University of West Bohemia, Pilsen
23rd February 2015

UT2004 bots made easy!

# Intelligent Virtual Agents by Pogamut 3

## Gentle introduction

# Intelligent Virtual Agents
## What?

- Software agent *(by Michael Wooldridge)*
  - Embodied intelligent autonomous entity

# Intelligent Virtual Agents
## What?

- Software agent *(by Michael Wooldridge)*
  - **Embodied** intelligent autonomous entity
    - Body that is subject to some (physical) laws within its environment

# Intelligent Virtual Agents
## What?

- Software agent *(by Michael Wooldridge)*
  - Embodied intelligent **autonomous** entity
    - Operating on an owner's behalf but without any interference of that ownership entity

# Intelligent Virtual Agents
## What?


- Software agent *(by Michael Wooldridge)*
  - Embodied **intelligent** autonomous entity
    - Reactive
    - Proactive
  - *Thermostat may be an agent too!*

# Intelligent Virtual Agents
## What?

- Software agent *(by Michael Wooldridge)*
  - Embodied **intelligent** autonomous entity
    - Reactive
    - Proactive
    - Social
  - *Okey… 'more' thermostats…*

# Intelligent Virtual Agents
## What?

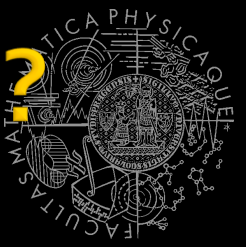
- Software agent *(by Michael Wooldridge)*
  - Embodied intelligent autonomous entity
    - Reactive
    - Proactive
    - Social
- Intelligent Virtual Agent *(IVA)*
  - **Specific** software agent **type**
    - Wholly and movably embodied within Complex virtual environment / world
    - Acts under bounded rationality

# Intelligent Virtual Agents
## What?

- Software agent *(by Michael Wooldridge)*
  - Embodied intelligent autonomous entity
    - Reactive
    - Proactive
    - Social
- Intelligent Virtual Agent *(IVA)*
  - Specific software agent type
    - Wholly and movably embodied within **Complex virtual environment (… ?)**
    - Acts under bounded rationality

# What is Complex V-Environment?

## How it can be classified?

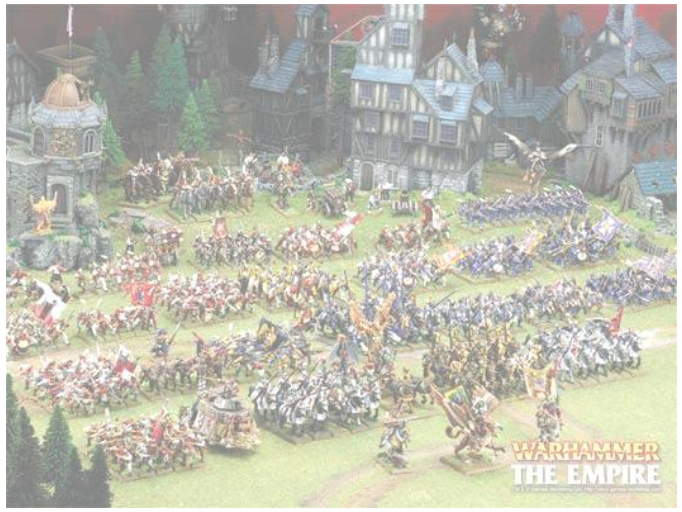


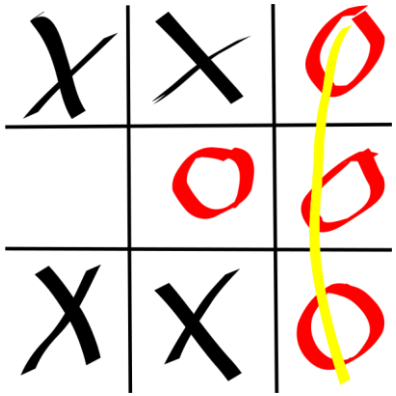
# Env. Classification
## Properties

- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy


# TicTacToe
## Properties

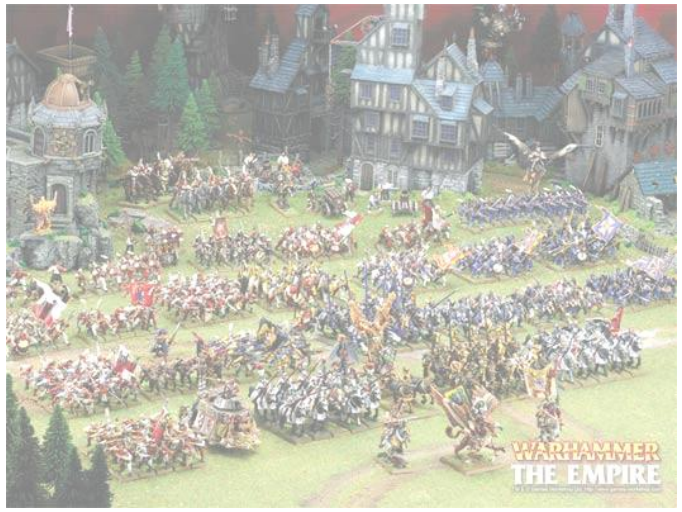
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

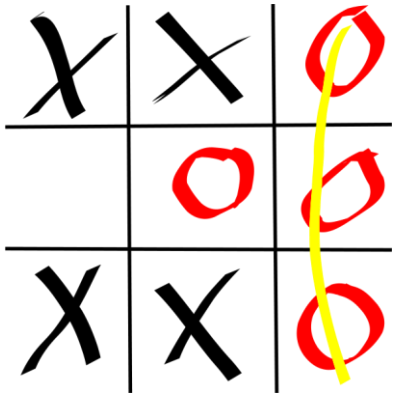


# TicTacToe
## Properties

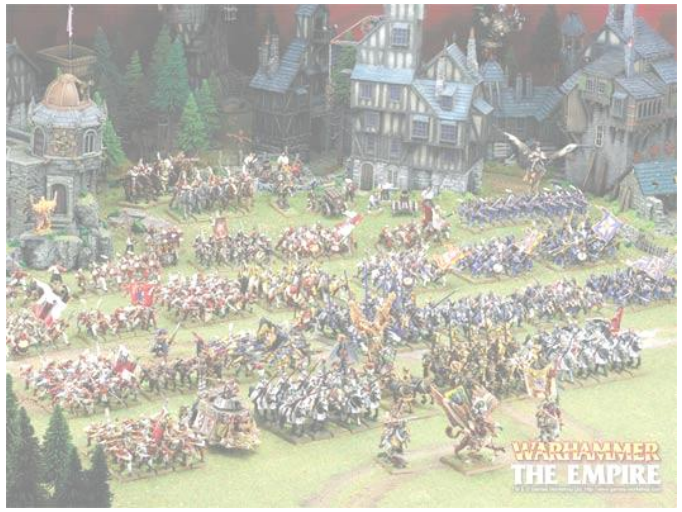
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

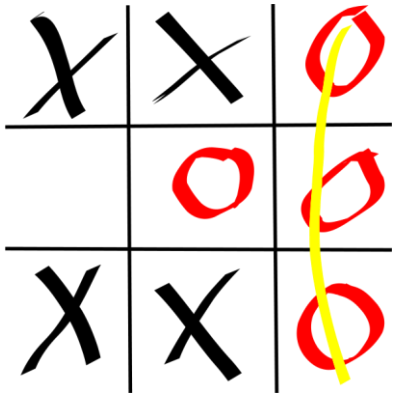


# TicTacToe
## Properties

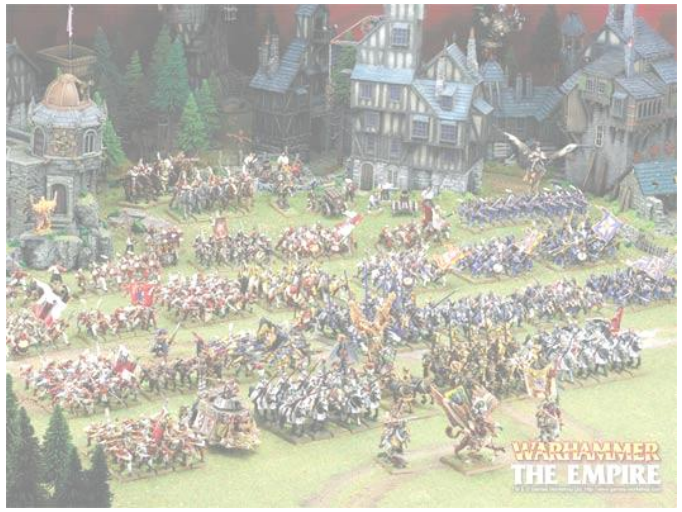
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

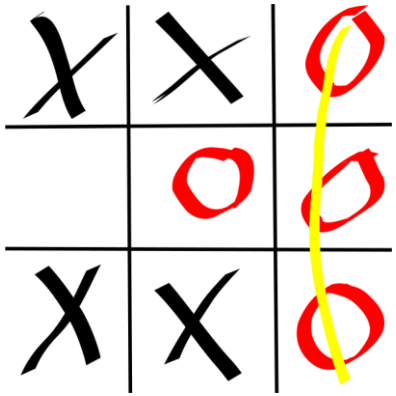


# TicTacToe
## Properties

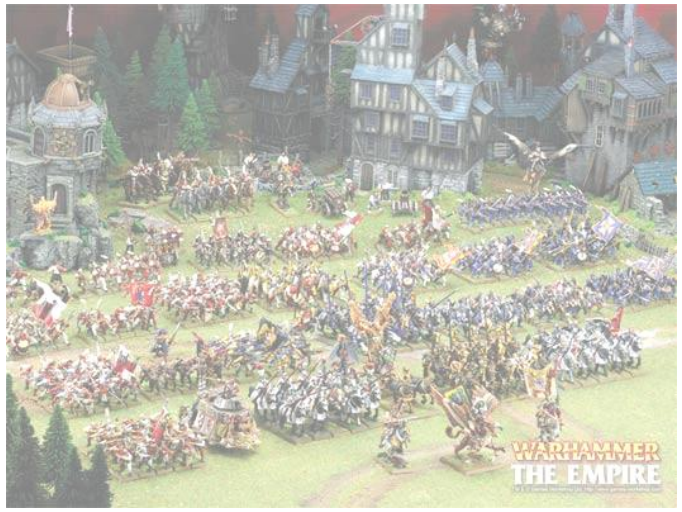
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

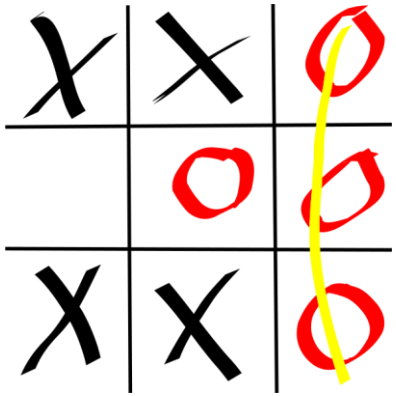


# TicTacToe
## Properties

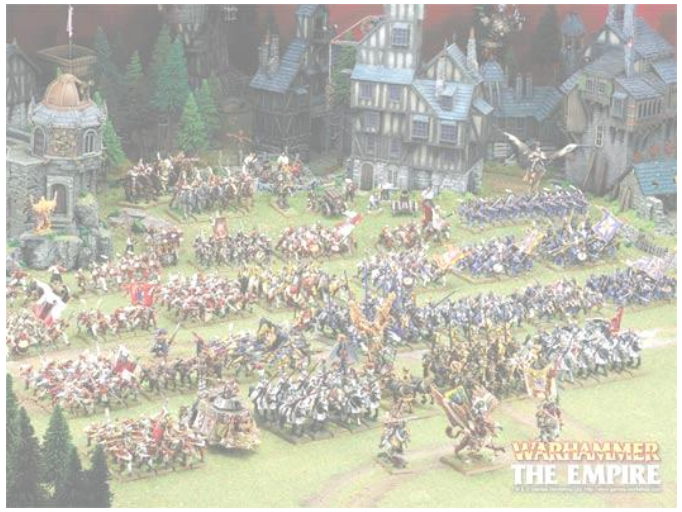
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

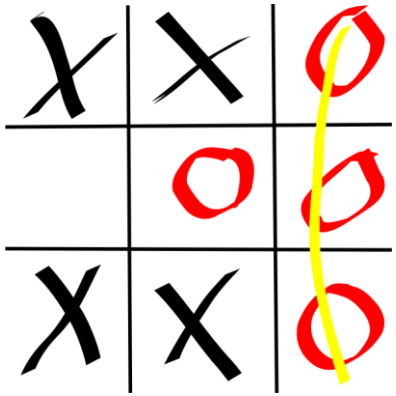


# TicTacToe
## Properties

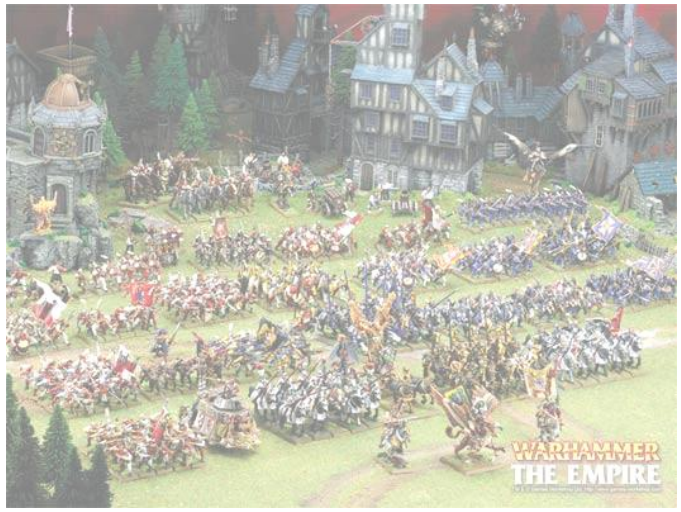
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

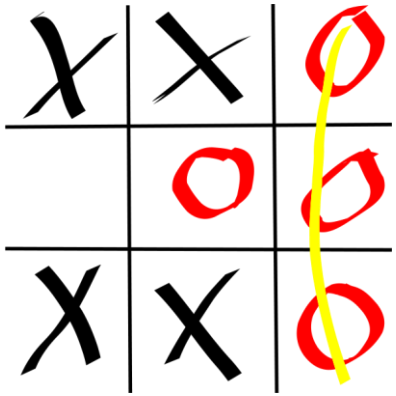


# TicTacToe
## Properties

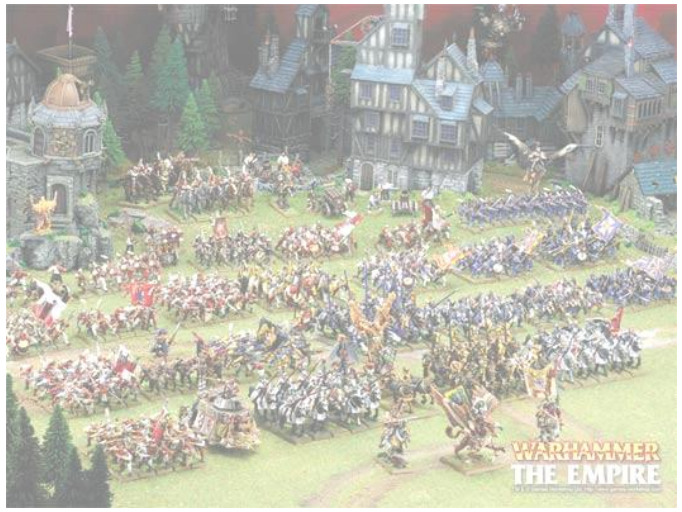
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

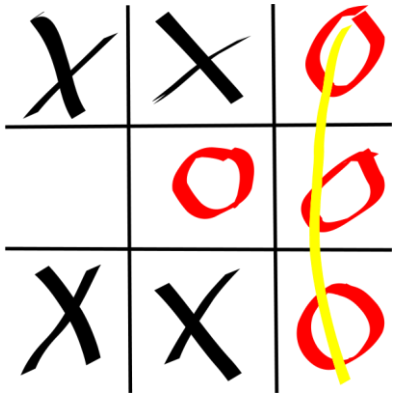


# TicTacToe
## Properties

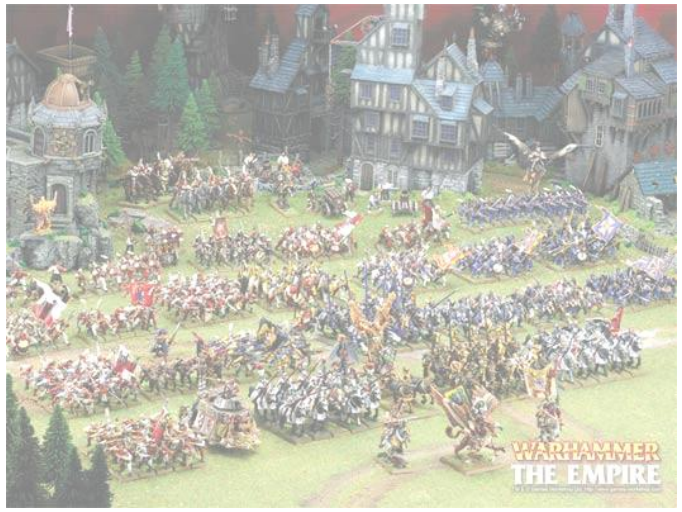
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

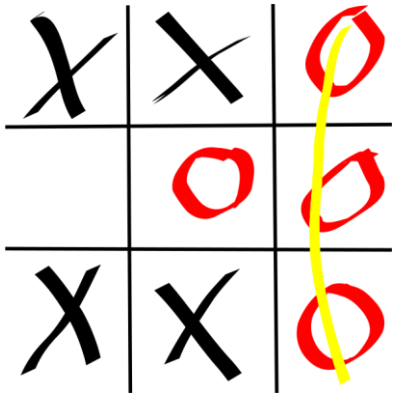


# TicTacToe
## Properties

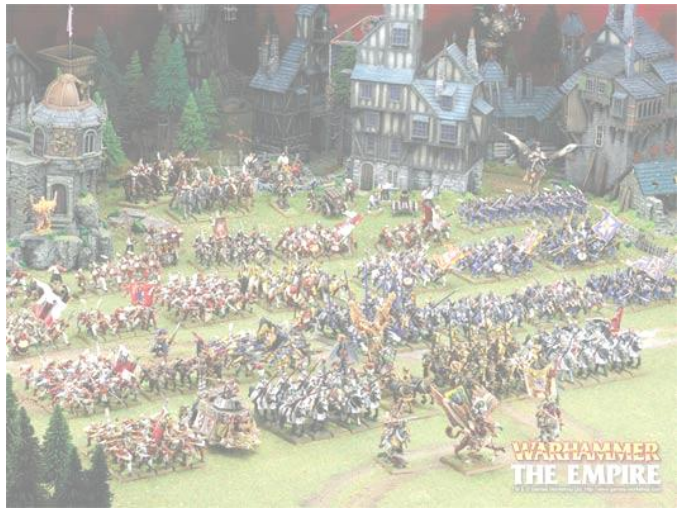
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

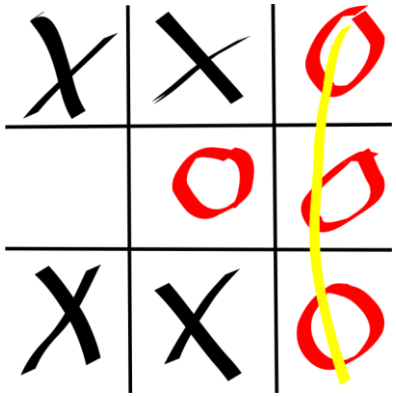


# TicTacToe
## Properties

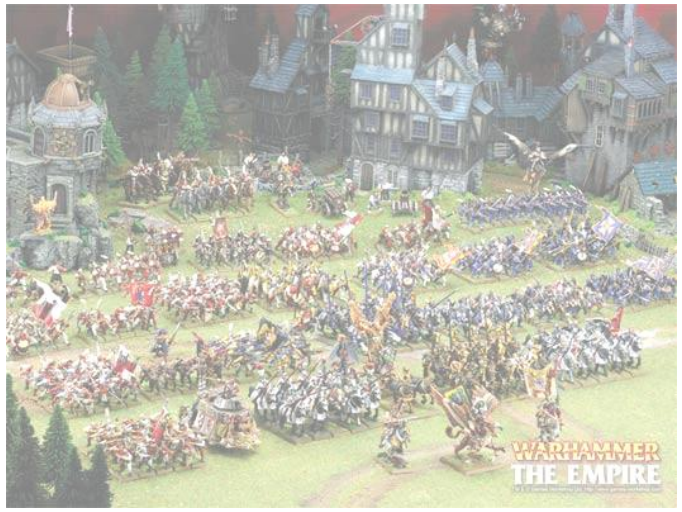
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

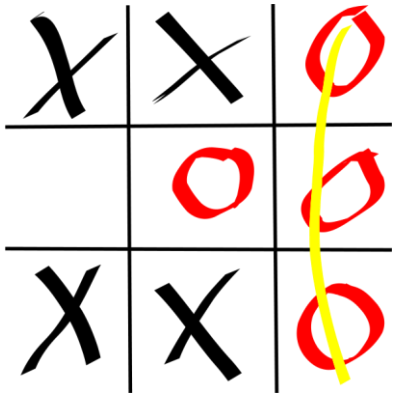


# TicTacToe
## What does it mean?

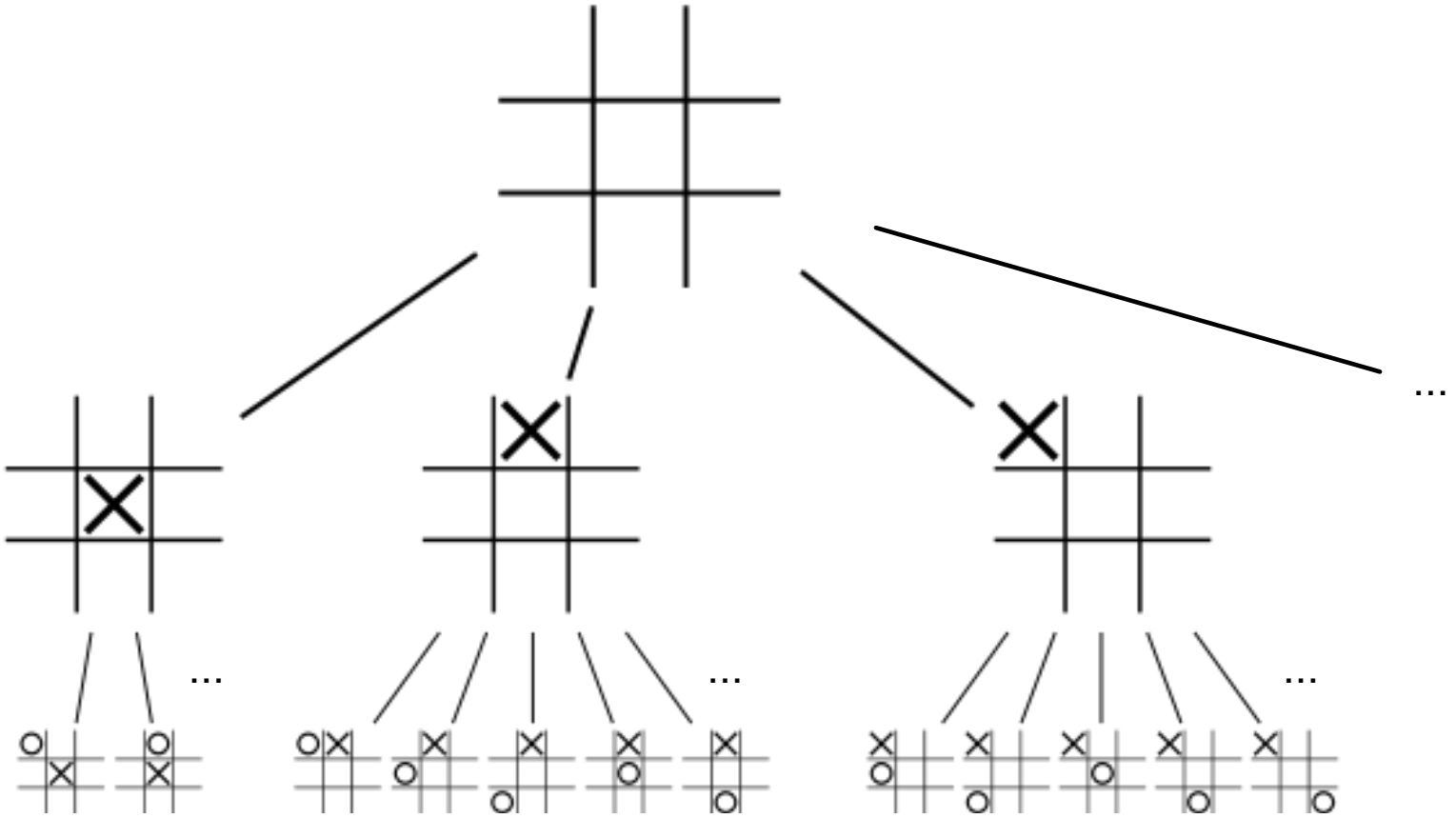
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy



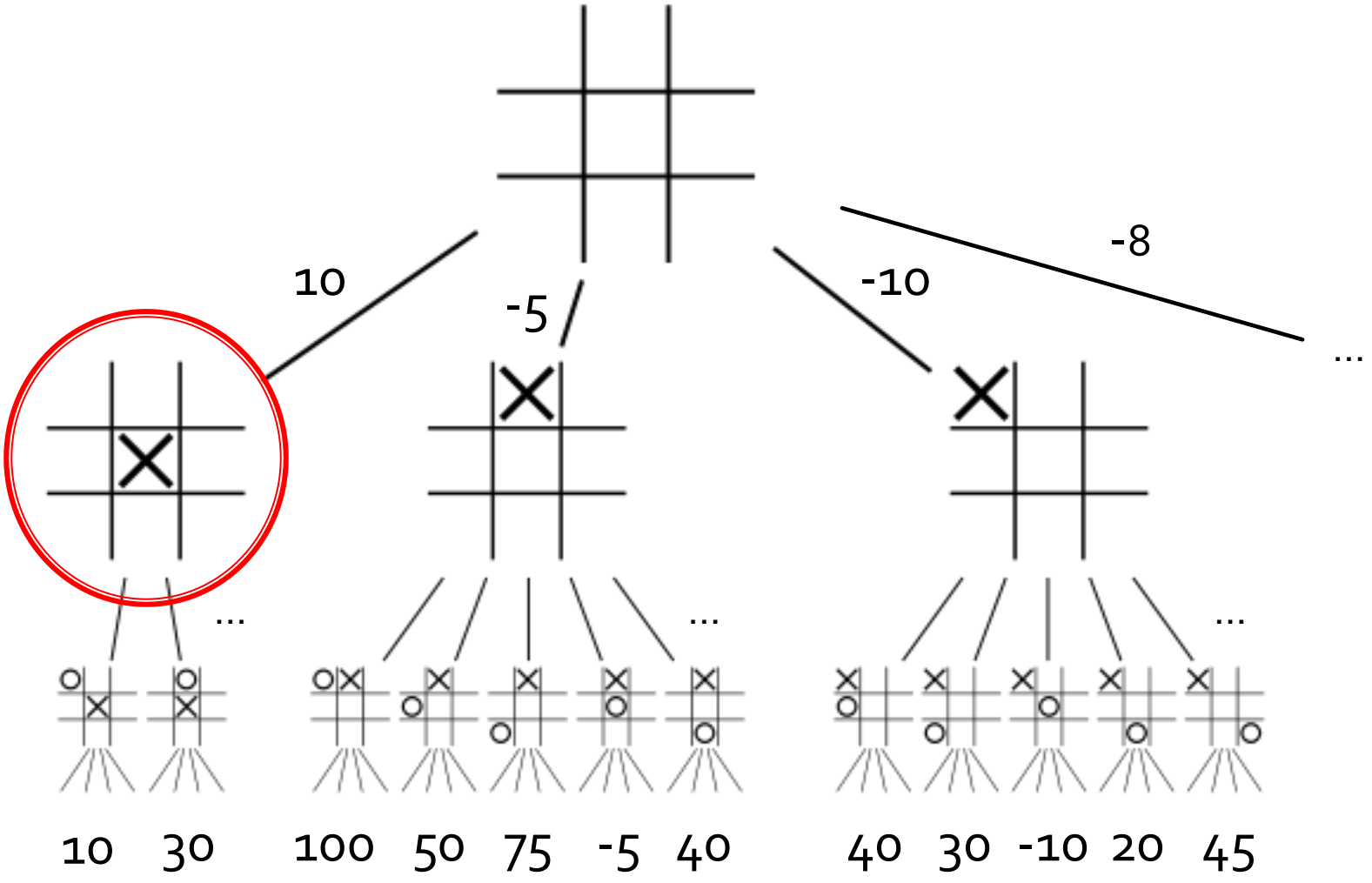
# „Reasoning as search“

## -- *Alan Newell*



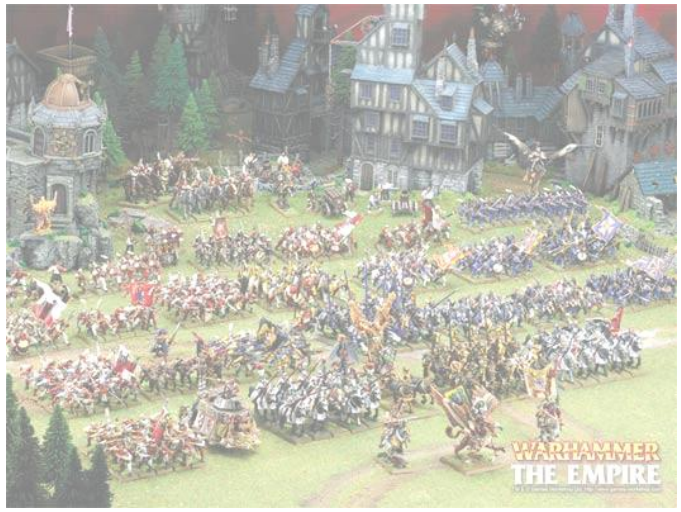
# „Reasoning as search“

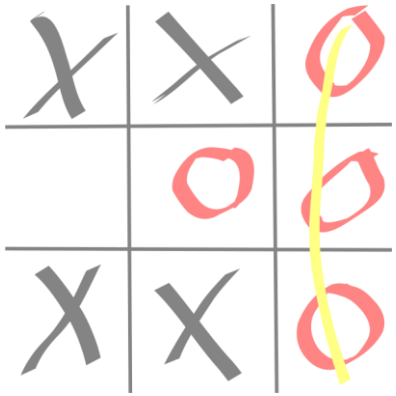
## => MIN-MAX algorithm + modifications



# UT2004
## What can be said?

- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic
- Discrete *vs.* Continuous
- Known *vs.* Unknown
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

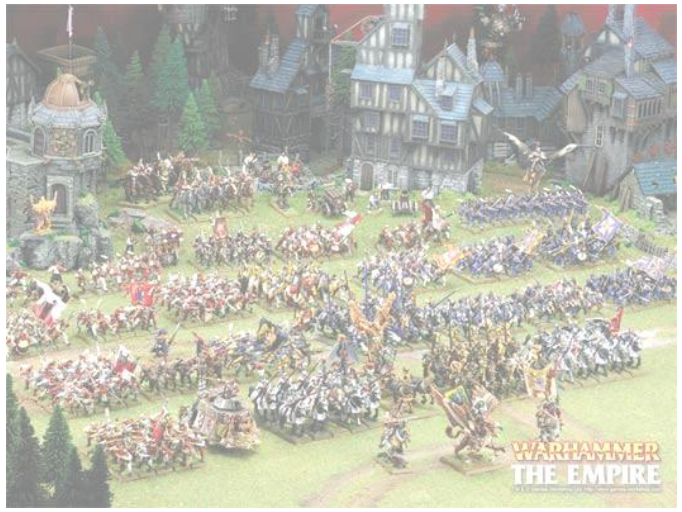


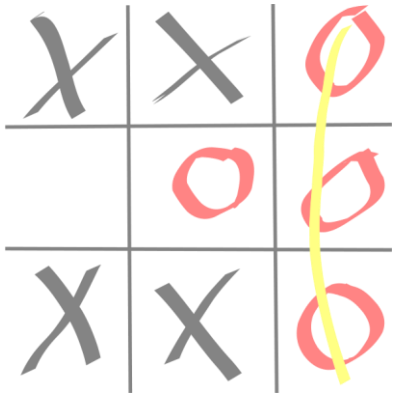



# UT2004

## The (almost) worst case imaginable!

- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic (weakly)
- Discrete *vs.* Continuous
- Known *vs.* Unknown (weakly)
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

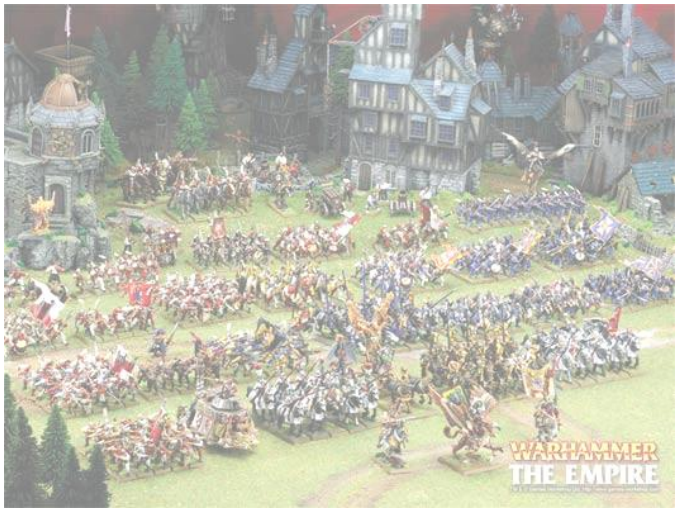


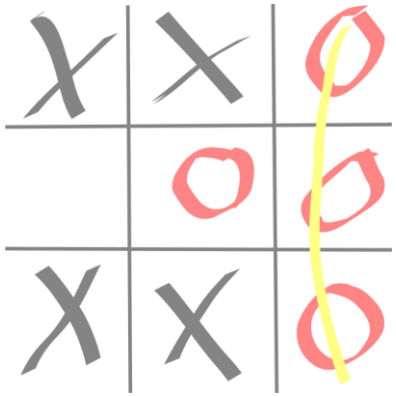



# UT2004

## => Hard to "search or plan"

- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic (weakly)
- Discrete *vs.* Continuous
- Known *vs.* Unknown (weakly)
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

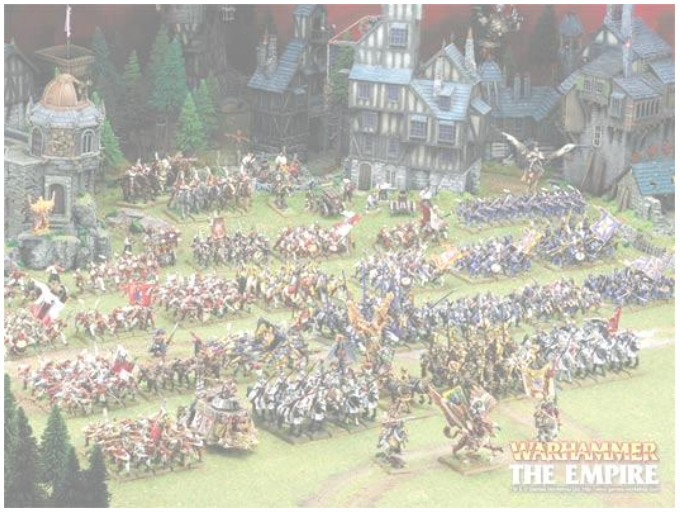


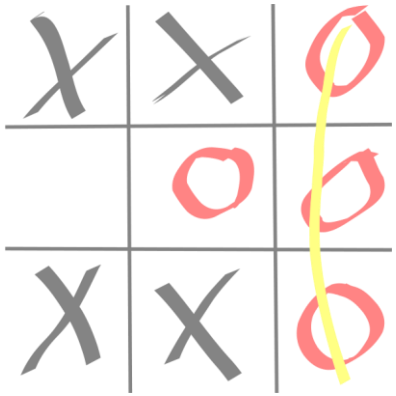



# UT2004

## => (Semi) Reactive Action-Selection

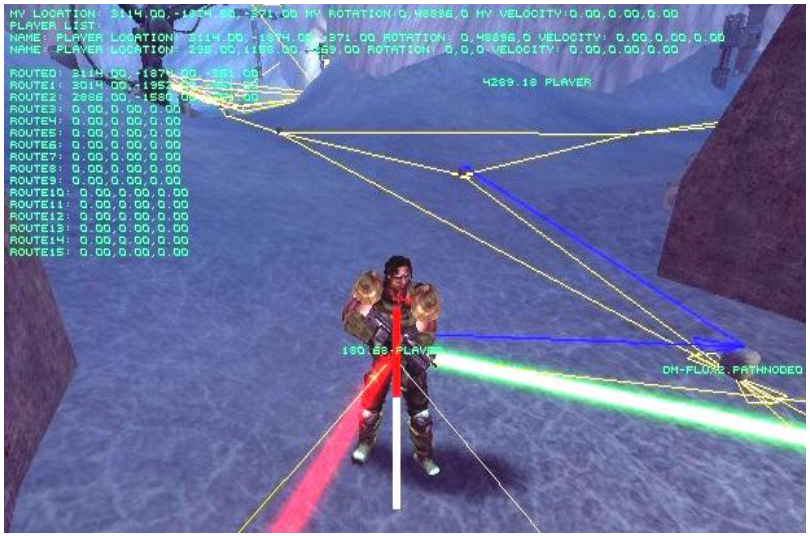
- Fully *vs.* Partially observable
- Episodic *vs.* Sequential
- Static *vs.* Dynamic
- Single *vs.* Multi agent
- Deterministic *vs.* Stochastic (weakly)
- Discrete *vs.* Continuous
- Known *vs.* Unknown (weakly)
- Turn-based *vs.* Real-time
- Noiseless *vs.* Noisy

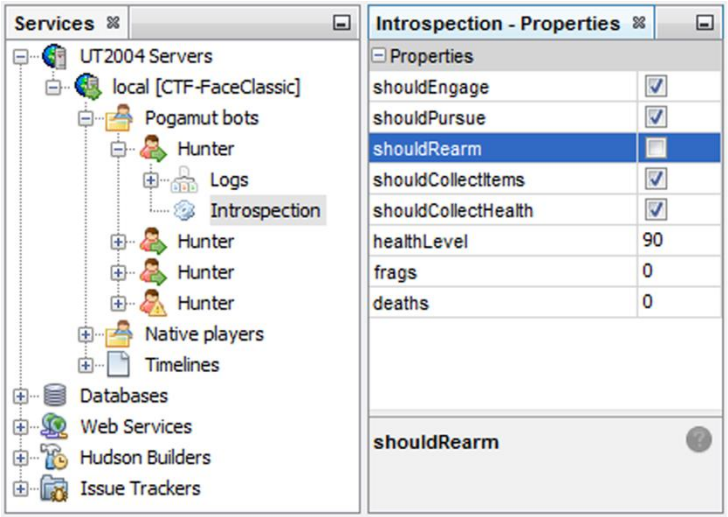


# Pogamut 3

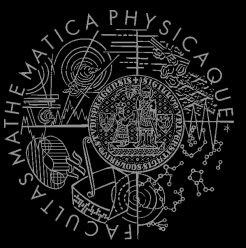
## What?



# PogamutCup 2015

## Detour…




Tournament between computer-controlled bots inside complex 3D virtual environment of Unreal Tournament 2004

Concrete setup

- 1v1, death-match, known maps
- match: up-to 10 frags or 10 minutes

Find us at http://www.pogamutcup.com

Registration opened till ? ? 2015 (to decide)

**Win the prize-money!**

# IVAs and Virtual Environments

## How it works?

Environment state (E)

Perception (P)

Action (A)

Memory / State (S)

Action execution

1. Observed state E is exported to the agent **p: E -> P**
2. Agent performs action-selection **f: PxS -> AxS**
3. Actions are simulated in the environment: **s: $A^n$ x E -> E**

One cycle for 2 agents: $\mathbf{E_{i+1} = s( [ f_1(p_1(E_i), S_1), f_2(p_2(E_i), S_2) ], E_i )}$

# IVAs and Virtual Environments

## How it works?

Environment state (E)

Perception (P)

Action (A)

Memory / State (S)

Action execution

Where is real-time?!

1. Observed state E is exported to the agent **p: E -> P**
2. Agent performs action-selection **f: PxS -> AxS**
3. Actions are simulated in the environment: **s: $A^n$ x E -> E**

One cycle for 2 agents: $\mathbf{E_{i+1} = s( [\ f_1(p_1(E_i), S_1), f_2(p_2(E_i), S_2)\ ], E_i )}$

# IVAs and Virtual Environments

## How it works?

Environment state (E)

Perception (P)

Action (A)

Memory / State (S)

Action execution

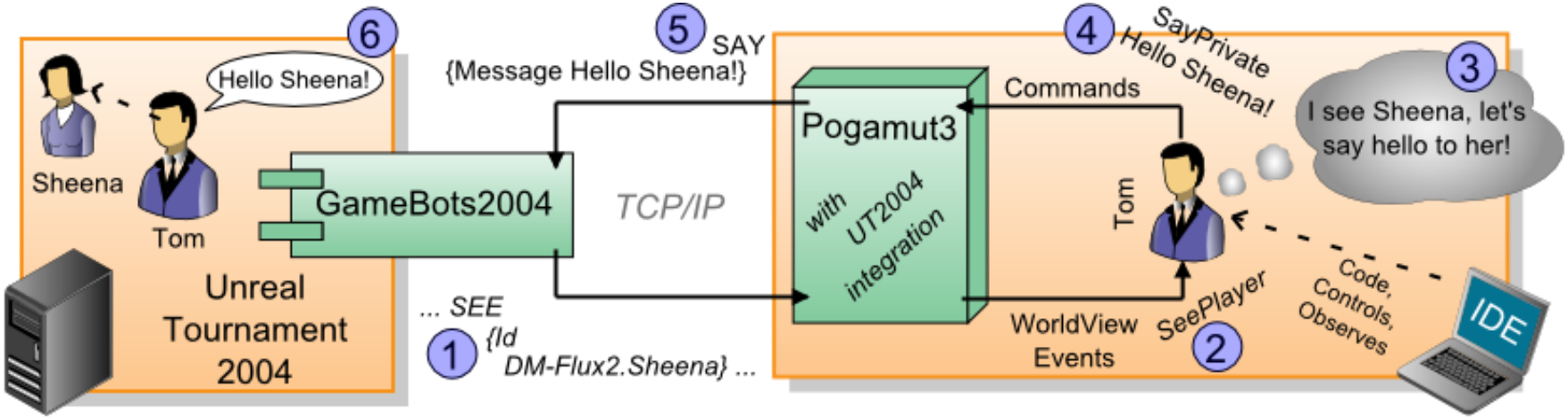
Fn s does not wait!

1. Observed state E is exported to the agent **p: E -> P**
2. Agent performs action-selection **f: PxS -> AxS**
3. Actions are simulated in the environment: **s: $A^n$ x E -> E**

One cycle for 2 agents: $E_{i+1} = s( [ f_1(p_1(E_i), S_1), f_2(p_2(E_i), S_2) ], E_i )$

# IVAs and Virtual Environments

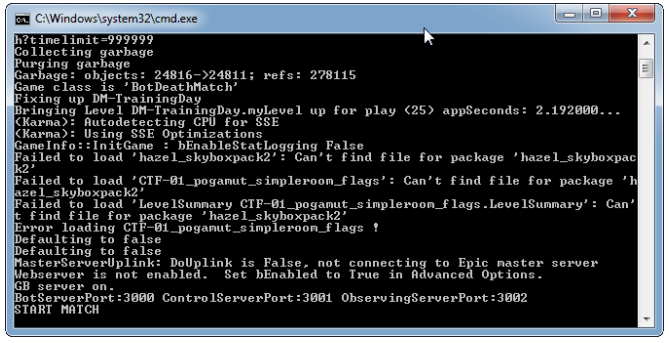
## How it is implemented?



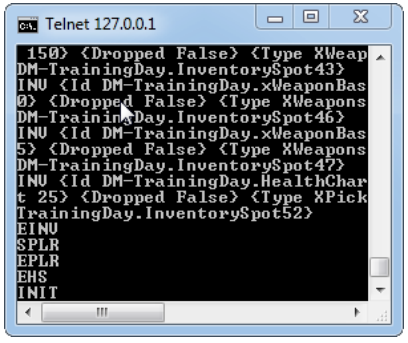

**Let's *TELNET*-operate the bot!**

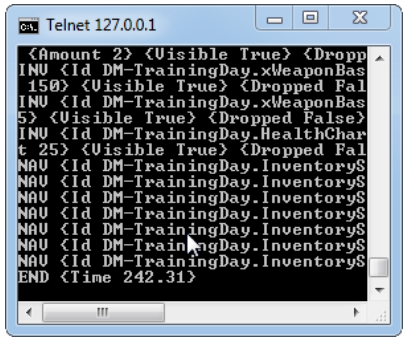
*UT2004/System/ucc.exe server DM-TrainingDay?game=GameBots2004.BotDeathMatch*

*telnet 127.0.0.1 3000*







# First Pogamut 3 Bot

## Show time!


- Let's import first Pogamut 3 example bot!
- Check the tutorial at home:
  - http://pogamut.cuni.cz/pogamut_files/latest/doc/tutorials/oo-EmptyBot.html

…

- See how easy is to code the bot! ☺

# Workshop website

## On Pogamut devel wiki…


- Visit workshop website
  - http://pogamut.cuni.cz/pogamut-devel/doku.php?id=lectures:pilsen_pogamut_2014-15_summer_semester
  - 1) Go to http://pogamut.cuni.cz
  - 2) Find a link to devel wiki (http://pogamut.cuni.cz/pogamut-devel)
  - 3) In main menu Click Lectures
  - 4) Find and click a link to this years Pilsen workshop website and …
- Get the First Empty Bot project template