# Aprendizaje por refuerzo para Sumobot - Q Learning

Caleb Trepowski
IA - FIUNA

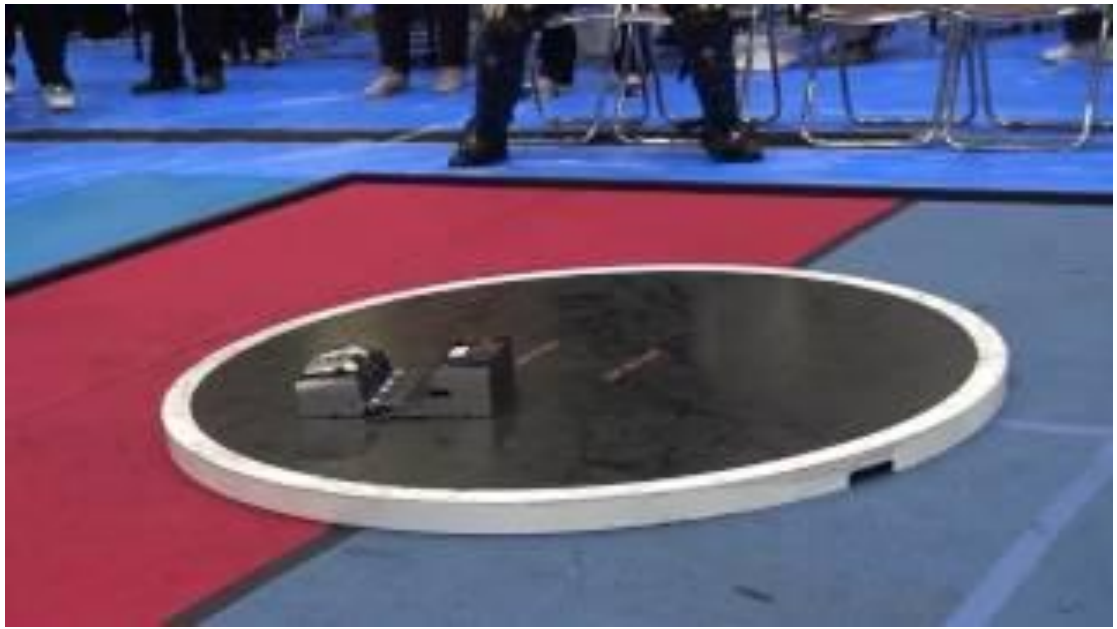


# Batallas de Sumobots

“Como en las artes marciales japonesas tradicionales, los robots intentan empujar al oponente fuera del ring”

*Regulaciones de competencia “Robochallenge”*

# Batallas de Sumobots



https://www.youtube.com/watch?v=QCqxOzKNFks

# Composición de un Sumobot

En su versión más básica, cuenta con:

- 2 motores, acoplados cada uno a una rueda
- Sensores de detección de objetos
- Sensores de detección de color (sensor de línea)

# Enfoque tradicional

Toma de acciones mediante árboles de decisiones

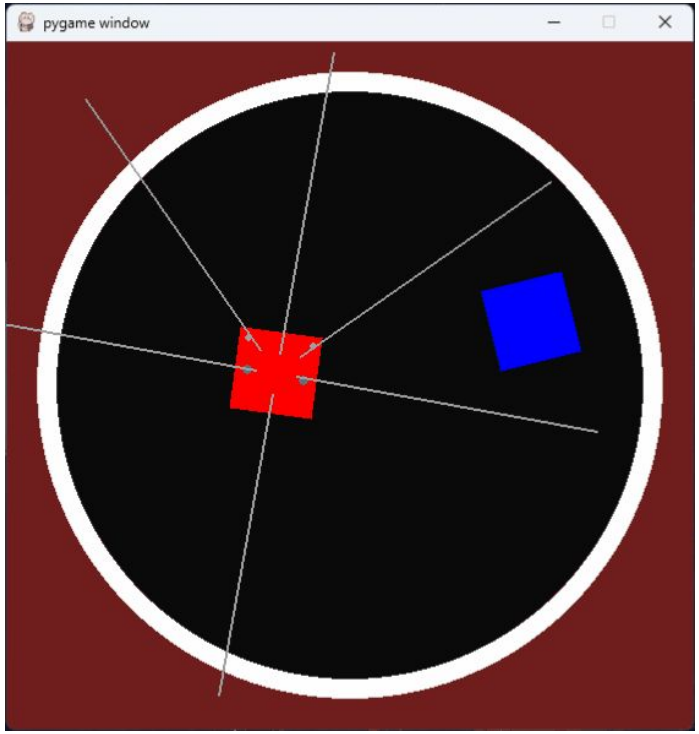
# Enfoque propuesto

Toma de decisiones según lo más conveniente dependiendo de la situación, entrenado según un sistema de recompensas

# Simulación del Entorno

- Mediante librería *gymnasium* (continuación de *OpenAI Gym*)
- Física simulada con *pymunk*
- Renderización para visualizar con *pygame*

# Simulación del Entorno: Demo



# Descripción del Entorno

- **Espacio de Observación: MultiBinary(8)**

  Los primeros 6 componentes son los sensores de proximidad, los últimos 2 los sensores de línea

# Descripción del Entorno

- **Espacio de Acción: Discrete(4)**

  Cuatro acciones posibles: ir hacia adelante, ir hacia atrás, girar hacia la izquierda, girar hacia la derecha

# Algoritmo Q-Learning

Q-Learning es una política de entrenamiento por refuerzo que busca la mejor siguiente acción, dado un estado actual.

Se utiliza una tabla (llamada Tabla Q), la cual se ajusta utilizando la ecuación de Bellman.

# Tabla Q

Cada fila representa un posible estado, cada columna una acción.

La mejor acción para un estado se decide por la acción que tenga mayor valor luego de haber entrenado.

# Tabla Q

Al tener 8 sensores booleanos, la cantidad posible de estados es $2^8 = 256$.

| Estado/Acción | 0 (Adelante) | 1 (Atrás) | 2 (Izquierda) | 3 (Derecha) |
|---|---|---|---|---|
| [0,0,0,0,0,0,0,0] : 0 | | | | |
| [0,0,0,0,0,0,0,1] : 1 | | | | |
| [0,0,0,0,0,0,1,0] : 2 | | | | |
| … | | | | |
| [1,1,1,1,1,1,1,1] : 255 | | | | |

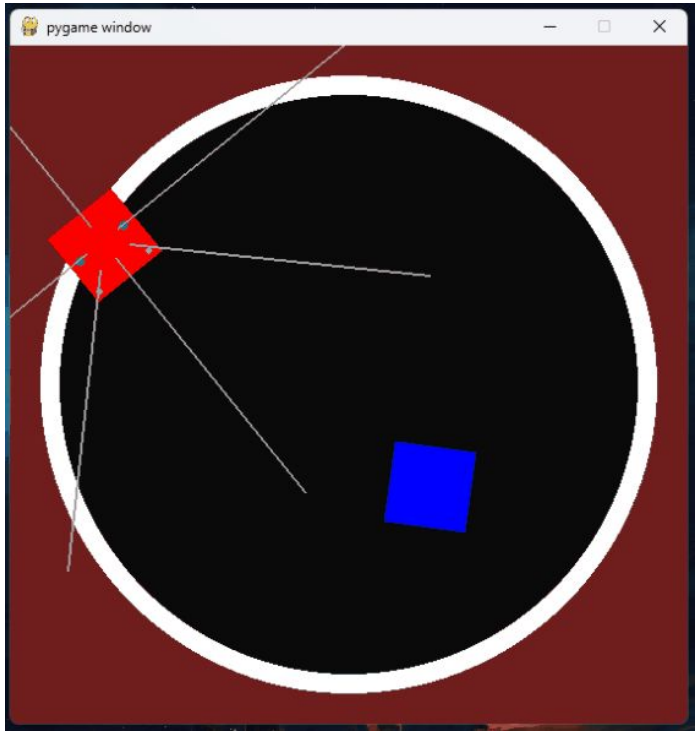
# Ecuación de Bellman

$$q^{new}(s,a) = (1-\alpha)\underbrace{q(s,a)}_{\text{old value}} + \alpha\overbrace{\left(R_{t+1} + \gamma \max_{a'} q(s',a')\right)}^{\text{learned value}}$$

# Exploración/Explotación

Al comienzo el agente no conoce el entorno, por lo que es mejor que tome acciones al azar (exploración). Conforme avance el entrenamiento, conviene que “aplique” lo aprendido (explotación).

# Resultado Final



# Repositorio del Proyecto

https://github.com/calebtrepowski/sumorobot-reinforcement-learning

# Fuentes

https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-q-learning

https://towardsdatascience.com/q-learning-algorithm-from-explanation-to-implementation-cdbeda2ea187