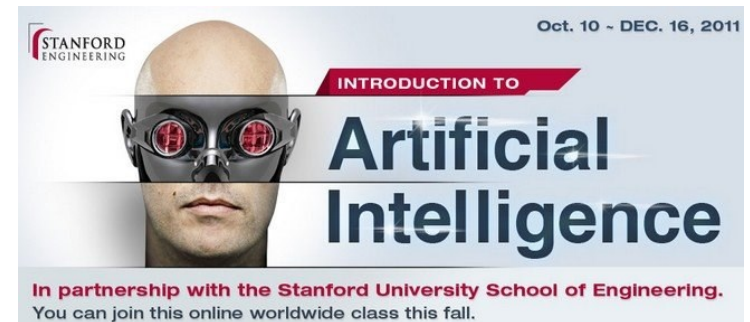


OpenNERO

Neuro-Evolving Robotic Operatives



Más de 160.000 participantes

Autor: Dersu García Sanz

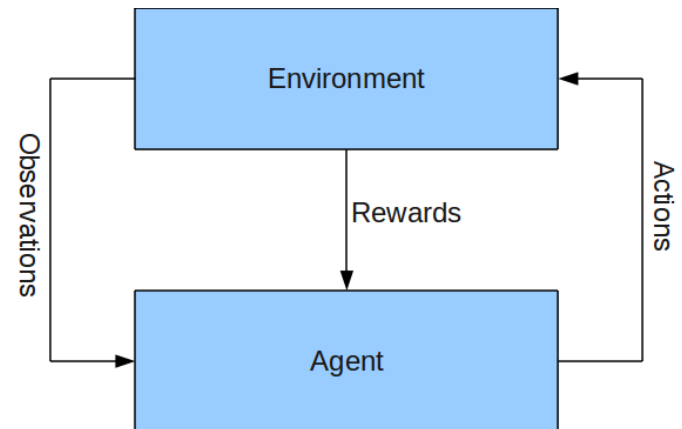
Introducción OpenNERO

- ▶ Basado en el juego de ML Neuro-Evolving Robotic Operatives
- ▶ Universidad de Austin, Texas en 2003
- ▶ Licencia BSD



Relación con los agentes

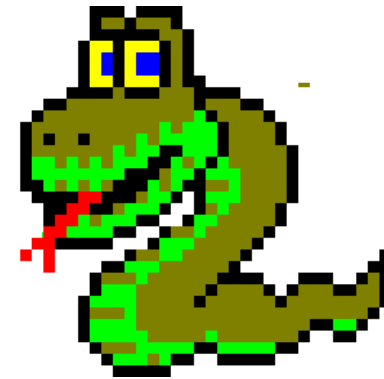
- ▶ Entorno virtual
- ▶ Entrenar una población de 50 agentes en diferentes áreas
- ▶ Editar a mano los agentes
- ▶ Enfrentar dos equipos



Instalación en Windows

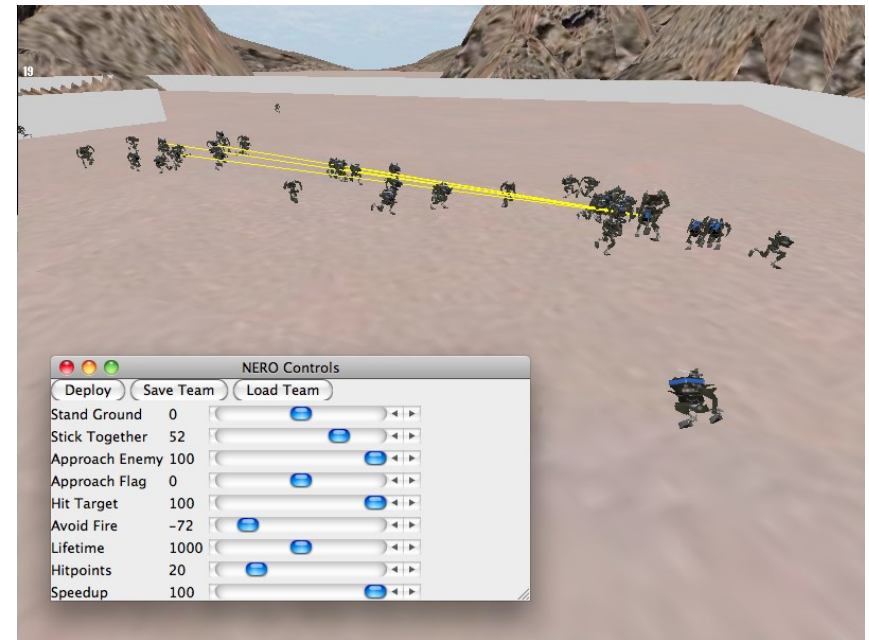


- ▶ Python 2.7 (win32)
- ▶ wxPython (GUI ToolKit)
- ▶ Añadir directorio con python.exe al PATH
- ▶ Instalar OpenNERO-20xx.xx.xx-win32.exe



Entrenar Agentes

- ▶ Añadir elementos al mapa
 - Torretas enemigas
 - Muros
 - Árboles
- ▶ Elegir tipo de agentes:
 - rtNEAT
 - Q-Learning
- ▶ Marcar objetivos



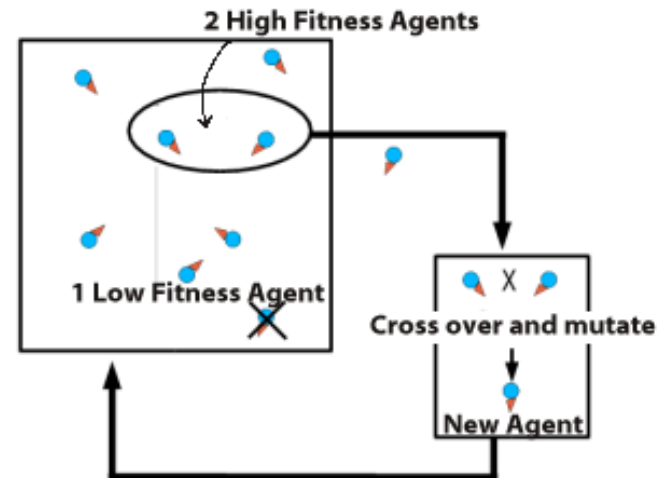
NERO Controls		
Deploy	Save Team	Load Team
Stand Ground	0	
Stick Together	52	
Approach Enemy	100	
Approach Flag	0	
Hit Target	100	
Avoid Fire	-72	
Lifetime	1000	
Hitpoints	20	
Speedup	100	

Entrenar Agentes: rtNEAT

► Real-time NeuroEvolution of Augmenting Topologies

- Algoritmo genético con redes neuronales
- Cada agente es una red neuronal
- Acumulan fitness según cumplen los objetivos asignados

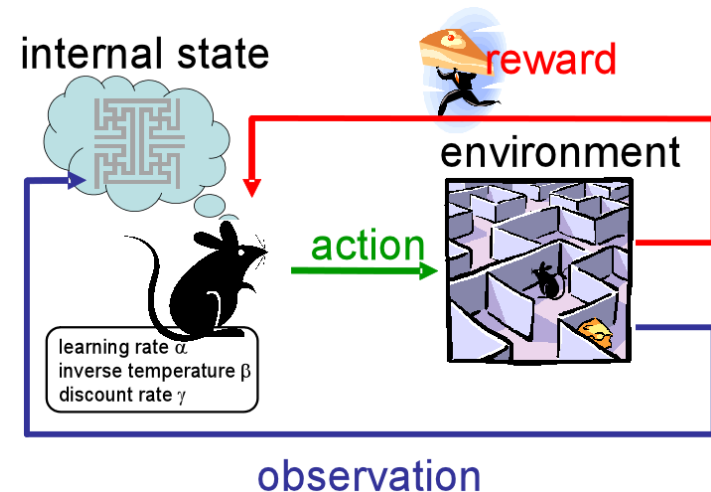
- En cada iteración:
 - Mutación y cruce
 - Los peores se descartan



Entrenar Agentes: Q-Learning

- ▶ Reinforcement Learning
- ▶ Se bonifica o penaliza al agente según cumpla los objetivos

- ▶ Por ejemplo:



- Si daña a un enemigo se bonifica
- Si recibe disparos se penaliza

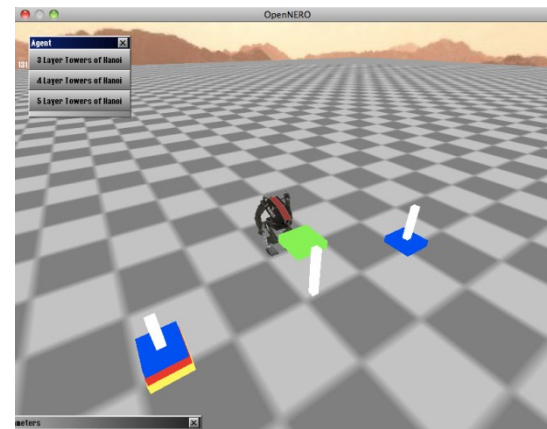
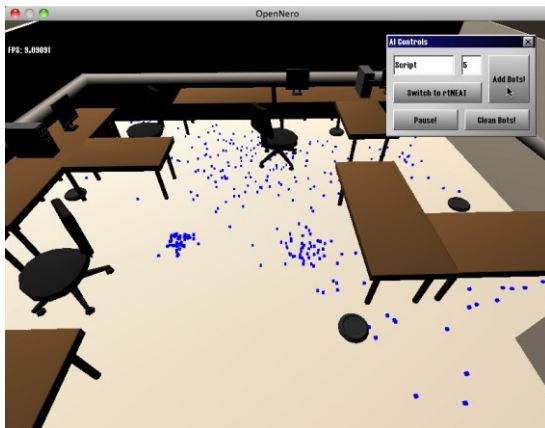
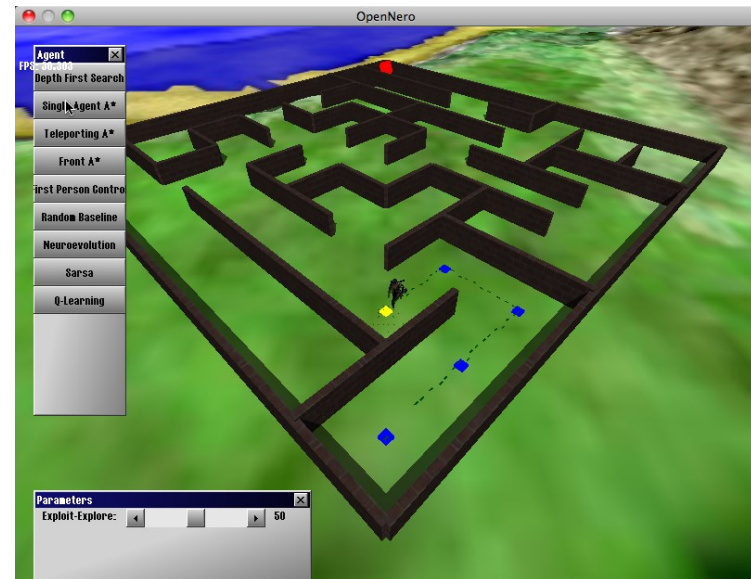
Enfrentar agentes

- ▶ Combates por equipos uno contra uno
 - Por tiempo
 - Por eliminación
- ▶ Torneo 2011 (156 participantes)



Otros juegos incluidos

- ▶ Maze (Búsqueda)
- ▶ Blocks
- ▶ Roomba



Enlaces de interés

- ▶ <https://www.udacity.com/course/cs271>
- ▶ <https://code.google.com/p/opennero/>
- ▶ <http://www.nerogame.org/>

Gracias