"Let's Save Resources!" A Dynamic, Collaborative AI for a Multiplayer Environmental Awareness Game

Pedro Sequeira, Francisco S. Melo and Ana Paiva

pedro.sequeira@gaips.inesc-id.pt

IEEE CIG 2015 – September 2, 2015

This work was supported by national funds through the FCT with reference UID/CEC/50021/2013 and grants CMUP-ERI/HCI/0051/2013.

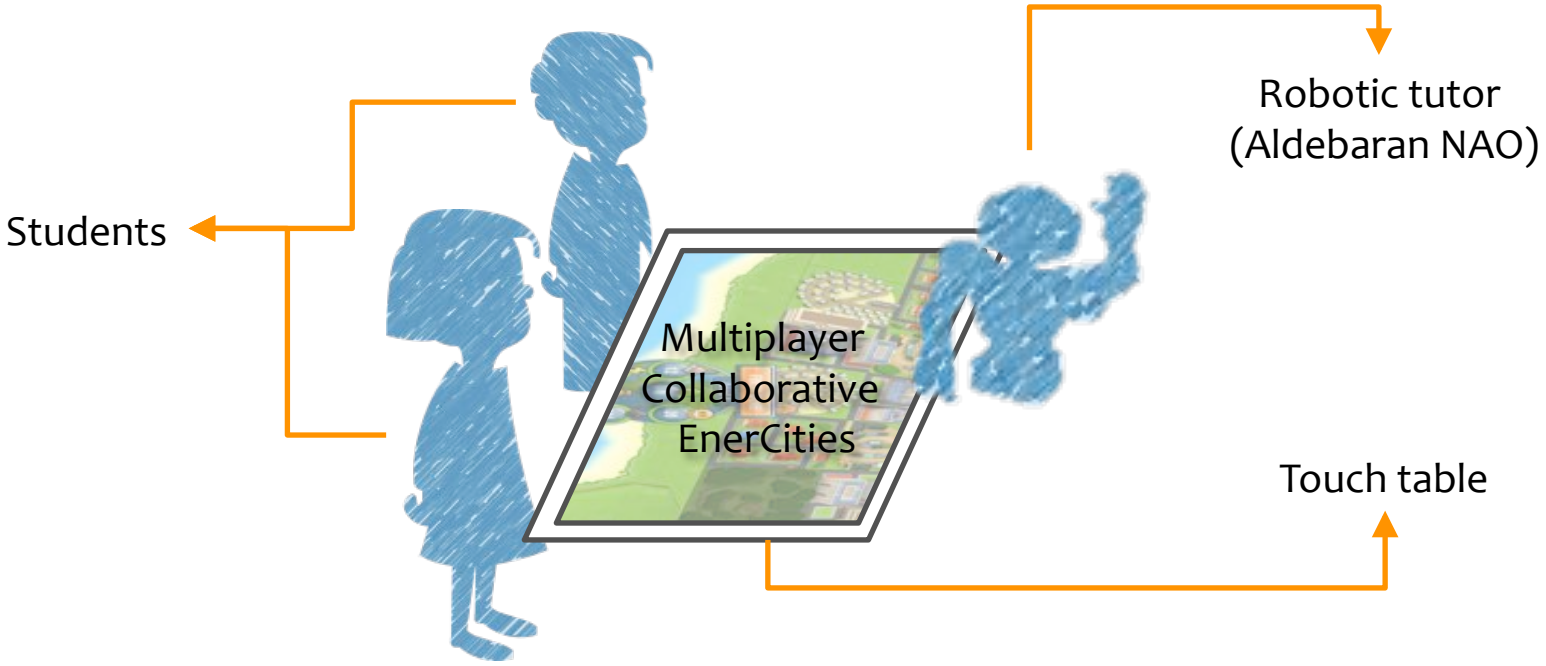
Introduction

EMOTE (1/2)

- EMbOdied-perceptive Tutors for Empathy-based learning
  - EU-FP7 project
  - http://www.emote-project.eu/
- Main objective
  - Artificial embodied **tutors**
  - Engage in **empathic** interactions
  - **Perceive** learning progress
  - **Intervene** in learning process

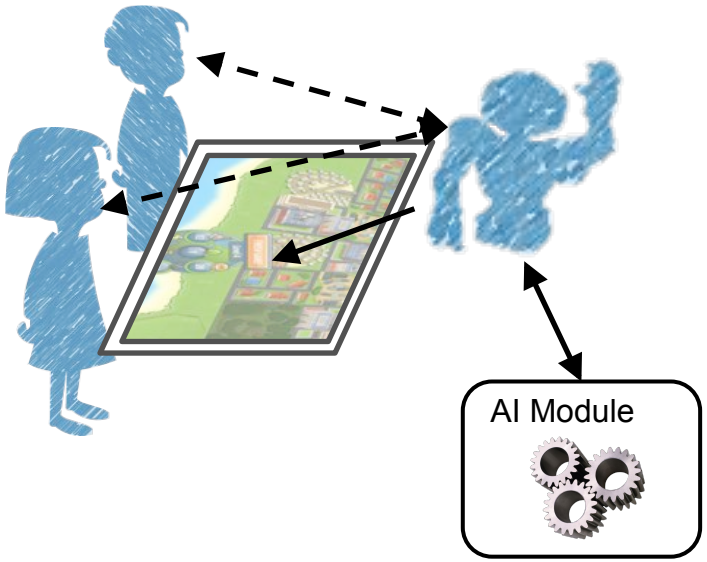
EMOTE (2/2)

- Collaborative learning scenario



MOTIVATION

- **AI module** objectives
  - **Control** tutor's actions
    - Contribute to **pedagogical** purpose
  - **Inform** robotic tutor
    - **Explain** tutor's decisions
    - **Focus** students' attention
    - **Influence** students' decisions
    - **Suggest** suitable actions
  - Contribute to **social, empathic** behavior
    - **Adapt** to group strategy
  - Interact in **timely** manner

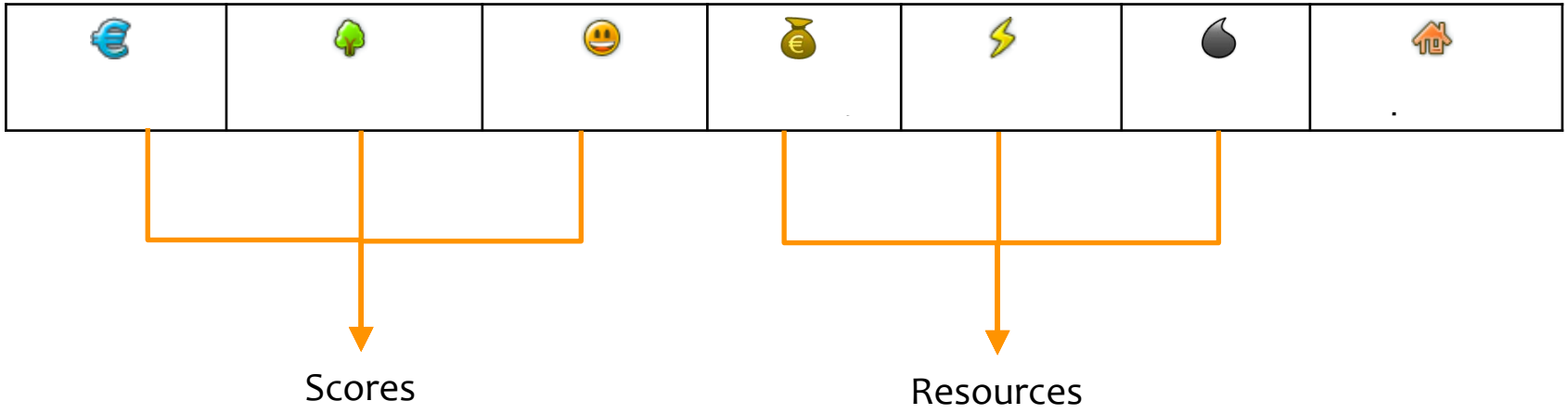


EnerCities (1/4)

- Original, single-player version
  - Developed in the context of EU project [Knol & De Vries, 2010]
  - Serious, turn-based game
- Objectives
  - **Raise awareness** on environmental issues
    - **Energy consumption** in a city
    - **Sustainable** city development
  - Each play influences the **city status**

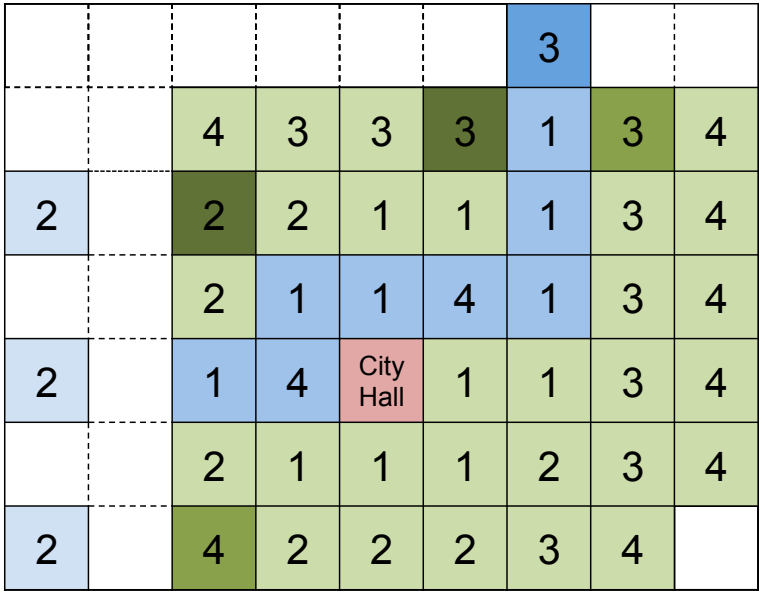
EnerCities (2/4)

- Game / city elements



EnerCities (3/4)

- Grid map
  - 45 distinct cells
    - Surface (land or water types)
    - Available in 4 game levels
  - 21 different structures
  - 56 distinct upgrades
    - 2.6 upgrades per structure avg.
- Actions at each turn
  - **Build** structure
  - **Upgrade** structure
  - **Implement** a policy
  - **Skip** turn



EnerCities (4/4)

- Dynamics
  - Contribution to scores and resources
    - Cell **surface**, **upgrades** performed, **neighbor** influence, **policies** implemented
    - Some **momentary** (buy a house), others **continual** (power spending)
  - **Level changes** when certain **population** amount is reached
  - **Game ends** when:
    - Natural resources **exhausted**
    - **Sustainable** city on **level 4** and **population level of 200**

Player's goal

M. C. EnerCities (MCEC)

- Multi-player version
  - Developed in the context of EMOTE [Ribeiro *et al.*, 2013]
  - 3 players / roles

Mayor

Economist

Environmentalist

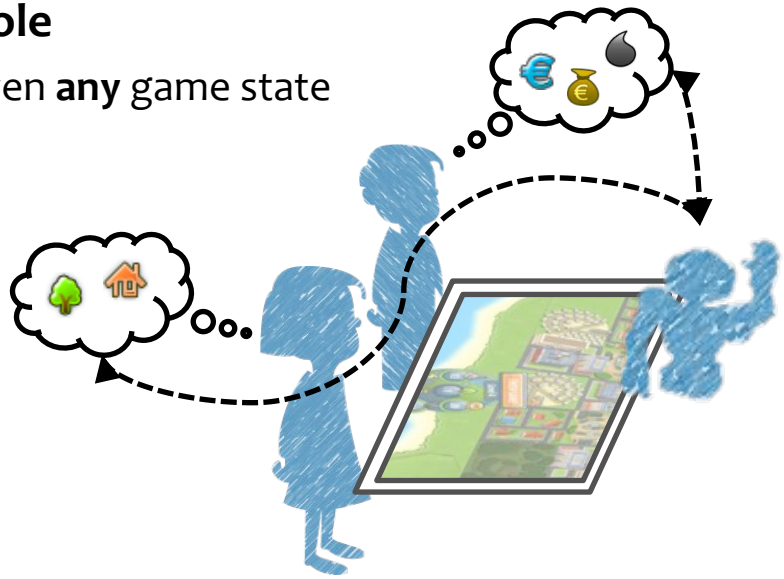
- Objectives
  - **Collaborative** game
    - Contribute to **sustainable** city
    - Each player contributes **differently**
  - Robotic **tutor** is active **player**

MCEC Challenges (1/2)

- Challenges associated with the game
  - No "global score"
    - **No actions maximize** all elements at the same time
  - Each player is **limited**
    - Actions affect **subset** of game elements
  - Each player influences many things
    - **Impact the future** states of the game
    - **Must not maximize** its score
  - **Collaboration** is necessary
    - Careful consideration to "win" the game

MCEC Challenges (2/2)

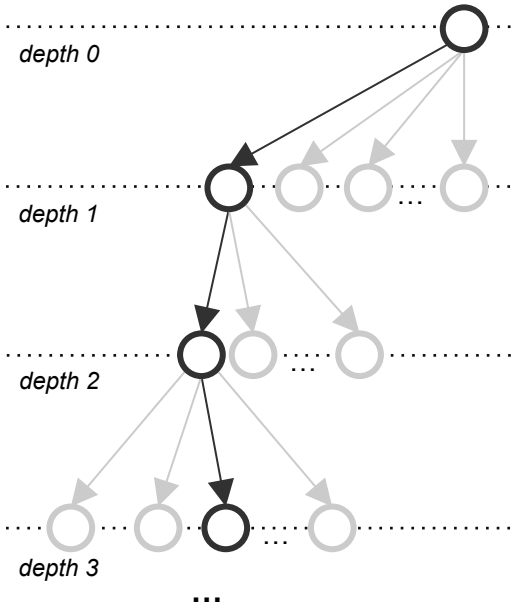
- Challenges associated with robotic tutor
  - Tutor must be able to play **any role**
    - Generate the **best** game plays given **any** game state
  - **Theory-of-mind** reasoning
    - **Predict** other players' moves
    - **Adapt** to "group" behavior
    - **Propose** alternatives
  - **Real-time** constraints
    - Make **timely** decisions
    - **Avoid breaks** in interaction flow



MCEC Analysis

- Complexity
  - Monte-Carlo analysis
    - 10^6 random-walks of MCEC
    - Random play order and actions
- Results

	Metric	Mean Value
	Game length	23.8 ± 16.4
Actions	Economist	113.3 ± 186.8
	Environmentalist	68.7 ± 100.2
	Mayor	83.4 ± 125.7
	Player avg.	88.4 ± 143.2



- **Complexity** of tree search: visit **2^{46} nodes**

AI Module

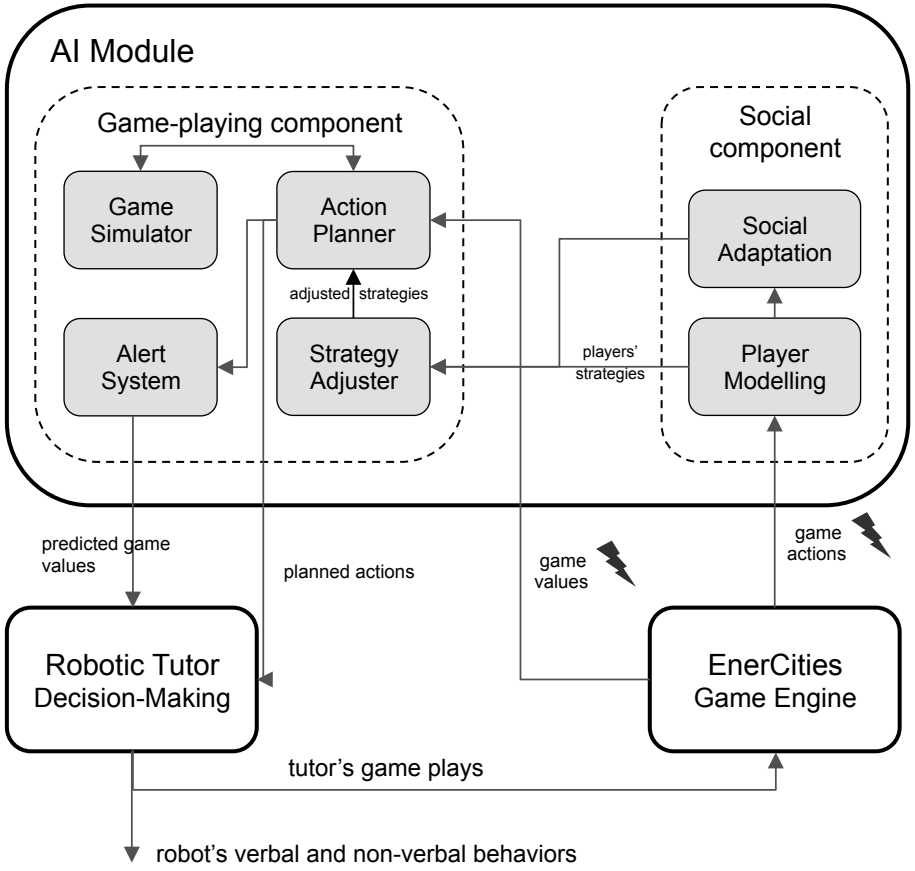
Game Strategy

- Idea
  - Problems
    - Reaching goal involves **several** game aspects
    - Robot's **strategy** depends on **context** and **pedagogical** factors
  - Solution
    - Strategy as **tendency** Θ to act upon **all city aspects**

  - Each weight asserts **importance** of factor
  - **Continuous space** of possible strategies

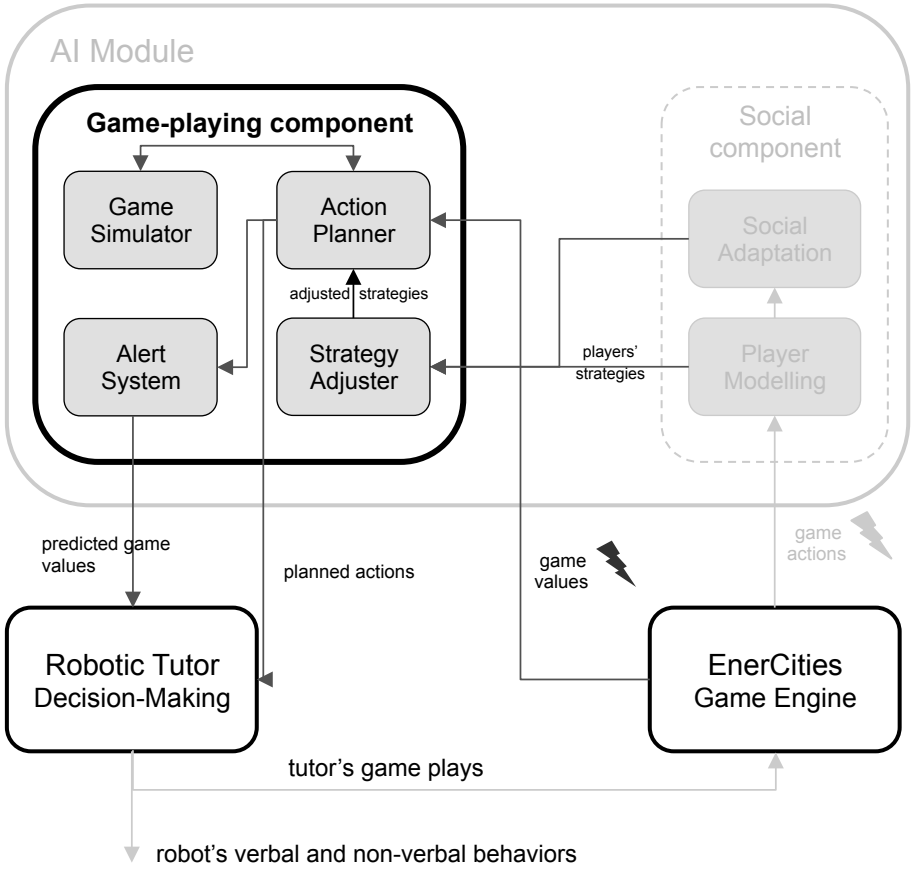
Architecture (1/3)

- 2 main components
  - Game-playing component
  - Social component
- Communication with
  - Robot's decision-making
  - Game engine



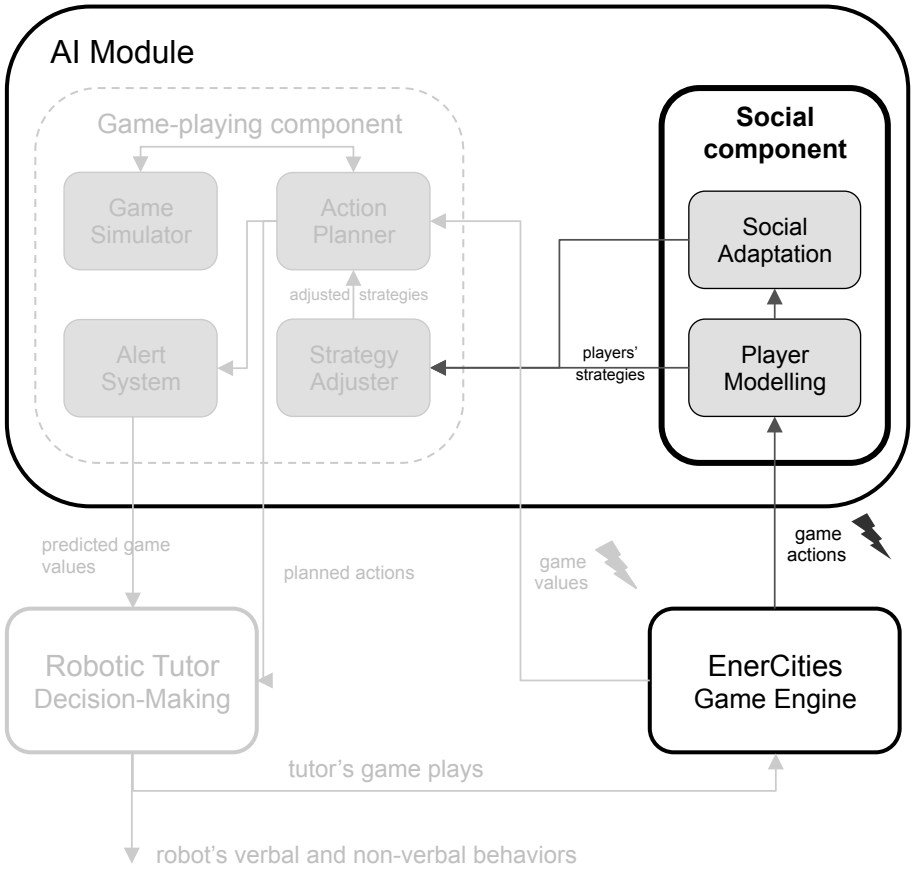
Architecture (2/3)

- Game-playing component
  - **Inform** decision-making
    - Predicted game values
    - Planned actions
  - Allow **expressive behavior**
    - **Verbal**, *e.g.*, inform and suggested game actions
    - **Non-verbal**, *e.g.*, point towards important game element
  - Game engine
    - Perform **actions**
    - **Collect information** on game values, players' moves, etc.



Architecture (3/3)

- Social component
  - Strategy **modelling**
    - Theory-of-mind
    - Collect players' actions information
  - Strategy **adjusting**
    - Adopt **group strategy**
    - Influence planning



Action Planner (1/4)

- Determine suitable game actions
  - Objective
    - Achieve **desirable** city status
    - **Maximize the gain** induced by some action
  - **Heuristic-based** search
  - Function of
    - Given **strategy**
    - Some **game state**

Action Planner (2/4)

- Action execution **gain** example
  - **Previous** game state, $\mathbf{s}$

2	2	2	90	2	1396	8	1

  - **Updated** game state after some action a, $T(\mathbf{s}, a)$

4	1	3	94	0	1396	8	0.7

  - **Normalized increment,** $Norm(T(\mathbf{s}, a) - \mathbf{s})$

0.2	-0.1	0.1	0.4	-0.2	0	0	0.9

  - **Strategy** used, Θ

0.5	-0.1	-0.1	0.3	0	0	0	0

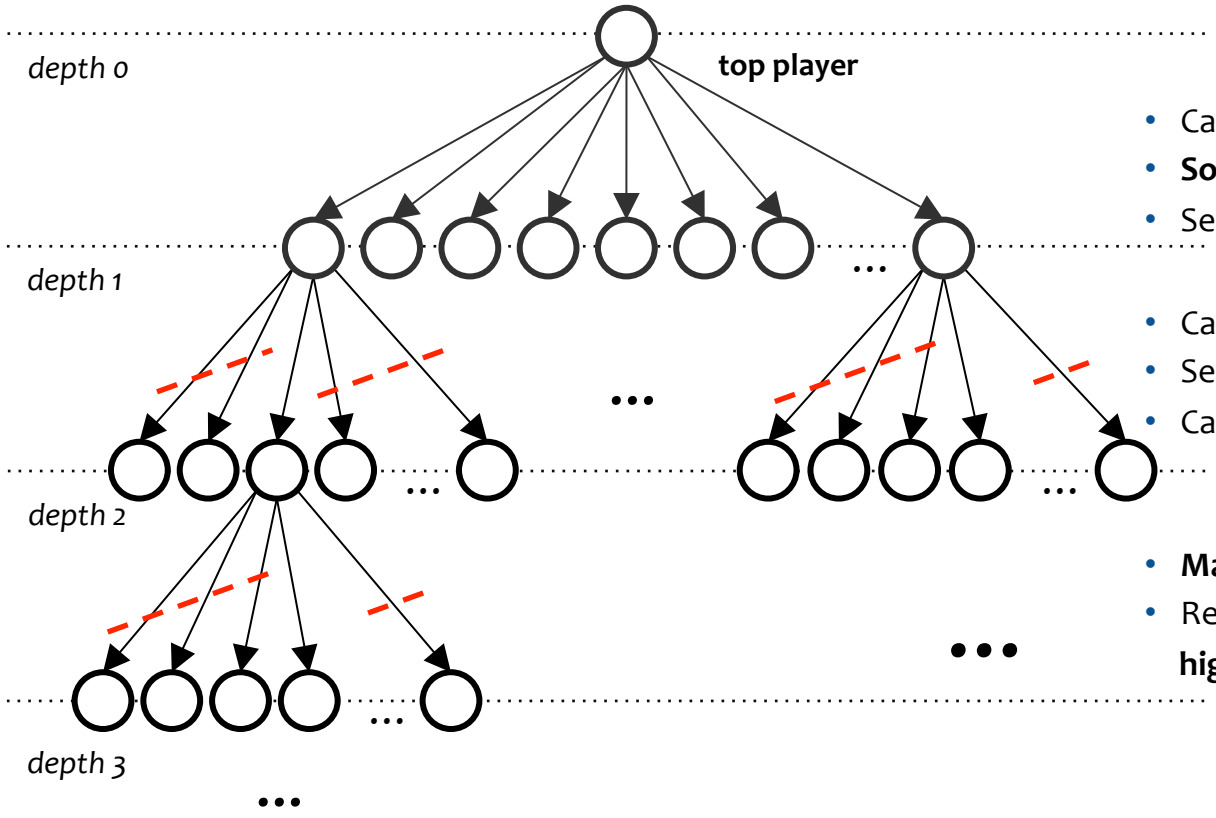
  - **Gain:** $\gamma(\mathbf{s}_t, \Theta, a) = Norm(T(\mathbf{s}, a) - \mathbf{s})^T \Theta = 0.22$

Action Planner (3/4)

- Planning algorithm
  - Objective
    - Search **optimal action**
  - Parameters
    - Player **role**
    - Current game values **state**
    - **Set of strategies** for all players
    - Maximum **depth**
    - Maximum **time**

Action Planner (4/4)

- Tree search visualization



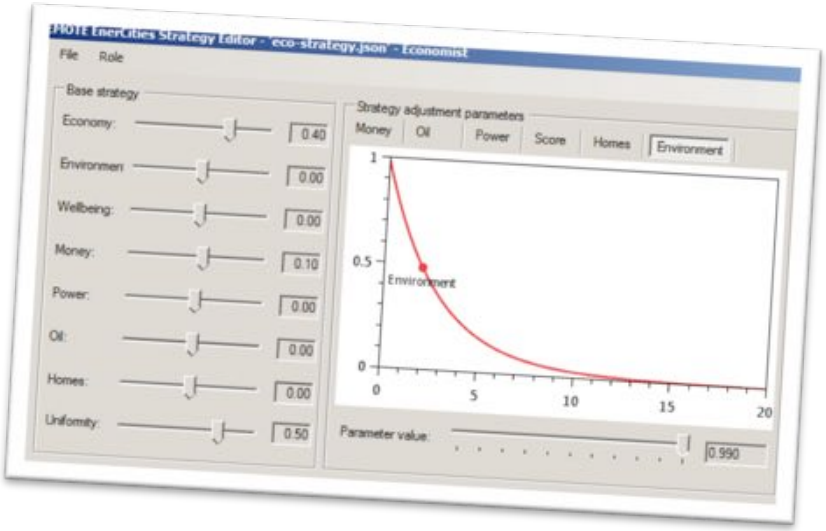
- Calculate gains for **top player**
- **Sort** descending
- Select next player

- Calculate gains for **d1 player**
- Select max, **prune** others
- Calculate gains for **top player**

- **Max** time or depth reached
- Return **top action** returning **highest future gain**

Strategy Adjuster

- Objective
  - Adjust tutor's strategy **online**
    - Strategy initialized according to some **profile**
  - **Hormonal-like** mechanism
    - **Modulates** behavior according to **city status**
  - Adjustment functions
    - Detect **below-threshold** game values
    - **Adjust** the importance of strategy component



Alert System

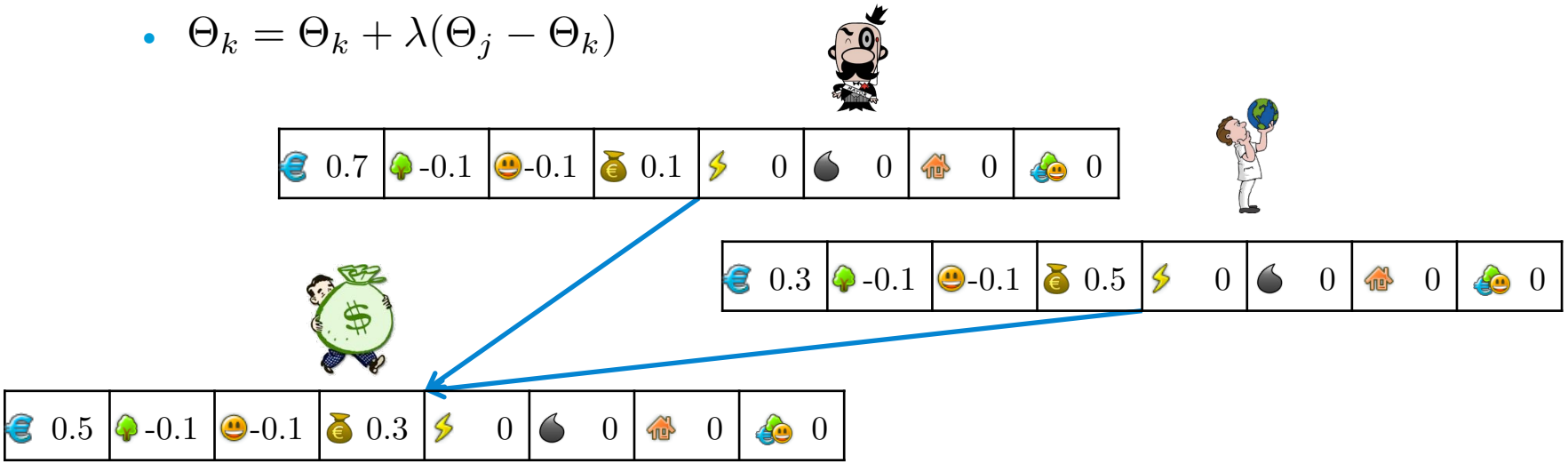
- Objective
  - Detect **"interesting"** game situations
  - **Inform** robot's decision-making
    - Tutor may intervene and *e.g.*, advise students
- Elements calculated during **planning**
  - **Predicted game values**
    - **Likelihood** of future city status
  - **No-action** probability
    - Reaching state where **Skip is only option**
  - **No-space** probability
    - Reaching state with **no cells** on which to **build**

Player Modelling

- Objective
  - **Model strategies** of human players
  - **Simplification:** players try to maximize **immediate** gains
- Procedure
  - Register **normalized increment**
    - $\Delta \mathbf{s} = Norm(T(\mathbf{s}, a_k) - \mathbf{s})$
  - **Update** player strategy **estimate**
    - $\Theta_k = \Theta_k + \alpha.\Delta \mathbf{s}$

Social Adaptation

- Objective
  - **Detect** group strategy
  - **Adapt** tutor's strategy accordingly
- Procedure
  - Approximate tutor's strategy
    - $\Theta_k = \Theta_k + \lambda(\Theta_j - \Theta_k)$



Conclusions

Summary (1/2)

- M. C. EnerCities
  - Multiplayer, **collaborative** game
  - Raise environmental awareness
- EMOTE project
  - **Social** and **pedagogical** interaction with humans
  - **Personalized** tutoring
- Influence tutor's behavior
  - **Explain** game actions taken
  - **Raise discussion** on interesting topics

Summary (2/2)

- AI Module
  - **Game-playing** component
    - **Simulate,** keep up-to-date information on city status
    - **Plan** actions for any role and state according to strategy
    - **Alert** about focus-demanding situations
  - **Social** component
    - **Theory-of-mind** mechanism
    - Learn **group strategy**
    - **Adapt** tutor's strategy accordingly

Example videos (1/2)

- "Balanced" strategies

Example videos (2/2)

- "Score-greedy" strategies

The End