

# PROYECTO FINAL – INTELIGENCIA ARTIFICIAL

Integrantes: Christian MORALES – Cedula: 8-896-891

**RESUMEN:** El proyecto final de inteligencia artificial abarcaremos algunos puntos tratados durante el semestre entre estos puntos está el uso del lenguaje Python, Python es un lenguaje de programación con múltiples funciones y adaptabilidad a diferentes problemas cotidianos en nuestro caso lo usaremos para trabajar inteligencia artificial, para nuestro proyecto lo usaremos mas concretamente para trabajar con algoritmos de aprendizaje reforzado, dicho algoritmo nos ayudara a resolver o terminar un nivel de vectorm2 de nuestra elección por medio de premios para nuestra inteligencia.

**PALABRAS CLAVES:** inteligencia artificial, Python , aprendizaje por refuerzo, técnicas de inteligencia artificial

## INTRODUCCION

En este breve pero instructivo proyecto resolveremos e implementaremos una técnica de inteligencia artificial en este proyecto final elegimos la técnica de aprendizaje reforzado, esta técnica como tal es un poco complicada ya que debemos tener muchas variables en cuentas, para nuestra fortuna es uno de los temas que tocamos durante todo el semestre, llegamos a ver desde ejemplos con el juego de Sonic de Sega Genesis, donde observamos un algoritmo de aprendizaje por

refuerzo por generaciones, donde si una generación no era acta como tal se hacía un genocidio, bueno en nuestro caso usaremos esta misma técnica y la implementaremos a nuestro juego vector man 2.

## OBJETIVOS :

El objetivo de este proyecto es la implementación de una técnica de aprendizaje de inteligencia artificial.

## DESARROLLO DEL PROYECTO

Para el desarrollo de este proyecto se debe seguir los siguientes pasos:

- La instalación de Python3.
- La instalación de openai retro.
- Creación y activación del entorno virtualenv.
- La instalación de gym-retro.
- La importación de nuestra rom

Siguiendo estos pasos podremos ejecutar y ver correr nuestro juego de una forma sencilla , estos pasos son importantes ya que nos ayuda a verificar que nuestra rom no tiene algún error .

Para la implementación de nuestra técnica de aprendizaje por refuerzo primero debemos definir que es aprendizaje por refuerzo, el aprendizaje por refuerzo es un área del aprendizaje automático inspirada en la psicología conductista, cuya ocupación es determinar qué

acciones debe escoger un agente de software en un entorno dado con el fin de maximizar alguna noción de "recompensa" o premio acumulado.

Ya sabiendo que es aprendizaje por refuerzo podemos pasar a la implementación para este necesitamos importar algunas librerías necesarias en nuestro código tales como:

```
import retro
import numpy as np
import cv2
import neat
import pickle
import gym
import random
```

Estas librerías nos ayudaran tanto como para controlar nuestro agente dentro del juego como son el cálculo matemático para el posicionamiento de X y Y de nuestro agente como lo es el entrenamiento de nuestra inteligencia artificial como lo es cv2, para generar números aleatorios como lo es random.

También se debe definir el nombre de nuestra rom para nuestro caso será Vector man 2 y del nivel que elegiremos para entrenar nuestra inteligencia artificial para esto usaremos la siguiente línea de comando:

```
env = retro.make('Vectorman2-Genesis', state='Lev
```

Ahora para controlar nuestro agente podemos controlarlo de una forma aleatoria usando la siguiente línea de comando:

```
ac = env.action_space.sample()
```

Para la evaluación de nuestra población ya que nuestro algoritmo esta basado en genes de una población usaremos la siguiente función llamada :

```
def eval_genomes(genomes, config):
```

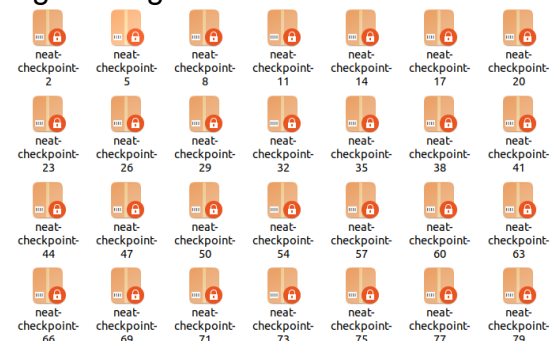
También usaremos un archivo texto para definir algunas variables como lo son las recompensas , el limite de población, cuantas después de cuantas generaciones se hace una limpia es decir un genocidio, la duración del algoritmo etc., dicho archivo es llamado config-feedforward y la línea de código que definiremos el nombre de nuestro archivo de configuración es:

```
config = neat.Config(neat.DefaultGenome, neat.DefaultReproduction,
                    neat.DefaultSpeciesSet, neat.DefaultStagnation,
                    'config-feedforward')
```

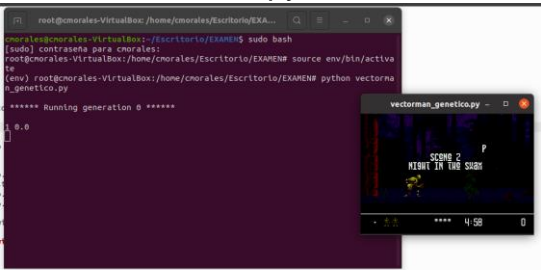
El ente ganador o quien finalice el nivel sera guardado por el siguiente líneas de comando:

```
with open('winner.pkl', 'wb') as output:
    pickle.dump(winner, output, 1)
```

Las generaciones que no sean eficientes o no cumplan con los requisitos sera eliminadas y las mejores se guardaran en un punto de guardado para ser evaluadas en la siguiente generacion .



para finalizar el desarrollo de este proyecto solo queda ejecutar nuestro archivo extension .py.



y observar como nuestro algoritmo de inteligencia artificial va probando y evaluando las acciones aleatorias de nuestro ente en el juego de esta forma va comenzar un ciclo de selección y eliminación ,puede pasar generacion sin llegar a un resultado eficiente dentro de nuestros margenes por eso este algoritmo de entrenamiento por refuerzo estan interesante porque el ente que recibe una cierta recompensa por distancia recorrida es quien prebalece mientras los demas son eliminados.

LIMPIAR DATA CON JUNIPER

La limpia de datos es importante en alguno algoritmo de aprendizaje en nuestro caso no la usaremos porque nuestro algoritmo esta basado en error y ensayo, pero lo mencionamos por si en el futuro se puede adaptar a otro tipo de aprendizaje que se encuentre dentro de la inteligencia artificial, como entorno usaremos junypiter es una herramienta muy utilin es mayor uso podemos ejecutar nuestros codigos python en ella, un ejemplo sencillo de esta es el uso de

mostrar una tabal capturada con la librería pandas de nuestro algortiomomo de aprendizaje mientras estaba trabajando.



TABLA DE RESULTADOS

La siguiente tabla muestra el mejor resultado hasta las prueba numero 363

ID	Generacion	Descripcion	Resultado distancia en X
1	2	Poco eficiente	1.00
2	40	Poco eficiente	2.50
3	200	Medio eficiente	20.00
4	250	Medio eficiente	24.00
5	363	Medio eficiente	50.00

CONCLUSION

En conclusión podemos decir que el desarrollo de este proyecto quedo de una forma inconclusa ya que no llegamos a una respuesta satisfactoria para resolver nuestro nivel del juego Vector man 2, aun que el algoritmo se puede mejorar para llegar a esa respuesta satisfactoria , también podemos recalcar la interesante que la inteligencia artificial y el potencial que tiene a futuro ,solo cabe con finalizar que seguiremos estudiando y probando mas de estos algoritmo para llegar a comprenderlos más.