technische universität
dortmund

CI enlivened ghosts for Pac-Man

Nicola Beume, Boris Naujoks, Georg Neugebauer, Nico Piatkowski, Mike Preuss
Computational Intelligence, Chair of Algorithm Engineering
Department of Computer Science, TU Dortmund

July 4, 2008

1 Introduction

Pac-Man is a well known classic arcade game, developed by the Japanese company Namco in 1980. In Pac-Man, the player controls a figure called Pac-Man navigating through a maze with the task to eat all dots (called food-pellets) lying in the tunnels of the maze, by which he wins the level and advances to the next. Pac-Man is chased by four computer controlled enemies (ghosts) who eat him if they catch him, resulting in the player losing one of his lives. If Pac-Man eats one of four special pellets called powerpills he is able to invert the situation for a short period of time and eat the ghosts. We used a reimplementation (clone) of the game under GNU General Public License (GPL) called NJam [Bab03] by M. Babuskov in which we rebuilt Namcos original Pac-Man level shown in figure 1. We focused on the control methods for the ghosts and developed some interesting behaviors. In our approach the ghosts are not only optimized regarding their ability to beat the player but also to behave in a way that the game is attractive for the player, providing more fun than playing against the ghosts controlled by the standard methods implemented in Pac-Man or in NJam.

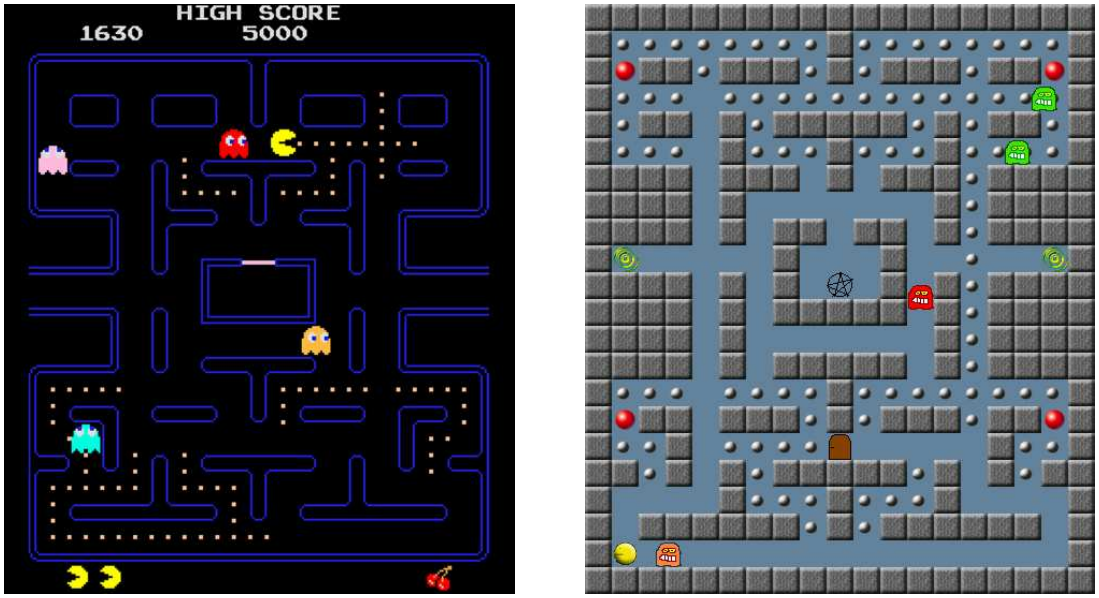

Figure 1: The original maze of Pac-Man (left) and the rebuild level in NJam (right).

2 Algorithms for controlling the ghosts

The algorithms designed are described in the following. Out of the larger implemented set, we focuse on the algorithms which will be presented at the workshop. Detailed information is available in the tech report [DEH+07].

2.1 Ghosts using Randomized-Deterministic Algorithms

Randomized and deterministic algorithms were used as a baseline for comparison to AI trained by Artificial Neural Networks (ANN) or Evolutionary Algorithms (EA). We called them Randomized-Deterministic (RD), because they are controlled by deterministic algorithms which use some randomness to make them less predictable. Overall we implemented five different controllers using RD algorithms. Two of them are presented for demonstration at the workshop.

Firou is a very simple RD ghost. He has two routes programmed which may be walked in either direction. At the starting point he chooses uniformly distributed one of the four possible routes and walks along the chosen route. This process is repeated every time he reached the starting point. The routes are shown in figure 2. This ghost will only be able to catch Pacman if Pacman walks directly into him.

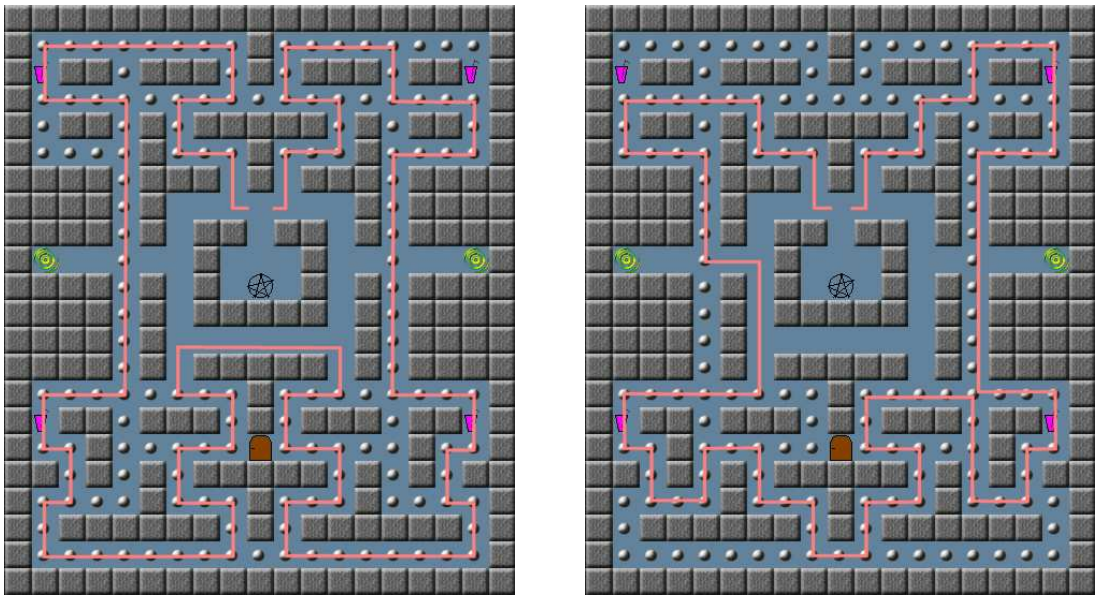

Figure 2: Routes which Firou may take through the labyrinth. The routes can be walked in either direction.

Old Shimer is an old and therefore experienced hunting ghost in the labyrinth. He has absolute information on the vertices that he has visited and their incident edges. Since he is very old, his memory is poor and he can only memorize ten vertices. His field of view is the area of his memory and the lines in all four directions of maximal ten tiles if not blocked by a wall. If he sees Pacman, he pursues him until he eats him or Pacman leaves his field of view.

2.2 Ghosts using Artificial Neural Networks

As one of the CI based approaches, ANNs were used to improve the behavior of the computer controlled ghosts in Pacman. The ghosts should learn how to handle certain situations and react in an appropriate manner, without the use of global knowledge. With respect to gameplay and fun this should lead to a more natural behaviour, compared to a non-beatable AI which dominates the human player throughout the entire game.

Training The process of learning is an important part of the concept of ANNs. We used the simple back-propagation algorithm for the training process. The data used was a set of encoded descriptions of possible game situations and the desired reactions, precisely the direction to move. This information had to be created manually based on our expertise.

Ingame The four ghosts were enlivened with four ANNs which were trained with different training sets overall leading to an average behavior increasing the fun for the player. In every game the ANNs are choosen randomly for the four ghosts. This makes the game more interesting as the ghost behavior is less predictability.

2.3 Ghosts using Evolutionary Algorithms

We selected three different approaches to gain an overview about the usability of EAs to generate game strategies. Two of the EAs use individuals which describe the full ingame behavior of a ghost, the third EA employs playing field coordinates as individuals and evolves them online every time a ghost needs to be moved.

Pure Strategies We designed a (4 + 4)-EA to generate game strategies by mutating randomly initialized strategies. A strategy is a mapping from situation-codes to legal moves. The selection, variation and fitness calculation is done every 180 game loops, so the ghost behavior will change during game play. The variation is done by remapping $^1/_5$ of all possible situation-codes to a new legal move.

Mixed Strategies Our mixed strategy approach works very similar to the pure Strategies approach with the difference that it deals with probability distributions over legal moves which results in a more chaotic and thus unpredictable behavior. The variation is done by adding a normal distributed value out of [0, 1] to one randomly selected move probability followed by a normalization of the distribution vector.

Online EA Our Online EA evaluates the calculated directions at runtime. Each individual consists of two coordinates and decodes one position in the maze. The mutation operator generates one new position by adding a randomly chosen number from the range [-5, 5] to the individuals coordinates. Since the mutation does not guarantee that generated positions are actually correct, a feasibility test is performed after each mutation. Each individual is evaluated according to its distance to Pacman. As the ghosts are supposed to chase Pacman, this distance is supposed to be minimized. Apart from the distance to Pacman the fitness function takes the relative position of the ghosts to each other into account as well as the availability of power pills. This solution leads to powerful ghosts because Pacman is always under pressure due to minimizing the distance to Pacman in the fitness function.

3 Demonstration content

We want to present the different algorithms mentioned above by playing against them. For this task the player chooses the algorithm he wants to play against. He can test the different solutions and make some experiences comparing the different behaviors. Additionally we will explain the characteristic moves while playing the game. The following algorithms will be presented:

- Randomized-Deterministic
- Different sets of ANNs
- Pure Strategies and Mixed Strategies
- Online EA

All approaches have specific characteristics especially regarding the degree of difficulty (see [BHN+08]) and the aspects of fun (see [BDE+08]). These developments belong to the very few that exist for ghosts in Pac-Man as most studies focussed on replacing the Pacman player by an artificial intelligence, resulting in a pure system without human interaction. We especially want to consider the aspects of interaction of human players and computer controlled players. For more information about our work visit also our homepage [PG5].

References

[Bab03] M. Babuskov. NJam, 2003. October 10, 2007.

[BDE+08] Nicola Beume, Holger Danielsiek, Christian Eichhorn, Boris Naujoks, Mike Preuss, Klaus Stiller, and Simon Wessing. Measuring flow as concept for detecting game fun in the pac-man game. In *Proc. 2008 Congress on Evolutionary Computation (CEC'08) within Fifth IEEE World Congress on Computational Intelligence (WCCI'08)*, 2008. (in print).

[BHN+08] Nicola Beume, Tobias Hein, Boris Naujoks, Georg Neugebauer, Nico Piatkowski, Mike Preuss, Raphael Stüer, and Andreas Thom. To model or not to model: Controlling pac-man ghosts without incorporating global knowledge. In *Proc. 2008 Congress on Evolutionary Computation (CEC'08) within Fifth IEEE World Congress on Computational Intelligence (WCCI'08)*. IEEE Press, 2008. (in print).

[DEH+07] Holger Danielsiek, Christian Eichhorn, Tobias Hein, Edina Kurtić, Georg Neugebauer, Nico Piatkowski, Michael Puchowezki, Jan Quadflieg, Sebastian Schnelker, Raphael Stüer, Andreas Thom, and Simon Wessing. Zwischenbericht der PG 511. Technical report of the collaborative research centre 531 computational intelligence (236/07), Technische Universität of Dortmund, 2007. in German.

[PG5] PG511. http://www.ciingames.de. July 4, 2008.