

Antonio de Jesús Covarrubias Sánchez

Registro: 22110347

T-7E

Ingeniería Mecatrónica

Sistemas Expertos

Profesor: Cabrera Arellano Mauricio Alejandro



ceti CENTRO DE ENSEÑANZA
TÉCNICA INDUSTRIAL



Enlace perfil de GitHub:

<https://github.com/tonycovasan22110347>

Enlace repositorio de GitHub:

<https://github.com/tonycovasan22110347/Akinator-Star-Wars->



Enlace perfil de itch.io:

<https://antonio-covarrubias.itch.io/>

Enlace juego en itch.io:

Manual de Usuario: Akinator Star Wars

1. Introducción

- Este programa simula el clásico juego de 'Adivina Quién' utilizando personajes del universo de Star Wars.
- El sistema se basa en un enfoque de sistemas expertos, utilizando una base de conocimiento en forma de reglas, inferencia por casos y encadenamiento hacia adelante.
- El usuario responde preguntas simples con 'Sí' o 'No' y el sistema reduce el conjunto de personajes posibles hasta llegar a una predicción.

2. Funcionamiento General

1. Al iniciar, el usuario ve una interfaz con un botón para comenzar el juego.
2. Se le presentarán preguntas basadas en características como '¿Es un Jedi?', '¿Usa blaster?', etc.
3. Las respuestas son usadas para filtrar posibles personajes utilizando un razonamiento tipo encadenamiento hacia adelante.
4. Al final, si queda un solo personaje, se muestra su nombre y su imagen (si está disponible).

3. Teoría de Reglas

El sistema usa reglas simples en forma de pares atributo-valor (por ejemplo: `es_jedi=True`).

Cada pregunta corresponde a un atributo booleano que se utiliza para filtrar la base de conocimiento (el archivo JSON con los personajes).

4. Razonamiento Basado en Casos

- Cada personaje puede verse como un caso compuesto por múltiples atributos booleanos.
- Durante el juego, las respuestas del usuario forman un caso parcial que se compara con los casos de la base de conocimiento para encontrar coincidencias.
- Este enfoque permite manejar múltiples resultados parciales o ambigüedad si varias coincidencias son posibles.

5. Encadenamiento Hacia Adelante

- El sistema aplica encadenamiento hacia adelante: a medida que el usuario responde preguntas, el sistema descarta personajes que no cumplen con las condiciones acumuladas.
- Este proceso se repite hasta que solo queda un personaje (o varios candidatos posibles).

6. Requisitos y Ejecución

- Requiere Python 3, las librerías tkinter y Pillow.
- Debe haber un archivo 'Personajes.json' con los datos de los personajes.
- Las imágenes deben estar en una carpeta 'imagenes/' y nombradas igual que los personajes.