

Agentes de Software



Agentes

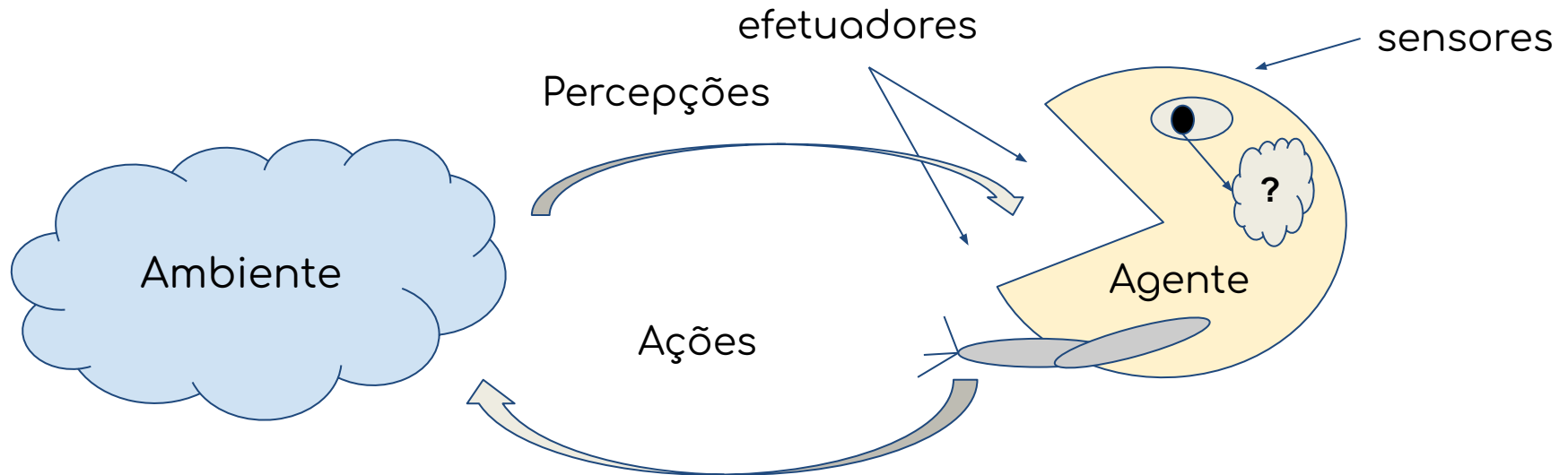


Um programa de IA pode ser visto como um **Agente Racional**

- **O que é um Agente Racional (inteligente)?**
- **Qual sua utilidade em IA?**
- **Ambientes e arquiteturas**
- **Aplicações**

O que é um agente

- **Agente** é qualquer entidade que:
 - **percebe** seu ambiente através de sensores (ex. câmeras, microfone, teclado, finger, ...)
 - **age** sobre ele através de efetadores (ex. vídeo, auto-falante, impressora, braços, ftp, ...)
- **Mapeamento: sequência perceptiva => ação**



Medida de Desempenho



- **Critério que define o grau de sucesso de um agente na realização de uma dada tarefa**
 - Esta medida deve ser imposta do exterior
 - Má escolha da MD pode acarretar comportamento indesejado
 - Compromissos entre objetivos múltiplos conflitantes
 - Resta o problema de saber quando avaliar o desempenho
 - Exs. aspirador de pó, provador de teoremas, filtragem de mails, policial de trânsito, avaliador de clima...

Agente Racional (McCarthy & Hayes 69, Newell 81)

- **Agente Racional:** fazer a melhor coisa *possível*
 - segue o **princípio da racionalidade:** dada uma **seqüência perceptiva**, o agente escolhe, segundo seus **conhecimentos**, as **ações** que satisfazem melhor seu **objetivo**.
- **Problema**
 - estado inicial + ações => estado final (objetivo)
- **Racionalidade ≠ Onisciência, limitações de:**
 - sensores
 - efetadores
 - raciocinador (conhecimento, tempo, etc.)
 - Agir para obter mais dados perceptivos é **racional**

Autonomia e utilidade



- **Autonomia**

- Capacidade de adaptação a situações novas, para as quais não foi fornecido todo o conhecimento necessário com antecedência

- **Para construir um sistema inteligente, utilizamos**

- linguagem
- inferência
- conhecimento

A Metáfora de agente decompõe



1) Problema em:

- percepções, ações, objetivos, e ambiente (e outros agentes)
- PAGE(**P**erceptions, **A**ctions, **G**oals, **E**nvironment)

2) Tipo de conhecimento em:

- Quais são as propriedades relevantes do mundo
- Como o mundo evolui
- Como identificar os estados desejáveis do mundo
- Como interpretar suas percepções
- Quais as consequências de suas ações no mundo
- Como medir o sucesso de suas ações
- Como avaliar seus próprios conhecimentos

Agente jogador de Futebol

Agente



raciocínio

Conhecimento:

- regras de futebol
- conhecimento sobre os jogadores,...

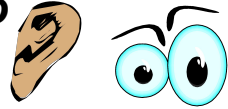
Objetivo:

- fazer gols e impedir que os adversários façam

Ações:

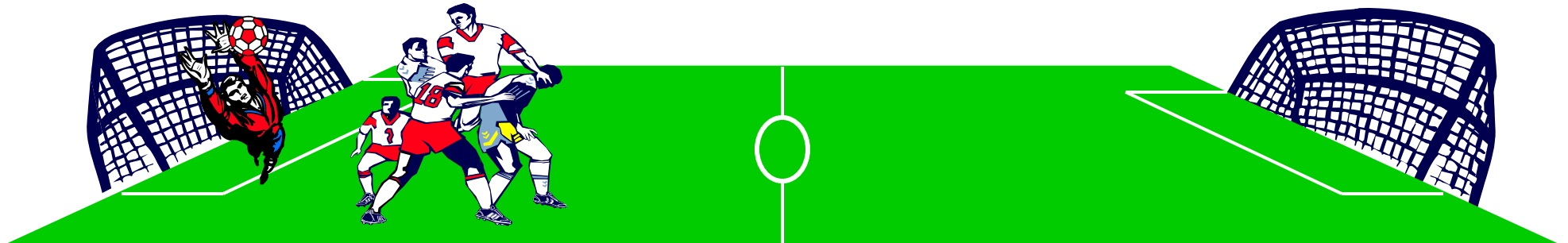
- chutar
- desarmar
- correr, ...

percepção



execução

Ambiente



Exemplos de agentes

Agente	acessível	determinista	episódico	estático	discreto
xadrez sem relógio	Sim	Sim	Não	Sim	Sim
xadrez com relógio	Sim	Sim	Não	Semi	sim
gamão	sim	não	não	sim	sim
motorista de taxi	Não	Não	Não	Não	Não
médico	Não	Não	Não	Não	Não
tutor	Não	Não	Não	Não	Sim
Analisador de imagem	Sim	Sim	Sim	Semi	Não
Busca na web	Não	Não	Sim	Não	Sim
Filtrador de mail	Sim	Não	Sim	Não	Sim
Músico	Sim	Não	Não	Não	Não

Ambiente



- **Classes de ambientes**

- Físico: robôs
- Software: softbots
- Realidade virtual (simulação do ambiente físico): softbots e avatares

- **Propriedades de um ambiente**

- Totalmente observável x parcialmente observável
- estático x dinâmico
- determinista x estocástico
- discreto x contínuo
- episódico x sequencial
- Único agente x multiagente
- tamanho: número de percepções, ações, objetivos,...

Ambientes: propriedades

- **Totalmente observável:** quando os *sensores* do agente conseguem perceber o estado completo do ambiente.
- **Determinista:** o próximo estado do ambiente pode ser completamente determinado pelo estado atual e as ações selecionadas pelo agente.
- **Episódico:** a experiência do agente é dividida em *episódios*. Cada episódio consiste em o agente perceber e então agir. Cada episódio não depende das ações que ocorreram em episódios prévios.

Ambientes: propriedades

- **Estático:** o ambiente não muda enquanto o agente está escolhendo a ação a realizar.
 - Semi-estático: o ambiente não muda enquanto o agente delibera, mas o "score" do agente muda.
- **Discreto:** quando existe um número distinto e claramente definido de percepções e ações em cada turno.
- **Contínuo:** percepções e ações mudam em um espectro contínuo de valores.
- **Multiagente:** ambiente composto por vários agentes que atuam de forma cooperativa e/ou competitiva.

Exemplos de ambientes

Agente	acessível	determinista	episódico	estático	discreto
xadrez sem relógio	Sim	Sim	Não	Sim	Sim
xadrez com relógio	Sim	Sim	Não	Semi	sim
gamão	sim	não	não	sim	sim
motorista de taxi	Não	Não	Não	Não	Não
médico	Não	Não	Não	Não	Não
tutor	Não	Não	Não	Não	Sim
Analizador de imagem	Sim	Sim	Sim	Semi	Não
Busca na web	Não	Não	Sim	Não	Sim
Filtrador de mail	Sim	Não	Sim	Não	Sim
Músico	Sim	Não	Não	Não	Não

+ Tamanho = número de percepções, ações e objetivos possíveis

Algoritmo básico

- **função** agenteSimples (percept) **retorna** ação
memória := atualizaMemória (memória, percept)
ação := escolheMelhorAção(memória)
memória := atualizaMemória (memória, ação)
retorna ação

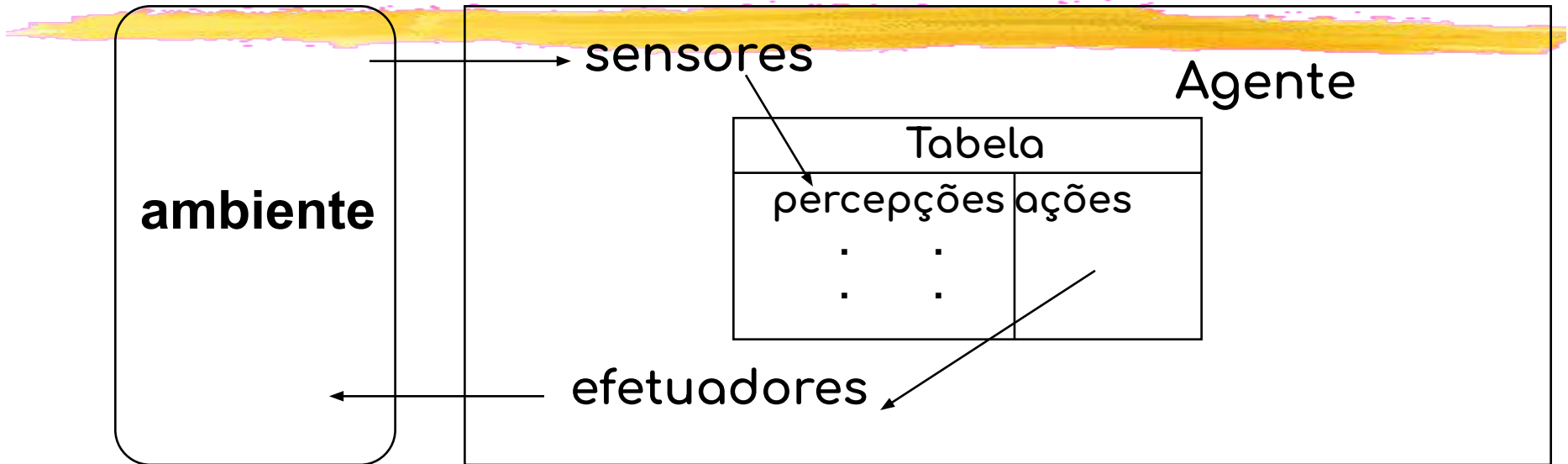
- **Arquiteturas**

- Agente tabela
- Agente reativo
- Agente reativo com estado interno (autômato)
- Agente cognitivo (baseado em objetivos)
- Agente otimizador
- Agente adaptativo



autonomia
complexidad
e

Agente tabela



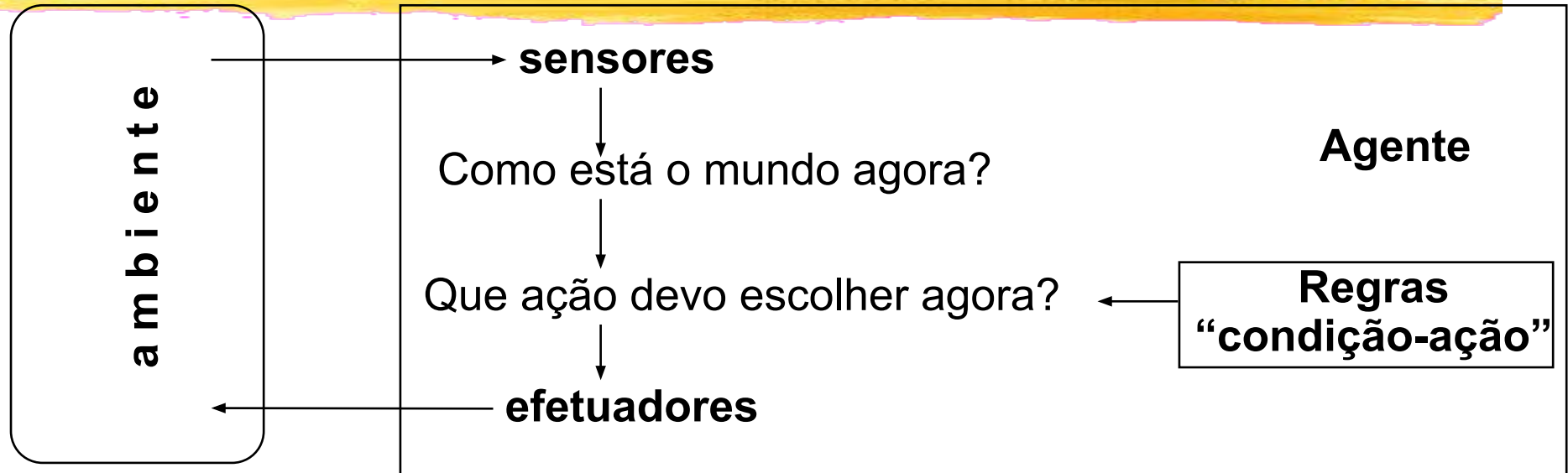
- **Limitações**

- Mesmo Problemas simples \rightarrow tabelas muito grandes (ex. xadrez 30^{100})
- Nem sempre é possível, por ignorância ou questão de tempo, construir a tabela
- Não há autonomia nem flexibilidade

- **Ambientes**

- acessível, determinista, episódico, estático, discreto e **minúsculo!**

Agente reativo



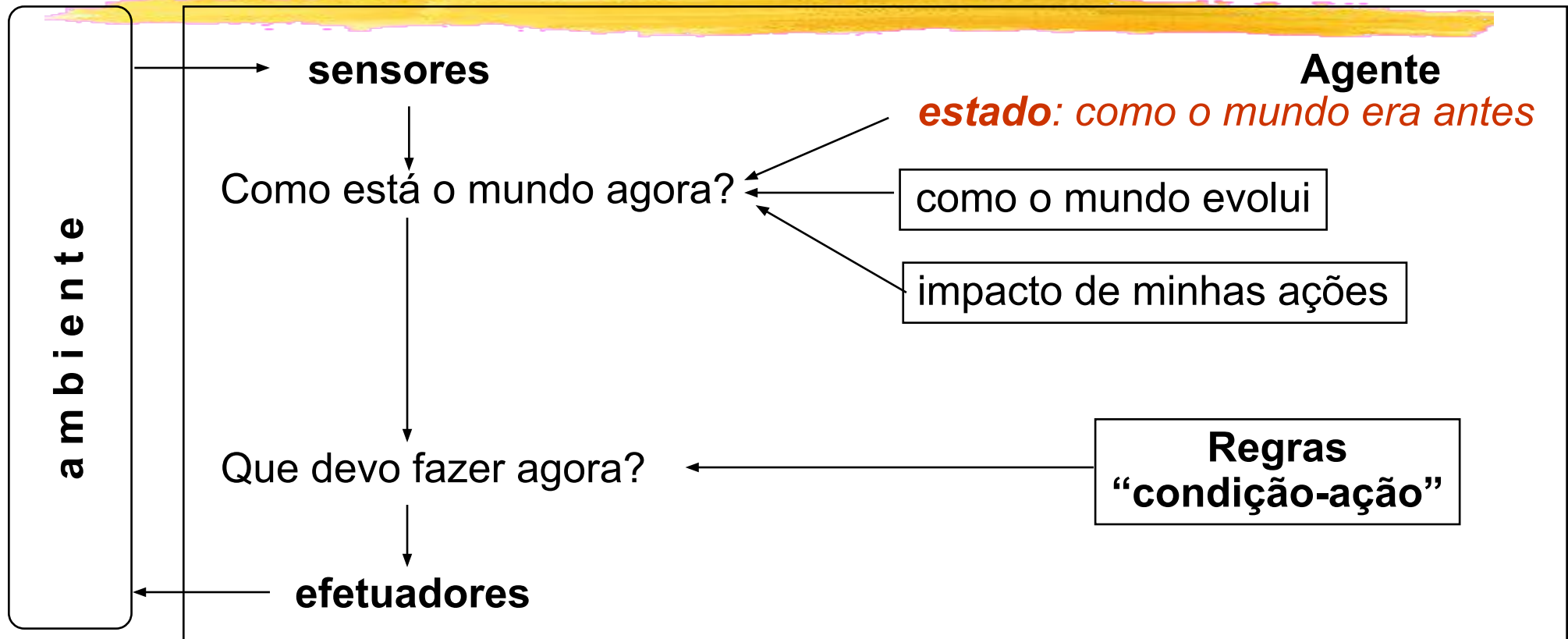
● Vantagens e desvantagens

- Regras condição-ação: representação inteligível, modular e eficiente
 - ex. **Se** velocidade > 60 **então** multar
- Não pode armazenar uma seqüência perceptiva, pouca autonomia

● Ambientes:

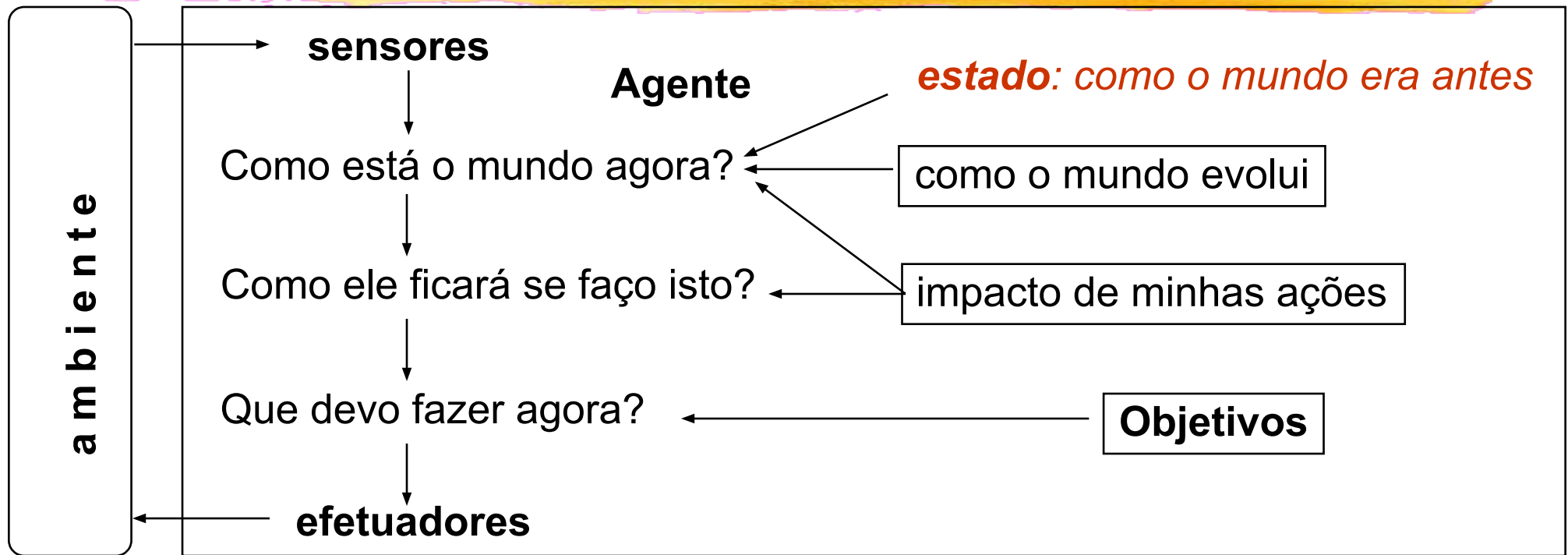
- Reflexo imprescindível em ambientes dinâmicos
- Acessível, episódico, pequeno

Agente reativo com estado interno



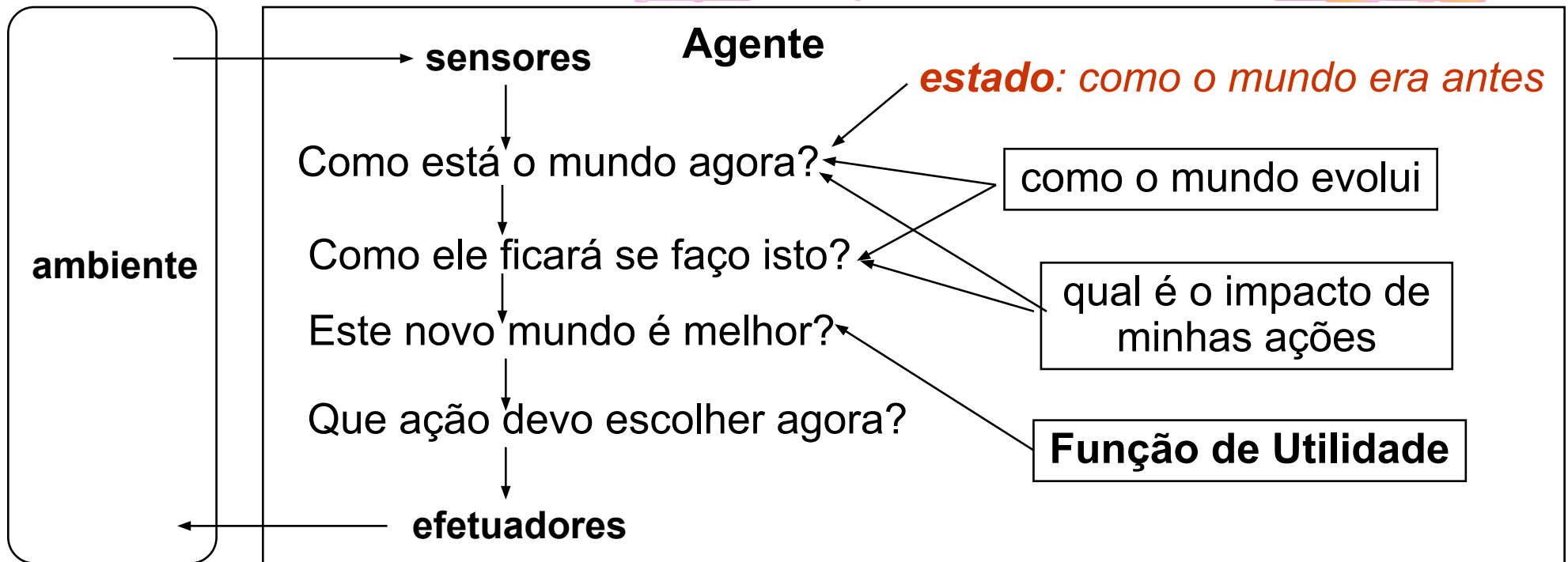
- **Desvantagem:** pouca autonomia
 - não tem objetivo, não encadeia regras
- **Ambientes:** determinista e pequeno
 - Ex. Tamagotchi

Agente cognitivo (goal-based)



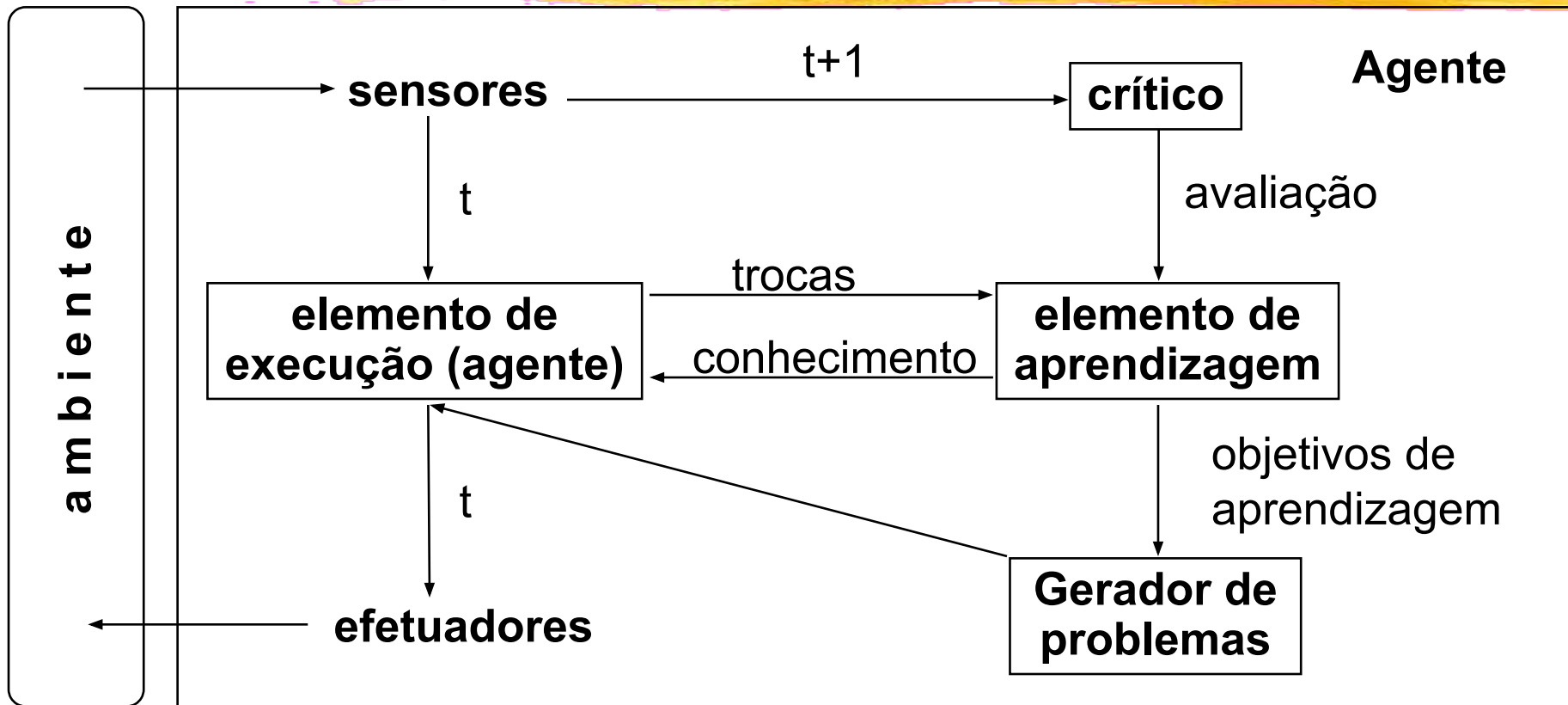
- Vantagens e desvantagens:
 - Mais *complicado* e *ineficiente*, porém mais *flexível*, *autônomo*
 - Não trata objetivos conflitantes
- Ambientes: determinista
 - ex.: xeque-mate no xadrez

Agente otimizador (utility based)



- **Ambiente:** sem restrição
- **Desvantagem:** não tem adaptabilidade
- **Ex.** motorista viçosence

Agente que aprende



- **Ambiente:** sem restrição
- **Vantagem:** tem adaptabilidade (aprende)
- **Ex.** motorista sem o mapa da cidade

Simulação do ambiente

- **Às vezes é mais conveniente simular o ambiente**
 - mais simples
 - permite testes prévios
 - evita riscos, etc...
- **O ambiente (programa)**
 - recebe os agentes como entrada
 - fornece repetidamente a cada um deles as percepções corretas e recebe as ações
 - atualiza os dados do ambiente em função dessas ações e de outros processos (ex. dia-noite)
 - é definido por um estado inicial e uma função de atualização
 - deve refletir a realidade

Simulação de ambiente

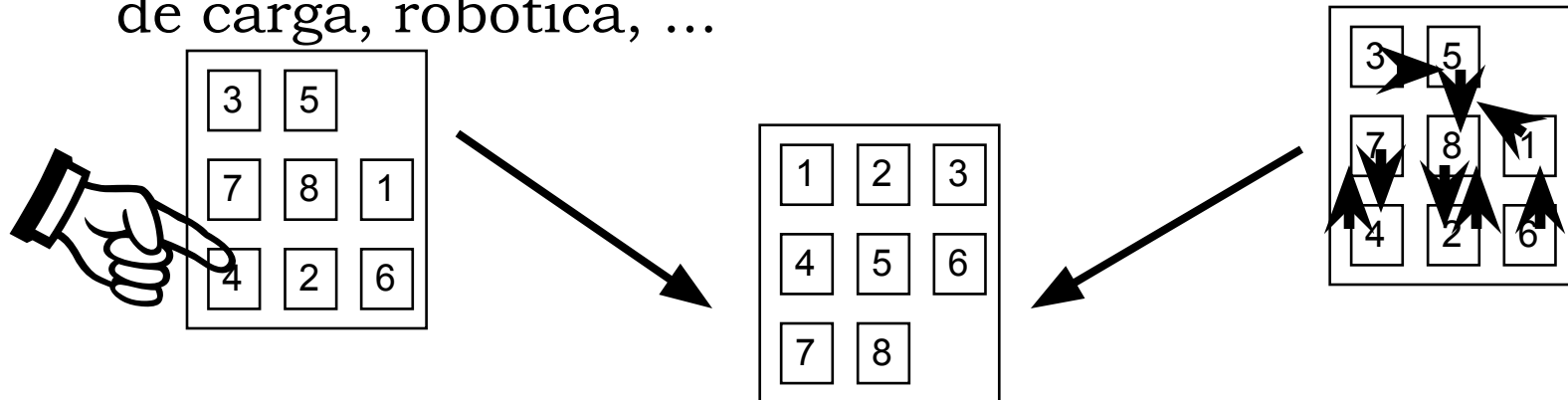
- **função** `simulaAmbiente (estado, funçãoAtualização, agentes, final)`
repita
 para cada *agente* **em** *agentes* **faça**
 `Percept[agente] := pegaPercepção(agente, estado)`
 para cada *agente* **em** *agentes* **faça**
 `Action[agente] := Programa[agente] (Percept[agente])`
 `estado := funçãoAtualização(ações, agentes, estado)`
 //opcional
 `scores := avaliaDesempenho(scores, agente, estado)`
até final
- **Observação:**
 - não cair em tentação “roubando” do ambiente a descrição do que aconteceu. Usar a memória do agente!

Inteligência coletiva

- **Porque pensar a inteligência/racionalidade como propriedade de um único indivíduo?**
- **Não existe inteligência ...**
 - Em um time de futebol?
 - Em um formigueiro?
 - Em uma empresa (ex. correios)?
 - Na sociedade?
- **Solução: IA Distribuída**
 - Agentes simples que juntos resolvem problemas complexos tendo ou não consciência do objetivo global
 - Proposta por Marvin Minsky e em franca expansão...
 - o próprio ambiente pode ser modelado como um agente

IA Distribuída: dois tipos de sistemas

- **Resolução distribuída de problemas**
 - consciência do objetivo global e divisão clara de tarefas
 - Exemplos: Robótica clássica, Busca na Web, Gerência de sistemas distribuídos, ...
- **Sistemas Multi-agentes**
 - não consciência do objetivo global e nem divisão clara de tarefas
 - Exemplos: n-puzzle, futebol de robôs, balanceamento de carga, robótica, ...



Conclusões



- **Agentes em IA**
 - Metodologia (metáfora) para projeto de sistemas
 - Sistemas multi-agentes e robótica
- **Agentes em computação**
 - Adoção de uma nova metáfora (antropomórfica e sociológica). Extrapolação de OOP
 - Integração de técnicas de IA
 - Novas tecnologias próprias à Web (ex. mobilidade)
 - Marketing (moda)
- **Agentes: técnica ou metodologia ?**

Desenvolver software inteligente



- **Projeto:**

- Modelar tarefa em termos de ambiente, percepções, ações, objetivos e utilidade
- Identificar o tipo de ambiente
- Identificar a arquitetura de agente adequada ao ambiente e tarefa

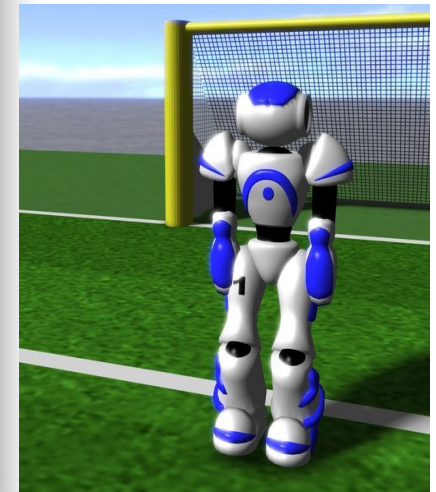
- **Implementação**

- o gerador e o simulador de ambientes
- componentes do agente (vários tipos de conhecimento)
- Testar o desempenho com diferentes instâncias do ambiente

Exemplos de ambientes multi-agentes

- **RoboCup (Simulation League)**

- SimSpark é um simulador multiagent dotado de física genérica para agentes em ambientes tridimensionais. Baseia-se no framework Spark. É utilizado como o servidor oficial da simulação RoboCup3D.
- http://wiki.robocup.org/wiki/Soccer_Simulation_League



Exemplos de ambientes multi-agentes

- **RoboCup (Simulation League)**



Exemplos de ambientes multi-agentes

● **Gamebots**

- Desenvolvido pelo instituto de Ciência da Informação da University of Southern California e busca utilizar o jogo Unreal Tournament em um domínio para a pesquisa em Inteligência Artificial.



Exemplos de ambientes multiagentes

● Gamebots

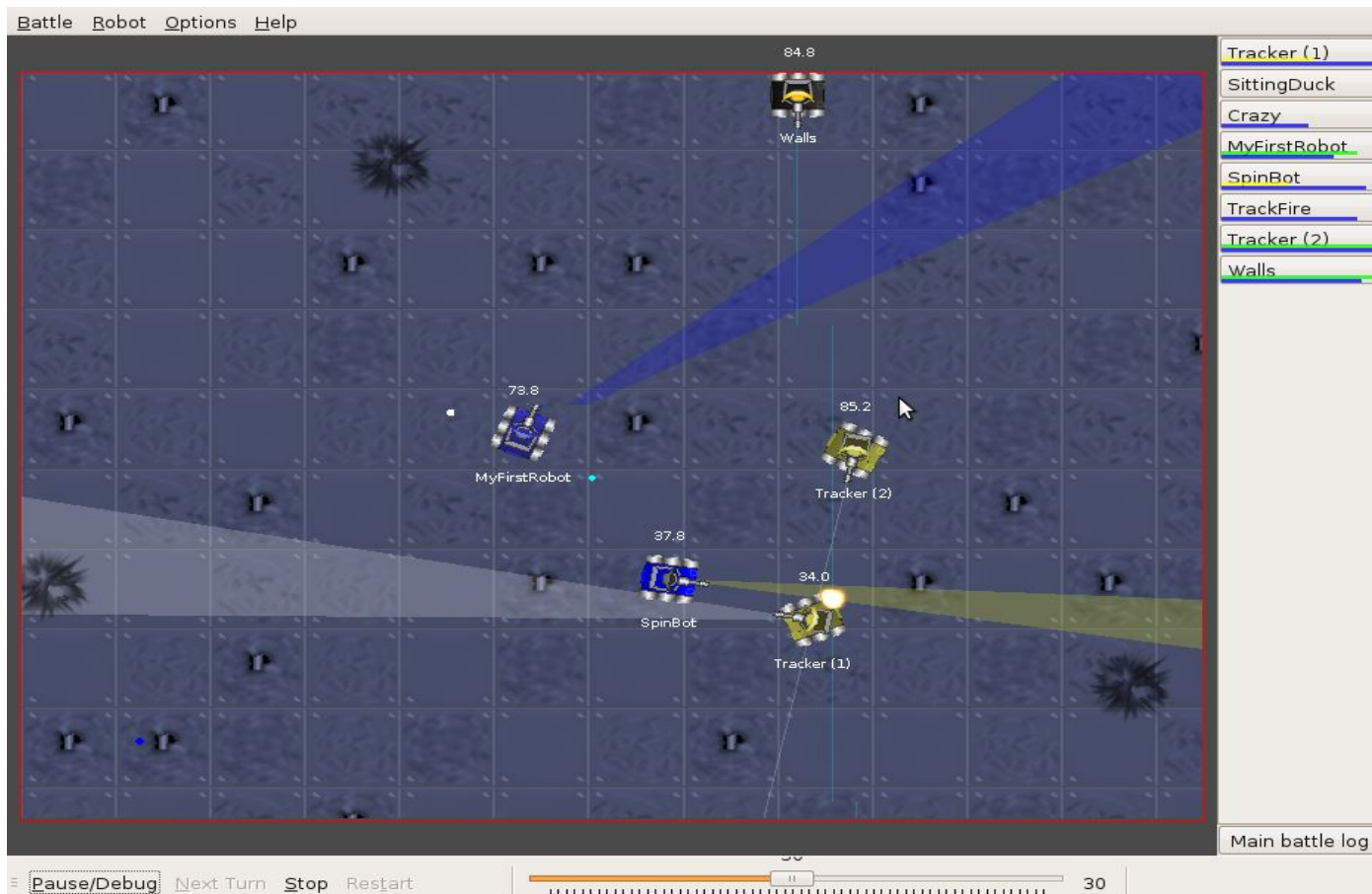
- <http://www-2.cs.cmu.edu/~galk/GameBots/WEB/index.html>



Exemplos de ambientes multiagentes

● Robocode

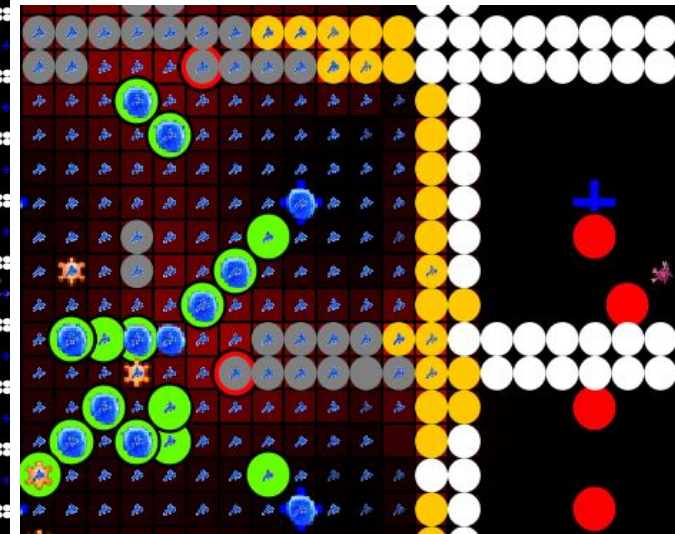
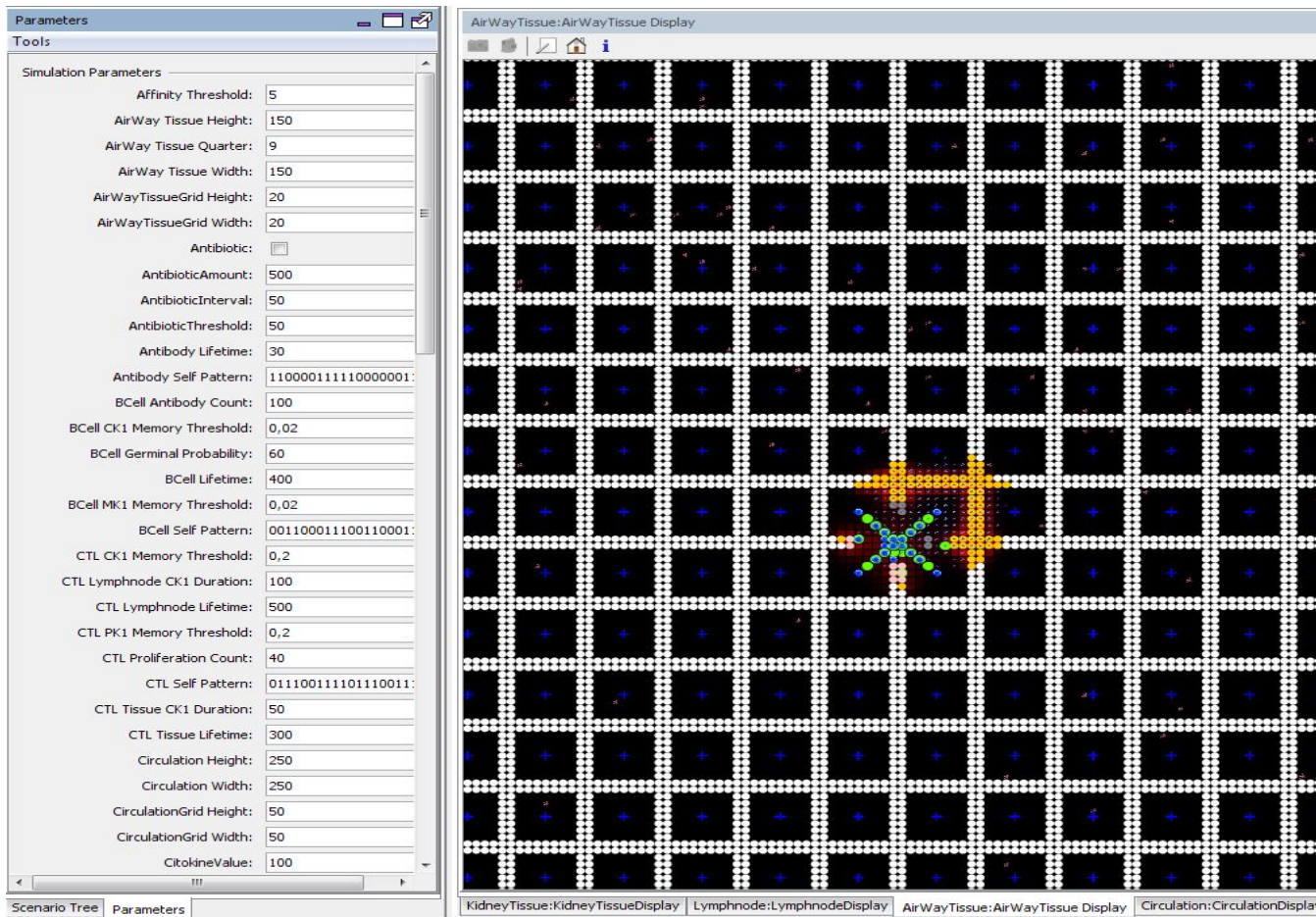
- Desenvolvido pela IBM para o ensino de Java. O usuário deve especificar o comportamento de tanques que irão combater em uma arena. <http://robocode.sourceforge.net/>



Exemplos de ambientes multiagentes

● Repast

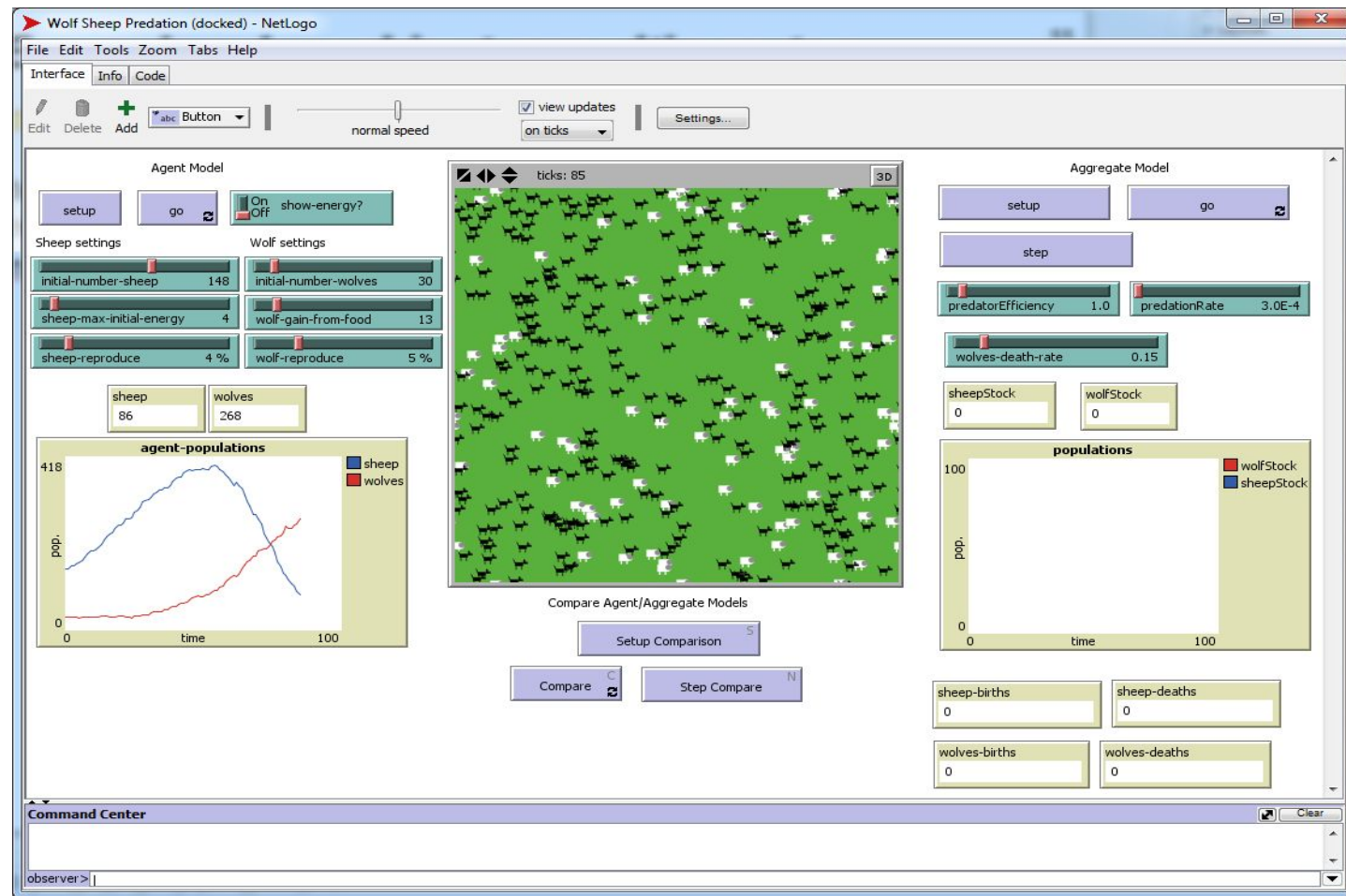
- Ambiente para criação de sistemas multiagentes reativos. É um framework gratuito e de código aberto.
- <http://sourceforge.net/projects/repast>



Exemplos de ambientes multiagentes

• NetLogo

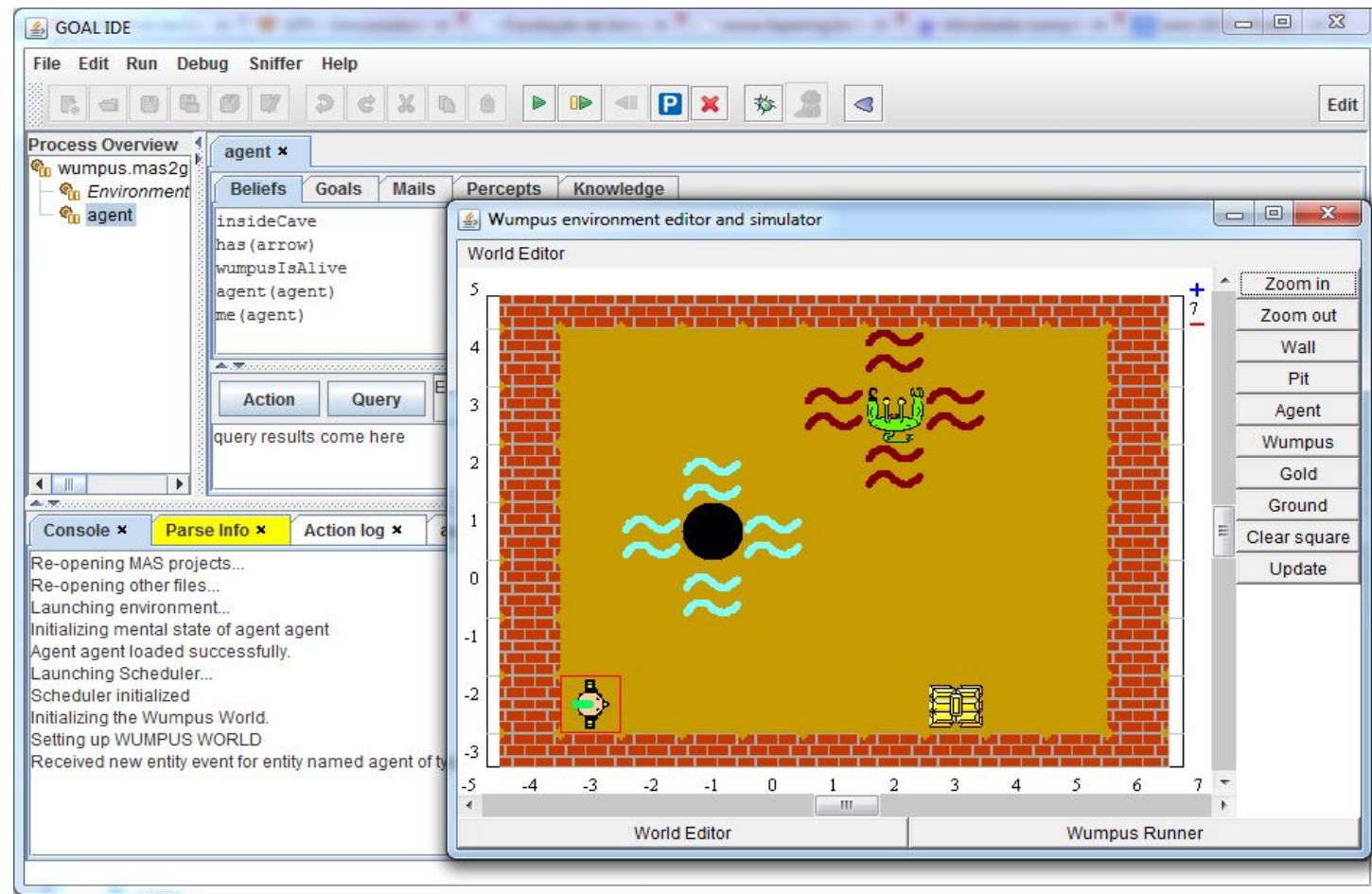
- Ambiente para criação de sistemas multiagentes reativos. Utiliza a linguagem logo para programar os agentes.
- <http://ccl.northwestern.edu/netlogo/>



Exemplos de ambientes multiagentes

● Goal

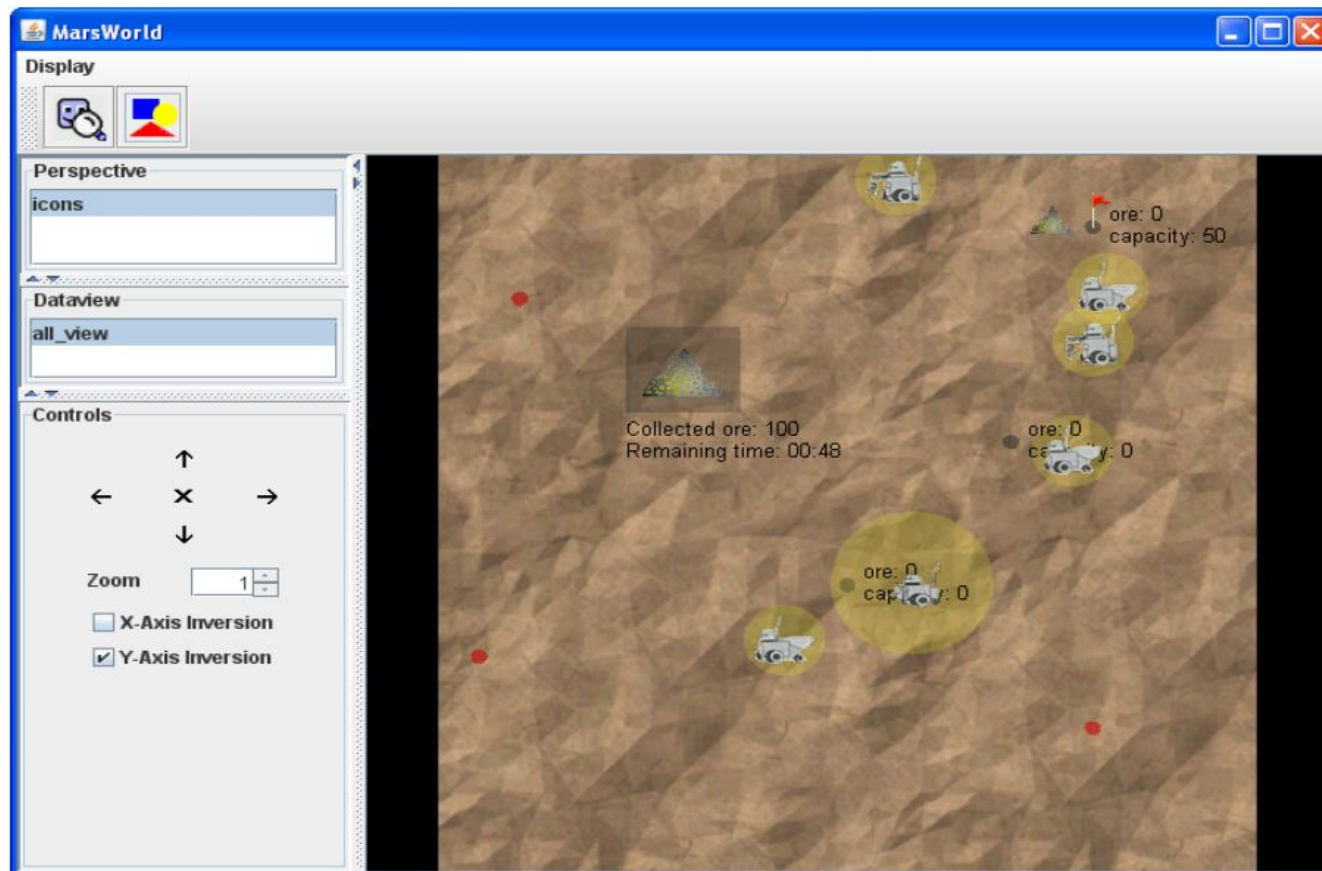
- Ambiente para criação de sistemas multiagentes deliberativos. Utiliza a linguagem prolog para programar os agentes.
- <http://mmi.tudelft.nl/trac/goal>



Exemplos de ambientes multiagentes

● Jadex

- Ambiente para criação de sistemas multiagentes deliberativos. Utiliza a linguagem Java para programar os agentes.
- <http://jadex-agents.informatik.uni-hamburg.de/xwiki/bin/view/About/Features>



Exemplos de ambientes multiagentes

● **Malmo**

- Ambiente para teste de agentes usando o ambiente Minecraft.
- <https://github.com/Microsoft/malmo>



Exemplos de ambientes multiagentes

● **StarCraft Brood war**

- Ambiente para teste de agentes usando o ambiente Starcraft Broodwar.
- <https://github.com/bwapi/bwapi>



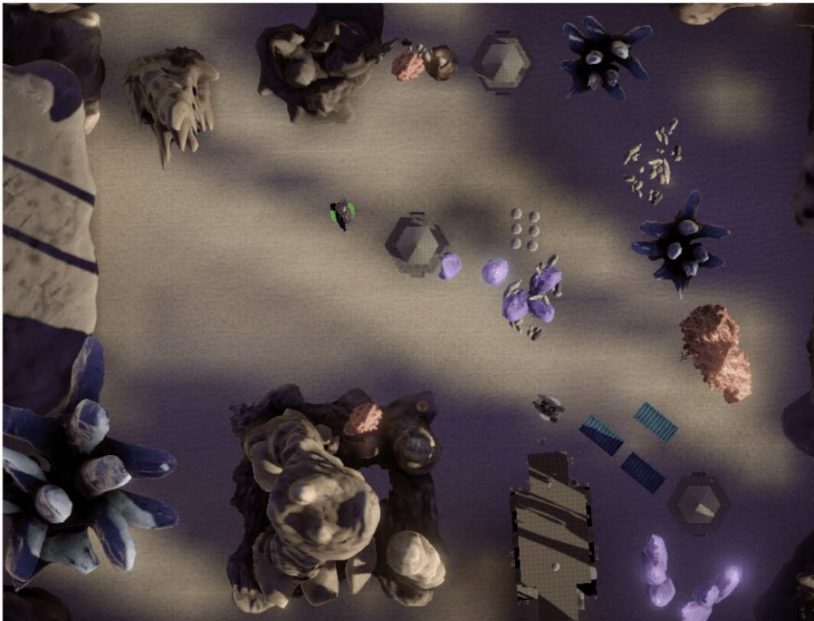
Exemplos de ambientes multiagentes

- **OpenAI: Emergent Tool Use from Multi-Agent Interaction**
 - Nesse ambiente, os agentes jogam um jogo de esconde-esconde baseado em equipe. Os agentes azuis têm a tarefa de evitar a linha de visão dos agentes vermelho.
 - <https://openai.com/blog/emergent-tool-use/>



Exemplos de ambientes multiagentes

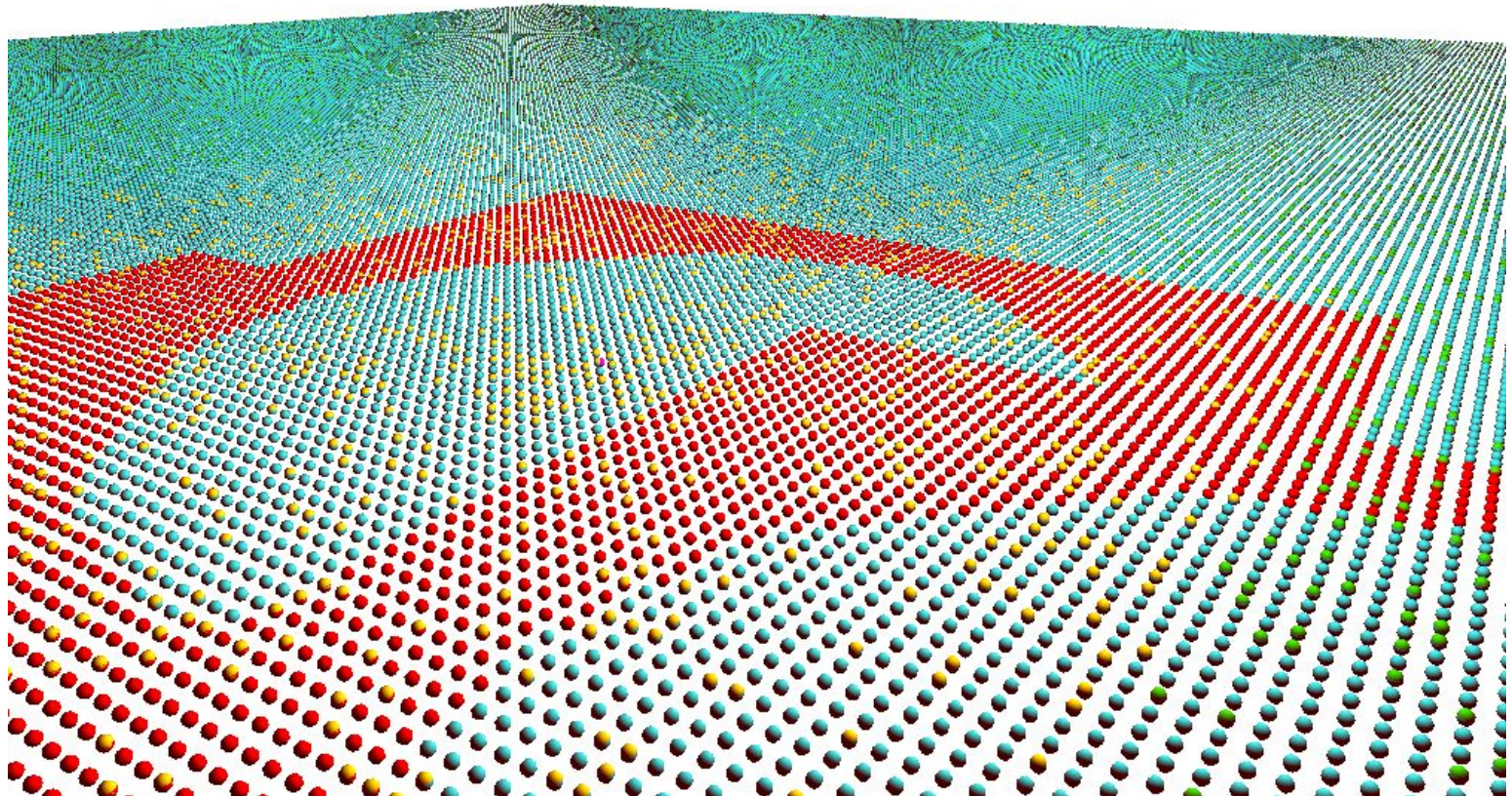
- **Unity Machine Learning Agents (ML-Agentes)**
 - É um plugin Unity de código aberto que permite que jogos e simulações sirvam como ambientes para treinar agentes inteligentes.
 - Os agentes podem ser treinados usando aprendizado de reforço, aprendizado de imitação, neuroevolução ou outros métodos de aprendizado de máquina
 - <https://unity3d.com/pt/how-to/unity-machine-learning-agents>



Exemplos de ambientes multiagentes

● **Flame GPU**

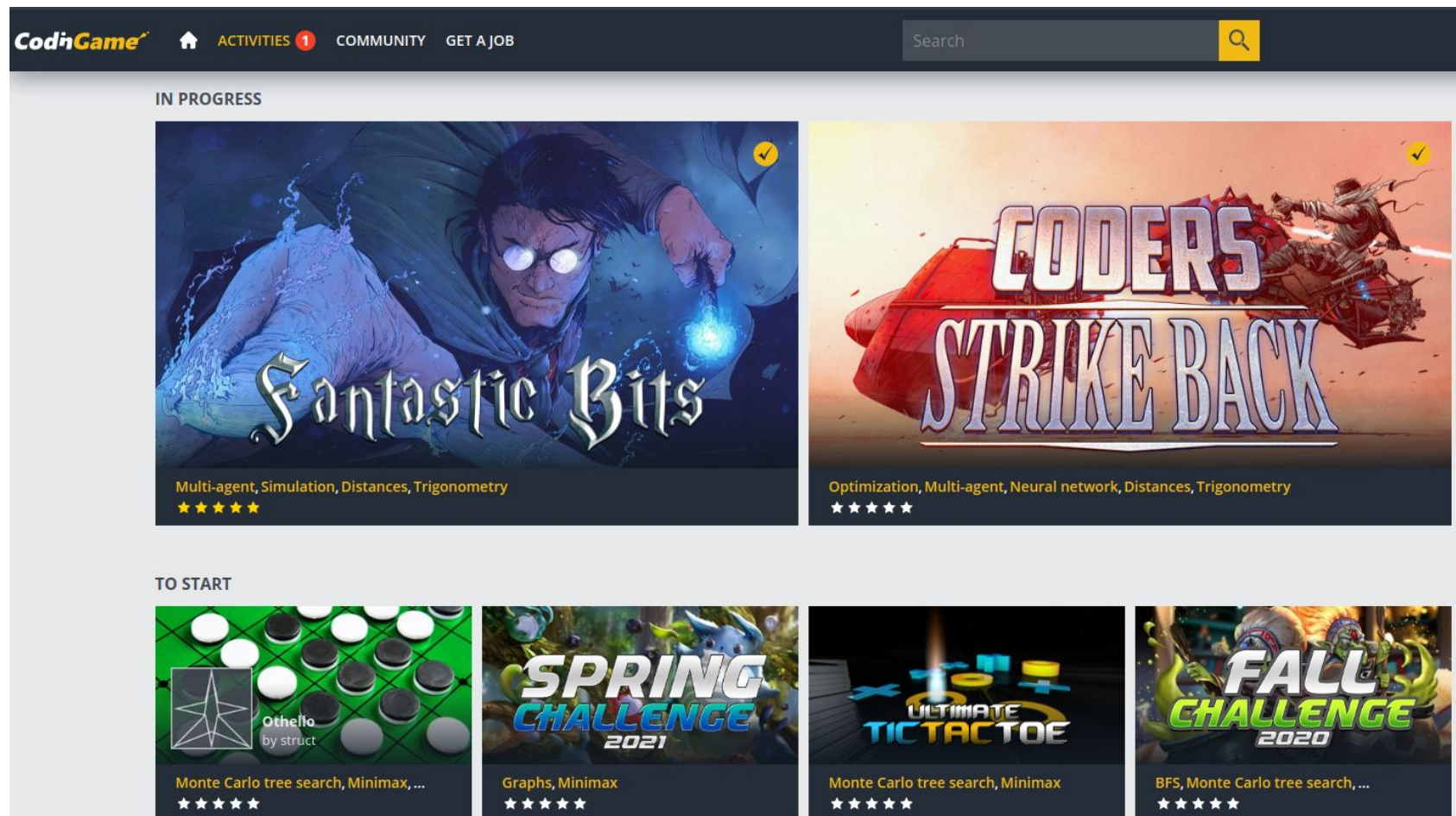
- Ambiente para criação de sistemas multiagentes reativos massivos executando em GPU.
- <http://www.flamegpu.com/>



Exemplos de ambientes multiagentes

● CODING GAME

- Ambiente para aprender a codificar por meio de games.
- <https://www.codinggame.com/>





FIM