Combining Influence Maps and Cellular Automata for Reactive Game Agents

Penelope Sweetser and Janet Wiles

School of ITEE, The University of Queensland, St Lucia 4072, Brisbane, Australia
{penny, j.wiles}@itee.uq.edu.au
http://www.itee.uq.edu.au/~penny

Abstract. Agents make up an important part of game worlds, ranging from the characters and monsters that live in the world to the armies that the player controls. Despite their importance, agents in current games rarely display an awareness of their environment or react appropriately, which severely detracts from the believability of the game. Some games have included agents with a basic awareness of other agents, but they are still unaware of important game events or environmental conditions. This paper presents an agent design we have developed, which combines cellular automata for environmental modeling with influence maps for agent decision-making. The agents were implemented into a 3D game environment we have developed, the EmerGEnT system, and tuned through three experiments. The result is simple, flexible game agents that are able to respond to natural phenomena (e.g. rain or fire), while pursuing a goal.

1 Introduction

Agents are an important part of game environments as they give the game life, story and atmosphere. Agents serve many different purposes and hold many different positions in games, which contributes to making the game world rich, interesting and complex. For example, strategy games include units (e.g. marines) that the player controls and role-playing games include agents that fill a wide range of different roles in society, from kings to cobolds. Game-players expect agents to behave intelligently by being cunning, flexible, unpredictable, challenging to play against and able to adapt and vary their strategies and responses [5]. However, players often find that agents in games are unintelligent and predictable [5]. Furthermore, players believe that agents' actions and reactions in games should demonstrate an awareness of events in their immediate surroundings [1]. However, many games are proliferated with agents that do not demonstrate even a basic awareness of the situation around them. These agents often occupy the landscape as glorified pieces of scenery and behave in exactly the same way in any number of situations, ranging from rain to open gun fire.

The question that arises is how can game agents be made to appear intelligent to the player by reacting sensibly to the game environment? First, we review various techniques that can be used for agents in games and identify influence maps as a potential solution. Then we describe a study in which we designed, implemented and tested reactive agents in the Emergent Games Engine Technology (EmerGEnT) system, a system we have developed that is based on cellular automata. Three structured experiments were conducted with the reactive agents in the EmerGEnT system to determine the design that would achieve the most appropriate agent behaviour, as indicated by criteria for efficiency, effectiveness and visible behaviour. The aim of the study was to assess the extent to which agents based on influence maps and cellular automata can exhibit behaviour that is appropriate, intelligent and realistic.

2 Reactive Agents in Current Games

The agents in most games are hard-coded, relying heavily on the prior knowledge of their designers and little on their current situation. Furthermore, many agents in current games simply do not react to the environment in any way. There are some games, however, in which the agents demonstrate situational awareness by actively sensing and reacting to other agents in their environment. For example, the agents in *Half-Life* and *Thief: The Dark Project* have sight and hearing and periodically "look" at and "listen" to the world [2]. However, the agents in these games are still hard-coded, as they periodically run through a list of rules to determine whether they sense an opponent. Also, as these agents must actively check to determine whether they can sense something periodically (real vision and hearing arrive at the senses continuously [2]), it is likely that events and actions will be missed.

Another game that requires the agents to sense and react to information in the environment is *The Sims*. Unlike *Half-Life* and *Thief*, the agents in *The Sims* constantly receive information from the environment. In *The Sims*, the AI is embedded in the objects in the environment, known as "Smart Terrain". Each agent has various motivations and needs and each object in the terrain broadcasts how it can satisfy those needs [7]. The agents in *The Sims* are not hard-coded like the agents in *Half-Life* and *Thief*. Instead, their behaviour is autonomous and emergent, based on their current needs and their environment.

Although the agents in each of these games are able to sense entities in the environment in some way, they are still unable to sense the state of the environment itself. The agents in *Thief* and *Half-Life* are limited to sensing other agents in the environment and the agents in *The Sims* are limited to sensing other agents and objects in the environment. These agents would still be unable to react to events and states of the environment such as rain, fire, gunfire and so on. Another approach that is more applicable to the problem of agents reacting to the game environment is a technique used in many strategy games, influence maps.

3 Influence Maps

Influence maps divide the game map into a grid of cells, with multiple layers of cells that each contain different information about the game world (see [3,6]). The values for each cell in each layer are first calculated based on the current state of the game and then the values are propagated to nearby cells, spreading the influence of each cell. Currently, influence maps are used in games for strategic, high-level decision-making. However, it would also be possible to use them for tactical, low-level decision-making, such as individual agents or units reacting to the environment.

The advantage of influence maps over methods that are currently used in games, such as Smart Terrain in *The Sims*, is that the agent is presented with a single value (calculated using the weighted sum to combine all the factors) instead of numerous messages being sent to the agent about the environment. Also, this approach has further advantages over the method used in games such as *Half-Life* and *Thief* as the agent is continuously adapting its behaviour to the environment (rather than probing at given time intervals) and its behaviour is a function of its environment (rather than following a prescribed set of rules). Finally, the influence map structure fits nicely with the cellular automata that are already being used to model the environment in the EmerGEnT system (see Sect. 4). Both use the same data structure and the raw values for the influence map are supplied by the calculations of the cellular automata. Therefore, the approach of using an influence map for tactical decision-making is investigated in this study as it accommodates passive sensing of a continuous environment (as opposed to discrete entities), allows the agents' situational awareness to evolve as a function of the environment, gives rise to reactive and emergent behaviour and combines well with the cellular automata model of the environment in the EmerGEnT system.

4 The EmerGEnT System

The Emergent Games Engine Technology (EmerGEnT) system is a 3D game world we have developed that models natural phenomena, such as fluid flow, heat, fire, pressure and explosions. The system is based on cellular automata, which divide the game world into a grid of cells and contain rules for how the cells interact. Each cell has a set of variables, including height, heat, pressure, fluid and terrain. The rules of the cellular automata are loosely based on thermodynamics and physics and use the properties of each cell to determine how the cells will exchange heat, pressure and fluid, which gives rise to explosions, fire and floods. The EmerGEnT world also includes game objects (e.g. buildings) and game agents (e.g. villagers) that have similar properties to the cells of the environment, which determine how the objects and agents act and interact in the game world.

5 Agent Design

For a game agent to react sensibly to the environment it must have two things: a way to sense the environment and a way to choose a suitable reaction, based on what it has sensed. An agent's understanding of its situation in the EmerGEnT system is represented as a weighted sum of the factors affecting each cell on the map. Based on the utility value of each cell, the agent chooses a cell to move to and reacts at a level that reflects its current situation (e.g. if the agent's current cell is on fire then it panics). After the agent chooses a destination, its task is simply to move towards it. This section discusses the "comfort" function that determines the utility of each cell, the agent's level of reaction and the agent's choice of destination cell.

The utility function for the agents in the EmerGEnT system determines how comfortable each cell is for the agents and is therefore called a *comfort* function. The comfort function is a weighted sum of the factors that affect the agents' comfort in each cell and includes *fire*, *heat*, *pressure* and *wetness*. Each of these factors is weighted according to how distressing it is for the agent. Fire is the most distressing, followed by temperature, pressure and wetness. The weights (W_1, W_2, W_3, W_4) can be tuned to reflect different priorities of different agents. For example, an alien might find water far more dangerous than heat. The comfort function returns a real value between zero and one, with a lower value representing a more comfortable cell.

$$Comfort = \mathrm{Min}\,((fire{*}W_1) + (heat{*}W_2) + (pressure{*}W_3) + (wetness{*}W_4),\ 1)$$

The comfort function provides an efficient alternative to the environment sending the agent multiple messages about its state, such as "it's hot" or "it's raining". Instead, the relevant factors are weighted and combined into a single value that gives the agent an estimate of the safety and comfort of its current location. The purpose of the comfort value is twofold. First, it provides a means for the agents to determine how comfortable they are in the current cell and to react accordingly. Second, it provides a means for the agents to assess surrounding cells and find a suitable destination. These two tasks are discussed in this section.

The comfort function returns a real value which allows the agent to react with varying degrees of distress, providing for diverse and interesting behaviour (see Table 1). The agent's level of reaction is denoted by its speed of movement, as well as its animation and sound. Scaling the agents' reactions allows the agents to react in varying ways to different situations, while greatly simplifying the process of determining how the agents will react. Instead of the agents considering each element in the environment individually, the comfort function determines the agents' level of discomfort and the agents respond accordingly by choosing the reaction level that corresponds to their comfort value.

Table 1. Agent reaction levels. Agents react with varying degrees of distress to provide more diverse behaviour

Value	*Level*	*Reaction*
< 0.1	comfortable	no reaction
0.1 - 0.3	uncomfortable	calmly moves to more comfortable cell
0.3 - 0.6	distressing	runs from the cell
> 0.6	painful	panics and runs quickly from cell

If the agents are not comfortable in their current cell then they must locate and move to a more comfortable cell. Each agent reassesses its situation each timestep, by calculating the comfort value for the cell it is standing in or passing through and finding a destination cell based on the comfort of its neighbour cells. As long as the agent is not comfortable, it will keep reassessing its situation and finding a new destination, which means that agents can change destination while they are moving towards their current destination, if they find a better destination. Also, as the state of the environment is continuously changing, the destination the agent found last cycle may no longer be a comfortable cell. In choosing a destination, the agents evaluate the comfort values of the cells in a neighbourhood of a given size and choose the cell with the lowest comfort value.

6 Agent Experiments

Three experiments were conducted to investigate and tune the behaviour of the agents in the EmerGEnT system, in terms of efficiency, effectiveness and observable behaviour. Several conditions were investigated in each experiment and ten trials with ten agents were run in each condition. The criteria that were used to evaluate the performance of the agents included whether or not the agents converged on a solution (i.e. agents located and reached comfortable cells), the number of cycles the EmerGEnT system ran before the agents converged, how efficiently the agents found a solution, what (if any) strategies or patterns agents exhibited and the number of local optima (comfortable cells) on which the agents converged. The initial state of each trial was randomly generated, including the position of the agents, the position of rain and the number and position of explosions (see Fig. 1). See [4] for detailed experiments.

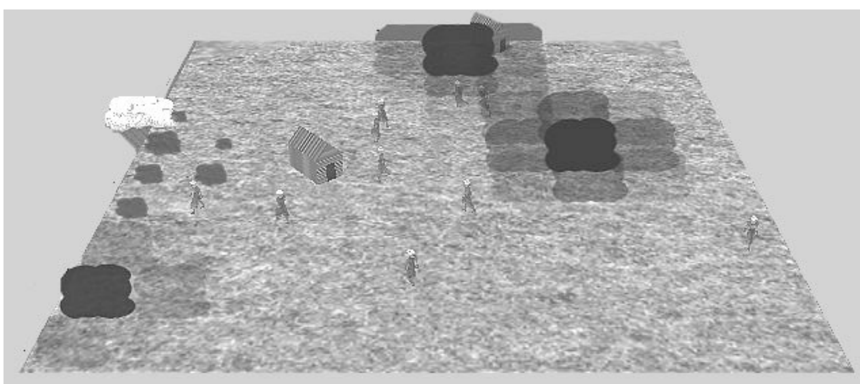

Fig. 1. The initial state of each trial was randomly generated, including position of agents, rain and explosions.

The aim of the first experiment was to determine the optimal neighbourhood size that agents should evaluate when choosing a destination (i.e. where to move to maximise comfort). Three conditions were tested, in which the agents evaluated neighbourhoods with a radius or one, two or three. Agents with each of these neighbourhood sizes demonstrated various advantages and drawbacks. The agents with a neighbourhood size of one performed the best at avoiding immediate danger. However, their short sight meant that they often ran towards more dangerous situations or became stuck in larger hazards as they were unable to find a way out. With a neighbourhood size of two, the agents were better at choosing safe destinations and appeared more organised, but still expressed the problems associated with short sight. The agents with a neighbourhood size of three were exceptional at picking particularly desirable cells and appeared organized, as many agents moved to similar locations. However, the problems for these agents were almost the opposite of the previous agents, as they performed the best at choosing a destination but were unable to avoid immediate hazards in getting to their destination. They would often put themselves in great danger (e.g. run through fire) to get to a safe destination cell.

From the first experiment, it was concluded that it would be desirable to combine the ability to find local optima of the longer-sighted agents with the ability to avoid immediate threats of the short-sighted agents. Consequently, the second experiment investigated whether a combination of immediate area (reactive) evaluation and greater area (goal) evaluation is more effective than either approach individually. The aim of the second experiment was to determine what combination of reactive (neighbourhood size = 1) and goal (neighbourhood size = 3) evaluation gave rise to the best agent behaviour. Three conditions were tested: evenly-weighted, goal-directed and reactive.

The agents in the second experiment displayed definite advantages over the agents in the first experiment. The agents in the evenly weighted and goal-directed conditions appeared far more intelligent, as they moved towards a goal rather than running back and forth randomly. Also, these agents appeared more realistic, as they moved around hazards on the way to their goal rather than simply running in a straight line, which made the agents in the previous experiment appear very flat and synthetic. Also, the agents in the evenly weighted condition displayed more depth as they did not always react in the same way, sometimes they would appear organised and at other times they would appear more independent, with their behaviour being heavily dependent on the current situation. The agents in the evenly weighted condition took the least amount of time to converge on safe cells. The agents in the goal-directed condition behaved in a similar way to the agents in the evenly weighted condition, but became stuck more often and still ran through hazards. The agents in the reactive condition had the least desirable behaviour as they often appeared to move randomly, did not appear organised and often became stuck. Therefore, it was concluded from the second experiment that the most suitable combination of reactive and goal-directed behaviour for the agents in the EmerGEnT system is approximately equal, where it is more desirable to err on the side of goal-directed than on reactive behaviour.

The first and second experiments gave rise to agents that efficiently, intelligently and realistically react to the environment by moving from danger to safety. However, in a computer game situation, it is also likely that agents will have greater goals or desires that they need to fulfil, apart from simply surviving and reacting sensibly to the environment. For example, marines in a strategy game might be on a mission to kill the enemy in a particular cell or a villager in a role-playing game might want to stay near its house or shop. Drawing on the notion of "desirability" values from influence maps, goal areas could be given high desirability values for the agents. Additionally, desirability values could then be propagated out to surrounding areas to indicate that these areas are more desirable as they are near the goal. Therefore, the aim of the third experiment was to combine the desire to reach a greater goal with the agents' current behaviour of reacting to the environment and avoiding hazards. The third experiment combined an influence map to propagate the desirability of the cells with the cellular automata to determine the comfort of the cells. The three conditions that were investigated in the third experiment were designed to test different influences of comfort and desirability on the agent's choice. The three conditions were evenly weighted, goal-oriented, and self-preserving.

The third experiment demonstrated that an equal weighting of desirability and comfort gave the agents the most acceptable observable behaviour, in terms of organisation, avoiding hazards and navigating the environment realistically and intelligently. When the weighting was tipped towards either comfort or desirability, the agents' behaviour appeared random, less organised and less intelligent. Only about half the equal-weighting agents found the goal as they opted for comfort over the goal. It is difficult to judge this as a success or a failure without a context for the agents. For example, it would be reasonable for villagers to prioritise their safety over achieving a specific goal, but marines would be expected to carry out the player's orders. It was concluded that the success of the agents must be judged with respect to the game situation, as different game types and scenarios have different requirements for successful agent behaviour. In general, agents should be able to reach their goal, while displaying appropriate behaviour (e.g. avoiding danger), but the relative importance of each of these aspects would be determined by the game situation. The third experiment produced an agent model that successfully integrates goal-directed behaviour (based on agent desires) with situation awareness (based on comfort), which enabled the agents to both react to the environment in an intelligent, realistic and organised way while simultaneously satisfying their desire to reach a goal.

7 Discussion and Conclusions

The outcome of the first two experiments was a model for agents that dynamically respond to the environment in an intelligent and realistic way, based on concepts from cellular automata and influence maps. The outcome of the third experiment was an extension of this model that also integrates goal-directed behaviour to enable the agents to respond to the environment while pursuing a goal. An advantage of the model developed through these experiments is extensibility, in that it can be extended to incorporate any aspects in the game world that are relevant to the agents' behaviour (e.g. other agents, terrain, events). It would also be possible to incorporate other models for agent behaviour, such as flocking, so that the agents also take into consideration the movement of other agents around them. The simplicity and flexibility of this model means that it can be used to govern the behaviour of almost any agent in any circumstance. The contribution of this research is a design that allows agents to dynamically react to the changing situation of their environment, as well as an intelligent pathfinding algorithm that allows agents to find a safe path to a goal, based on aspects of their environment. The agent parameters were tuned through the experiments discussed, but future work will be required to optimise their behaviour for specific game situations.

In conclusion, this research provides a possible solution for incorporating agents that appear intelligent to the player, by reacting sensibly to the game environment, into game worlds. First, reactive agents can be incorporated into game worlds by giving the agents a measure of comfort in their current situation (via cellular automata or other means), as well as a map for deciding where they might move to maximise their comfort. As this design closely resembles an influence map, it is also possible to integrate goal-directed behaviour and potentially personality, group movement and various other behaviours into the agent model. Whereas current agents in games do not demonstrate an awareness of their situation or react appropriately to events in their immediate surroundings, the reactive agents presented in this paper maintain a model of the comfort of their environment and react according to the changing state of their situation. The reactive agent model developed in this study allows agents to dynamically react to the changing situation of their environment and to intelligently find a path to a goal, increasing their visible level of intelligent, realistic and responsive behaviour.

References

1. Drennan, P., Viller, S. and Wyeth, P.: Engaging Game Characters: Informing Design with Player Perspectives. Entertainment Computing - ICEC 2004: Third International Conference, Lecture Notes in Computer Science, Vol. 3166. Springer-Verlag, Berlin Heidelberg New York (2004) 355-358
2. Leonard, T.: Building an AI Sensory System: Examining the Design of Thief: The Dark Project. Gamasutra (March 7, 2003). Available online 2 February, 2005, at http://www.gamasutra.com/gdc2003/features/20030307/leonard_pfv.htm (2003)
3. Sweetser, P.: Strategic Decision-Making with Neural Networks and Influence Maps. In: S. Rabin (ed.): AI Game Programming Wisdom 2. Charles River Media, Inc., Hingham, MA (2004) 439-446
4. Sweetser, P.: PhD Thesis Chapter 6: Reactive Agents. In: An Emergent Approach to Game Design – Development and Play (in preparation). Available online, 25 April, at http://www.itee.uq.edu.au/~penny/publications.htm
5. Sweetser, P., Johnson, D., Sweetser, J. and Wiles, J.: Creating Engaging Artificial Characters for Games. Proceedings of the Second International Conference on Entertainment Computing. Carnegie Mellon University, Pittsburgh, PA (2003) 1-8
6. Tozour, P.: Influence Mapping. In: M. Deloura (ed.): Game Programming Gems 2. Charles River Media, Inc., Hingham, MA (2001) 287-297
7. Woodcock, S.: Game AI: The State of the Industry. Gamasutra (November 1, 2000). Available online 2 February, 2005, at http://www.gamasutra.com/features/20001101/woodcock_01.htm (2000)