



**NOMBRE:**

ALEX BENAVIDEZ

**CARRERA:**

INGENIERIA EN SISTEMAS

**MATERIA:**

INTELIGENCIA ARTIFICIAL

**PROFESOR:**

ING. DIEGO QUISI

**FECHA:**

09/07/2020

## **INTRODUCCION**

La inteligencia artificial en si siempre ha estado ligada a los videojuegos, debido a que varias aplicaciones que se desarrollaron buscaban de cierta manera crear agentes capaces de realizar tareas que requieran inteligencia. Una de estas tareas eran precisamente resolver juegos con cierto grado de complejidad como por ejemplo el ajedrez o las damas.

Uno de los primeros programas que se realizaron fue el juego de tres en raya, el cual fue programado por A. S Douglas en el año 1952. Después Arthur Samuel invento una forma de aprendizaje de maquina denominado aprendizaje de refuerzo, en donde utilizo un programa que tenía la capacidad de aprender a jugar damas chinas contra sí mismo.

En la actualidad y hace ya tiempo se han utilizado diferentes algoritmos para la realización de juegos que son muy conocidos, como por ejemplo se ha utilizado una red neuronal para el conocido juego de Mario Bros o un algoritmo genético para el juego tan famoso de la serpiente.

## **DESARROLLO DE CONTENIDOS**

### **Universe**

Es una plataforma que permite medir y entrenar la inteligencia general de una Inteligencia Artificial en el siministro mundial de lo que son juegos, sitios web y otras aplicaciones. Proporciona una interfaz simple de Gym para cada entorno de Universe.

Universe permite a un agente de IA utilizar una computadora como lo hace un humano: mirando los píxeles de la pantalla y operando un teclado y un ratón virtuales. Debemos entrenar los sistemas de IA en toda la gama de tareas que esperamos que resuelvan, y Universe nos permite entrenar a un solo agente en cualquier tarea que un humano pueda completar con una computadora.

Universe permite que cualquiera pueda entrenar y evaluar agentes de Inteligencia Artificial en una gama realmente alta de entornos complejos en tiempo real.

Universe hace posible que cualquier programa que ya exista pueda convertirse en un entorno OpenAI Gym. Esto lo hace posible empacando el programa en un contenedor Docker y presentando a la Inteligencia Artificial con la misma interfaz que utiliza una persona, es decir enviando eventos de teclado y mouse y sucesivamente recibiendo pixeles de pantalla.

## Gym

Es un kit de herramientas para poder desarrollar y comparar algoritmos de aprendizaje de refuerzo. Es una colección de problemas de prueba que se puede usar para poder elaborar algoritmos de aprendizaje de refuerzo.

Gym viene con un conjunto amplio de entornos que van desde fácil a difícil e involucran muchos tipos diferentes de datos. Como, por ejemplo:

- Control clásico y texto de juguete: completan tareas a pequeña escala, principalmente de la literatura RL. Están aquí para empezar.
- Algorítmico: realiza cálculos como la adición de números de varios dígitos y la inversión de secuencias. Uno podría objetar que estas tareas son fáciles para un equipo. El reto es aprender estos algoritmos únicamente a partir de ejemplos. Estas tareas tienen la propiedad agradable que es fácil variar la dificultad variando la longitud de la secuencia.
- Atari: jugar juegos clásicos de Atari. Hemos integrado el entorno de aprendizaje Arcade (que ha tenido un gran impacto en la investigación de aprendizaje de refuerzo) en una forma fácil de instalar.
- Robots 2D y 3D: controlan un robot en simulación. Estas tareas utilizan el motor de física MuJoCo, que fue diseñado para la simulación rápida y precisa del robot. Se incluyen algunos entornos de un punto de referencia reciente de los investigadores de UC Berkeley (que por cierto se unirán a nosotros este verano). MuJoCo es un software propietario, pero ofrece licencias de prueba gratuitas.

## Ejemplos de Juegos utilizando Gym

```
import gym

env = gym.make("Phoenix-v3")

observation = env.reset()

for _ in range(1000):
    env.render()

    action = env.action_space.sample() # your agent here (this takes random actio$
    observation, reward, done, info = env.step(action)

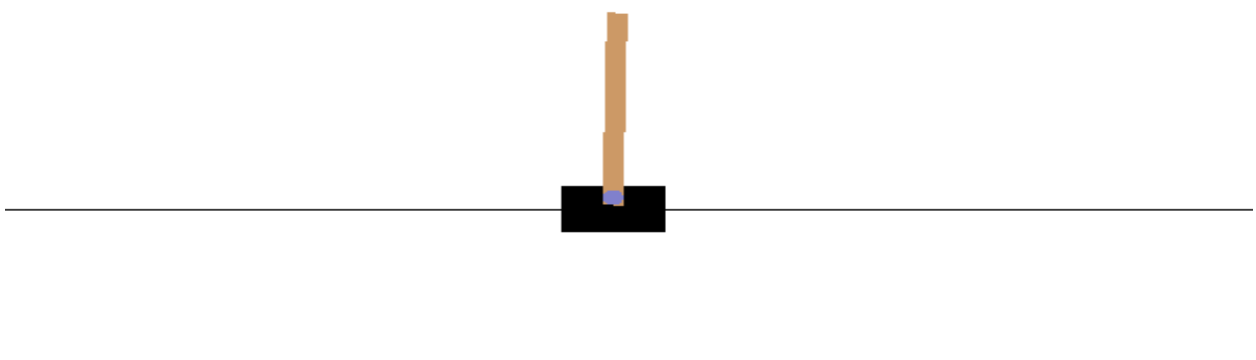
    if done:
        observation = env.reset()

env.close()
```



```
import gym
env = gym.make("CartPole-v1")
observation = env.reset()
for _ in range(1000):
    env.render()
    action = env.action_space.sample() # your agent here (this takes random actio$
    observation, reward, done, info = env.step(action)
    if done:
        observation = env.reset()
env.close()
```

/home/alex/anaconda3/lib/python3.7/site-packages/ipykernel\_launcher.py



## **CONCLUSIONES:**

- Gym nos permite realizar videojuegos utilizando la inteligencia artificial, lo que permite que estos videojuegos puedan aprender mediante el transcurso de su ejecución
- Cabe recalcar que la librería universo no está disponible en este momento.