Proyecto AI 1

Piedra, papel, tijeras, lagarto, Spock

Programación de Inteligencia Artificial

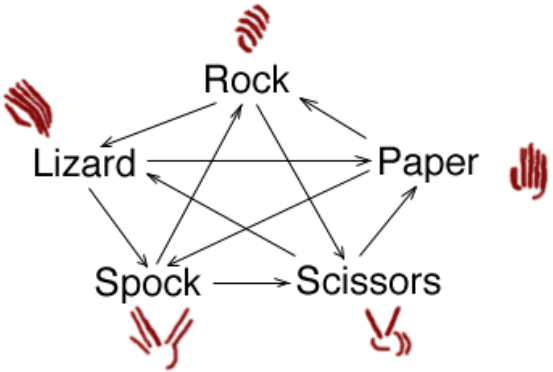
Programación de
Inteligencia_Artificial

Curso Especialización
Inteligencia_Artificial y
Big_Data

Especificación Mini-Proyecto 1

Piedra, papel, tijeras, lagarto, Spock

- Partiendo del código disponible en 05_RPS_More_AI.py, añade la funcionalidad necesaria para ofrecer la variante *lagarto, Spock* del juego *piedra, papel o tijeras*.
- Modela las diferentes situaciones de juego, junto con su resultado, en un archivo victories.xml.

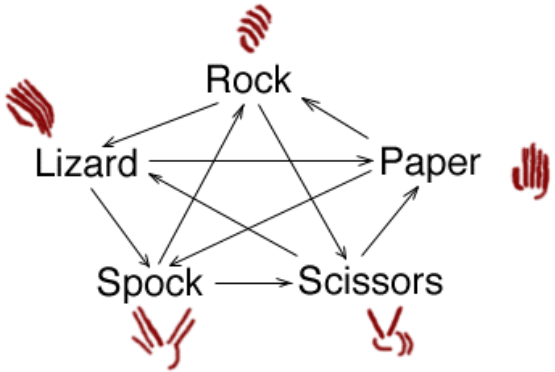


Scissors cuts Paper covers Rock crushes Lizard poisons Spock smashes Scissors decapitates Lizard eats Paper disproves Spock vaporizes Rock crushes Scissors.

Especificación Mini-Proyecto 1

Piedra, papel, tijeras, lagarto, Spock

- Se recomienda generalizar la función assess_game() para limitar el número de sentencias condicionales.
- En victories.xml, se recomienda incluir la información que se mostrará por pantalla con respecto a las diferentes situaciones de juego.

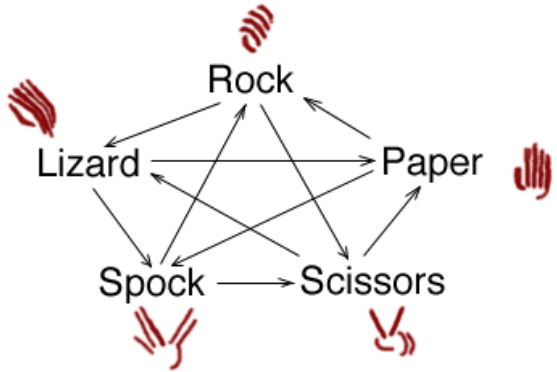


Scissors cuts Paper covers Rock crushes Lizard poisons Spock smashes Scissors decapitates Lizard eats Paper disproves Spock vaporizes Rock crushes Scissors.

Especificación Mini-Proyecto 1

Piedra, papel, tijeras, lagarto, Spock

- Valora sustituir el diccionario Victories por alguna referencia que le permita acceder al contenido de victories.xml.
- Se recomienda usar el módulo xml.etree.ElementTree para procesar el archivo victories.xml.



Scissors cuts Paper covers Rock crushes Lizard poisons Spock smashes Scissors decapitates Lizard eats Paper disproves Spock vaporizes Rock crushes Scissors.

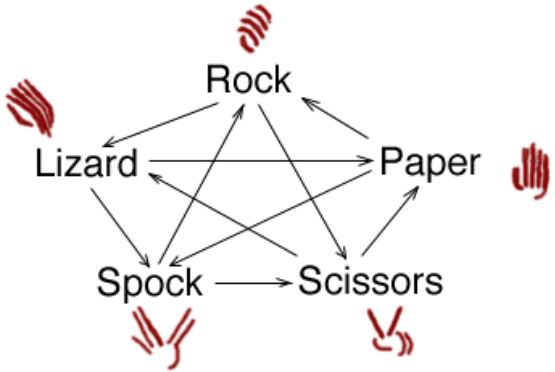
Especificación Mini-Proyecto 1

Piedra, papel, tijeras, lagarto, Spock

- Definir las distintas situaciones de victoria por parte del usuario en un fichero victories.xml.

XML mode | Format | Save

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<victories>
    <victory choice="Scissors" against="Paper">
        Scissors cuts Paper.
    </victory>
</victories>
```



Scissors cuts Paper covers Rock crushes Lizard poisons Spock smashes Scissors decapitates Lizard eats Paper disproves Spock vaporizes Rock crushes Scissors.

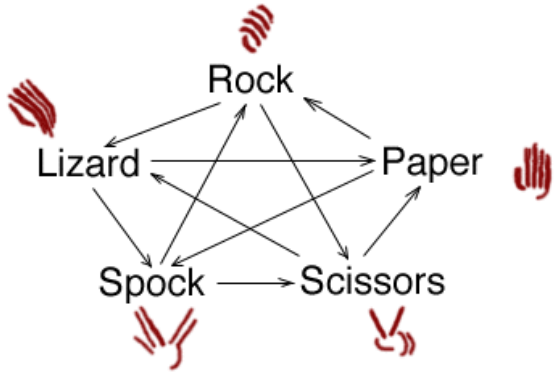
Especificación Mini-Proyecto 1

Piedra, papel, tijeras, lagarto, Spock

- Rellenar el diccionario de victorias utilizando la librería estándar de Python ElementTree. Clic en la imagen para ir a implementación Google Colab.

```
from xml.etree import ElementTree

Victories = ElementTree.parse('victories.xml').getroot()

def main():
  user_action = 'Scissors'
  computer_action = 'Paper'
  test_xpath = f"./victory[@choice='{user_action}'][@against='{computer_action}']"

  test_match = Victories.find(test_xpath)
  choice = test_match.attrib['choice']
  against = test_match.attrib['against']

  print(f"choice: {choice}")
  print(f"against: {against}")
  print(test_match.text.strip())


if __name__ == "__main__":
  main()
```

```
choice: Scissors
against: Paper
Scissors cuts Paper.
```

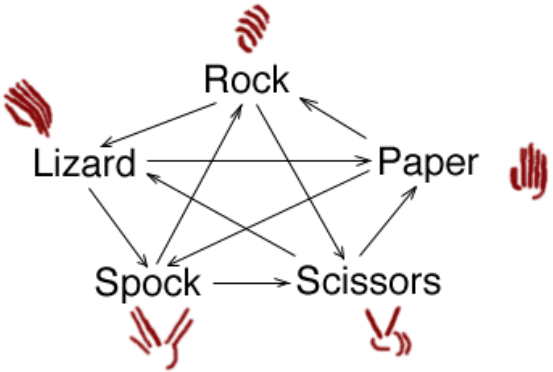


Scissors cuts Paper covers Rock crushes Lizard poisons Spock smashes Scissors decapitates Lizard eats Paper disproves Spock vaporizes Rock crushes Scissors.

Especificación Mini-Proyecto 1

Piedra, papel, tijeras, lagarto, Spock

- Generalizar el método asses_game() utilizando XPATH para buscar en el diccionario victories (cargado desde xml) preguntando por los atributos @choice y @against utilizando el método find, de tal forma que si el resultado del método find es positivo (lo encuentra) gana el usuario y si no gana el ordenador.
- ¿Cómo modelo el empate?
- La entrega será el fichero py y xml resultante.

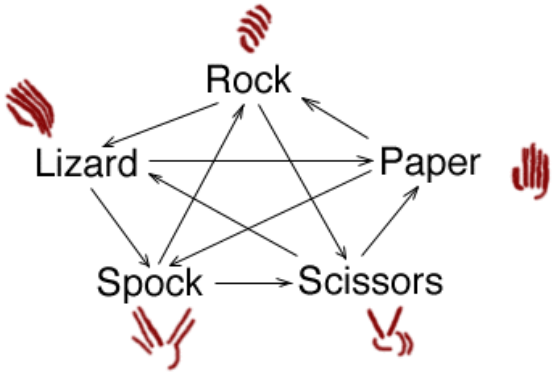


Scissors cuts Paper covers Rock crushes Lizard poisons Spock smashes Scissors decapitates Lizard eats Paper disproves Spock vaporizes Rock crushes Scissors.

Especificación Mini-Proyecto 1

Piedra, papel, tijeras, lagarto, Spock

- Criterios de calificación:
  - Es condición para seguir evaluando el resto de criterios y poder superar la tarea que el código compile y ejecute. 2 puntos.
    - La lógica del juego es correcta. 1 punto.
    - Se han modelado los casos de juego correctamente usando un fichero XML. 1 punto.
    - Se ha generalizado la función de evaluación. 2 puntos.
    - Uso de ElementTree y XPATH para explorar el XML. 3 puntos.
    - Claridad del código y uso de comentarios. 1 punto.



Scissors cuts Paper covers Rock crushes Lizard poisons Spock smashes Scissors decapitates Lizard eats Paper disproves Spock vaporizes Rock crushes Scissors.