

# Ms. Pacman Competition Entry

Tómas Guðmundsson

14. maí 2009

## 1 Introduction

Ms. Pacman is a very dynamic game. It is a spinoff the famous Pacman game, unauthorized it was published by Midway in 1981 and like many other games it featured a female protagonist. The aim in Ms. Pacman is to eat all pills, catch the powerpills to eat the Ghosts and occasionally eat the candy or prize that drop into the screen. I made a Ms. Pacman automated player as a part of a final project for one course in my studies as a Software Engineer at the university of Iceland.

## 2 Ms. Pacman player

I decided to look at the problem from the standpoint of a human player. What makes a good decision in pacman ? Why is it better to catch the powerpill now instead of waiting for another ghost to come close. Is it good to get all the pills as quickly as possible and just try to avoid the ghost. I though I could use the ideas from Weigthed Peace Counter method. It is quite obvious that a road that pacman can travel that contains pills is a better road than a road that contains none. Therefor I made the connection from pills to pieces on a board. However since the dynamic nature of Ms. Pacman I decided to try and cook up a dynamic version of the WPC, I call dWPC. This package can calculate and read information of every node, wether it contained a pill, powerpill, ghost and the candy.

$$\frac{\text{pills}}{\text{length}} + \frac{\text{powerpills}}{\text{length}} + \frac{\text{ghost edible}}{\text{length}} - \frac{\text{ghosts}}{\text{length}} \quad (1)$$

Equation 1 was the first method I used. I calculated this, at every junction, recursively for a fixed amount of length and a fixed amount of recursive step Ms. Pacman was allowed to take. Length in Equation 1 is the total length from Ms. Pacman for the node that was being calculated. I then changed it to better suit my understanding and used a total budget of nodes

Ms. Pacman was allowed to go. The problem with recursive step and allowed length could be different for different actions. If the road it calculated for was shorter than the allowed length then it stopped and got the reward for the next junction. So evidently it calculated a different number of nodes every time.

A total amount of nodes, called budget was used and is decremented for every node visited. It is then split in so many ways as are available to Ms. Pacman. If Ms. Pacman could go in 3 direction and budget of nodes he had left, would be split equally for every direction. This way he would calculate the quality of the road, branching from his position. This is done for every possible action at every possible time to determine which is the best way to go.

$$\text{budget} * \text{pill} + \text{budget} * \text{powerpill} + \text{budget} * \text{ghost edible} - \text{budget} * \text{ghost} \quad (2)$$

Here, budget get's decremented for every node as mentioned before therefor we put the biggest weight on the closest nodes. For example if the budget is initially 50, and the reward for a pill is 1, then a pill in the next node would give 50 in reward. A ghost reward of 50 would then be negative  $50 * 50$  in the next node, but positive  $50 * 50$  if the ghost is edible.

### 3 Results

Evidently, this Ms. Pacman player can ignore ghosts (setting the reward for ghosts equal to zero) or put all the emphasis on getting the powerpill. Therefor some combination of these factors would lead to a good player. This was evident in early development when I changed the rewards. Therefor I set up an evolutionary strategy, ES(1+10) to try and find the best player. My initial tweaking lead to a player that got 22500 points. After training, the ES found values for the weights so that Ms. Pacman got 39400 points. The player plays exactly the same, every time for the same values and many runs of the game show that the ghosts play exactly the same as well in regards to how Ms. Pacman plays. The player I am submitting gets 37140 points and should get it every single time.