

Replica ver jugar a la IA de mario windows en tiempo real

Descarga el proyecto: <https://github.com/jorgecasase/mariobrosplay-windows-gym-RL>

Pasos necesarios para correr un modelo de RL pre-entrenado con interfaz gráfica para verlo jugar. Funciona para windows nativo, no funciona en colab.

El modelo puedes entrenarlo, o cargarlo del github:

- notebook para entrenar:
https://github.com/pedroconcejero/deep_learning_2024/blob/main/mario_RL_pytorch_tutorial_400_episodes_save_every_1e4.ipynb
- modelo entrenado:
https://github.com/pedroconcejero/deep_learning_2024/blob/main/mario_net_8.chkpt
- o en mi github hay un modelo con 40000 episodios que termina el nivel:
https://github.com/jorgecasase/mariobrosplay-windows-gym-RL/blob/main/trained_mario.chkpt

1. Descarga miniconda para el entorno virtual

(tienes que poner tu email)

<https://www.anaconda.com/download/>

En la instalación añadimos al path

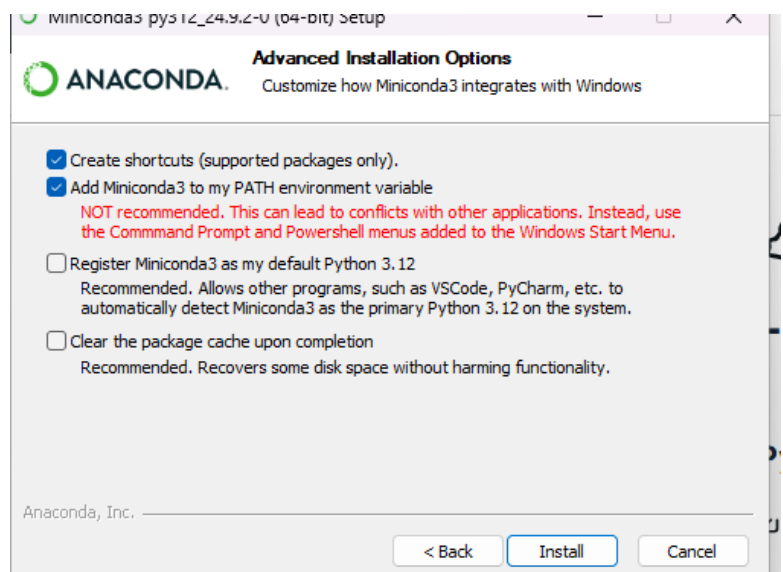
Miniconda Installers



Windows

Python 3.12

64-Bit Graphical Installer

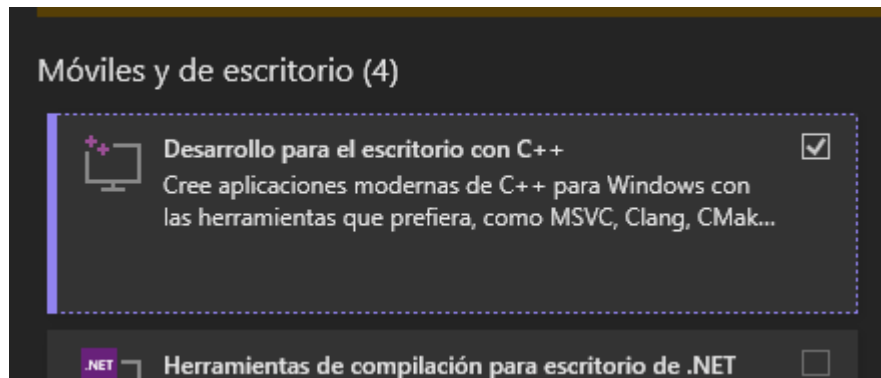


2. Descarga e instala visual studio build tools

Hay que instalar visual studio build tools para c++ porque la librería nes-py necesita compilar código c++

<https://visualstudio.microsoft.com/es/visual-cpp-build-tools/>

Marcas esta casilla e instalas (tarda un rato)



3. Creas el entorno virtual con el archivo enviroment.yml

Creas el entorno con el documento environment.yml (tarda un rato)

conda env create -f enviroment.yml

```
Directorio de C:\Users\Jorge\Downloads\mariobrosplay-windows-gym-RL-main
11/12/2024 23:20 <DIR> .
11/12/2024 23:20 <DIR> ..
11/12/2024 23:20      601 environment.yml
11/12/2024 23:20     7.272 play.ipynb
11/12/2024 23:20     233 README.md
      3 archivos      8.106 bytes
      2 dirs 404.034.191.360 bytes libres

C:\Users\Jorge\Downloads\mariobrosplay-windows-gym-RL-main>conda env create -f environment.yml
```

4. Instalamos Rust y jupyter

Hay que instalar rustup para usar jupyter, instalación estándar (1)

<https://rustup.rs/>

```
profile: default
modify PATH variable: yes

1) Proceed with standard installation (default - just press enter)
2) Customize installation
3) Cancel installation
>1
```

y ahora reinicias el cmd

conda activate mario

pip install matplotlib

pip install scikit-image

pip install jupyter

5. Creas un kernel para jupyter con tu entorno

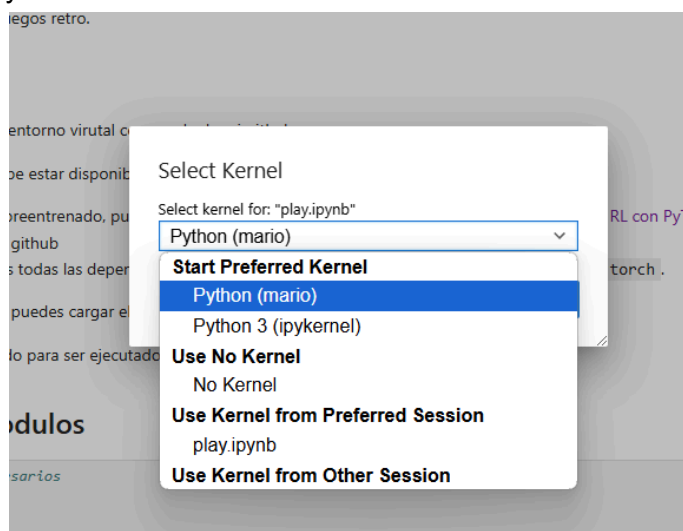
pip install ipykernel

python -m ipykernel install --user --name=mario --display-name "Python (mario)"

ahora abres jupyter con el comando

jupyter notebook

y seleccionas el kernel mario creado



6. Cargar modelo y ejecutar

Abrimos play.ipynb

Ponemos la ruta de nuestro modelo.chkpt en esta celda y ejecutamos todo el notebook.

Cargar modelo preentrenado

```
: # Configuración del directorio de guardado para puntos de control y métricas
save_dir = Path('checkpoints') / datetime.datetime.now().strftime('%Y-%m-%dT%H-%M-%S')
save_dir.mkdir(parents=True) # Crear el directorio si no existe

# Inicialización del agente Mario con un punto de control preentrenado **AQUI CARGAS TU MODELO**
checkpoint = Path('mario_net_13.chkpt')
mario = Mario(state_dim=(4, 84, 84), action_dim=env.action_space.n, save_dir=save_dir, checkpoint=checkpoint)

# Fijar la tasa de exploración al mínimo
mario.exploration_rate = mario.exploration_rate_min

# Inicialización del registro de métricas
logger = MetricLogger(save_dir)
```

Ahora se nos abre una ventana y podemos ver la interfaz gráfica del juego y los entrenamientos de la IA.

