



Competición Final

APRENDIZAJE AUTOMÁTICO

Andrés Vinagre Blanco - 100383414
Alejandro Parrado Arribas - 100383453

Curso 2019 - 2020

Grupo 83

Contenido

| | |
|-------------------|---|
| Introducción..... | 2 |
| Agente | 3 |
| Pruebas | 4 |
| Tipo 1 | 4 |
| Tipo 2 | 4 |
| Tipo 3 | 5 |
| Tipo 4 | 5 |

Referencias

Este documento esta basado en la memoria de la práctica 2 de Aprendizaje Automático.

Introducción

El documento explica la implementación de un agente automático de PacMan usando Aprendizaje por Refuerzo, concretamente, el algoritmo Q-Learning. A lo largo del documento se explican los diferentes atributos usados. También, se explica el proceso de entrenamiento del agente con los mapas creados por nosotros.

Agente

Este agente es el mejor de los resultantes tras entrenar durante la práctica de Q-Learning, esto se debe a que conseguimos obtener atributos que otorgan al agente suficiente información para jugar bien.

Uno de los atributos es **Dirección al Fantasma más cercano**, en un principio este atributo lo pensamos sin tener en cuenta los muros lo cual es un error ya que habiendo un muro delante le estabas dando información errónea al Pacman es por ello por lo que se pensó en varias posibilidades como añadir otro atributo que sea si hay muro delante con la desventaja de estar duplicando el número de estados o simplemente darle la dirección real. Esta última posibilidad nos pareció la más correcta ya que te evitabas tener muchísimos más estados. Como este atributo nos devuelve una dirección, el espacio de estados se va a ver multiplicado por 4

El segundo atributo del agente es parecido al anterior ya que es la **Dirección a la comida más cercana**, en este caso como puede haber o no comida en el mapa, se considera que hay un estado más para tratar el caso de que no haya comida. Esto con los fantasmas no sucede porque si no hay fantasmas en el mapa, el juego acaba.

El último atributo consiste en **Cosa más Cerca**. Gracias a la clase *Distancer* a que nos da la distancia real de una casilla origen a una destino teniendo en cuenta los muros porque volvemos al problema anterior de dar información imprecisa si no se tienen en cuenta los muros. En un principio se valoró de discretizar las distancias pero no fue muy útil porque se perdía bastante información y si querías un grado decente de precisión, los estados de la qtable crecían exponencialmente y se salía del rango recomendado de los 200. Finalmente se optó por la variable binaria de la cosa mas cercana.

Pruebas

Tras conocer el agente que mejor jugaba se creo un nuevo set de pruebas para entrenar el aspecto de que el agente al entrenar maximice la puntuación. Para ello se entreno desde cero el agente con los nuevos mapas y requirió un tiempo de 4 horas en total.

Se pueden diferenciar varios tipos de pruebas dependiendo de su propósito general.

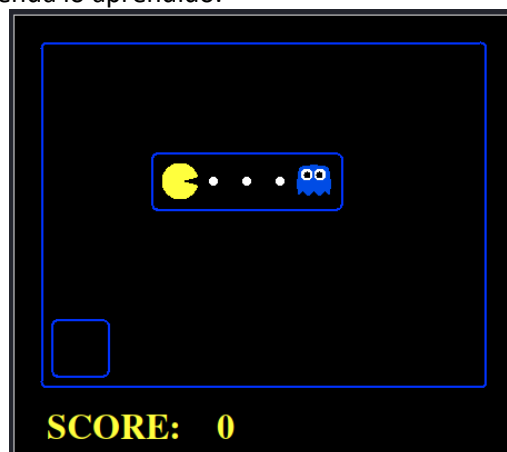
Tipo 1

Este tipo de pruebas ayudan al Pacman realizar los pasos básicos de movimiento para comerse los fantasmas. Se realizan varias con distintas direcciones empezando con unos valores de ϵ 0,4 y Alpha 0,2



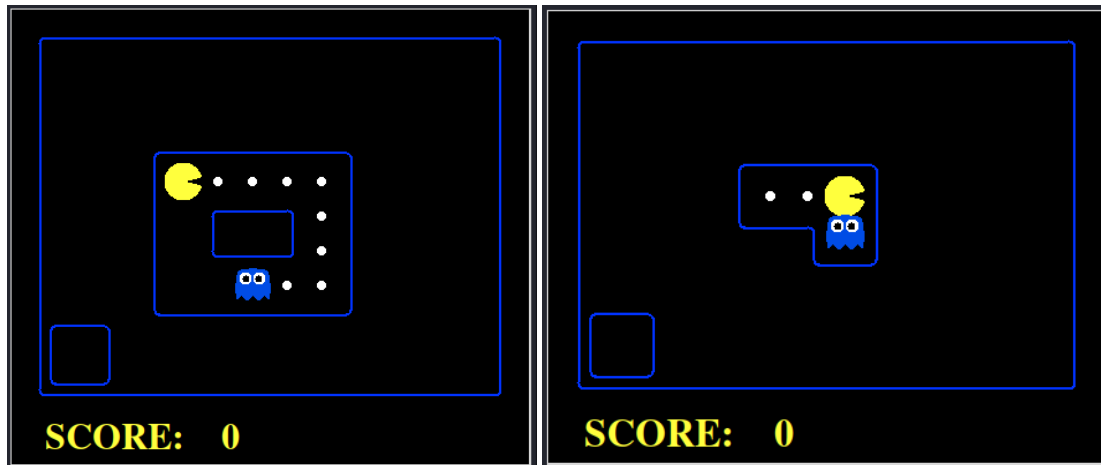
Tipo 2

Las pruebas tipo 2 son parecidas a las anteriores pero con comida y valores de ϵ 0,3 y Alpha 0,1 para prevenir que se desaprenda lo aprendido.



Tipo 3

Las pruebas de este tipo se centran en **maximizar** la puntuacion ya que los algoritmos de refuerzo tratan de maximizar el refuerzo a largo plazo. La mayoría del tiempo de entrenamiento se ha empleado en estos tipos de mapa. Se han empleado unos valores de aprendizaje mas relajados para que no desaprenda el agente; ϵ 0,2 y α 0,02



Tipo 4

El ultimo tipo de pruebas se centran más en un comportamiento general del agente en pruebas reales de rendimiento. Los valores de aprendizaje relajados para no desaprender igual que los mapas anteriores, ϵ 0,2 y α 0,02.

