INFO8006: Project 2 - Report

Romain LAMBERMONT - s190931
Arthur LOUIS - s191230

October 26, 2021

1 Problem statement

a. In this section, we'll discuss the problem as an adverserial search problem and we'll define all elements of the problem

- Set of states : A state in Pacman can be described as :

$$s = \{\text{pacmanPos}, \text{ghostPos}, \text{ghostDirection}, \text{foodMatrix}, \text{score}\}$$

Where all terms are respectively : the position of Pacman, the position of the ghost, the direction of the ghost, the food matrix (representing food position with boolean values) and the current score. In this case we can define the initial state s_0 like this :

$$s_0 = \{\text{pacmanPos}_0, \text{ghostPos}_0, \text{ghostDirection}_0, \text{foodMatrix}_0, 0\} \tag{1}$$

Where all terms with a 0 index being initial paramaters given by the layout.

- Player function : The player can easely be determined this way :

$$\text{player}(s) = \begin{cases} 0 \text{ if Pacman} \\ 1 \text{ if Ghost} \end{cases}$$

- Legal actions : In a specific state s, the set pf legal actions is :

$$\text{action}(s) = \{\text{North, East, South, West}\}$$

These actions are only available under the condition that we don't hit a wall if we take the action, thus it depends of the layout.

- Transition model : The transition model can be described as the result of the action taken on the previous state, giving us the new state s':

$$s' = \text{result}(s, a) = \{\text{pacmanPos} + a, \text{ghostPos'}, \text{ghostDirection'}, \text{foodMatrix'}, \text{score'}\}$$

Where all parameters with ' are computed as a response from the action taken a (the ghost reacts at the action taken by Pacman and by taking consideration of its behaviour and the food and score are also updated as a result from the action a).

- Terminal test : The test to check if the state is terminal is simple :

$$\text{terminal}(s) = \begin{cases} 1 \text{ if ghost eats Pacman OR Pacman has eaten all dots in foodMatrix} \\ 0 \text{ Otherwise} \end{cases}$$

- Utility function : The utility function of a Pacman is simply given by the score :

$$\text{utility}(s, p) = \begin{cases} \text{score if } p = \text{Pacman} \\ -\text{score if } p = \text{Ghost} \end{cases}$$

As a reminder the score value is computed using this formula :

score = $-$time steps$+10*$eaten dots$-5*$eaten capsules$+200*$eaten ghost$+(-500$ if lost$)+(500$ if won)

b. In the case of a zero-sum game as Pacman the utility function can be described as :

$$\text{utility}(s, p) = \begin{cases} 1 \text{ if Ghost eats Pacman} \\ \text{-1 if Pacman eats all dots} \end{cases}$$

2 Implementation

a.

b. ***Just refer to your code.***

c. ***Just refer to your codes, and define your three evaluation/cut-off functions pairs.***

3 Experiment

a.

b.