

> TECNOLOGÍA

Inteligencia artificial revela cuáles son los barrios más 'cool' de Colombia

El estudio ha generado gran interés entre los urbanistas y amantes de la moda.

20 de octubre de 2023 Por: Redacción El País

Inteligencia artificial

La IA bien utilizada podría mejorar tratamientos para la salud: OMS

Según el organismo, la IA es muy prometedora para la atención sanitaria.

La inteligencia artificial reveló cuál es la mejor hora para despertarse

La inteligencia artificial continúa ofreciendo valiosos consejos para mejorar la calidad de vida de las personas.



JESSE COGLAN

El Pentágono de Estados Unidos está probando si la IA puede planificar una respuesta a una guerra global

How The Games Industry Can Leverage Advances In AI To Revolutionize NPCs



Kylan Gibbs Forbes Councils Member

Forbes Technology Council COUNCIL POST | Membership (Fee-Based)

TECH

Artificial intelligence is helping old video games look like new

Generative AI in Games Will Create a Copyright Crisis

Titles like *AI Dungeon* are already using generative AI to generate in-game content. Nobody knows who owns it.

Una startup japonesa fue premiada por crear un sensor gustativo basado en IA

ACTUALIDAD / Noticias

Nvidia presenta DLSS 3.5, la nueva versión de su IA que mejora la calidad y rendimiento de los videojuegos

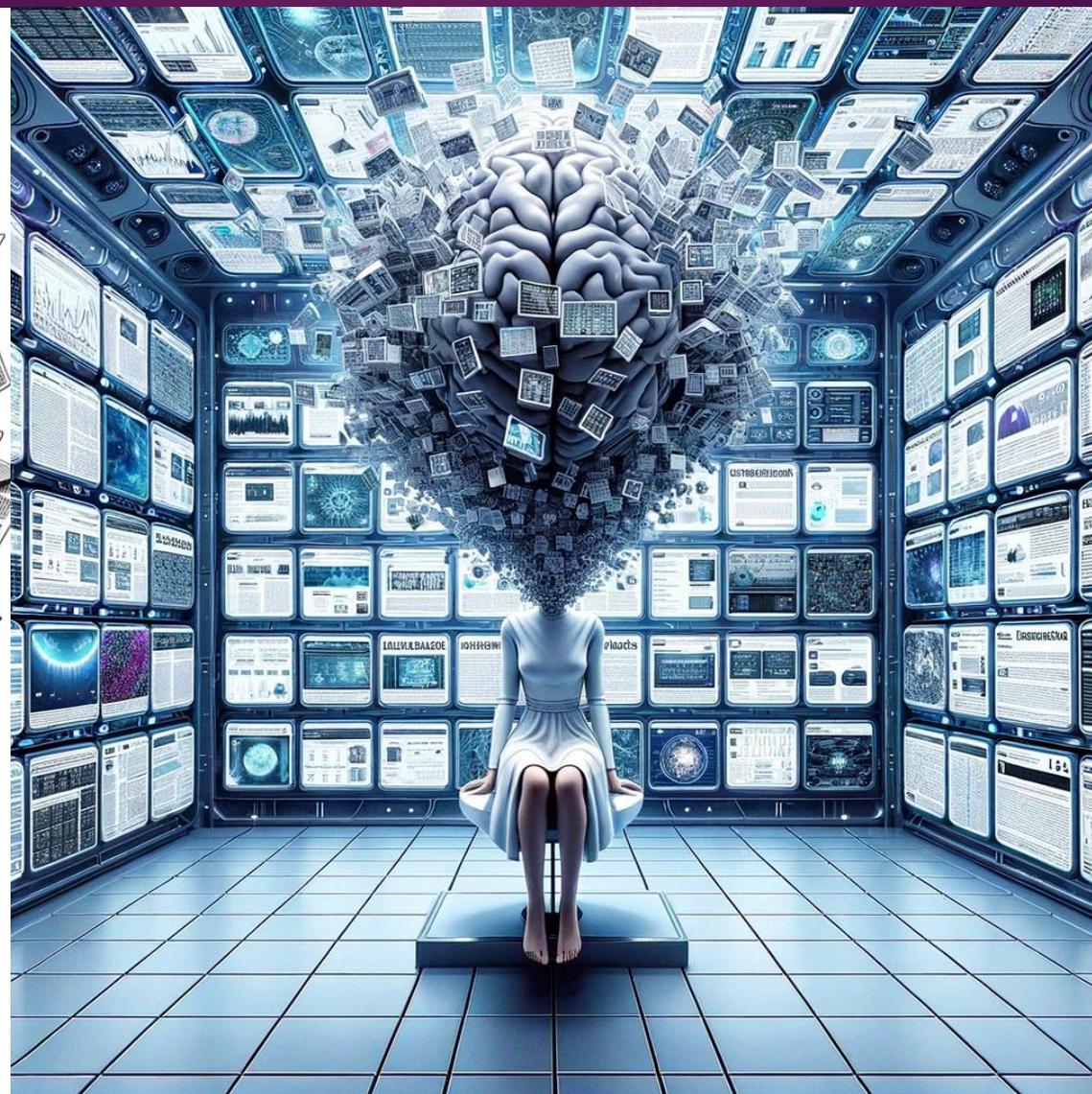
Generative AI is a game changer for higher education

As in most areas of society, generative AI presents risks and opportunities for the higher education sector. But regardless of which side of the debate you fall, it will be a revolutionary technology

HACIA EL
FUTURO
INTELIGENCIA ARTIFICIAL & EDUCACIÓN

UK school gets AI chatbot as its 'principal headteacher'

The school headmaster told that AI chatbot Abigail has been developed with an abundance of knowledge in machine learning and educational management and an ability to process huge collections of data.



Perspectivas
HACIA EL
FUTURO
INTELIGENCIA ARTIFICIAL & EDUCACIÓN



Más que aprender a saber
como usarlo, el deber ser es
intentar entender **que es y**
como funciona

Investigadores de IA, los videojuegos son tus amigos

INTELIGENCIA ARTIFICIAL Y VIDEOJUEGOS

Oswaldo Andrés Ordóñez Bolaños
Octubre, 23



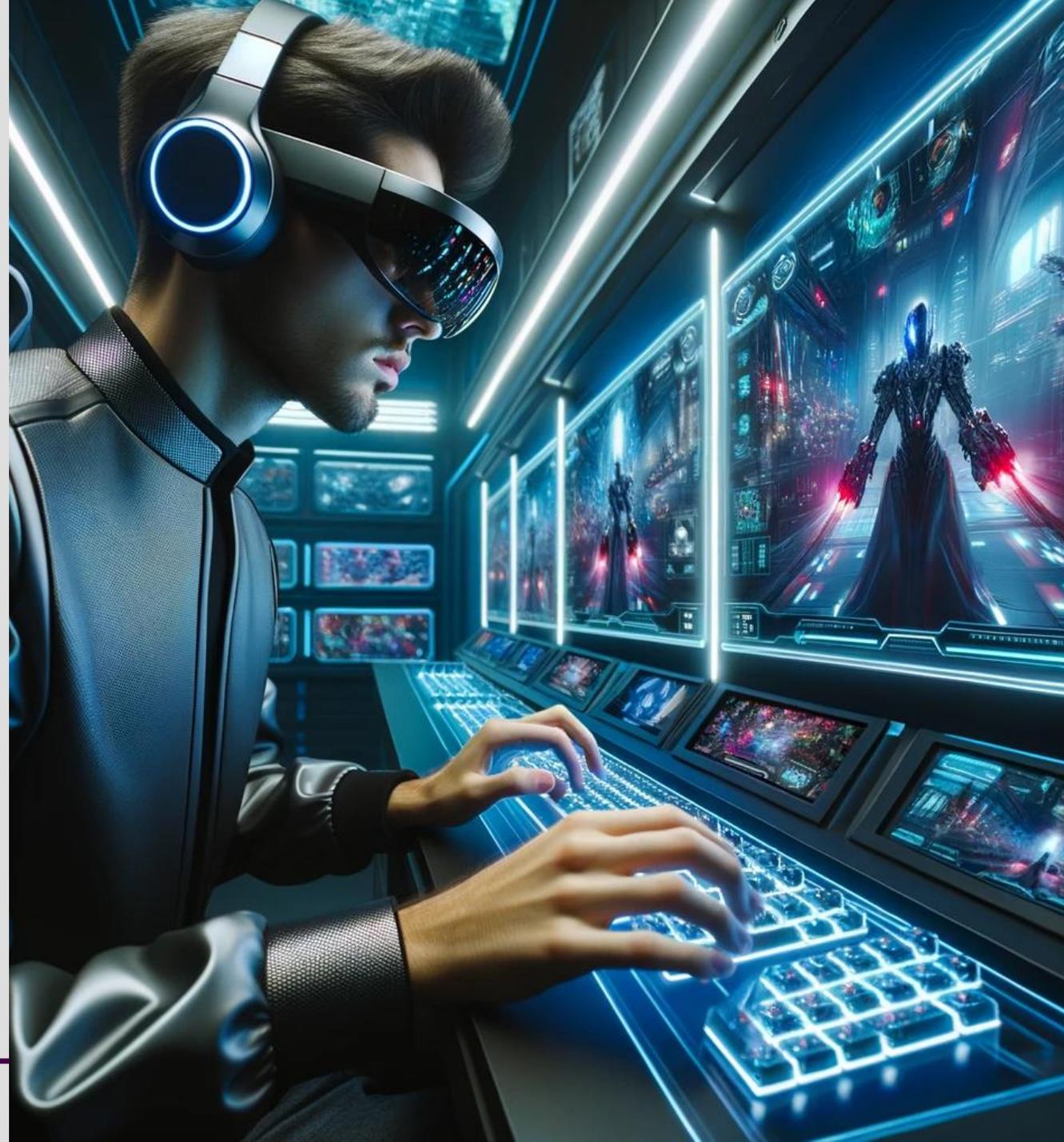
Oswaldo Andrés Ordóñez Bolaños

Ingeniero electrónico – Magister en computación

Líder de innovación y nuevas tecnologías del Instituto Departamental de Bellas Artes – Docente Universidad del Valle – Investigador del grupo de investigación Smart Data Analysis Systems Group

Técnicas de machine learning para la detección de emociones a través de señales del cerebro – Gemelos digitales aplicados a escenarios educativos – reinforcement learning para el desarrollo de juegos pervasivos – Desarrollo de experiencias XR para juegos serios

- ¿ Qué es AI ?
- ¿ Qué hacen los humanos con los juegos?
- ¿ Qué hace la IA con los juegos ?





La inteligencia artificial

la ciencia de hacer que las computadoras o máquinas **piensen y aprendan** como lo hacen los humanos.

Pensar

Capacidad de una máquina para procesar información y tomar decisiones basadas en ella.

Aprender

La máquina puede mejorar su rendimiento en una tarea específica con el tiempo y la experiencia.

Técnicas

Aprendizaje automático

Redes neuronales

Procesamiento de lenguaje natural

Visión por computador

**Algoritmos de
búsqueda y
optimización**

Aprendizaje por refuerzo

¿ Qué hacemos los humanos con los juegos ?

Estudiarlos, testearlos

Jugarlos

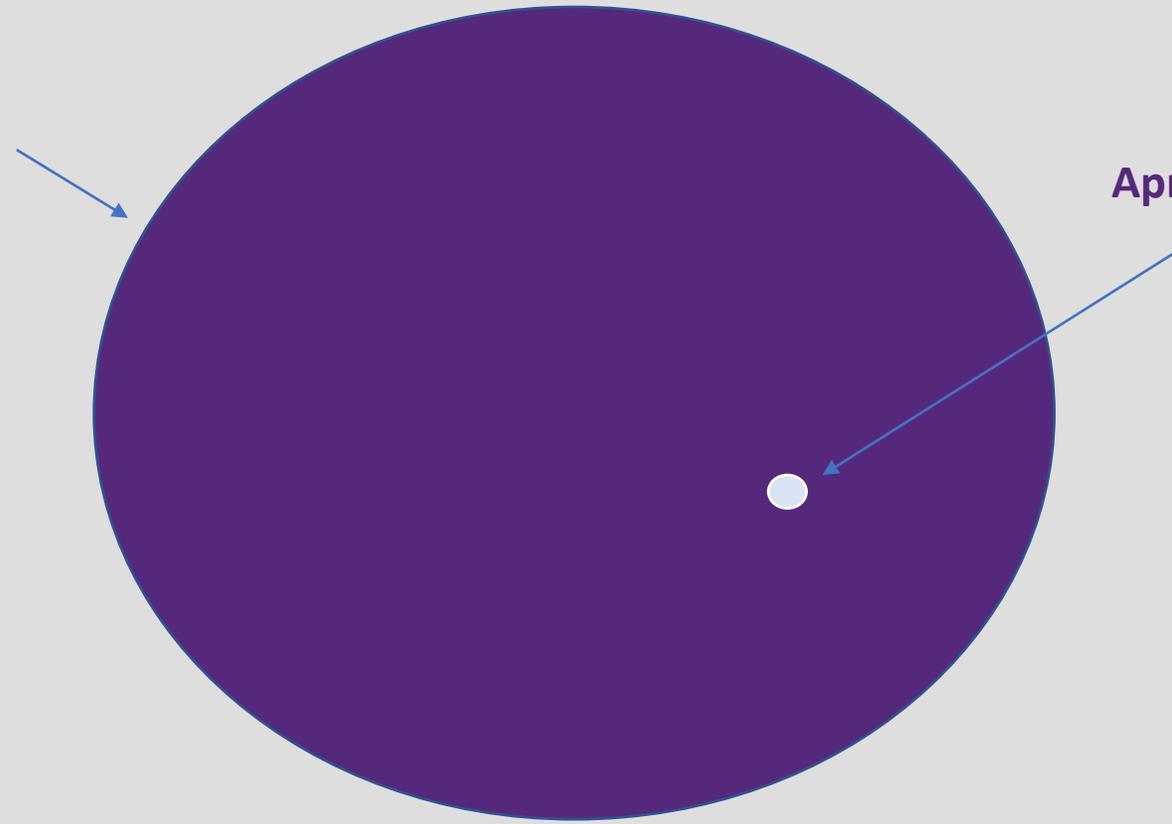
Desarrollar contenidos para ellos (props)

Diseñarlos y desarrollarlos

¿ Qué hace la IA con los juegos ?



IA aplicado a juegos

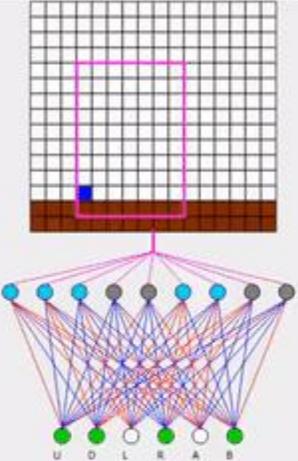


Aprender a jugar juegos de mesa

- Videojuegos como testbeds / benchmarks
(Estudiarlos)
- IA jugando videojuegos *(Jugarlos)*
- IA para nuevos mundos / nuevas posibilidades *(Desarrollar contenidos)*
- Juego con personajes con manera humana/creíble
- Rutas inteligentes *(Desarrollar contenidos)*
- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador *(Diseñarlos y desarrollarlos)*



- Videojuegos como testbeds / benchmarks



The diagram shows a 16x16 grid representing the game state. A pink rectangle highlights a 4x4 area. Below the grid is a neural network with 8 input nodes (green circles labeled U, D, L, R, A, B) and 8 output nodes (blue and grey circles). Lines connect the input nodes to the output nodes.

Generation:	1214	Offspring:	10, 90
Individual:	Replay	Lifespan:	Infinite
Best Fitness:	0	Mutation:	Static 5.0%
Max Distance:	48	Crossover:	Roulette
Num Inputs:	80	SBX Eta:	100.0
Trainable Params:	789	Layers:	[80, 9, 4]



- IA jugando videojuegos

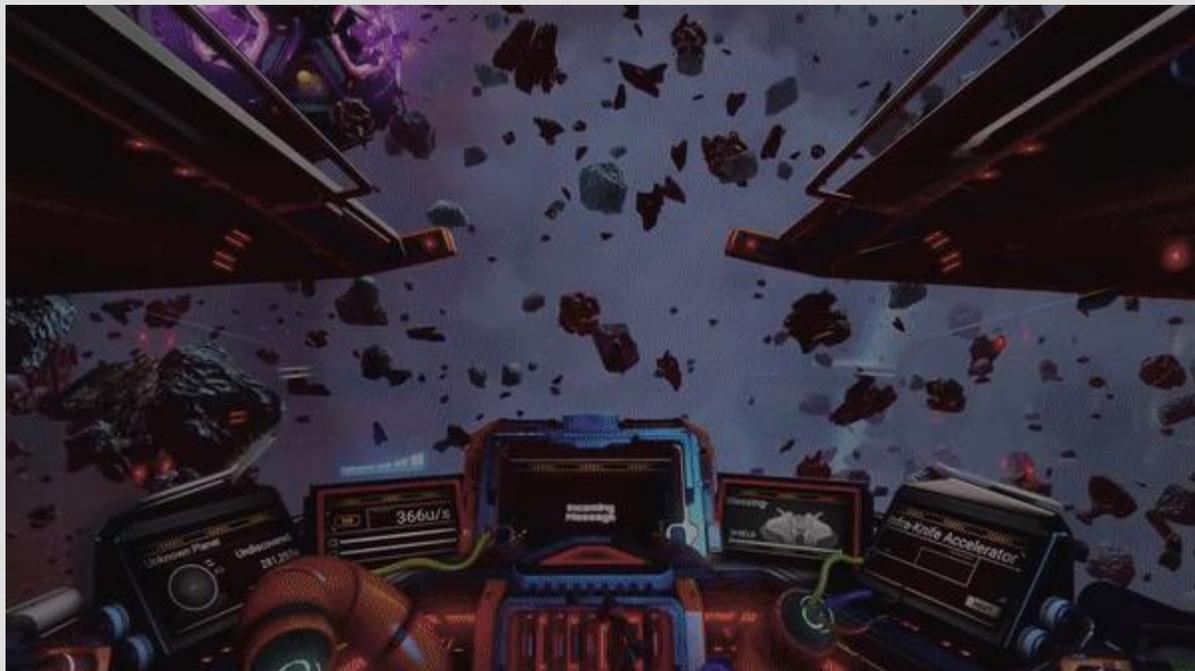
Playing Atari 2600 with deep reinforcement learning
Deep mind ,2013



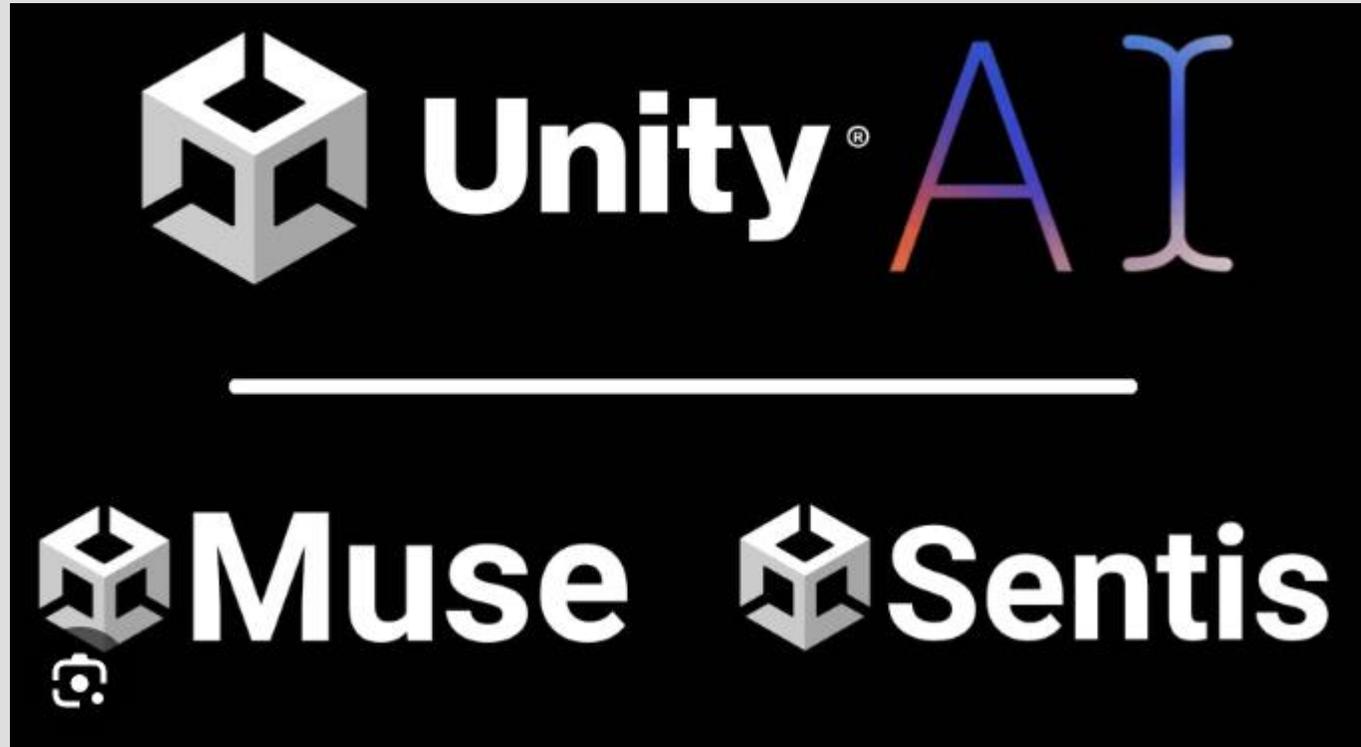
Dota 2 with Large Scale Deep Reinforcement Learning+
Open AI, 2019



- IA para nuevos mundos / nuevas posibilidades



- IA para nuevos mundos / nuevas posibilidades



- IA para nuevos mundos / nuevas posibilidades

IA TOOLKIT

Herramientas de inteligencia artificial organizadas

Explora una caja de herramientas de inteligencia artificial, meticulosamente categorizada y enriquecida con información esencial de cada IA. Accede a enlaces y tutoriales para una introducción efectiva. Además, encuentra vídeos informativos que te orientarán antes de aventurarte en el fascinante mundo de la IA.

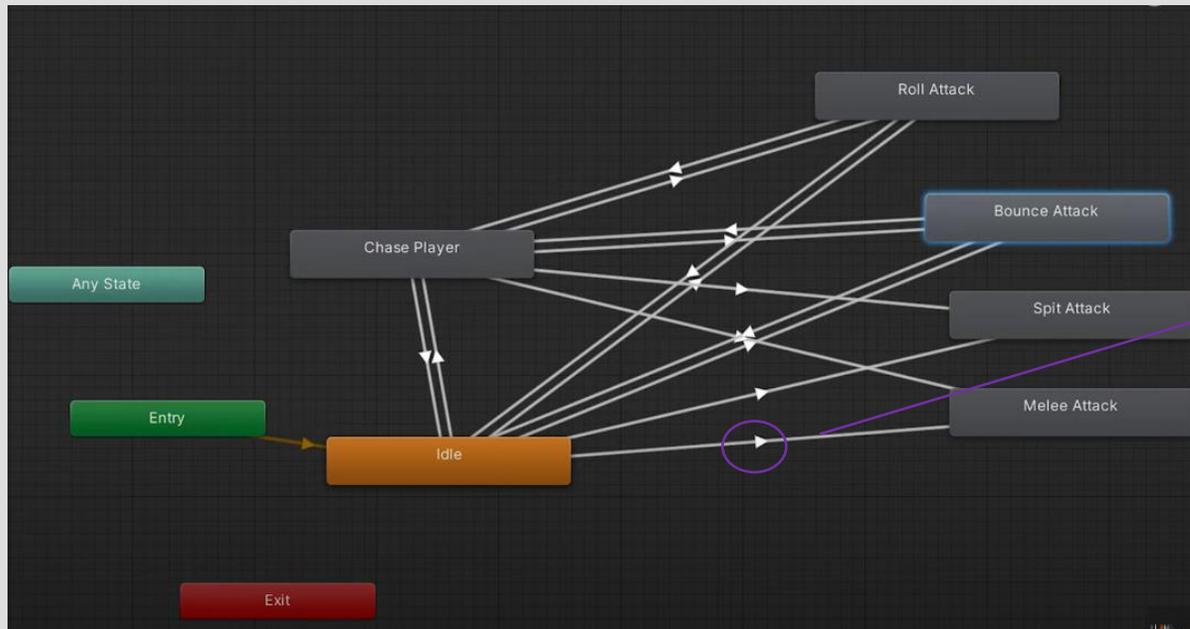


[IA TOOLKIT MILENOTE](#) 

[IA Toolkit - Milanote](#)

- Juego con personajes con manera humana/creíble

Maquina de estados finitos



Agente inteligente

- Dotarlo de sentidos al NPC
- Implementar estados
- Reaccionar dependiendo de una condición

* Rutas inteligentes

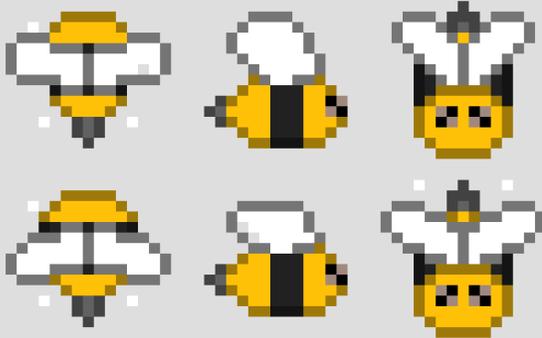
- Rutas inteligentes (Pathfinding)



Videojuego basado en los conceptos de explotación, exploración y genética para lograr que unas abejas lleguen a su nido de manera inteligente.

- Rutas inteligentes (Pathfinding)

AGENTE



OBJETIVO



- Rutas inteligentes (Pathfinding)



- Rutas inteligentes (Pathfinding)



- Rutas inteligentes (Pathfinding)



Ruta desde una fuente a un destino evitando obstáculos y minimizando los costos.

- Rutas inteligentes (Pathfinding)

GRAY WOLF OPTIMIZER (GWO)

Distancia euclidiana

$$d(P1, P2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

solución = [X1, Xn] *posición del agente*

$$n = [\text{Objetivo} - \text{Origen}]$$

Algoritmo 1. GWO PATHFINDING

Inicializar a, A y C (variables GWO)

Inicializar punto de partida

Inicializar ubicación objetivo

Inicializar ubicación obstáculos

for (i=0; i<=n; i++) do

 Generar población Xi(t)

 Evaluar **función objetivo** para cada agente de búsqueda (Search Agent)

end for

X_{α} = Mejor agente de búsqueda.

X_{β} = Segundo mejor agente de búsqueda.

X_{δ} = Tercer mejor agente de búsqueda.

Repeat

 for each search agent

 Actualizar posiciones de cada uno de los agentes de búsqueda

 end for

 update a, A y C (Variables GWO)

 Calcular fitness de los agentes de búsqueda

 update $X_{\alpha}, X_{\beta}, X_{\delta}$

 t=t+1

Until (t<Maxiter)

Return Mejor solución X_{α} - Mejor posición

- Rutas inteligentes (Pathfinding)

Número de agentes = 50

Número máximo de iteraciones = 5000

Límite inferior = -100

Límite superior = 11

Punto de inicio = [0,0]

Ubicación objetivo = [18,15]

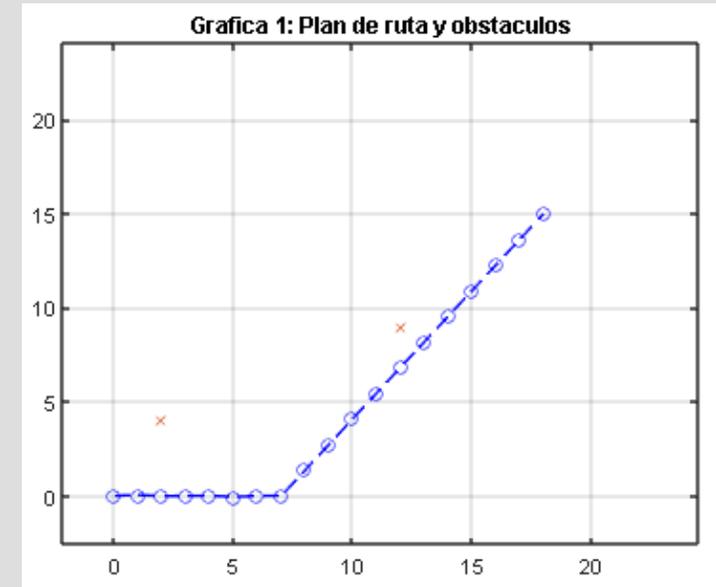
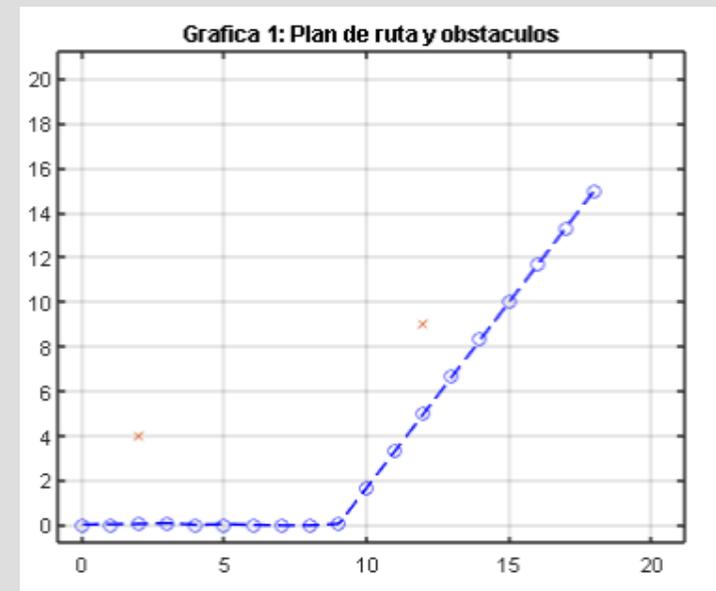
Número de obstáculos = 2

Ubicación obstáculo 1 = [2,4]

Ubicación obstáculo 2 = [10,9]

Rango obstáculo = 2

Valor obstáculo = 10



- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

APRENDIZAJE POR REFUERZO

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Aprendizaje automático

Aprendizaje supervisado: El maestro son los **ejemplos**.

- Hay algunos ejemplos etiquetados, se aprende patrones basados en esos ejemplos.

Aprendizaje no supervisado: **No** hay maestro.

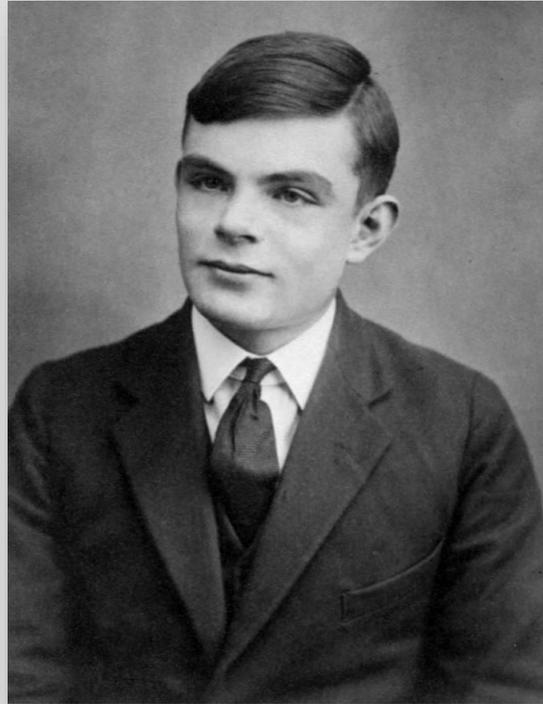
- Aprender desde ejemplos sin etiqueta.

Aprendizaje reforzado: El maestro es la **experiencia**. “Aprendiendo con un crítico”

- Hay un mundo, se aprende patrones explorándolo.

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Intelligent Machinery, 1948



¿ Puede llegar una maquina a ser inteligente?
Maquina de estados discretos
Sistema de placer/dolor que podría guiar su aprendizaje

The Hedonistic Neuron — A Theory of Memory, Learning, and Intelligence, 1982.



Harry Klopf evidencia que los humanos, después de haber asegurado la homeostasis (la búsqueda de un bien), buscan maximizar el placer, no estabilizarlo

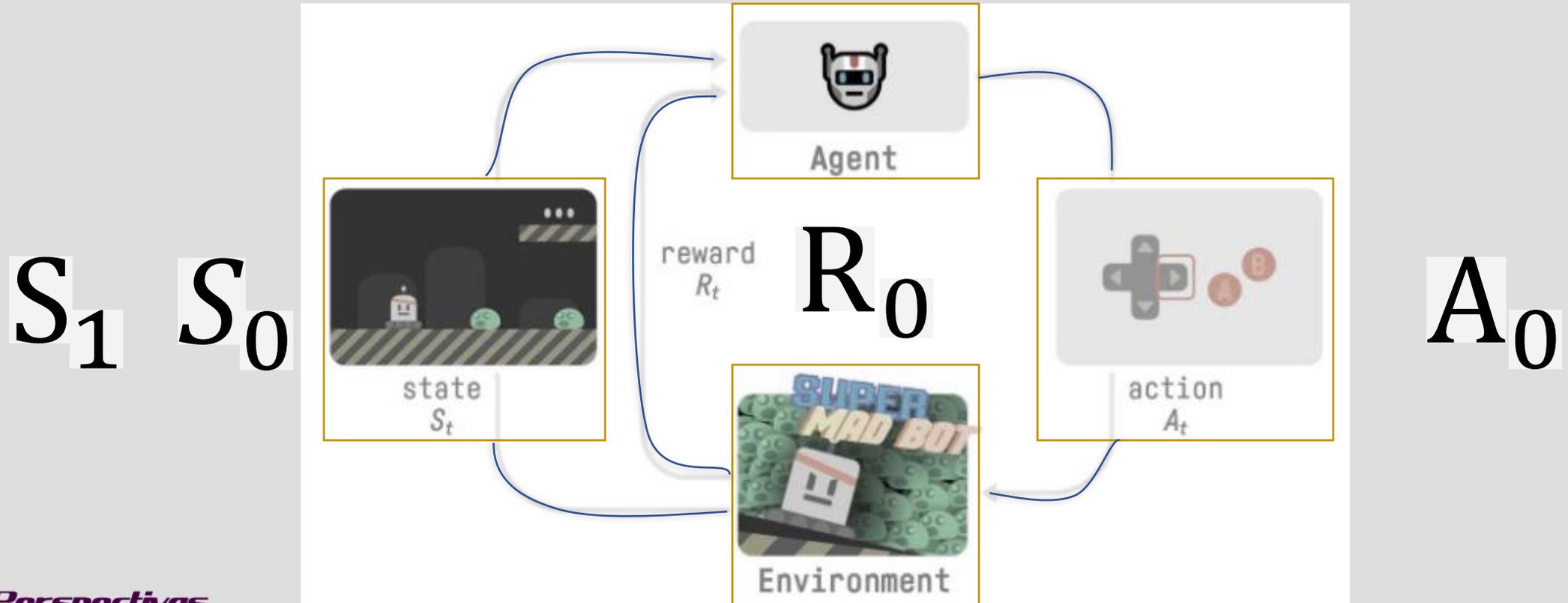
- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

La gran pintura

El aprendizaje por refuerzo es un marco de trabajo para resolver **tareas de control** (también llamadas problemas de decisión) mediante la construcción de **agentes** que aprenden del entorno al interactuar con él a través de **prueba y error** y recibir recompensas (**positivas o negativas**) como retroalimentación única.

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Aprendizaje por refuerzo



- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Marco de trabajo algunos conceptos

Estado (s)

Descripción completa del estado del mundo (no hay información oculta). Entorno completamente observado.

Observación (O)

Descripción **parcial** del estado. Entorno parcialmente observado.

Espacio discreto

El número de acciones posibles son **finitas**.

Action Space

El espacio Acción es el conjunto de todas las acciones posibles en un entorno.

Espacio Continuo

el número de acciones posibles son **infinitas**.

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Marco de trabajo algunos conceptos



- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Recompensas y descuentos

La recompensa es fundamental para el RL porque es la única **información para el agente**. Gracias a ella, nuestro agente sabe si la acción realizada fue buena o no.

$$G_T = R(\tau) = r_{t+1} + r_{t+2} + r_{t+3} + r_{t+4} + \dots$$

Trayectoria: Secuencia de estados y acciones.

*Return: Recompensas acumuladas.
Rendimiento esperado*

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

El objetivo del agente es
**maximizar las recompensas
acumuladas**

No solo las recompensas inmediatas

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Recompensas y descuentos



$$G_T = R(\tau) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \dots$$

Recompensas acumuladas esperadas con descuento

Gamma: tasa de descuento.

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Recompensas y descuentos

Las recompensas que llegan antes son más probables (Mayor gamma, menor descuento) porque que son más predecibles que la recompensa futura a largo plazo, depende del gamma decidimos si exploración o explotación.

$$G_T = R(\tau) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \dots$$

Recompensas acumuladas esperadas con descuento

Gamma: tasa de descuento.

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

El objetivo del agente es **maximizar**
las recompensas esperadas
acumuladas con descuento

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Compensación de exploración y explotación

Exploración

La exploración consiste en explorar el medio ambiente probando acciones aleatorias para encontrar más información sobre el medio ambiente.

Explotación

La explotación consiste en explotar información conocida para maximizar la recompensa.

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Compensación de exploración y explotación



- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Q-learning

Q-learning ayuda a aprender la mejor acción a tomar en cada situación, basándose en las recompensas que has recibido en experiencias pasadas, para maximizar tus recompensas totales en el futuro.



- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Q-Learning con Taxi-v3

El objetivo es capacitar a **un agente de taxi** para que navegue en esta ciudad para transportar a sus pasajeros desde el punto **A** al punto **B**.

```
env = gym.make("Taxi-v3")
env.render()
```

- Ambiente 5x5
- El taxi aparecerá de manera aleatoria
- El pasajero aparecerá de manera aleatoria en (R, B, G, Y) y desea ir a algunos de los lugares.
- **6** acciones (N,S,W,E,Recoger, dejar)

Sistema de recompensas:

- **-1** por cada tiempo de paso
- **+20** por entregar con éxito al pasajero
- **-10** por acciones ilegales (recoger o dejar al pasajero en el exterior del destino).

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Q-Learning con Taxi-v3

Crear la Q-table e inicializarla

```
state_space = env.observation_space.n
action_space = env.action_space.n
# Crear nuestra Q-tabla con state_size filas and action_size columnas (500x6)
Q = np.zeros((state_space, action_space))
print(Q)
print(Q.shape)
```

```
[[0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 ...
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
(500, 6)
```

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Q-Learning con Taxi-v3

Se define parámetros

```
total_episodes = 25000      # Numero total de episodios
total_test_episodes = 50    # Numero total de episodios test
max_steps = 200             # Maximo de pasos por episodio

learning_rate = 0.01        # Taza de aprendizaje
gamma = 0.99                # Taza de descuento

# Parametros de exploración / explotación
epsilon = 1.0                # Taza de exploración epsilon
max_epsilon = 1.0           # Probabilidad de exploración al empezar
min_epsilon = 0.001         # Probabilidad minima de exploración
decay_rate = 0.01           # Tasa de decaimiento exponencial para exploración
```

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Q-Learning con Taxi-v3

Se define la estrategia de Epsilon Greedy

```
def epsilon_greedy_policy(Q, state):  
    if(random.uniform(0,1) > epsilon):  
        # explotación  
        action = np.argmax(Q[state])  
    else:  
        # exploración  
        action = env.action_space.sample()  
  
    return action
```

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Q-Learning con Taxi-v3

Se implementa el algoritmo de Q-learning y entrenamos el agente

```
for episode in range(total_episodes):
    # Reinicio ambiente
    state = env.reset()
    step = 0
    done = False
    # Reducimos para cada episodio el epsilon (Se necesita más explotación cada vez)
    epsilon = min_epsilon + (max_epsilon - min_epsilon)*np.exp(-decay_rate*episode)
    for step in range(max_steps):
        # Se inicia con la politica epsilon greedy
        action = epsilon_greedy_policy(Q, state)
        # Dependiendo de la interacción de la acción con el ambiente, tenemos un nuevo estado y una recompensa
        new_state, reward, done, info = env.step(action)

        # Actualizamos Q(s,a):= Q(s,a) + lr [R(s,a) + gamma * max Q(s',a') - Q(s,a)]
        Q[state][action] = Q[state][action] + learning_rate * (reward + gamma *
                                                                np.max(Q[new_state]) - Q[state][action])

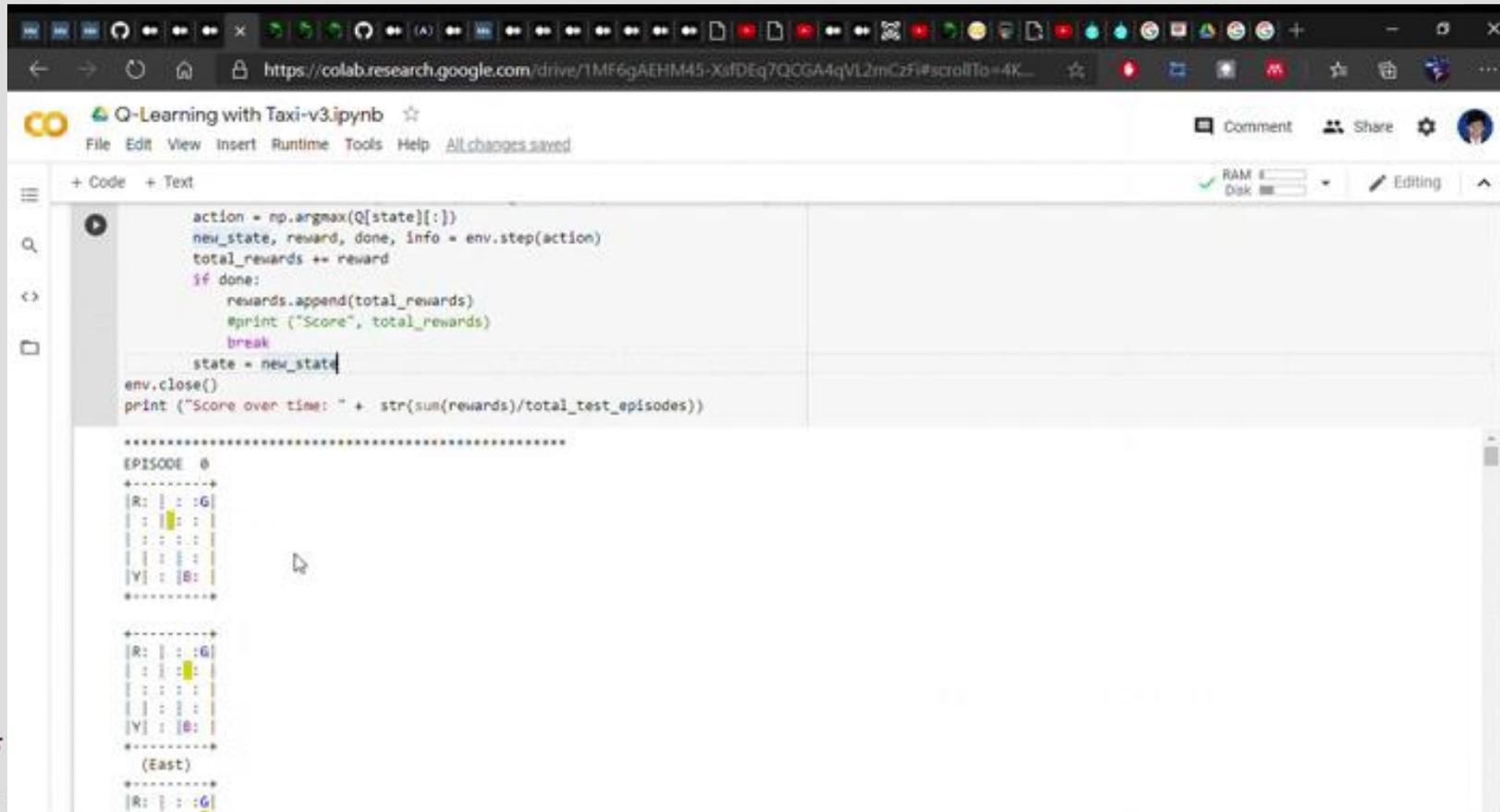
        # Se analiza si ya terminó el episodio
        if done == True:
            break

    state = new_state
```

- Modelar/Diseñar/Desarrollar la experiencia, estilo o preferencia del jugador

Q-Learning con Taxi-v3

Taxi autónomo



```
action = np.argmax(Q[state][:])
new_state, reward, done, info = env.step(action)
total_rewards += reward
if done:
    rewards.append(total_rewards)
    #print ("Score", total_rewards)
    break
state = new_state
env.close()
print ("Score over time: " + str(sum(rewards)/total_test_episodes))
```

```
*****
EPISODE 0
[R: | : :G]
| : | : | |
| : | : |
| : | : |
|Y| : |B|
*****

[R: | : :G]
| : | : | |
| : | : |
|Y| : |B|
*****
(East)
[R: | : :G]
```

Reinforcement learning en educación

Generar un **aprendizaje personalizado** de acuerdo a los intereses y la forma de aprender del estudiante. Shawsy et al. - Towards a Personalized Learning Experience Using Reinforcement Learning (2019)

Elegir el contenido siguiente de acuerdo a los **estados**, pero de acuerdo a un simulador basado en modelos cognitivos. Siddarth et al. – Accelerating human learning with deep reinforcement learning(2017)

Aula inteligente con múltiples sensores, como cámaras y un creador de cuestionarios para **rastrear el proceso de aprendizaje** de los estudiantes y aplique técnicas de aprendizaje por refuerzo para **proporcionar una guía de aprendizaje**. Lui et al. – Towards Smart Educational Recommendations with Reinforcement Learning in Classroom(2018)

MUCHAS GRACIAS

Oswaldo.ordonez@correounivalle.edu.co

+57 315 719 1848