Journal of Korea Game Society 2015 Oct; 15(5): 89-98
http://dx.doi.org/10.7583/JKGS.2015.15.5.89

게임 콘텐츠, 산업, 마케팅

Making Levels More Challenging with a Cooperative Strategy of Ghosts in Pac-Man

Taeyeong Choi, Hyeon-Suk Na
School of Computer Science and Engineering, Soongsil University

고스트들의 협력전술에 의한 팩맨게임 난이도 제고

최태영, 나현숙
숭실대학교 컴퓨터학부
ctyeong@gmail.com, hsnaa@ssu.ac.kr

ABSTRACT

The artificial intelligence (AI) of Non-Player Companions (NPC), especially opponents, is a key element to adjust the level of games in game design. Smart opponents can make games more challenging as well as allow players for diverse experiences, even in the same game environment. Since game users interact with more than one opponent in most of today's games, collaboration control of opponent characters becomes more important than ever before. In this paper, we introduce a cooperative strategy based on the A* algorithm for enemies' AI in the Pac-Man game. A survey from 17 human testers shows that the levels with our collaborative opponents are more difficult but interesting than those with either the original Pac-Man's personalities or the non-cooperative greedy opponents.

요 약

NPC, 특히 적 캐릭터들의 인공지능은 게임의 설계 단계에 있어 난이도를 조절하기 위해 핵심적인 요소이다. 지능적인 적들은 게임을 보다 도전적으로 만들 뿐 아니라, 동일한 게임 환경에서도 유저들에게 다양한 경험을 제공할 수 있다. 오늘날 대부분의 게임 유저들은 다수의 적들과 상호작용을 하기 때문에, 적 캐릭터들의 협업을 제어하는 것은 이전 어느 때보다 그 중요성이 크다고 할 수 있다. 본 연구는 팩맨 게임의 적 인공지능에 구현될 수 있는 A* 알고리즘 기반의 협력전술을 제안한다. 17명의 피실험자로부터 얻은 설문 결과는 제안된 협력전술을 따르는 적으로 구성된 레벨이, 기존 팩맨게임에서의 적들 또는 비협력적인 적들로 구성된 레벨들보다 더 어렵고 흥미로웠음을 보여준다.

Keywords : Artificial Intelligence(인공지능), Game Level Design(게임 레벨 디자인), Behavior Pattern of NPC(NPC 행동 패턴), Cooperative Multi-agents(협력적 다중 에이전트)

Received: Jul, 28, 2015 Accepted: Sep, 10, 2015
Corresponding Author: Hyeon-Suk Na(Soonsil University)
E-mail: hsnaa@ssu.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

1. Introduction

Designing behavior patterns of NPCs is a crucial issue in the game level design. Intelligent opponents deliver more fun to game users and make them more absorbed in the game play. In most of trendy video games, players are to fight against several opponent NPCs at a time. Thus using collaborative AI for opponent characters becomes more important and interesting in the game design.

In the last decades, cooperative AI strategies have been studied in the context of automatic control of Multi-Agent Systems (MAS)[1]. In the computer game design, however, little of them has been utilized so far, thus both game designers and users feel the more and more necessity of smart AI of multiple enemies to maximize the variation of levels and to deliver more users' fun.

In this note, we propose a simple but effective collaborative strategy of multiple opponents and test our approach in a simplified version of the Pac-Man game (Namco 1980). We chose this game as a test-bed since its game environment is simple and similar to that of "Predator/Prey" pursuit problem, which is widely used for illustration of techniques of MAS; The domain is just an orthogonal grid-based connected graph, and one player character, Pac-Man, has a goal to collect all dots, called pallets, avoiding four enemies, called ghosts.

To help the four ghosts cooperatively capture the Pac-Man, we implemented a centralized system for controlling all the ghosts. At every moment, the controller decides whether the player is inside a tetragon formed by the ghosts and thus surrounded by them (Siege mode) or not (Free mode). Depending on the mode, the central supervisor assigns an appropriate target spot to each ghost, so as to not only keep this sieging situation but also efficiently threaten and kill the Pac-Man.

Using a survey from 17 human testers, we evaluate the effectiveness of our strategy. The levels with our collaborative opponents are more difficult but interesting than those with either the original Pac-Man's personalities or the non-cooperative greedy opponents.

2. Related Works

Cooperative AI strategies have been studied for automatic control of MAS[1]. As for the real-time pursuit problem introduced by Brenda et al.[2], many algorithms have been employed to make four predators pursue a moving prey in non-hazy environments. In [3, 4] it was shown by experiments that if the prey is effectively slower than the predators, then the prey is eventually caught in any initial configurations by a centralized greedy strategy of coordinating the predators nearer to the current position of the prey.

The authors of [5,6,7] investigated reinforcement learning algorithms in non-hazy environments, by using for inputs the relative angle or the distance of the prey and the center of the predators and for rewards whether or not the prey is caught[5] or the distances from the prey to the predators or their center[6,7]. They claimed that the predator team was successful in catching the prey as well as learning how to cooperate each other, but their ways of rewarding for cooperative behaviors seem too implicit to train independent agents in effective collaboration. Undeger and Polat[8] introduced two coordination strategies in hazy environment, namely Blocking Escape Directions (BES) and Using Alternative Proposals (UAL), which help the predators waylay the possible escape directions of the prey in coordination. They compared their coordination strategy with the uncoordinated one, against a prey controlled by Prey A* algorithm, and observed an impressive reduction in the number of moves to catch the prey.

Collaboration strategies for predators in computer games seem so far rarely put in use. As for the Pac-Man game, several authors applied Monte Carlo Tree Search (MCTS) techniques to controlling the Pac-Man or the team of ghosts. MCTS makes a decision based on tree search where nodes are evaluated through random simulations of future movements. For instance, the authors of [9], who won the first Ms. Pac-Man Versus Ghost Team Competition in the 2011 IEEE Congress on Evolutionary Computation (CEC), formed their ghost team of a ghost controlled by the original rules and three ghosts controlled separately by different MCTS. In order to increase the reliability of simulation results, they developed a mechanism for predicting the Pac-Man's movements based on its similar previous movements, and used them during Monte Carlo simulations.

In the next section, we describe a simple centralized strategy (similar to that of [3]) for controlling all the ghosts. At every moment, the controller decides whether the player has been sieged by the ghosts or not, and depending on the decision, it assigns an appropriate target spot to each ghost, so as to not only keep the Siege mode but also efficiently threaten and kill the Pac-Man. Our work is very simple and easy to implement, and treats hazy environment as in [8,9]. To prove the effectiveness of predators' cooperative strategy, we use 17 human players and examine a survey from their feedback instead of using prey AIs as in [8,9]. Most players felt the levels with our collaborative opponents more difficult but interesting than those with either the original Pac-Man's personalities or the non-cooperative greedy opponents.

3. Cooperative Strategy

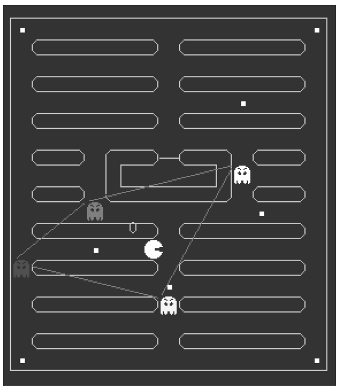
We utilize the Pac-Man game as a test-bed since it provides a simple and dynamic environment for the typical predator/prey pursuit problem, where the player controls an unit, the Pac-Man moves horizontally or vertically, and should collect all the dots, avoiding 4 opponents, the ghosts, in order to win a game.

In the original Pac-Man game, enemies act according to their own policies, namely personalities. One ghost just chases the Pac-Man and another tries to shift to the location 4 points straight ahead of the Pac-Man. The third one chases the Pac-Man if farther than some distance, but otherwise, moves towards the bottom-left corner of the maze. The last ghost draws a vector from the first ghost to the location two points ahead of the Pac-Man, doubles it, and takes as the target point the location the extended vector points at. In fact, there are three modes, called chase, scatter, and frightened, in the original version of the game, but we focus on chase mode only described above, because its duration time is the most relevant factor for the level adjustment.

Such original mechanism is equipped with a certain degree of collaboration of ghosts, but stronger cooperation of ghost team would provide more fun to advanced users. Hence, in this section, we will introduce a new strategy for the ghost team that can ameliorate the ghosts performance.

3.1 Centralized System

We model a centralized system for first monitoring the current status and then controlling all the ghosts' movement. As a preprocessing step, the system computes the shortest-distance action tables by A* algorithm for all the maps that output an optimal action to go from A to B along the shortest path in a given map. Referring to this table, the controller and the ghosts can determine which ghost is nearest to a given point and what is the next position nearest to the assigned target point.



[Fig. 1] Tetragon-shaped area of the ghosts (colored with red) showing Siege mode

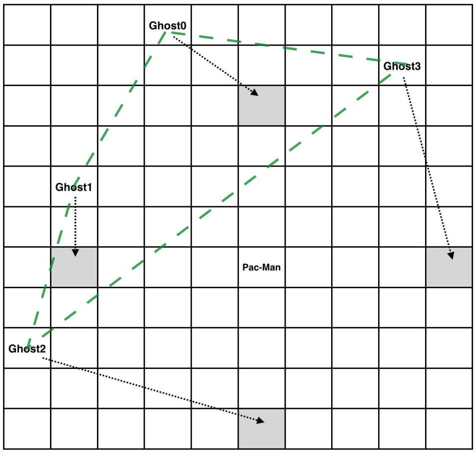
During the game, the system checks the current locations of the Pac-Man and the ghosts, and determines if the current status is Siege mode or Free mode, depending on whether the ghost team is besieging the Pac-Man or not. For this judgement, the controller computes a tetragon-shaped area based on the coordinates of all ghosts, and if the Pac-Man is inside the area, judges the status as Siege mode, and Free mode otherwise. As shown in [Fig. 2], the method of forming the tetragon is to connect all the ghosts in counterclockwise direction from the ghost with the largest y-coordinate.

3.2 Free Mode

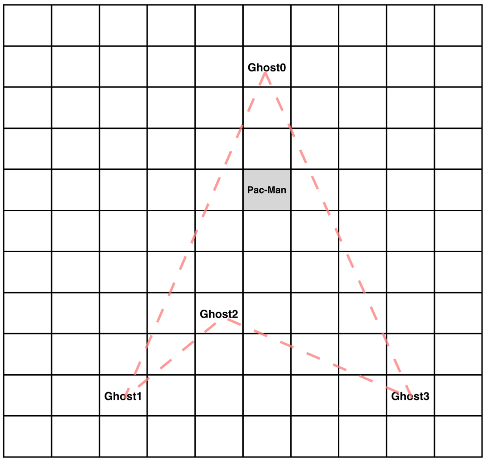
In Free mode, the centralized controller works in a similar way to the strategy of [3]. It calculates four target points and allots them to the ghosts, one for a nearest ghost, which aims to help the ghost team lay siege to the

```
Area getGhostArea ( ghostSet )
  leaderGhost := a ghost with the largest
                    y-coordinate in ghostSet
  ghostSet := ghostSet - leaderGhost
  For( i = 0; i <= 2; i++ )
    ghostSet[i].v := getVector( leaderGhost, ghost )
    ghostSet[i].a := getAngle( x-axis, ghost.v )
  newGhostSet[1...3] := ghostSet sorted by angle in
                          descending order
  newGhostSet[0] := leaderGhost
  For( j = 1; j <= 4; j++ )
    area.addLine( ghostSet[j-1], ghostSet[j%4] )
  return area
```

[Fig. 2] Computing the tetragon of ghosts



[Fig. 3] Ghost control in Free mode



[Fig. 4] Ghost control in Siege mode

Pac-Man character as soon as possible. The four points are calculated by adding 4 to or subtracting 4 from the x- and y-coordinates of the Pac-Man (as in [Fig. 3]). Using the value 4 larger than 1 (which was used in [3,4]) helps the ghosts make a siege more effectively and quickly, but the specific value of 4 used here is empirically determined for the best result in our settings, thus needs revision depending on the size and the structure of the mazes.

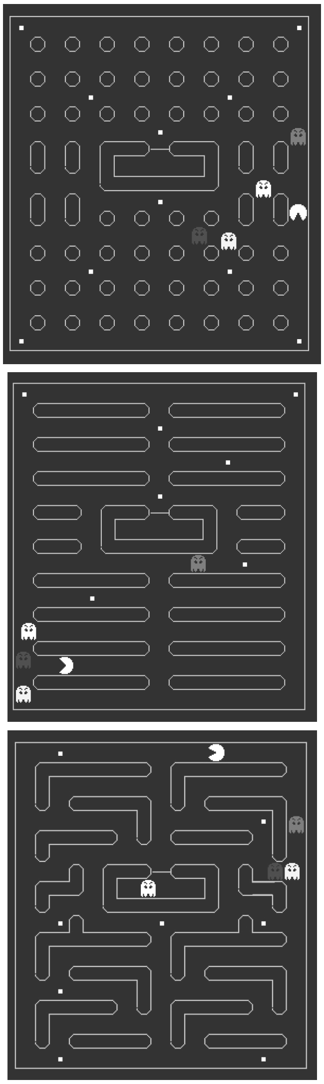
3.3 Siege Mode

Siege mode encourages the ghosts currently surrounding the Pac-Man to kill the Pac-Man quickly. The controller changes all the ghosts' targets to the location of the Pac-Man as in [Fig. 4], so that the ghosts try to eat the Pac-Man as swiftly as possible.

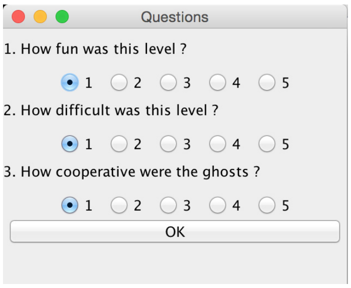
Adjusting the target points according to the two modes turns out to be very effective for enhancing multiple agents' teamwork. Furthermore, it is very time-efficient since the controller and the ghosts just need to look up the precomputed table for the shortest distance or path between any two points.

4. Experiments & Results

To perform our experiments, we used an open source code of the game Pac-Man, and modified only part of the ghosts' intelligence. Three types of AI for ghosts were implemented. The first AI is that of the original Pac-Man where the ghosts act according to their unique personalities (described in Section 3). The second one is that at every moment all the ghosts independently and greedily move nearer to the Pac-Man, referring to the shortest-distance action table pre-computed by A* algorithm. The last AI, our approach, is that the controller coordinates the next positions of the ghosts, referring to not only the shortest-distance table but also the modes in which they are. In Free mode, the target points are chosen as the points at distance 4 from the Pac-Man, and in Siege mode, the target points changed to the Pac-Man's position itself.



[Fig. 5] Map A, B, and C (from top to bottom)



[Fig. 6] Questionnaire given after each level

To gain non-biased results, we generated three types of maze environments [Fig. 5] and produced 9 different levels by combining the mazes with the three AIs. The mazes vary in terms of the spatial complexity. To be specific, in Map A, 21 percent of the space is filled with blocks, and in Map B and Map C 39 percent is filled with blocks. Maps B and C have the same ratio of obstacles, but the blocks in Map B are parallel and those in Map C have complex structures.

To get more realistic results, we used 17 human players for the test, instead of Pac-Man AIs used in [8,9]. We taught them how to play the Pac-Man game, but we did not notify the purpose of this test. We asked them to successively play 9 levels given in a random order. The type of selected AI was not exposed to the subjects in each trial to prevent them from trying to learn and predict ghosts' behavior patterns.

In each level, the player and the ghosts start at random points. The player is to avoid the ghosts and collect 10 pellets. If the player is killed four times, the ghosts win the game, but if all pellets are eaten by the player, the player wins the game. We set the speed of the ghosts slightly slower than the Pac-Man's, in order to make the ghosts catch the player only with cooperative tactics.

After each level ends, information of subject's play was automatically logged, and the questionnaire [Fig. 6] is given to be completed, that is designed to qualitatively evaluate three kinds of aspects: difficulty, interestingness, and teamwork of ghosts.

[Table 1] Winning rates of the testers

AI type	Map		
	A	B	C
Original	0.59	0.94	1.00
Greedy	0.65	0.94	0.82
Free/Siege	0.47	0.64	0.70

[Table 2] Average time taken for testers' win

AI type	Map		
	A	B	C
Original	44.00	42.75	44.82
Greedy	39.09	35.63	36.86
Free/Siege	51.00	46.36	46.58

[Table 3] Average Pac-Man's remains

AI type	Map		
	A	B	C
Original	2.30	2.75	2.65
Greedy	3.00	3.06	3.14
Free/Siege	1.88	2.27	2.25

4.1 Objective Performance

Based on the logs of the players, we first examine the objective measurements of the levels. Three types of data are extracted for measuring the difficulty of a level.

The first is the winning rate of the players at the level, being lower for the more difficult level. As shown in [Table 1], the subject players felt our ghost team most difficult in every map; the players obtained the lowest winning rate when our ghost AI was adopted. The fact that the subject players felt our ghost team most difficult in every map is also ensured in [Table 2] and [Table 3], which illustrate the time elapsed for users′ victory and the remained lives of the Pac-Man, respectively. The more time elapsed, the harder the player won the level, and the less remained lives of the Pac-Man at the victory, the more killed lives of the Pac-Man by the ghosts before the victory.

[Table 4] Average scores in difficulty

AI type	Map		
	A	B	C
Original	3.71	2.94	3.00
Greedy	3.00	1.71	2.53
Free/Siege	4.24	3.35	3.41

[Table 5] Average scores interestingness

AI type	Map		
	A	B	C
Original	3.82	3.41	3.47
Greedy	3.24	2.94	2.94
Free/Siege	3.65	3.53	3.53

[Table 6] Average scores in teamwork skill

AI type	Map		
	A	B	C
Original	3.71	3.24	3.00
Greedy	2.65	2.29	2.82
Free/Siege	3.71	3.24	3.65

These tables show that in every map our cooperative units were much harder than the original ghosts or the non-cooperative greedy ghosts for the player to win the games. The original ghosts outperform the non-cooperative greedy ghosts, enlightening the importance of collaborative strategies.

Interestingly, in both original AI and our strategy, the difficulty decreases as the complexity of the maze increases, which implies that the effect of collaboration and threatening reduces in a more complex maze. Compared with the original ghosts against which every player won in Maze C, our collaborative strategy gave reasonable difficulties to the players in all mazes.

4.2 Subjective Performance

Based on the questionnaire data, we now investigate the subjective evaluation of the levels. After each level, the questionnaire window shows up to be filled in. A questionnaire set includes three kinds of questions, to which answers are in five-point scales.

1) Difficulty: [Table 4] shows the difficulty evaluation of the players for each level. As expected, our cooperative strategy is regarded as making the levels most difficult. Again, the original ghosts scored higher than the greedy team did in every level, which supports that users have difficulty in playing against collaborative tactics.

2) Interestingness: The next question is to measure how much fun the users had. As shown in [Table 5], the subject gamers selected as the most interesting enemies the original ghosts in Map A but our ghosts in Maps B and C. The non-cooperative opponents were counted as the most tedious ones, which might be because such simple greedy behavior patterns are easily predictable ones to human users.

The reason why the players ranked the original ghosts higher than ours is because Map A is an open space, so makes the behaviors of the original ghosts based on the personalities more unexpected and interesting, whereas in open space our ghosts always easily keep their tetragon-shaped area, swiftly attack the users, and thus made the gamers bored. In more complicated maps (B and C), however, our ghosts offered more fun to the subjects than the original ones did. The gaining of our strategy against the original AI in the scores of Interestingness might look negligible, but in fact is highly meaningful because combining the result in [Table 4] together with the results in [Table 1] and [Table 5] for Mazes A and C shows that our ghost AI can be regarded as a method for increasing the level of games and (at least keeping) the fun of gamers simultaneously in any maps.

3) Teamwork: Lastly, the testers were to evaluate the collaboration of ghosts. As seen in [Table 6], the greedy ghost team was rated as the worst cooperative team for every level. In addition, compared to the original ghost team, our ghosts obtained the same score in Maps A and B, but much higher score in Map C. As noticed above, it is interesting that the more sophisticated the maps become, the weaker the original AI's teamwork becomes, while our ghosts demonstrated strong teaming even in the most complicated maze, Map C.

5. Conclusions & Future Work

In this note, we suggest a method for cooperative tactics of opponents in the Pac-Man game. The approach requires one main controller system that can determine whether the ghosts are sieging the player unit or not, and assign appropriate coordinates to each ghost.

Through experiments with 17 human game users, it is concluded that cooperativeness in enemies' action pattern is necessary to make levels interesting. Furthermore, our collaborative strategy can provide users with more difficult but more interesting game experiences than original ghosts can.

One further research direction is to develop more various and interesting collaborative strategies by combining ours with up-to-date collaborative AI techniques. In addition to MCTS as related works, machine learning based models and algorithms can be utilized since it is relatively easy to collect users' play data for training. Such an approach could offer customized-levels for gamers and more exciting game experiences.

Another interesting topic is to handle more complex game environments, such as various size of maps and items potentially influential to plays, and as in the real Pac-Man game, three kinds of different modes (Chase, Scatter, and Frighten).

REFERENCES

[1] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective", Autonomous Robots, Vol. 8, No. 3, pp. 345-383, 2000.

[2] M. Benda, V. Jagannathan, and R. Dodhiawalla, "On optimal cooperation of knowledge sources", Technical Report No.BCS-G2010-28, Boeing Advanced Technology Center, 1986.

[3] L.M. Sephens and M.B. Merx, "The effect of agent control strategy on the performance of a DAI pursuit problem", In Proceeding of the the 10th International Workshop on Distributed Artificial Intelligence, 1990.

[4] T. Haynes and S. Sen, "Evolving behavioral strategies in predators and prey", Adaptation and Leaming in Multiagent System, Springer Verlag, Berlin, pp.113-126, 1996.

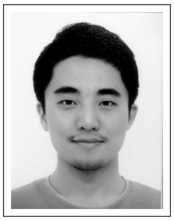
[5] Y. Ishiwaka, T. Sato, and Y. Kakazu, "An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning", Elsevier Journal on Robotics and Autonomous Systems, Vol. 43, No. 4, pp. 245-256, 2003.

[6] Hyong-Ill Lee and Byung-Cheon Kim, "Multiagent Control Strategy Using Reinforcement Learning", The KIPS Transactions: PartB, Vol. 10B, No. 3, pp. 249-256, 2003.

[7] D. Xiao and A. Tan, "Cooperative cognitive agents and reinforcement learning in pursuit game", In Proceedings of 3rd Int'l Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS'05), 2005.

[8] C. Undeger and F. Polat, "Multi-agent real-time pursuit", Autonomous Agents and Multi-Agent Systems, Vol. 21, No. 1, pp. 69-107, 2010.

[9] K. Q. Nguyen and R. Thawonmas, "Monte carlo tree search for collaboration control of ghosts in Ms. Pac-Man", IEEE Transactions on Computational Intelligence and AI in Games, Vol 5, No. 1, pp. 57-68, 2013.



최 태 영(Taeyeong Choi)

2007-2015 숭실대학교 컴퓨터학부 공학사
2015-현재 애리조나주립대학교 박사과정

관심분야: 인공지능, 기계학습,다중 에이전트 플래닝

나 현 숙(Hyeon-Suk Na)

1993 서울대학교 수학과 이학사
1995-2002 포항공과대학교 수학과 이학 석·박사
2001-2003 프랑스 INRIA, HK UST Post Doc.
2003-현재 숭실대학교 IT대학 컴퓨터학부 교수

관심분야: 알고리즘, 계산기하학, 컴퓨터그래픽스