

Goal Based Agents In A Competitive Environment

2006 CS310 Project
Luca F Beltrami

Agenda

- What is this project about
- Academic aims
- World Framework
- Spreading Activation Agent Theoretical Design
- Agent Implementation
- Model Shortcomings
- Workarounds

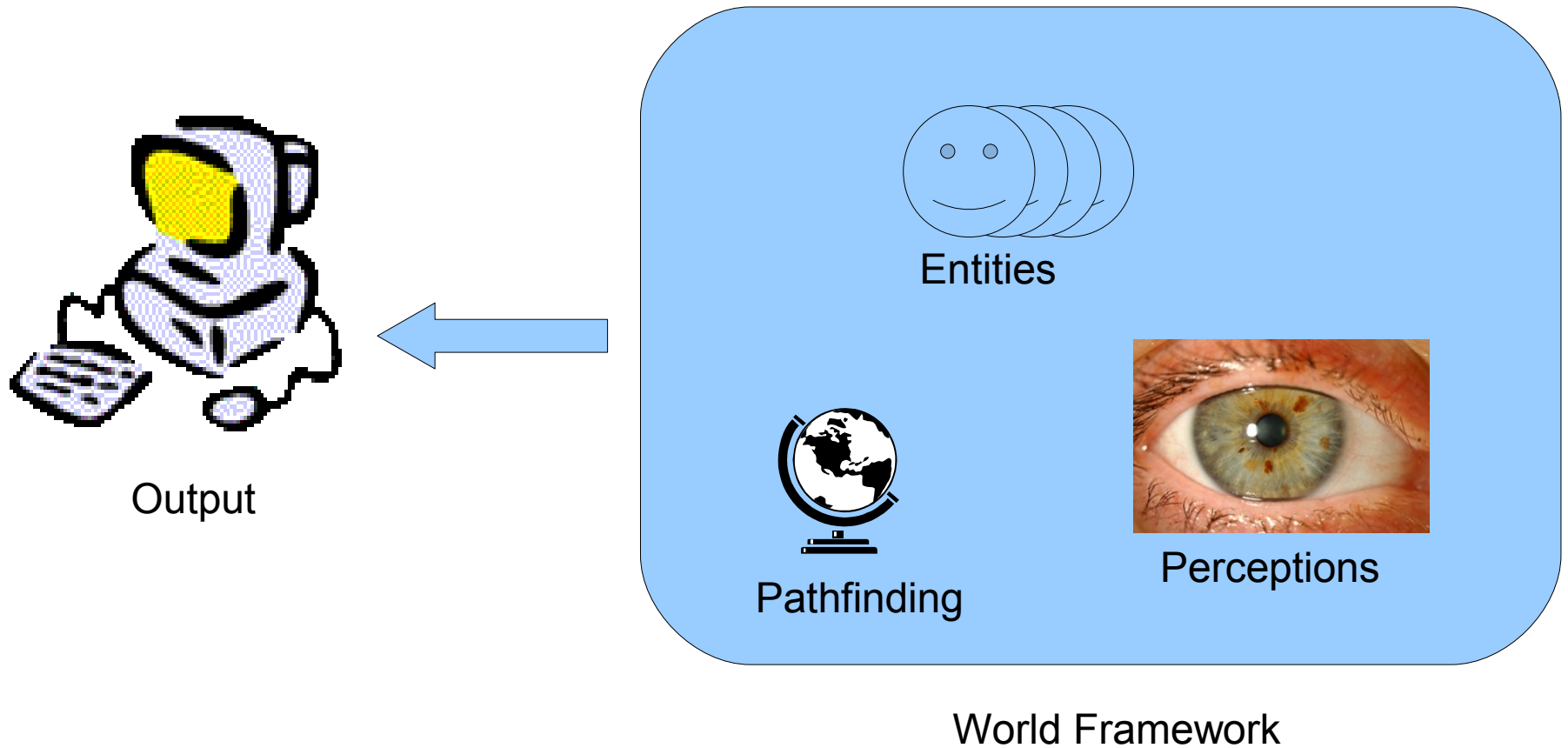
Project Description

- An implementation of Patti Maes' Spreading Activation agents
- A (simple) world where the agents can act
- A (very simple) graphical output module to show the state of the world

Academic Aims

- The basic Spreading Activation model makes a assumption about the world
- The paper describing the model leaves questions unanswered
- The project tries to adapt the model to a world where the restrictions are relaxed
- The project tries to fill in the gaps

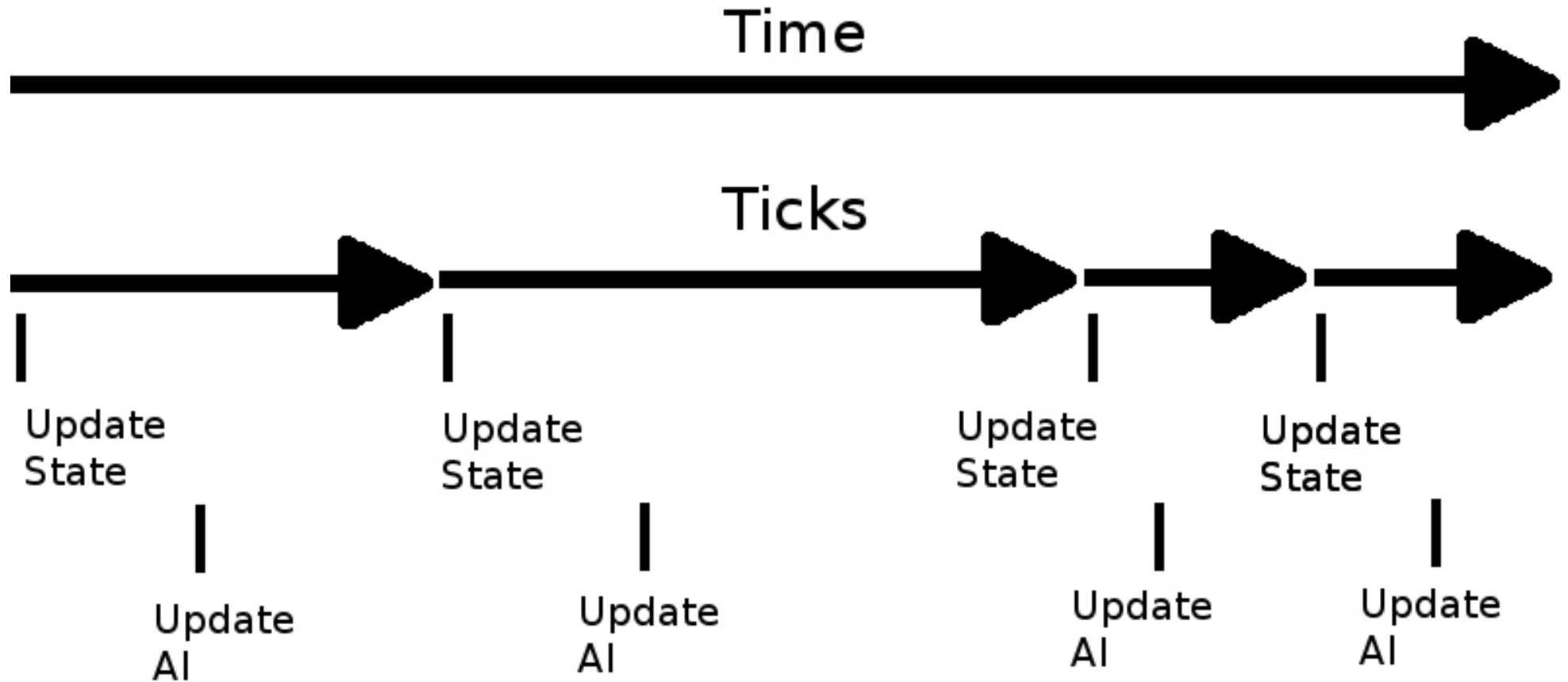
World Framework: Basic Design



World Framework: Entities

- An Entity is anything that exists in the world
- Basic entities only know of their position and name
- Entities can support extensions that let them do other things, such as combat, moving, being rendered by the output or supporting an agent

World Framework: Timing



World Framework: Graphical Output

- Uses OpenGL because I was familiar with it
- Very simple, only supports 2D world
- Theoretically the World / Output interface should allow many other output modules in place of this one

Agent / Entity Communication

- Entity to Agent – Percepts
 - Theoretically agent never access entity, only percepts from entity
 - In fact for performance and ease of coding this is often broken
- Agent to Entity – Orders
 - Agent ignores structure of world framework
 - Orders represent concrete commands
 - Allow for very loose coupling between agent and entity

The Theory

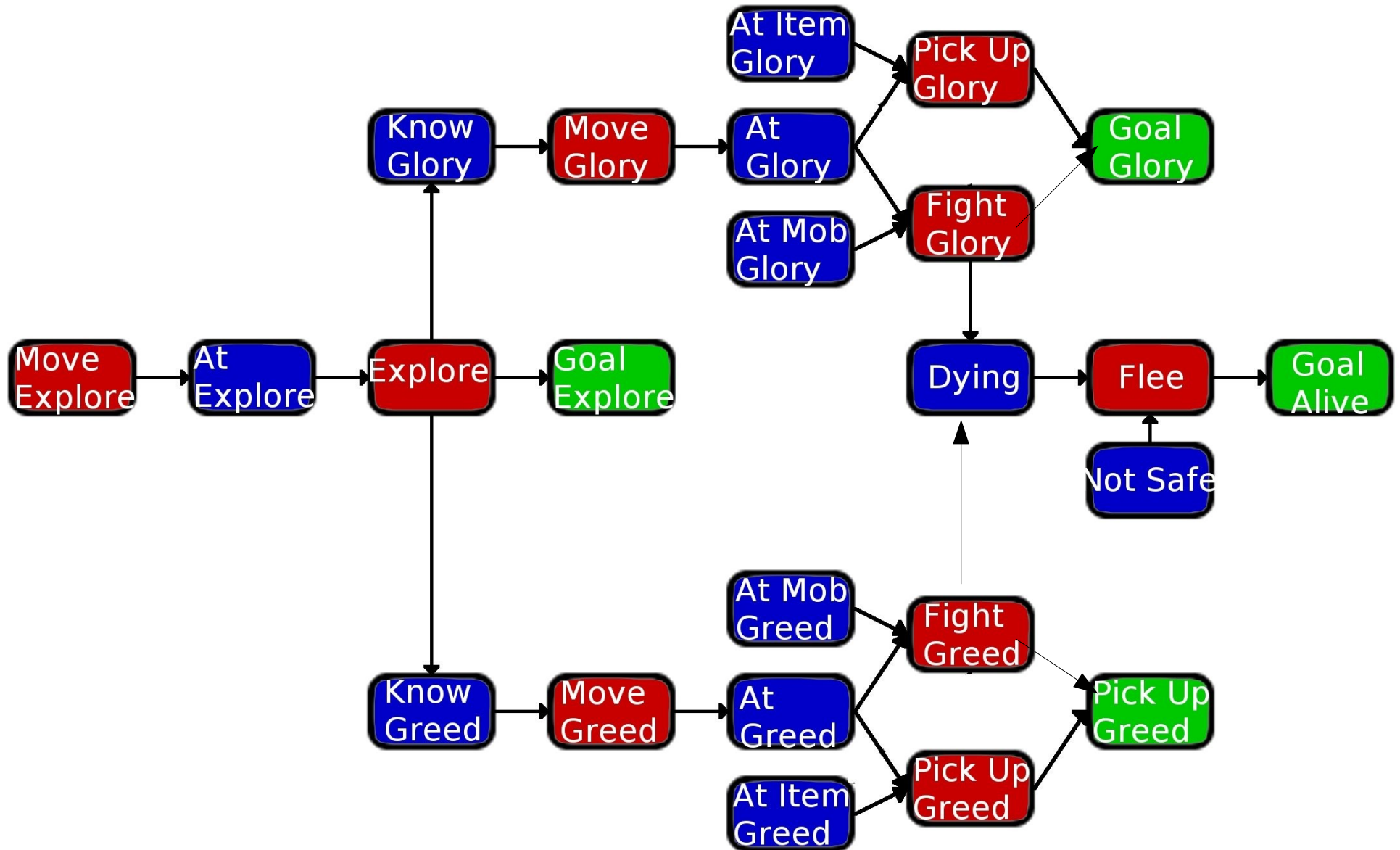
- Agents are networks of Competences
- Agent state is described by Symbols
- Each Competence requires certain Symbols to be true
- Each Competence has a set of Symbols it activates and a set of Symbols it deactivates after execution
- Competences execute when they have enough energy
- Symbols spread energy to Competences that can activate them or that require them
- Competences spread energy to prerequisite Competences

The Practice

- Implemented as a network
- Symbols and Competences are nodes
- Competences only link to Symbols and vice versa
- Each node encapsulates the appropriate energy spread algorithm



The Agent Network



Problems And Solutions:

Atomic Actions

- Atomic Actions: the model assumes actions taken by agents are atomic
- Fixed Network Structure: the network cannot be changed at runtime, making “learning” problematic
- No Goal priorities: all goals have the same importance
- No permanent Goals

Evaluation

- Still in progress
- Evaluation parameters:
 - Performance
 - Ease of coding
 - “Smartness”

Conclusions

- The main objectives of the project were achieved
- Energy Spread parameters require tweaking
- Some “quirks” left that require further study
- Extensions: grouping, better selection algorithms, inter agent communication...

And now onto the Demos

- Feel free to ask questions at any time
- Demo 1: just a bunch of agents in the world
- Demo 2: “stress test”
- Demo 3: selector quirks
- Demo 4: priorities
- Demo 5: combat