

Goal based agents in a competitive environment Project specifications.

20th October 2005

1 The problem

To design and code an AI system that allows agents to act based on given goals and the framework necessary to simulate the agents and the environment around them. The environment simulated in this project will be a fantasy setting where the agents are heroes, beasts and monsters but the design and functionality of the AI will not be tied to this specific setting. Ideally agents will be able to learn which activities further their goals the most and will act in groups when this is advantageous to them.

2 Technical objectives

2.1 Research

The main area of research for the project will be on Spreading Activation and Partial Programming based systems. Some research on AI in games and A* pathfinding will also be required.

2.2 World simulation framework

The world will be represented as a flat patch of terrain populated by entities. World coordinates will be expressed in a continuous floating point coordinate system. Time will also be assumed to flow continuously rather than in discrete steps.

2.2.1 Entity storage

Entities in the world must be stored so that entities in given areas (volumes) are easy and quick to retrieve. This is needed to efficiently implement perceptions among other things.

2.2.2 Entities

Entities represent the physical objects in the world. Entities might be either static (such as buildings, trees, etc) or movable (such as agents). In general any entity might be "driven" by an agent and will provide the agent with a set of methods to act on its decisions.

Entity movement Movable entities in the world will provide methods to move themselves. These methods should enforce whatever restrictions on movement the entity has (limit maximum movement speed, prevent vertical movement for entities that are walking, etc.)

Entity interaction Entities must provide methods that implement any interaction with other entities they are capable of (speech?, combat)

Entity state An entity is responsible for keeping track of its internal state (for example health) and must provide methods for the agent to inspect this state.

Entity perception Different from interaction in that perceptions are passive while interactions are active. An entity will provide methods to determine what the entity is currently perceiving with each of its senses.

2.2.3 Graphical output

A simple OpenGL based output system. Each entity will store a reference to a "model" that will represent the entity on screen. A single model instance might be shared by multiple entities and each should be able to customise the rendering of the model so that it represents selected parts of the internal state. The output will use a fixed camera angle (vertically looking down) and might allow limited movement and zoom functionality.

2.2.4 Framework testing

Before proceeding further the entire framework will be extensively tested since its functionality are fundamental to the rest of the project. Every component of the framework will undergo unit testing as soon as it is written and integration testing will be performed as soon as possible.

2.3 Pathfinding

Pathfinding will be implemented as a separate module from the rest of the AI since it is "lower level" functionality and might require tighter integration with the world framework.

2.3.1 Algorithm

Pathfinding will be implemented using the A* algorithm. Since this algorithm requires a quantized world to work correctly the pathfinding module will maintain a grid based representation of the world that will store navigation data.

2.3.2 OPTIONAL Pathfinding improvements

Adaptive subdivision To improve performance and path detection the "passability grid" (the data structure that stores the navigational data) could use different levels of subdivision depending on the density of objects in the area. This would greatly reduce the number of cells needed to describe wide open areas and possibly allow the detection of additional paths in very dense areas.

"Island" detection To speed up detection of unreachable areas the "passability grid" should store which cells are reachable from other cells (for example by assigning an id number to each "island").

2.3.3 Pathfinding testing

Testing on the pathfinding module will be carried out much in the same way as for the world framework. A very simple AI might be implemented to drive the entities around the map.

2.4 AI

The AI will be implemented as a number of independent agents acting independently from each other. Each agent will be associated with a single entity and will perceive the world only through that entity senses.

2.4.1 Simple comparison between AI models

Basic implementations of Spreading Activation based AI and Partial Planning based AI will be compared to decide which one (or combination of the 2) is best suited to the goal.

Spreading Activation AI implementation Basic implementation of AI based on the Spreading Activation model by Patti Maes

Partial Planning AI implementation Basic implementation of AI based on partial planning methods.

Evaluation of the comparison The evaluation will focus on which AI implementation best allows a straightforward implementation of the desired behaviours, has rapid response times and uses less system resources.

2.4.2 Full AI implementation

The AI will be fully implemented using whatever technique was deemed best in the previous stage.

Full AI Design The AI system design will be finalised, using either the Spreading Activation model or Partial planning or a combination of both possibly in a multilayer architecture.

Implementation Multiple agent archetypes will be implemented, each with different goals and priorities (for example a warrior archetype might prioritize hunting big monsters for the "glory" while a wild animal archetype will prioritize survival).

2.4.3 AI testing

At this point a round of testing will be carried out to ensure that the AI implementation is correct from a purely "is the code (reasonably) bug free" point of view.

2.4.4 AI assessment

A number of agents will be placed in the world and observed. The assessment will determine if the agents are actually doing what their archetype "likes and dislikes" would suggest and if not why they aren't.

2.4.5 OPTIONAL Grouping

Implement and asses the ability for agents to group together to further their goals. Agents in a group should act in a reasonably coordinated manner and break away from the group if it is not actually better than acting alone (in the agent's opinion).

2.4.6 OPTIONAL Learning

Implement and asses limited learning functionality, specifically the ability to tell what kind of behaviour furthers the agent's goal the most in the given environment.

3 Methods

The development method will in general use a brief design phase before the start of the implementation. Additional design will be carried out during the implementation phase and extensive refractoring will be used if needed.

4 Timetable

Note that the workload due to modules in term 2 is roughly half of that in term 1

Week	Objectives
Term 1	
Week 3	Specifications
Week 4	Research
Week 5	Research, World framework design
Week 6	Research, Graphical output for world framework
Week 7	Entity storage for world framework
Week 8	Entities for world framework
Week 9	Entities for world framework, Pathfinding
Week 10	Pathfinding
Christmas Vacation	World framework and Pathfinding testing
Term 2	
Week 1	Basic AI implementation for comparison
Week 2	Basic AI implementation for comparison, evaluation
Week 3	Full AI design
Week 4	Full AI implementation
Week 5	Full AI implementation
Week 6	AI assessment
Week 7	Optional Objectives
Week 8	Optional Objectives, Report draft
Week 9	Optional Objectives
Week 10	Optional Objectives

5 Resources

gcc c++ compiler 3.3+ with libraries

OpenGL 1.4+

SDL 1.2.8+

Scons 0.96+