

StrathPac - An Automated Player for Ms. Pac-Man

*L. Kelly, L. Dicken, J. Levine
Department of Computer and Information Science
University of Strathclyde*

Introduction

This paper describes a software agent designed to play the arcade game Ms. Pac-Man. This game provides two major challenges in its environment which governed the approach taken in this agent, namely the consistent map layouts which allowed for application of search techniques to reason about the agents current situation and also the non-determinism and rapid change presented by ghost movements, which preclude a purely deterministic agent (for example, one with pre-set paths to follow, which would work well in the original Pac-Man), instead requiring a more intelligent approach.

Execution

This agent was developed and tested using the implementation of Ms. Pac-Man found on <http://www.webpacman.com> and version 1.6 of the Sun SE JVM. To run the agent, ensure that a Ms. Pac-Man game is completely on screen and unobscured then run this command from PacAgent/bin:

```
java Main
```

The agent will then find the game, cause a mouse click event over it to give it keyboard focus, add coins and start the game. If for some reason the game does not have keyboard focus or loses it during operation it should be regained manually.

The agent begins by analysing the layout of the map via a screen scraping technique and abstracting this to a grid of 28 by 31 squares each representing an 8 by 8 pixel area, offset such that each pill lies in the centre of a square. Every grid square which it is possible for Ms. Pac-Man to reach is then further abstracted to form a roughly 300 node graph which the agent uses to reason about its environment using the methods described below. It is worth noting that this graph structure treats the tunnels at the side of a map as it does any other connected nodes but does not account for the 'ghost-slowng' effect they have.

Once this initial setup is complete, the agent then executes the following loop repeatedly:

- Screen scrape: The agent captures the section of the screen containing the Ms. Pac-Man game and examines it for the locations of the ghosts and Ms. Pac-Man as well as checking which grid squares still contain pills. This information is used to update the game state for use in the next section.
- Reason: The agent then uses the game state and the graph structure to apply an adapted breadth first search which accounts for distance from ghosts and the points value of game objects (edible ghosts have priority over pills for example) to create a path for the agent to follow in game. See fig 1. for an example of such a path.
- Actuate: Pass the resulting path to the actuator, which will control Ms. Pac-Man by triggering key press events.

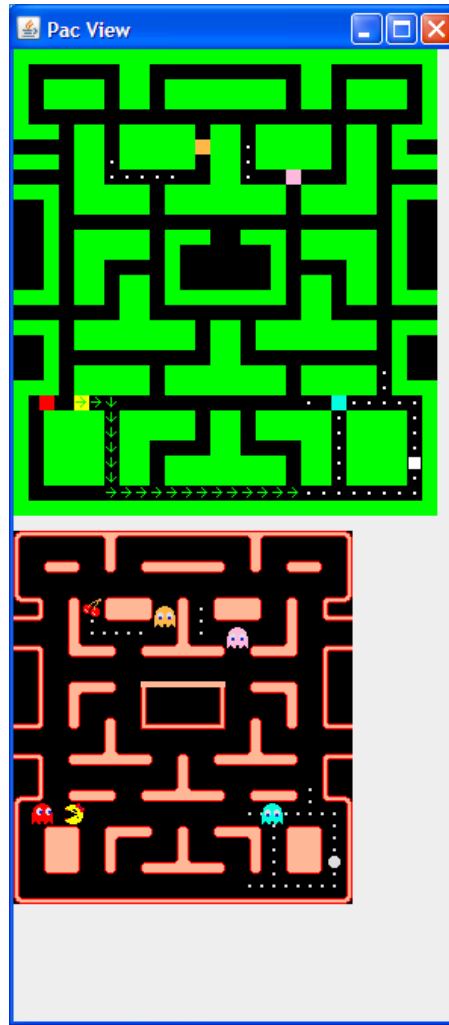


Figure 1. - StrathPac's UI Highlighting the Path Planing System

Controller Logic

The agent presented here was designed with the purpose of maximising its score in a single game of Ms. Pac-Man. To achieve this, it acts towards two occasionally conflicting goals:

1. Achieving an optimal score for each level.
2. Completing a level while avoiding capture by the ghosts.

These goals offer two alternative approaches to collecting points in a game of Ms. Pac-Man, either taking risks to achieve high scores in early levels or playing it safe for as long as possible and hoping to gain points while doing so. The next sections will explain the extent to which these approaches were used and provide justifications for this design.

As the levels of Ms. Pac-Man become progressively more difficult and because capturing ghosts offers so many points (in particular all four in a single power pill which will result in a total of 3000 points) in comparison to simple pill sweeping, it is clear that correct use of power pills is key to maximising score.

However, achieving an optimal score - that is, by consuming all four ghosts on each power pill - is a difficult

and potentially risky task. An agent must choose to activate the power pill at the correct time so that it can reach all the ghosts before the effect wears off, but in doing so should not allow itself to be caught by a ghost while waiting for an ideal situation.

For this reason the Ms. Pac-Man agent operates under 'modes' based on the current goal it is pursuing. These modes are:

1. Hunting Ghosts.
2. Progressing through level.

The primary concern of mode 2 is survival and collecting pills is considered secondary to this. In this mode the agent will progress to the nearest safe pill using a modified breadth first search which accounts for the threat the ghosts pose to a given location based on their distance to it. Left permanently in this mode the agent will clear the map of pills and move onto the next, so while in this mode the agent is waiting for an opportunity to switch into hunting mode. The agent decides when best to do this based on its distance to the ghosts and the reachability of the nearest power pill. Based upon this information it may decide to 'pill-park', (that is wait near a power pill until it is ready to consume it.) to consume the power (possible after having spent some time parking) or to continue pill sweeping until the situation changes.

In hunting mode the agent prioritises only consuming edible ghosts and avoiding non-edible ones (the result of a consumed ghost visiting the pen and re-entering the game). It does this because of the limited time in which it has before the effects of the power pill wear off and the high points value associated with capturing ghosts. As mentioned previously, there are risks in chasing ghosts and when the effects of the power pill do wear off the ghosts become dangerous again and potentially closer than they would normally be in mode 2 due to the fact the agent was previously chasing them. For this reason, the agent will switch back into mode 2 when it deems it unprofitable to continue to chase edible ghosts (when they are both distant and about to become inedible for example.)

Limitations

Our approach suffers from a fundamental limitation at this point, in that it is not aggressive enough when it eats a power pill - it will frequently ignore a ghost that would have been easy to kill in favour of eating additional pills. This is a major deficiency since the majority of the score is gained by sequential killing of ghosts in a single power pill as noted above. This means that the importance of making such a clean sweep cannot be overstated, but at the current time our agent is unable to do this. This is largely due to the current tuning of the prioritisation of the goals, but also in part caused by the inability of the agent to accurately gauge where the ghosts are heading and intercept them efficiently, instead attempting to follow behind them until it has caught up to them. There is a trade-off here between runtime and quality of results - our current agent does not strike this balance appropriately.

Conclusions

In this paper we have presented our approach to designing an intelligent and robust agent capable of playing Ms. Pac-Man. Although this system has certain weaknesses (detailed above), it has been developed from the ground up to provide an extensible, efficient player capable of making rational decisions in a fast-paced environment. In this we believe we have had a degree of success, although there remains considerable scope for improvement.