

Examen Segundo Interciclo
Inteligencia Artificial
Nombre:Roberto Pacho

Objetivo:

1. Desarrollar un juego (tema libre) empleando una de las 2 siguientes alternativas:

- 1.easyAI
- 2.Universe + GYM

El juego deberá implementar algún algoritmo de IA y de igual forma, generar un informe de movimientos, puntajes y quién gana la partida. Se debe tener un juego en donde se tenga un jugador humano y otro utilizando Inteligencia Artificial, finalmente no se puede repetir el juego por más de tres personas por lo que se debe publicar en el foro el juego seleccionado.

2. Dentro del juego el usuario puede registrar y ingresar los gustos de alguna área basadas en el lugar geográfico por ejemplo: comida, películas, lugares turísticos etc.

3. En base a la información proporcionada se deberá generar un sistema que permita mostrar lugares de interés, para ello tomar los datos de las tareas y pruebas dentro de una base de datos orientadas a grafos.

4. Realizar el sistema con una interfaz gráfica y almacenar los puntajes y datos de los usuarios o jugadores.

Código y documentos de entrega: Se deberá entregar un informe con el proceso dentro del mismo tener capturas del uso del juego y generar un documento en PDF de validación y pruebas. Finalmente subir todo al repositorio incluido los códigos fuentes

Desarrollo:

- Creamos la interfaz Login para guardar los datos de los jugadores

LOGIN IA

LOGIN

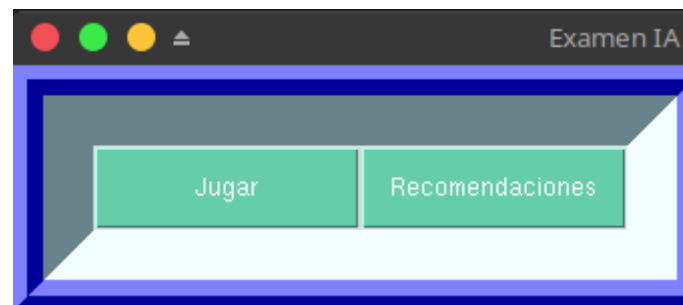
Usuario

Edad:

Pais:

correo

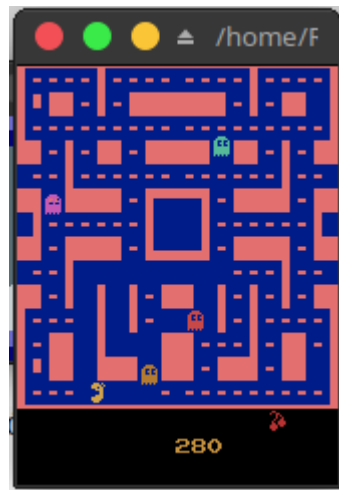
- Creamos una interfaz de opciones para el jugador como: Jugar/Recomendar



- Usamos la librería Gym para desarrollar el juego
Instalamos Gym

```
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
(base) root@Roberto:/home/RobertoPacho# pip install gym
```

- Importamos la librería en el cuaderno de python
- Al presionar el botón jugar tendremos lo siguiente



Al presionar el botón Recomendaciones tendremos la siguiente interfaz.

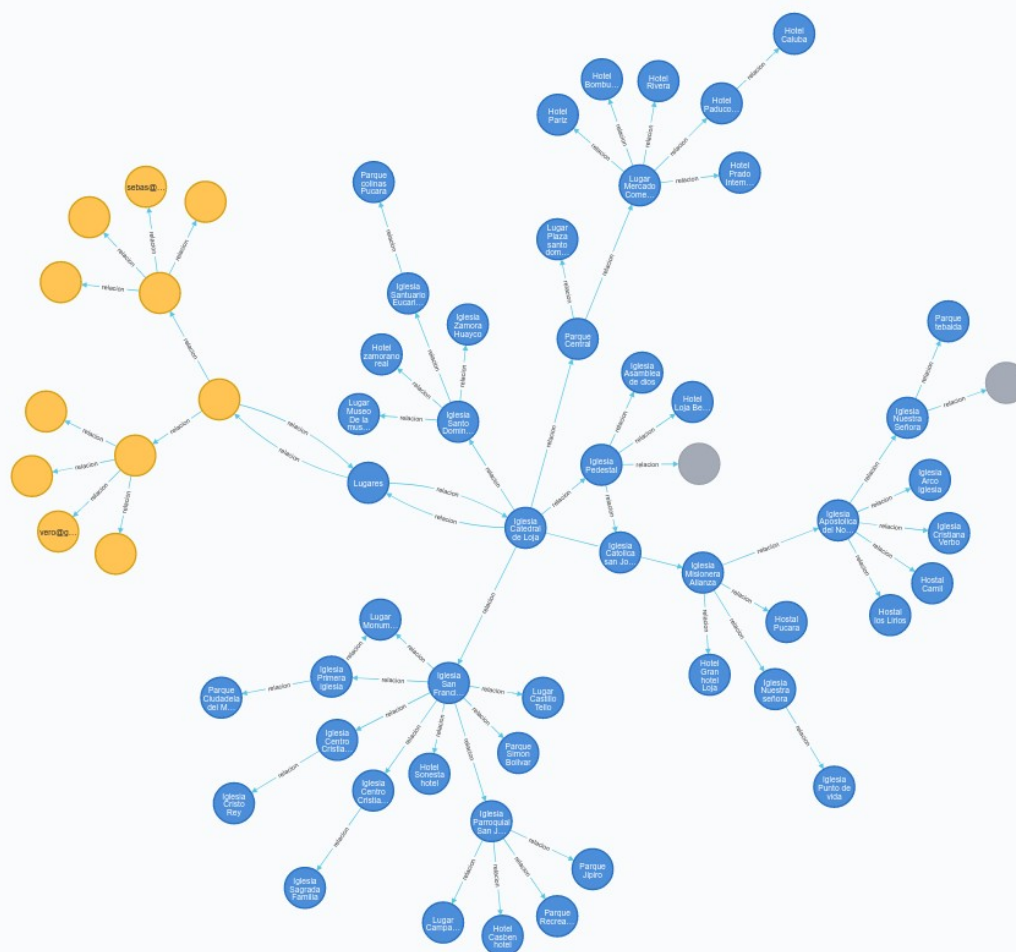


Almacenamiento de Datos en neo4j

```
neo4j$ MATCH (n) RETURN n LIMIT 75
```

Graph *(60) Recomendacion(49) Jugador(11)
*(64) relacion(64)

Table
Text
Code



```

from tkinter import *
import tkinter.font as tkFont
import gym
from neo4j import GraphDatabase

def opciones(self):
    usr=self
    neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'examen_ia')
    # Configuración de la raíz

    root1 = Tk()
    root1.title("Examen IA")
    root1.resizable(1,1)

    frame = Frame(root1, width=480, height=320)
    frame.pack(fill='both', expand=1)
    frame.config(cursor="pirate")
    frame.config(bg="lightblue")
    frame.config(bd=25)
    frame.config(relief="sunken")

    def juego():
        def OpenAI_Gym(game,num_episodios,num_inter):
            env= gym.make(game)
            puntos=""
            for i_episodio in range(num_episodios):
                env.reset()
                recompensa=0
                for t in range(num_inter):
                    env.render()
                    action=env.action_space.sample()
                    observation,reward,terminal,info=env.step(action)
                    recompensa += reward
                    if terminal:
                        recompensa += reward
                        print("recompensa obtenida: ",recompensa)
                        break
                puntos=str(recompensa)
            env.close()
            with neo4j._driver.session() as session:
                session.write_transaction(neo4j.crear_puntaje,puntos)
                session.write_transaction(neo4j.crear_relacion_usuario_puntaje,usr,puntos)
            OpenAI_Gym(game= 'MsPacman-v0', num_episodios=5,num_inter= 1000)

        def recomendacion():

            vjuego = Frame(frame, width=480, height=320)
            vjuego.grid(row=0, column=2, padx=10, pady=3, ipadx=20, ipady=20)
            fontStyle = tkFont.Font(family="Lucida Grande", size=12)

            tipo=Label(vjuego,text='! TE GUSTARIA VISITAR LUGARES !', font=fontStyle)
            tipo.place(x=50,y=50)
            etomo=Label(vjuego,text='TE RECOMENDAMOS LOS SIGUIENTE: ', font=fontStyle)
            etomo.place(x=50,y=100)

            def par(self):
                lug=str(self)
                neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'examen_ia')
                with neo4j._driver.session() as session:
                    session.read_transaction(neo4j.buscar,vjuego,lug)

            btn1 = Button(vjuego, text="PARQUES", bg="MediumAquamarine", fg="white", height = 2, width =
15,command=par("Parque"))
            btn2 = Button(vjuego, text="Iglesias", bg="MediumAquamarine", fg="white", height = 2, width =
15,command=par("Igle"))
            btn3 = Button(vjuego, text="Hoteles", bg="MediumAquamarine", fg="white", height = 2, width =
15,command=par("Ho"))
            btn4 = Button(vjuego, text="LUGARES DE INTERES", bg="MediumAquamarine", fg="white", height = 2,
width = 15,command=par("Luga"))

            btn1.place(x=50, y=150)
            btn2.place(x=50, y=200)
            btn3.place(x=50, y=250)
            btn4.place(x=50, y=300)
            #ImageTk.PhotoImage(new_img)

            btn5 = Button(frame, text="Jugar", bg="MediumAquamarine", fg="white", height = 2, width =
15,command=juego)
            btn6 = Button(frame, text="Recomendaciones", bg="MediumAquamarine", fg="white", height = 2, width =
15,command=recomendacion)

            btn5.grid(column=1, row=0)
            btn6.grid(column=2, row=0)

            root1.config(cursor="arrow")
            root1.config(bg="blue")
            root1.config(bd=15)
            root1.config(relief="ridge")

# Finalmente bucle de la aplicación
root1.mainloop()

```

```

root = Tk()
root.title("LOGIN IA")
root.geometry('400x400')
root.resizable(width=0, height=0)
fontStyle = tkFont.Font(family="Lucida Grande", size=12)
fontStyle1 = tkFont.Font(family="Lucida Grande", size=30)
fontStyle2 = tkFont.Font(family="Lucida Grande", size=15)

eusuario=Label(root,text='LOGIN', font=fontStyle1)
eusuario.place(x=150,y=20)

eusuario=Label(root,text='Usuario', font=fontStyle)
eusuario.place(x=50,y=100)
usuario = Entry(root, fg='black', bg='white', justify='center')
usuario.place(x=110, y=100, width=200)

edad=Label(root,text='Edad: ', font=fontStyle)
edad.place(x=50,y=150)
edad = Entry(root, fg='black', bg='white', justify='center')
edad.place(x=110, y=150, width=100)

epais=Label(root,text='Pais: ', font=fontStyle)
epais.place(x=50,y=200)
pais = Entry(root, fg='black', bg='white', justify='center')
pais.place(x=110, y=200, width=200)

ecorreo=Label(root,text='correo', font=fontStyle)
ecorreo.place(x=50,y=250)
correo = Entry(root, fg='black', bg='white', justify='center')
correo.place(x=110, y=250, width=200)

class Neo4jService(object):

    def __init__(self, uri, user, password):
        self._driver = GraphDatabase.driver(uri, auth=(user, password))

    def close(self):
        self._driver.close()

    def crear_usuario(self, tx, nombre):
        tx.run("CREATE (Usuario:Jugador {usuario: $nombre})", nombre=nombre)

    def crear_usuarioj(self, tx, usr):
        tx.run("CREATE (Usr:Jugador {nombre: $usr})", usr=usr)

    def crear_edad(self, tx, edad):
        tx.run("CREATE (Edad:Jugador {edad: $edad})", edad=edad)

    def crear_pais(self, tx, pais):
        tx.run("CREATE (Pais:Jugador {pais: $pais})", pais=pais)

    def crear_correo(self, tx, correo):
        tx.run("CREATE (Correo:Jugador {correo: $correo})", correo=correo)

    def crear_puntaje(self, tx, puntaje):
        tx.run("CREATE (Puntaje:Jugador {puntaje: $puntaje})", puntaje=puntaje)

    def crear_relacion_usuario_usr(self, tx, usuario, usr):
        tx.run("MATCH (a:Jugador {usuario: $usuario}) "
              "MATCH (b:Jugador {nombre: $usr}) "
              "MERGE (a)-[r:relacion]->(b)",
              usuario=usuario, usr=usr)

    def crear_relacion_usuario_edad(self, tx, usuario, edad):
        tx.run("MATCH (a:Jugador {nombre: $usuario}) "
              "MATCH (b:Jugador {edad: $edad}) "
              "MERGE (a)-[r:relacion]->(b)",
              usuario=usuario, edad=edad)

    def crear_relacion_usuario_pais(self, tx, usuario, pais):
        tx.run("MATCH (a:Jugador {nombre: $usuario}) "
              "MATCH (b:Jugador {pais: $pais}) "
              "MERGE (a)-[r:relacion]->(b)",
              usuario=usuario, pais=pais)

    def crear_relacion_usuario_correo(self, tx, usuario, correo):
        tx.run("MATCH (a:Jugador {nombre: $usuario}) "
              "MATCH (b:Jugador {correo: $correo}) "
              "MERGE (a)-[r:relacion]->(b)",
              usuario=usuario, correo=correo)

    def crear_relacion_usuario_puntaje(self, tx, usuario, puntaje):
        tx.run("MATCH (a:Jugador {nombre: $usuario}) "
              "MATCH (b:Jugador {puntaje: $puntaje}) "
              "MERGE (a)-[r:relacion]->(b)",
              usuario=usuario, puntaje=puntaje)

    def buscar(self,tx,vjuego,lug):
        result = tx.run("MATCH(p:Recomendacion) WHERE p.name STARTS WITH $p RETURN p.name", p=lug)
        etomo=Label(vjuego,text="EN LOJA:")
        etomo.place(x=200,y=130)

        frame1 = Frame(vjuego, width=400, height=40)
        frame1.place(x=200,y=150)

        scrollbar = Scrollbar(frame1)
        scrollbar.pack( side = RIGHT, fill = Y )

        mylist = Listbox(frame1, yscrollcommand = scrollbar.set )
        for line in result:
            mylist.insert(END, str(line))

        mylist.pack( side = LEFT, fill = BOTH )
        scrollbar.config( command = mylist.yview )

neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'examen_ia')
with neo4j._driver.session() as session:
    session.write_transaction(neo4j.crear_usuario , "Usuario")

def cerrar():
    neo4j = Neo4jService('bolt://localhost:7687', 'neo4j', 'examen_ia')
    usr=str(usuario.get())
    ed=str(edad.get())
    pa=str(pais.get())
    mail=str(correo.get())

    with neo4j._driver.session() as session:
        session.write_transaction(neo4j.crear_usuarioj , usr)
        session.write_transaction(neo4j.crear_edad , ed)
        session.write_transaction(neo4j.crear_pais , pa)
        session.write_transaction(neo4j.crear_correo , mail)

        session.write_transaction(neo4j.crear_relacion_usuario_usr,"Usuario",usr)
        session.write_transaction(neo4j.crear_relacion_usuario_edad,usr,ed)
        session.write_transaction(neo4j.crear_relacion_usuario_pais,usr,pa)
        session.write_transaction(neo4j.crear_relacion_usuario_correo,usr,mail)

    #root.destroy()
    opciones(usr)

btn1 = Button(root, text="Ingresar", bg="MediumAquamarine", fg="white", height = 2, width =
15,font=fontStyle2,command=cerrar)
btn1.place(x=120, y=300)

root.mainloop()

```