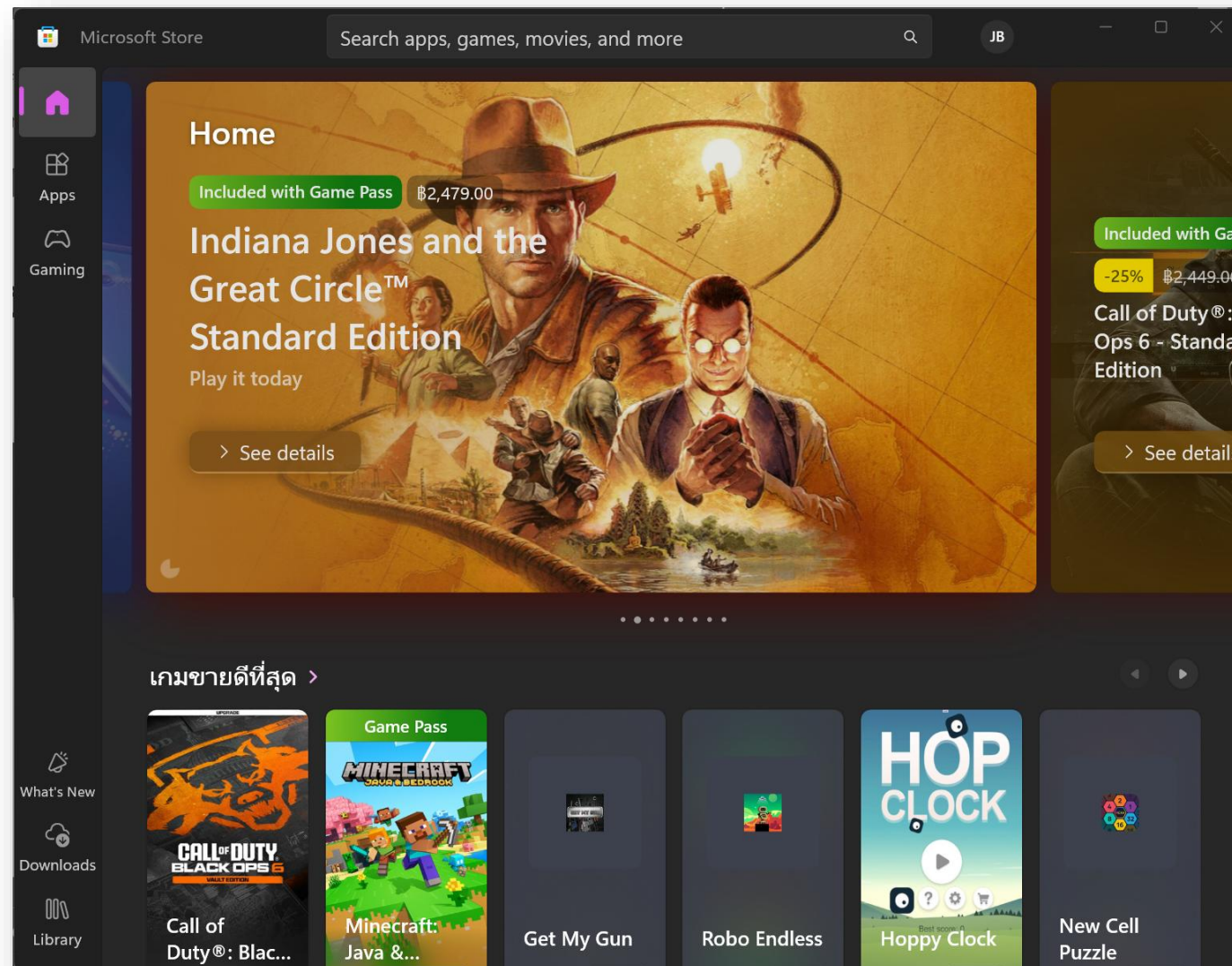
```
C:\Program Files\Windo x + v
(plab) PS C:\plab\Scripts> pip install --upgrade scipy scikit-learn scikit-image

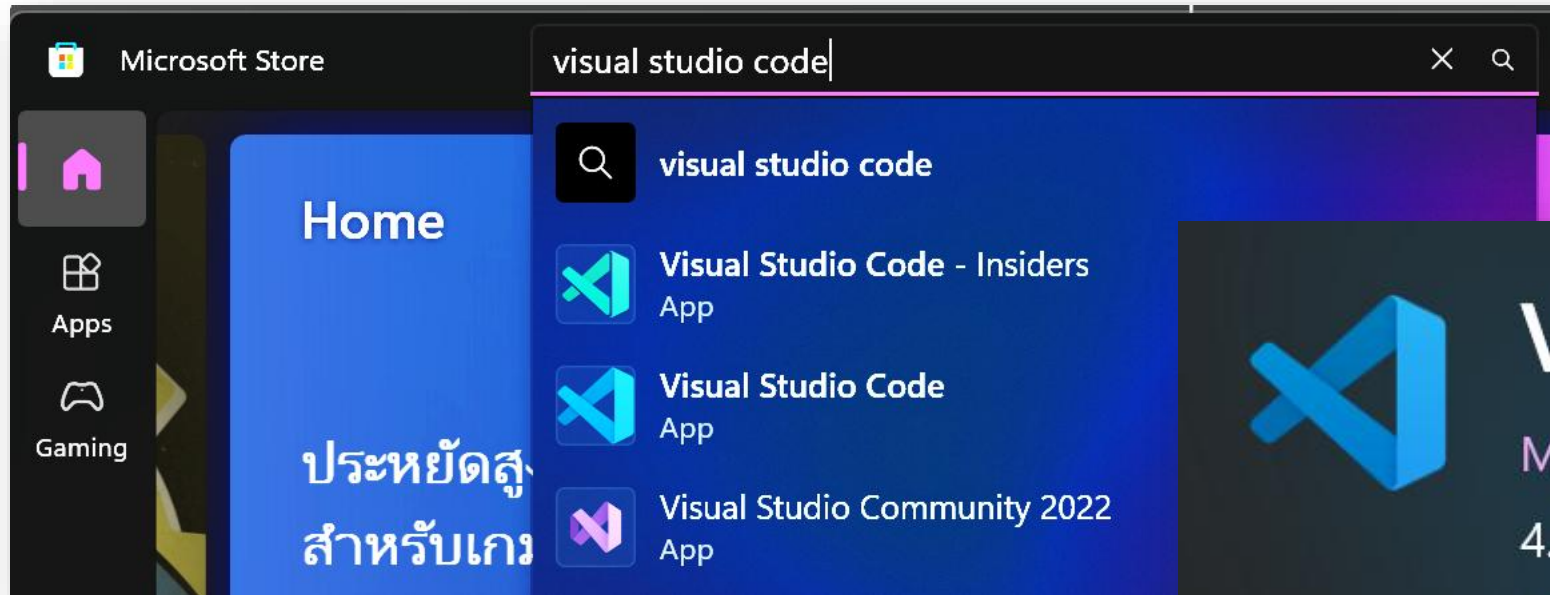
C:\Program Files\Windo x + v
(plab) PS C:\plab\Scripts> pip install --upgrade scipy scikit-learn scikit-image
Collecting scipy
  Downloading scipy-1.14.1-cp312-cp312-win_amd64.whl.metadata (60 kB)
Collecting scikit-learn
  Downloading scikit_learn-1.6.0-cp312-cp312-win_amd64.whl.metadata (15 kB)
Collecting scikit-image
  Downloading scikit_image-0.25.0-cp312-cp312-win_amd64.whl.metadata (14 kB)
Requirement already satisfied: numpy<2.3,>=1.23.5 in c:\plab\lib\site-packages (from sci
py) (2.2.0)
Collecting joblib>=1.2.0 (from scikit-learn)
  Downloading joblib-1.4.2-py3-none-any.whl.metadata (5.4 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn)
  Downloading threadpoolctl-3.5.0-py3-none-any.whl.metadata (13 kB)
Collecting networkx>=3.0 (from scikit-image)
  Downloading networkx-3.4.2-py3-none-any.whl.metadata (6.3 kB)
Requirement already satisfied: pillow>=10.1 in c:\plab\lib\site-packages (from scikit-im
age) (11.0.0)
Collecting imageio!=2.35.0,>=2.33 (from scikit-image)
  Downloading imageio-2.36.1-py3-none-any.whl.metadata (5.2 kB)
Collecting tifffile>=2022.8.12 (from scikit-image)
  Downloading tifffile-2024.12.12-py3-none-any.whl.metadata (31 kB)
Requirement already satisfied: packaging>=21 in c:\plab\lib\site-packages (from scikit-i
```



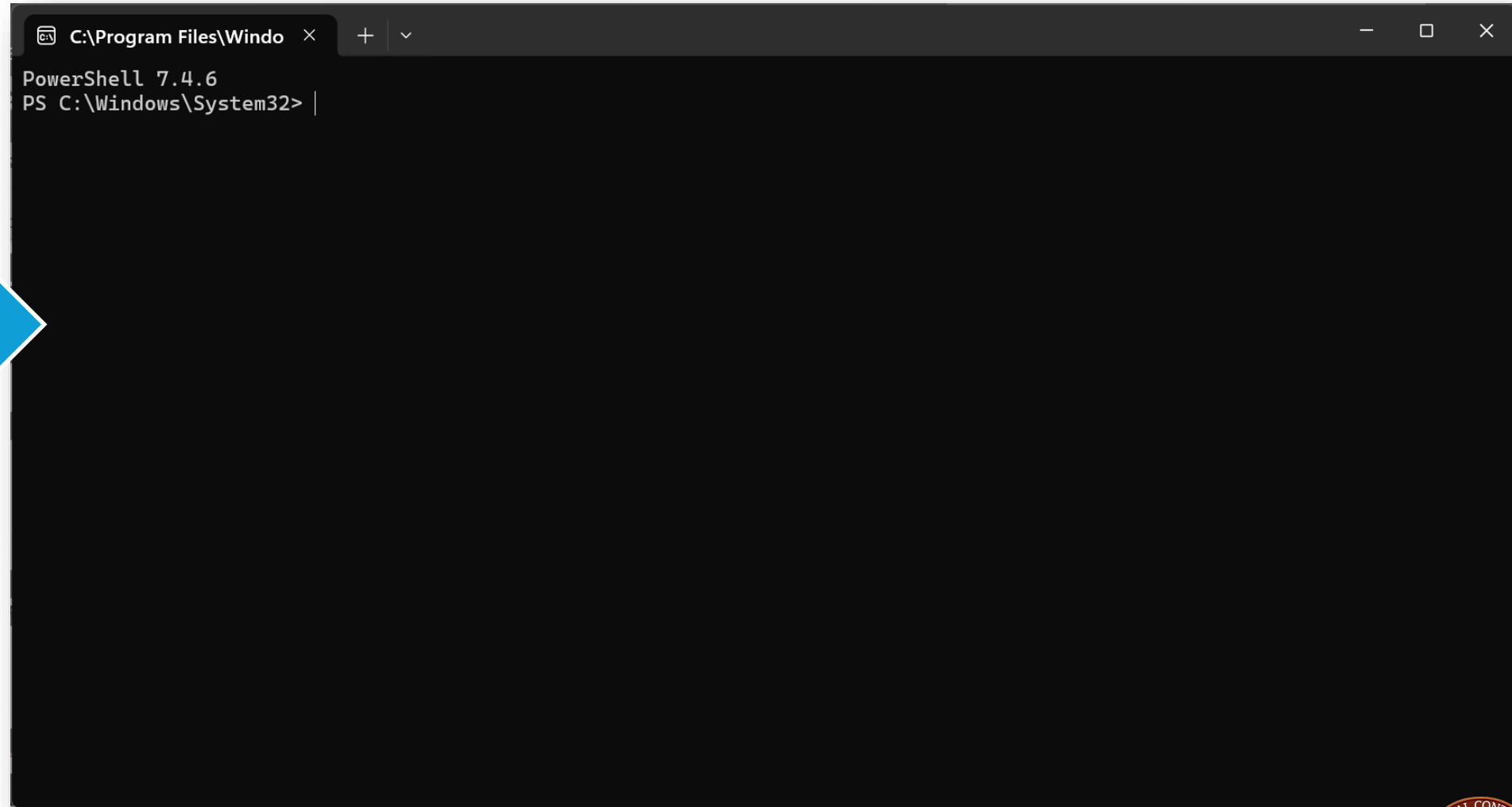
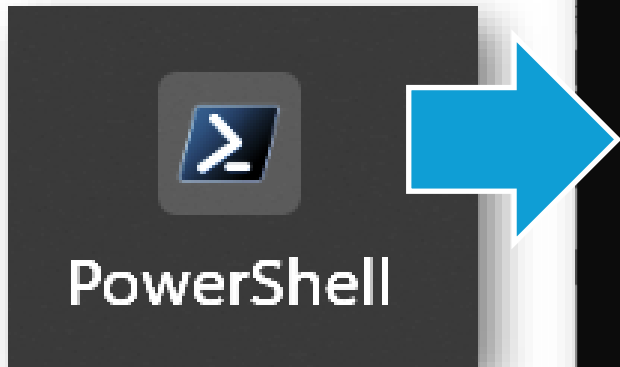
```
C:\Program Files\Wind... x + v
Requirement already satisfied: packaging>=21 in c:\plab\lib\site-packages (from scikit-image) (24.2)
Collecting lazy-loader>=0.4 (from scikit-image)
  Downloading lazy_loader-0.4-py3-none-any.whl.metadata (7.6 kB)
Downloaded scipy-1.14.1-cp312-cp312-win_amd64.whl (44.5 MB)
  _____ 44.5/44.5 MB 34.1 MB/s eta 0:00:00
Downloaded scikit_learn-1.6.0-cp312-cp312-win_amd64.whl (11.1 MB)
  _____ 11.1/11.1 MB 58.0 MB/s eta 0:00:00
Downloaded scikit_image-0.25.0-cp312-cp312-win_amd64.whl (12.9 MB)
  _____ 12.9/12.9 MB 50.5 MB/s eta 0:00:00
Downloaded imageio-2.36.1-py3-none-any.whl (315 kB)
Downloaded joblib-1.4.2-py3-none-any.whl (301 kB)
Downloaded lazy_loader-0.4-py3-none-any.whl (12 kB)
Downloaded networkx-3.4.2-py3-none-any.whl (1.7 MB)
  _____ 1.7/1.7 MB 91.3 MB/s eta 0:00:00
Downloaded threadpoolctl-3.5.0-py3-none-any.whl (18 kB)
Downloaded tifffile-2024.12.12-py3-none-any.whl (227 kB)
Installing collected packages: tifffile, threadpoolctl, scipy, networkx, lazy-loader, joblib, imageio, scikit-learn, scikit-image
Successfully installed imageio-2.36.1 joblib-1.4.2 lazy-loader-0.4 networkx-3.4.2 scikit-image-0.25.0 scikit-learn-1.6.0 scipy-1.14.1 threadpoolctl-3.5.0 tifffile-2024.12.12
(plab) PS C:\plab\Scripts>
```

# ติดตั้ง VS Code





# สวัสดี Python



C:\Program Files\Windo × + ▾

PowerShell 7.4.6

PS C:\Windows\System32> cd \plab\scripts

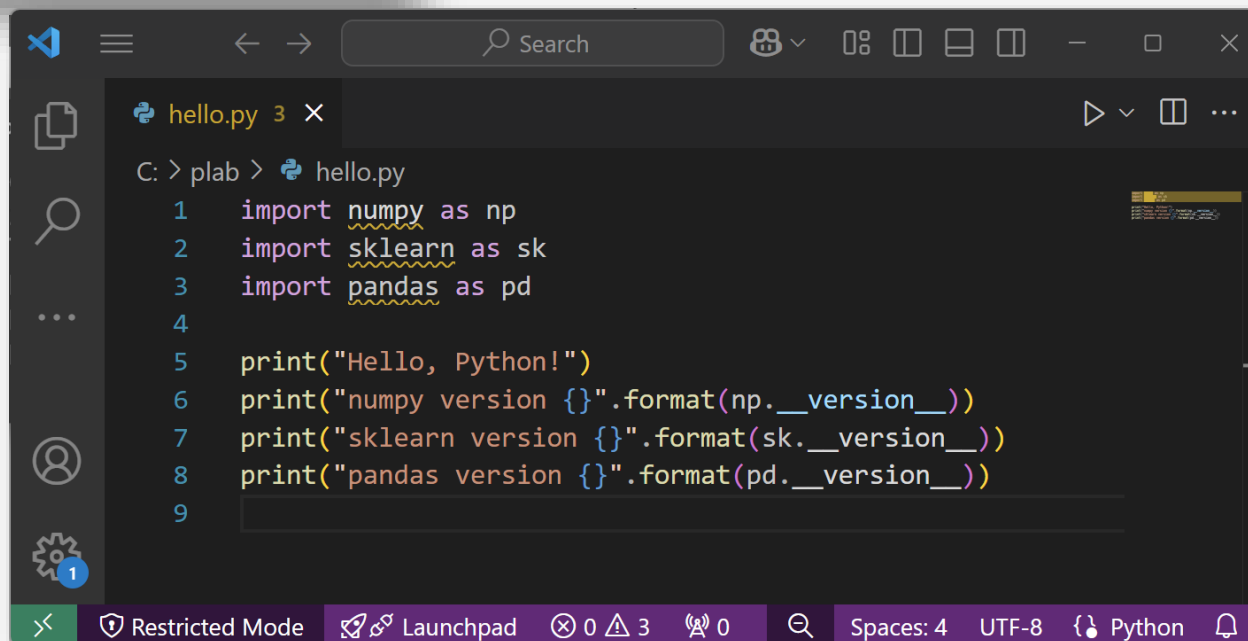
PS C:\plab\Scripts> .\activate.ps1

(plab) PS C:\plab\Scripts> cd ..

(plab) PS C:\plab> code hello.py

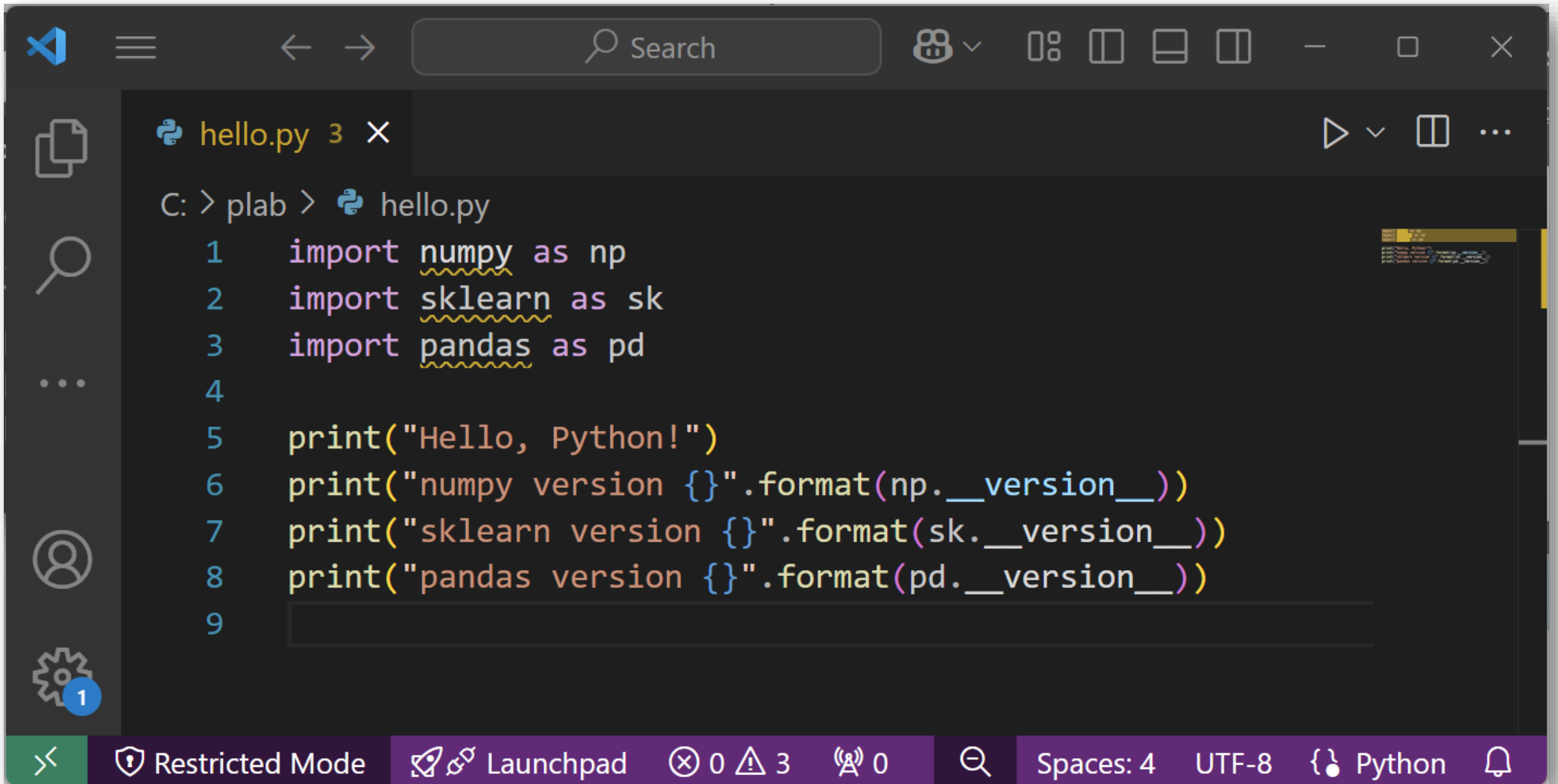
(plab) PS C:\plab> |

เรียก VS Code พร้อมสร้างหรือเปิด  
(ถ้าสร้างไว้แล้ว) ไฟล์ hello.py



```
C: > plab > hello.py
1 import numpy as np
2 import sklearn as sk
3 import pandas as pd
4
5 print("Hello, Python!")
6 print("numpy version {}".format(np.__version__))
7 print("sklearn version {}".format(sk.__version__))
8 print("pandas version {}".format(pd.__version__))
9
```





```
C: > plab > hello.py

1  import numpy as np
2  import sklearn as sk
3  import pandas as pd
4
5  print("Hello, Python!")
6  print("numpy version {}".format(np.__version__))
7  print("sklearn version {}".format(sk.__version__))
8  print("pandas version {}".format(pd.__version__))
9
```

Restricted Mode Launchpad 0 3 0 Spaces: 4 UTF-8 Python



# เรียกให้โปรแกรม hello.py ทำงาน

```
(plab) PS C:\plab> python .\hello.py  
Hello, Python!  
numpy version 2.2.0  
sklearn version 1.6.0  
pandas version 2.2.3  
(plab) PS C:\plab> |
```

# การติดตั้งและการตั้งค่า

# เตรียม Unity และ ML-Agents Toolkit

- ดาวน์โหลด Unity Editor 2023+
- ติดตั้ง ML-Agents Toolkit ผ่าน Package Manager
- ตรวจสอบว่า Python และไลบรารี TensorFlow หรือ PyTorch ติดตั้งบนเครื่องแล้ว

```
pip install mlagents
```

```
pip install tensorflow
```

```
pip install torch torchvision torchaudio
```

# ติดตั้ง Unity Barracuda:

- หากต้องการใช้โมเดล Pre-trained ให้เพิ่ม Barracuda Package ผ่าน Unity Package Manager
- Barracuda รองรับไฟล์โมเดลแบบ ONNX (.onnx)

# การออกแบบ AI ใน Unity

# การตั้งค่าการเรียนรู้ (Training Environment)

- สร้าง Agent ในเกมที่สามารถเรียนรู้พฤติกรรม เช่น NPC หรือ หุ่นยนต์
- กำหนด Observation (ข้อมูลที่ Agent เห็น) และ Action (การกระทำที่สามารถทำได้)
- ตั้งค่า Reward System เพื่อกำหนดรางวัลสำหรับการกระทำที่ถูกต้อง

# การเทรนโมเดล

- ใช้ ML-Agents Toolkit เทรนโมเดล AI ใน Python
- ตัวอย่างการเทรนด้วยคำสั่ง

```
mlagents-learn config.yaml --run-id=my_first_run
```

- เมื่อการเทรนเสร็จสิ้น โมเดลจะถูกเซฟเป็นไฟล์ .nnt ซึ่งสามารถนำมาใช้ใน Unity ได้

# การนำโมเดลเข้ามาใช้

- นำไฟล์โมเดล .nnt ใส่ใน Unity
- ใช้ Behavior Parameters ใน Inspector ของ Unity เพื่อเชื่อมโยงโมเดลกับ Agent

# การใช้งาน AI แบบ Pre-trained

# การนำเข้าโมเดล ONNX

- นำโมเดลจาก TensorFlow หรือ PyTorch ที่เทรนเสร็จแล้ว ให้แปลงเป็น ONNX และนำเข้า Unity ผ่าน Barracuda
- ตัวอย่างการแปลงจาก PyTorch

```
import torch  
torch.onnx.export(model, input_tensor, "model.onnx")
```

# การใช้ใน Unity

- เพิ่มคอมโพเนนต์ NN Model และตั้งค่าอินพุต/เอาต์พุตใน Inspector
- ใช้โมเดลในการควบคุมพฤติกรรมของตัวละครหรือระบบเกม

# ตัวอย่างโปรเจกต์

# NPC ที่ปรับตัวเองได้

- สร้าง NPC ที่เรียนรู้พฤติกรรมของผู้เล่น เช่น การหลบหลีก, การโจมตี
- ใช้ Reinforcement Learning กับ ML-Agents

# ระบบเกมที่ปรับความยากอัตโนมัติ

(Dynamic Difficulty Adjustment)

- ใช้ AI วิเคราะห์ข้อมูลของผู้เล่น เช่น เวลาที่ใช้ผ่านด่าน ความสำเร็จ/ล้มเหลว
- AI สามารถปรับเปลี่ยนความยากของเกมแบบเรียลไทม์

# เกมจำลองที่ใช้ AI

- เช่น การสร้างเกมจำลองเศรษฐกิจที่ AI ควบคุมตลาด หรือการพัฒนาเกมการศึกษาให้ AI เป็นคู่ฝึก

# แหล่งเรียนรู้เพิ่มเติม

# Unity ML-Agents Documentation

## Barracuda Documentation

### TensorFlow for Unity

### PyTorch Tutorials



# ตัวอย่าง

การใช้ Machine Learning กับข้อมูลอุณหภูมิรายวัน  
ย้อนหลัง 5 ปี และใช้ UI Toolkit สร้างหน้าจอรับวัน  
เดือน และปีที่ต้องการทำนายค่าอุณหภูมิ แล้วรายงานผล  
ค่าที่ทำนายหลังจากคลิกกดปุ่มทำนายผล

# รวบรวมและเตรียมข้อมูลอุณหภูมิ

- ข้อมูลควรมีลักษณะเช่น
  - วันที่ (ปี/เดือน/วัน)
  - ค่าอุณหภูมิสูงสุด, ต่ำสุด, เฉลี่ย (ตามความเหมาะสม)

# สร้างและเทรนโมเดล Machine Learning

- ใช้ TensorFlow หรือ PyTorch สำหรับการสร้างโมเดล Time Series Forecasting

1/3

```
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split

# โหลดข้อมูล data = pd.read_csv("temperature_data.csv")
data['Date'] = pd.to_datetime(data['Date'])
data.set_index('Date', inplace=True)
```



**# สร้างชุดข้อมูล**

```
X = data.index.values
```

```
y = data['Temperature'].values
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,  
                                                    test_size=0.2, random_state=42)
```

**# สร้างโมเดล**

```
model = tf.keras.Sequential([  
    tf.keras.layers.Dense(64, activation='relu', input_shape=(1,)),  
    tf.keras.layers.Dense(64, activation='relu'),  
    tf.keras.layers.Dense(1)])  
model.compile(optimizer='adam', loss='mse')
```

```
# ฝึกฝน  
model.fit(X_train, y_train, epochs=100,  
          validation_data=(X_test, y_test))  
model.save("temperature_predictor.h5")
```

# บันทึกโมเดลในรูปแบบ .h5

- โมเดลนี้สามารถโหลดใน Unity ผ่าน Barracuda สำหรับการพยากรณ์

# การสร้าง UI ด้วย Unity UI Toolkit

- สร้าง Layout ด้วย UI Toolkit:
  - เพิ่มฟอร์มที่รับวัน/เดือน/ปี และปุ่ม "ทำนายผล"
  - ตัวอย่างโค้ด UXML

```
<ui:UXML xmlns:ui="UnityEngine.UIElements">
  <ui:VisualElement>
    <ui:Label text="ทำนายค่าอุณหภูมิ" />
    <ui:TextField name="dayInput" label="วัน" />
    <ui:TextField name="monthInput" label="เดือน" />
    <ui:TextField name="yearInput" label="ปี" />
    <ui:Button text="ทำนายผล" name="predictButton" />
    <ui:Label name="resultLabel" text="ผลลัพธ์จะแสดงที่นี่" />
  </ui:VisualElement>
</ui:UXML>
```

```
using UnityEngine;
using UnityEngine.UIElements;
using Unity.Barracuda;
public class TemperaturePredictor : MonoBehaviour{
    public NNModel temperatureModel; // โมเดลที่นำเข้าใน Unity
    private IWorker worker;
    void Start() {
        // โหลดโมเดล
        worker = WorkerFactory.CreateWorker(temperatureModel);
        // เข้าถึง UI
        var root = GetComponent<UIDocument>().rootVisualElement;
        var dayInput = root.Q<TextField>("dayInput");
        var monthInput = root.Q<TextField>("monthInput");
        var yearInput = root.Q<TextField>("yearInput");
        var predictButton = root.Q<Button>("predictButton");
        var resultLabel = root.Q<Label>("resultLabel");
```

```
// เพิ่มฟังก์ชันให้ปุ่ม
predictButton.clicked += () => {
    // ดึงข้อมูลจากฟอร์ม
    int day = int.Parse(dayInput.value);
    int month = int.Parse(monthInput.value);
    int year = int.Parse(yearInput.value);
    // สร้างข้อมูลอินพุต
    Tensor input = new Tensor(1, 3); // ตัวอย่างอินพุต [ปี, เดือน, วัน]
    input[0, 0] = year;
    input[0, 1] = month;
    input[0, 2] = day;
    // รับโมเดล
    Tensor output = worker.Execute(input).PeekOutput();
    float predictedTemperature = output[0];
}
```

```
// แสดงผลลัพธ์  
resultLabel.text = $"อุณหภูมิที่คาดการณ์: {predictedTemperature:F2} °C";  
// ทำความสะอาด Tensor  
input.Dispose()  
output.Dispose();
```

```
};
```

```
}
```

```
void OnDestroy()
```

```
{
```

```
    // ปิด Worker
```

```
    worker.Dispose();
```

```
}
```

```
}
```

# เชื่อมต่อโมเดลกับ Unity

- นำโมเดล .onnx หรือ .h5 เข้าใน Unity ผ่าน Barracuda
- เพิ่ม Script TemperaturePredictor ใน GameObject

# การทดสอบและปรับปรุง

- ทดสอบ  
ใส่ข้อมูลวันที่และตรวจสอบว่าโมเดลสามารถทำนายค่าอุณหภูมิได้อย่างแม่นยำ
- ปรับปรุง  
หากผลลัพธ์ไม่น่าพอใจ อาจต้องปรับโมเดลหรือเพิ่มข้อมูลฝึกเพิ่มเติม

# การใช้งาน

- หลังจากตั้งค่าทั้งหมดแล้ว คุณจะได้น้ำจอที่สามารถป้อนข้อมูล "วัน/เดือน/ปี" และแสดงผลการคำนวณค่าอุณหภูมิที่คาดการณ์ได้ทันทีหลังจากกดปุ่ม!

