Agustín Barruti
Jorge Salazar
Ander Carrera

PL1c

1. Agente Reflex:

En este primer ejercicio tenemos que devolver un valor el cual nos refleja lo bueno que es la posible acción del pacman el cual es obtenido por una serie de valores de diferente peso, a saber; distancia a la comida más cercana y distancia al fantasma más cercana.
Estos valores se toman en negativo ya que se busca la distancia mínima, pero la mayor de estas. Se han dado diferentes pesos (importancia) sobre los valores calculados a base de diferentes testeos. A la comida más cercana se le multiplica por 3 porque tiene más importancia sobre la distancia de los fantasmas. A la distancia de los fantasmas se utiliza el valor recíproco (multiplicado por 10) ya que su valor influye, pero no con la misma importancia que la comida.

Este valor total lo acabamos obteniendo por al siguiente expresión:

```
valorTotal = comidaCerca + fantasCerca - (50*len(listaComida))
```

El valor principal estará definido por la cantidad de comida restante (se le multiplica por un valor grande, a base de pruebas hemos decidido usar el 50, para que los valores calculados no influyan de forma tan significativa, pero aún así influyan).

```
jorge@jorge-Aspire7:~/Codigos/I.A/ia/PL1c/multiagent$ python3 pacman.py -p ReflexAgent -l testClassic
Pacman emerges victorious! Score: 564
Average Score: 564.0
Scores:        564.0
Win Rate:      1/1 (1.00)
Record:        Win
```

```
jorge@jorge-Aspire7:~/Codigos/I.A/ia/PL1c/multiagent$ python3 pacman.py --frameTime 0 -p ReflexAgent -k 1
Pacman emerges victorious! Score: 565
Average Score: 565.0
Scores:        565.0
Win Rate:      1/1 (1.00)
Record:        Win
```

```
jorge@jorge-Aspire7:~/Codigos/I.A/ia/PL1c/multiagent$ python3 pacman.py --frameTime 0 -p ReflexAgent -k 2
çPacman emerges victorious! Score: 1460
Average Score: 1460.0
Scores:        1460.0
Win Rate:      1/1 (1.00)
Record:        Win
```

2. Minimax:

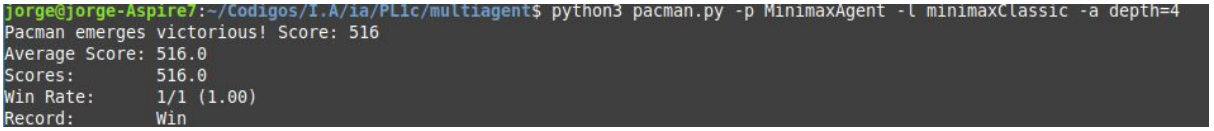
Para este segundo ejercicio desarrollaremos el algoritmo proporcionado en clase, a través de dos funciones complementarias: max_value y min_value.

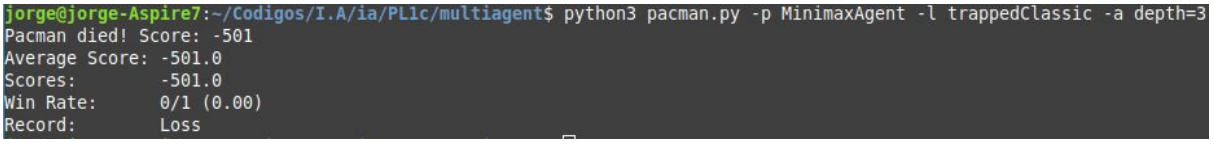
Max_Value: En este caso inicializamos nuestra variable ‘v’, a -infinito e iremos aumentando la misma. Para obtener una correcta recursividad, inicializamos esta

variable como una tupla guardando en su segunda posición la acción a desarrollar por el pacman. Para ello, iteramos sobre las acciones disponibles de nuestro pacman llamando a la función complementaria min_value con el sucesor y en caso de obtener un valor mayor al registrado guardandolo en la variable local 'v'. Finalmente, devolveremos dicha tupla como resultado.

Min_value: Similar a nuestro caso anterior inicializamos está a +infinito junto con las acciones a realizar e iremos disminuyendo este valor. A diferencia de la anterior, tal y como se especifica en el algoritmo, si nos hallamos en una profundidad igual a el número de nodos menos 1 llamaremos a la función max_value de vuelta o por el contrario actuaremos recursivamente. Finalmente, del mismo modo, devolveremos la tupla como resultado de esta función.

Por último, como resultado devolveremos la acción a realizar resultante por el pacman.

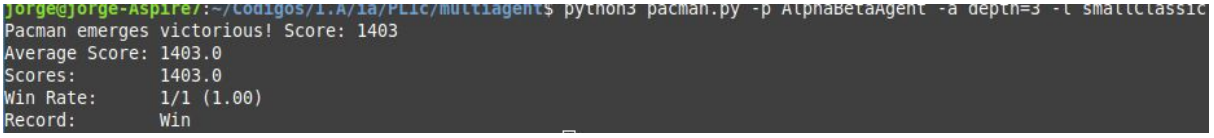




3. Podado Alpha-Beta:

Este algoritmo es exactamente igual al anterior pero aplicando una regla de podado, el cual no tiene efecto sobre el valor min-max calculado para la raíz.

Del mismo modo aplicaremos el algoritmo proporcionado, devolviendo v en caso de que el valor del mismo sea mayor a beta para el max_value o menor que alpha en min_value. Finalmente modificaremos los valores opuestos siendo estos los el máximo entre alpha e v y el mínimo entre beta e v en min.



4. Expectimax:

Este último algoritmo, sustituye nuestra función min por una "aleatoria", siendo esta compuesta de la misma manera, a diferencia de que v lo calcularemos utilizando el valor proporcionado a través de la recursividad tanto para exp_value como max_value dividido entre el número de hijos (media ponderada de los hijos).

Agustín Barruti
Jorge Salazar
Ander Carrera

```
jorge@jorge-Aspire7:~/Codigos/I.A/ia/PL1c/multiagent$ python3 pacman.py -p ExpectimaxAgent -l minimaxClassic -a depth=3
çPacman emerges victorious! Score: 514
Average Score: 514.0
Scores:        514.0
Win Rate:      1/1 (1.00)
Record:        Win
```

```
jorge@jorge-Aspire7:~/Codigos/I.A/ia/PL1c/multiagent$ python3 pacman.py -p AlphaBetaAgent -l trappedClassic -a depth=3 -q -n 10
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Pacman died! Score: -501
Average Score: -501.0
Scores:        -501.0, -501.0, -501.0, -501.0, -501.0, -501.0, -501.0, -501.0, -501.0, -501.0
Win Rate:      0/10 (0.00)
Record:        Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss, Loss
jorge@jorge-Aspire7:~/Codigos/I.A/ia/PL1c/multiagent$ python3 pacman.py -p ExpectimaxAgent -l trappedClassic -a depth=3 -q -n 10
Pacman died! Score: -502
Pacman died! Score: -502
Pacman died! Score: -502
Pacman died! Score: -502
Pacman emerges victorious! Score: 532
Pacman emerges victorious! Score: 532
Pacman emerges victorious! Score: 532
Pacman died! Score: -502
Pacman emerges victorious! Score: 532
Pacman died! Score: -502
Average Score: -88.4
Scores:        -502.0, -502.0, -502.0, -502.0, 532.0, 532.0, 532.0, -502.0, 532.0, -502.0
Win Rate:      4/10 (0.40)
Record:        Loss, Loss, Loss, Loss, Win, Win, Win, Loss, Win, Loss
```

5. Función de evaluación:

En este ejercicio, nos hemos basado en la misma función de evaluación para la pregunta 1, pero añadiendo más valores a tener en cuenta, como que se coma las cápsulas para comer fantasmas, si los fantasmas están asustados que busque comérselos...
También hemos tenido en cuenta que si el fantasma está a una distancia menor de 3 casillas, se le tenga en cuenta, y si no lo está que lo ignore y siga preocupándose de la comida y las cápsulas. Los valores “aleatorios” mencionados en la pregunta 1, también se han ajustado para un mejor funcionamiento en esta pregunta. A la distancia de las cápsulas se le ha dado un peso multiplicado por 2, ya que tiene una importancia grande, pero no mayor que la propia comida, que en esta pregunta tiene un peso multiplicado por 4 (a diferencia de 3, como en la pregunta 1). A la distancia de los fantasmas, el valor recíproco se multiplica por 30 en vez de por 10. El valor total, que se basaba en la cantidad de comida restante, de la misma forma se le ha multiplicado por un valor grande para que los valores previamente calculados le influyan levemente, pero en este caso es por 60, en vez de 50:

```
valorTotal = capsulasCerca + comidaCerca + fantasCerca -
(60*len(listaComida))
```

Agustín Barruti
Jorge Salazar
Ander Carrera

AUTOGRADER

Autograder q1:

```
jorge@jorge-Aspire7:~/Codigos/1.A/ia/PLIc/multiagent$ python3 autograder.py -q q1 --no-graphics
autograder.py:17: DeprecationWarning: the imp module is deprecated in favour of importlib; see
  import imp
Starting on 3-21 at 22:13:29

Question q1
===========

Pacman emerges victorious! Score: 1249
Pacman emerges victorious! Score: 1258
Pacman emerges victorious! Score: 1238
Pacman emerges victorious! Score: 1247
Pacman emerges victorious! Score: 1248
Pacman emerges victorious! Score: 1256
Pacman emerges victorious! Score: 1251
Pacman emerges victorious! Score: 1255
Pacman emerges victorious! Score: 1255
Pacman emerges victorious! Score: 1246
Average Score: 1250.3
Scores:        1249.0, 1258.0, 1238.0, 1247.0, 1248.0, 1256.0, 1251.0, 1255.0, 1255.0, 1246.0
Win Rate:      10/10 (1.00)
Record:        Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
*** PASS: test_cases/q1/grade-agent.test (4 of 4 points)
***     1250.3 average score (2 of 2 points)
***         Grading scheme:
***          < 500:  0 points
***         >= 500:  1 points
***         >= 1000:  2 points
***     10 games not timed out (0 of 0 points)
***         Grading scheme:
***          < 10:  fail
***         >= 10:  0 points
***     10 wins (2 of 2 points)
```

Agustín Barruti
Jorge Salazar
Ander Carrera

Autograder q2:

```
Question q2
===========

*** PASS: test_cases/q2/0-eval-function-lose-states-1.test
*** PASS: test_cases/q2/0-eval-function-lose-states-2.test
*** PASS: test_cases/q2/0-eval-function-win-states-1.test
*** PASS: test_cases/q2/0-eval-function-win-states-2.test
*** PASS: test_cases/q2/0-lecture-6-tree.test
*** PASS: test_cases/q2/0-small-tree.test
*** PASS: test_cases/q2/1-1-minmax.test
*** PASS: test_cases/q2/1-2-minmax.test
*** PASS: test_cases/q2/1-3-minmax.test
*** PASS: test_cases/q2/1-4-minmax.test
*** PASS: test_cases/q2/1-5-minmax.test
*** PASS: test_cases/q2/1-6-minmax.test
*** PASS: test_cases/q2/1-7-minmax.test
*** PASS: test_cases/q2/1-8-minmax.test
*** PASS: test_cases/q2/2-1a-vary-depth.test
*** PASS: test_cases/q2/2-1b-vary-depth.test
*** PASS: test_cases/q2/2-2a-vary-depth.test
*** PASS: test_cases/q2/2-2b-vary-depth.test
*** PASS: test_cases/q2/2-3a-vary-depth.test
*** PASS: test_cases/q2/2-3b-vary-depth.test
*** PASS: test_cases/q2/2-4a-vary-depth.test
*** PASS: test_cases/q2/2-4b-vary-depth.test
*** PASS: test_cases/q2/2-one-ghost-3level.test
*** PASS: test_cases/q2/3-one-ghost-4level.test
*** PASS: test_cases/q2/4-two-ghosts-3level.test
*** PASS: test_cases/q2/5-two-ghosts-4level.test
*** PASS: test_cases/q2/6-tied-root.test
*** PASS: test_cases/q2/7-1a-check-depth-one-ghost.test
*** PASS: test_cases/q2/7-1b-check-depth-one-ghost.test
*** PASS: test_cases/q2/7-1c-check-depth-one-ghost.test
*** PASS: test_cases/q2/7-2a-check-depth-two-ghosts.test
*** PASS: test_cases/q2/7-2b-check-depth-two-ghosts.test
*** PASS: test_cases/q2/7-2c-check-depth-two-ghosts.test
*** Running MinimaxAgent on smallClassic 1 time(s).
Pacman died! Score: 84
Average Score: 84.0
Scores:        84.0
Win Rate:      0/1 (0.00)
Record:        Loss
*** Finished running MinimaxAgent on smallClassic after 0 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases/q2/8-pacman-game.test

### Question q2: 5/5 ###


Finished at 22:16:53

Provisional grades
==================
Question q2: 5/5
------------------
Total: 5/5
```

Agustín Barruti
Jorge Salazar
Ander Carrera

Autograder q3:

```
Question q3
===========

*** PASS: test_cases/q3/0-eval-function-lose-states-1.test
*** PASS: test_cases/q3/0-eval-function-lose-states-2.test
*** PASS: test_cases/q3/0-eval-function-win-states-1.test
*** PASS: test_cases/q3/0-eval-function-win-states-2.test
*** PASS: test_cases/q3/0-lecture-6-tree.test
*** PASS: test_cases/q3/0-small-tree.test
*** PASS: test_cases/q3/1-1-minmax.test
*** PASS: test_cases/q3/1-2-minmax.test
*** PASS: test_cases/q3/1-3-minmax.test
*** PASS: test_cases/q3/1-4-minmax.test
*** PASS: test_cases/q3/1-5-minmax.test
*** PASS: test_cases/q3/1-6-minmax.test
*** PASS: test_cases/q3/1-7-minmax.test
*** PASS: test_cases/q3/1-8-minmax.test
*** PASS: test_cases/q3/2-1a-vary-depth.test
*** PASS: test_cases/q3/2-1b-vary-depth.test
*** PASS: test_cases/q3/2-2a-vary-depth.test
*** PASS: test_cases/q3/2-2b-vary-depth.test
*** PASS: test_cases/q3/2-3a-vary-depth.test
*** PASS: test_cases/q3/2-3b-vary-depth.test
*** PASS: test_cases/q3/2-4a-vary-depth.test
*** PASS: test_cases/q3/2-4b-vary-depth.test
*** PASS: test_cases/q3/2-one-ghost-3level.test
*** PASS: test_cases/q3/3-one-ghost-4level.test
*** PASS: test_cases/q3/4-two-ghosts-3level.test
*** PASS: test_cases/q3/5-two-ghosts-4level.test
*** PASS: test_cases/q3/6-tied-root.test
*** PASS: test_cases/q3/7-1a-check-depth-one-ghost.test
*** PASS: test_cases/q3/7-1b-check-depth-one-ghost.test
*** PASS: test_cases/q3/7-1c-check-depth-one-ghost.test
*** PASS: test_cases/q3/7-2a-check-depth-two-ghosts.test
*** PASS: test_cases/q3/7-2b-check-depth-two-ghosts.test
*** PASS: test_cases/q3/7-2c-check-depth-two-ghosts.test
*** Running AlphaBetaAgent on smallClassic 1 time(s).
Pacman died! Score: 84
Average Score: 84.0
Scores:        84.0
Win Rate:      0/1 (0.00)
Record:        Loss
*** Finished running AlphaBetaAgent on smallClassic after 1 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases/q3/8-pacman-game.test

### Question q3: 5/5 ###
```

Agustín Barruti
Jorge Salazar
Ander Carrera

Autograder q4:

```
Question q4
===========

*** PASS: test_cases/q4/0-eval-function-lose-states-1.test
*** PASS: test_cases/q4/0-eval-function-lose-states-2.test
*** PASS: test_cases/q4/0-eval-function-win-states-1.test
*** PASS: test_cases/q4/0-eval-function-win-states-2.test
*** PASS: test_cases/q4/0-expectimax1.test
*** PASS: test_cases/q4/1-expectimax2.test
*** PASS: test_cases/q4/2-one-ghost-3level.test
*** PASS: test_cases/q4/3-one-ghost-4level.test
*** PASS: test_cases/q4/4-two-ghosts-3level.test
*** PASS: test_cases/q4/5-two-ghosts-4level.test
*** PASS: test_cases/q4/6-1a-check-depth-one-ghost.test
*** PASS: test_cases/q4/6-1b-check-depth-one-ghost.test
*** PASS: test_cases/q4/6-1c-check-depth-one-ghost.test
*** PASS: test_cases/q4/6-2a-check-depth-two-ghosts.test
*** PASS: test_cases/q4/6-2b-check-depth-two-ghosts.test
*** PASS: test_cases/q4/6-2c-check-depth-two-ghosts.test
*** Running ExpectimaxAgent on smallClassic 1 time(s).
Pacman died! Score: 84
Average Score: 84.0
Scores:        84.0
Win Rate:      0/1 (0.00)
Record:        Loss
*** Finished running ExpectimaxAgent on smallClassic after 1 seconds.
*** Won 0 out of 1 games. Average score: 84.000000 ***
*** PASS: test_cases/q4/7-pacman-game.test

### Question q4: 5/5 ###
```

Agustín Barruti
Jorge Salazar
Ander Carrera

Autograder q5:

```
Question q5
===========

Pacman emerges victorious! Score: 1261
Pacman emerges victorious! Score: 975
Pacman emerges victorious! Score: 953
Pacman emerges victorious! Score: 1139
Pacman emerges victorious! Score: 1037
Pacman emerges victorious! Score: 1118
Pacman emerges victorious! Score: 970
Pacman emerges victorious! Score: 910
Pacman emerges victorious! Score: 966
Pacman emerges victorious! Score: 1116
Average Score: 1044.5
Scores:        1261.0, 975.0, 953.0, 1139.0, 1037.0, 1118.0, 970.0, 910.0, 966.0, 1116.0
Win Rate:      10/10 (1.00)
Record:        Win, Win, Win, Win, Win, Win, Win, Win, Win, Win
*** PASS: test_cases/q5/grade-agent.test (6 of 6 points)
***     1044.5 average score (2 of 2 points)
***         Grading scheme:
***          < 500:  0 points
***         >= 500:  1 points
***         >= 1000:  2 points
***     10 games not timed out (1 of 1 points)
***         Grading scheme:
***          < 0:  fail
***         >= 0:  0 points
***         >= 10:  1 points
***     10 wins (3 of 3 points)
***         Grading scheme:
***          < 1:  fail
***         >= 1:  1 points
***         >= 5:  2 points
***         >= 10:  3 points

### Question q5: 6/6 ###


Finished at 21:56:07

Provisional grades
==================
Question q5: 6/6
------------------
Total: 6/6
```