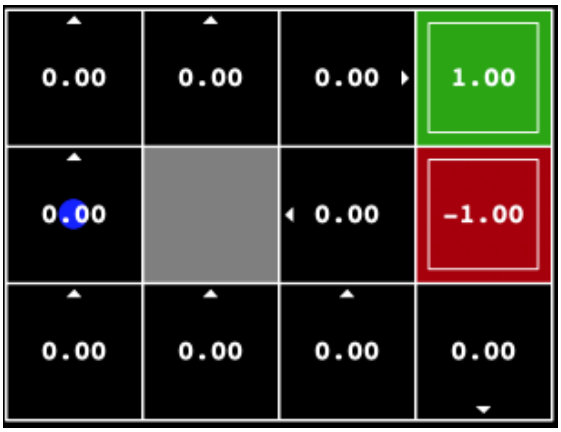
Laura Quiroga - 201922965

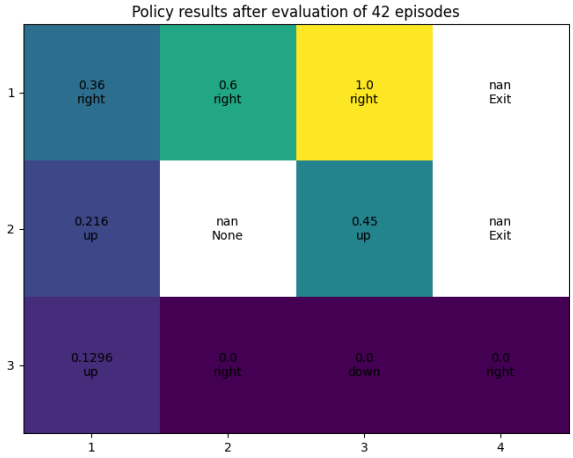
# Assignment 6: Q-Learning

Escenario 1: Gridworld

## 1. Gridworld pequeño



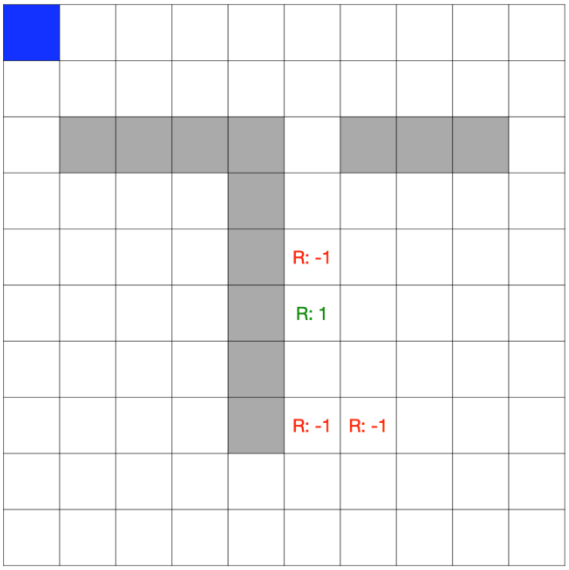

Resultados de la política:



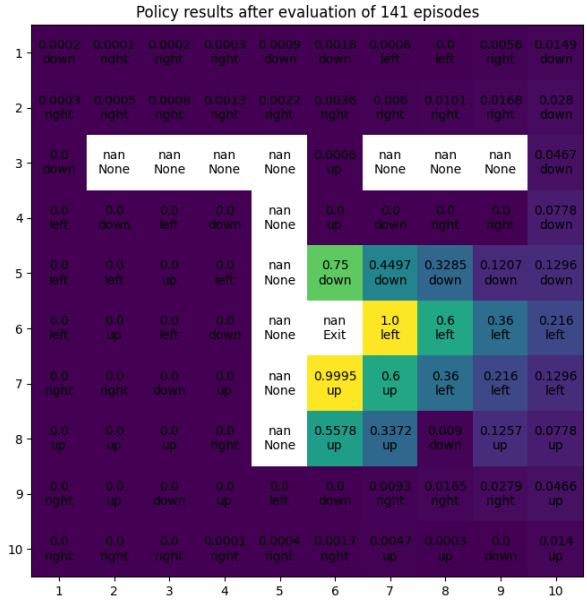

Q-tabla:

| State | up | right | down | left |
|---|---|---|---|---|
| (1, 1) | 0.108 | 0.36 | 0.0 | 0.0 |
| (1, 2) | 0.216 | 0.0 | 0.05832 | 0.0 |
| (1, 3) | 0.1296 | 0.0 | 0.067927 | 0.043282 |
| (2, 1) | 0.224975 | 0.6 | 0.179998 | 0 |
| (2, 2) | 0 | 0 | 0 | 0 |
| (2, 3) | 0.0 | 0.0 | 0.0 | 0.0 |
| (3, 1) | 0.449982 | 1.0 | 0.09 | 0.179999 |
| (3, 2) | 0.45 | 0 | 0.0 | 0.0 |
| (3, 3) | 0 | 0.0 | 0 | 0 |
| (4, 1) | 0 | 0 | 0 | 0 |
| (4, 2) | 0 | 0 | 0 | 0 |
| (4, 3) | -0.996094 | 0.0 | 0.0 | 0.0 |

## 2. Gridworld 10x10




Resultados de la política:



Q-tabla:

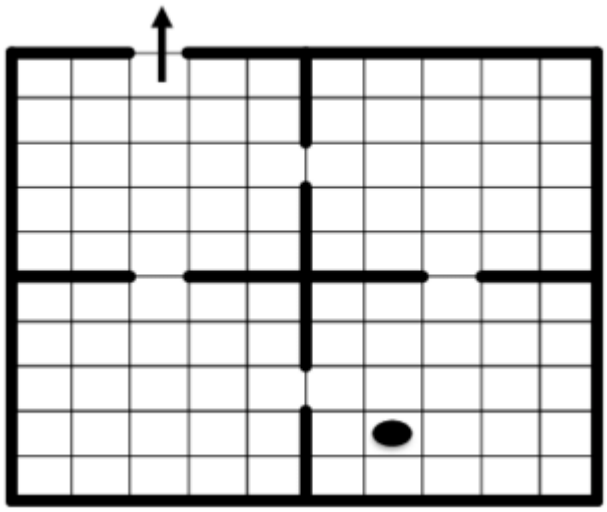
| State | up | right | down | left |
|---|---|---|---|---|
| (1, 1) | 3.6e-05 | 1.1e-05 | 0.000168 | 4.3e-05 |
| (1, 2) | 0.0 | 0.000282 | 0.0 | 0.000124 |
| (1, 3) | 0.0 | 0.0 | 0.0 | 0.0 |
| (1, 4) | 0.0 | 0.0 | 0.0 | 0.0 |
| (1, 5) | 0.0 | 0.0 | 0.0 | 0.0 |
| (1, 6) | 0.0 | 0.0 | 0.0 | 0.0 |
| (1, 7) | 0.0 | 0.0 | 0.0 | 0.0 |
| (1, 8) | 0.0 | 0.0 | 0.0 | 0.0 |
| (1, 9) | 0.0 | 0.0 | 0.0 | 0.0 |
| (1, 10) | 0.0 | 0.0 | 0.0 | 0.0 |
| (2, 1) | 2e-06 | 5.8e-05 | 0.0 | 7e-06 |
| (2, 2) | 0.0 | 0.00047 | 0.0 | 0.000127 |
| (2, 3) | 0 | 0 | 0 | 0 |
| (2, 4) | 0.0 | 0.0 | 0.0 | 0.0 |
| (2, 5) | 0.0 | 0.0 | 0.0 | 0.0 |
| (2, 6) | 0.0 | 0.0 | 0.0 | 0.0 |
| (2, 7) | 0.0 | 0.0 | 0.0 | 0.0 |
| (2, 8) | 0.0 | 0.0 | 0.0 | 0.0 |
| (2, 9) | 0.0 | 0.0 | 0.0 | 0.0 |
| (2, 10) | 0.0 | 1e-06 | 0.0 | 0.0 |
| (3, 1) | 0 | 0.000154 | 0 | 0 |
| (3, 2) | 1.7e-05 | 0.000783 | 4.6e-05 | 0.00015 |
| (3, 3) | 0 | 0 | 0 | 0 |
| (3, 4) | 0.0 | 0.0 | 0.0 | 0.0 |
| (3, 5) | 0.0 | 0.0 | 0.0 | 0.0 |
| (3, 6) | 0.0 | 0.0 | 0.0 | 0.0 |
| (3, 7) | 0.0 | 0.0 | 0.0 | 0.0 |
| (3, 8) | 0.0 | 0.0 | 0.0 | 0.0 |
| (3, 9) | 0.0 | 0.0 | 0.0 | 0.0 |
| (3, 10) | 0.0 | 1.1e-05 | 0.0 | 0.0 |
| (4, 1) | 0.000117 | 0.000282 | 7.6e-05 | 0.0 |
| (4, 2) | 0.000169 | 0.001306 | 0.000427 | 0.00025 |
| (4, 3) | 0 | 0 | 0 | 0 |
| (4, 4) | 0 | 0.0 | 0.0 | 0.0 |
| (4, 5) | 0.0 | 0.0 | 0.0 | 0.0 |
| (4, 6) | 0.0 | 0.0 | 0.0 | 0.0 |
| (4, 7) | 0.0 | 0.0 | 0.0 | 0.0 |
| (4, 8) | 0.0 | 0.0 | 0.0 | 0.0 |
| (4, 9) | 0.0 | 0.0 | 0.0 | 0.0 |
| (4, 10) | 0.0 | 8.2e-05 | 0.0 | 0.0 |
| (5, 1) | 0.00014 | 0.000146 | 0.000867 | 7.7e-05 |
| (5, 2) | 0.000141 | 0.002177 | 0.000879 | 0.000164 |
| (5, 3) | 0 | 0 | 0 | 0 |
| (5, 4) | 0 | 0 | 0 | 0 |
| (5, 5) | 0 | 0 | 0 | 0 |
| (5, 6) | 0 | 0 | 0 | 0 |
| (5, 7) | 0 | 0 | 0 | 0 |
| (5, 8) | 0 | 0 | 0 | 0 |
| (5, 9) | 0.0 | 0.0 | 0.0 | 0.0 |
| (5, 10) | 0.0 | 0.000417 | 0.0 | 0.0 |

| State | up | right | down | left |
|---|---|---|---|---|
| (6, 1) | 0 | 0.0 | 0.001809 | 0.0 |
| (6, 2) | 0.000189 | 0.003628 | 9.1e-05 | 0.001238 |
| (6, 3) | 0.000552 | 0.0 | 0.0 | 0 |
| (6, 4) | 0 | 0 | -0.35 | 0 |
| (6, 5) | 0 | 0 | 0.75 | 0 |
| (6, 6) | 0 | 0 | 0 | 0 |
| (6, 7) | 0.999512 | 0 | -0.41 | 0 |
| (6, 8) | 0.557812 | 0 | 0 | -0.5 |
| (6, 9) | -0.712813 | 0.0 | 0.0 | 0.0 |
| (6, 10) | 0.0 | 0.001747 | 0.0 | 0.0 |
| (7, 1) | 7e-06 | 0.0 | 0 | 0.00056 |
| (7, 2) | 8.9e-05 | 0.006047 | 0.001635 | 0.000605 |
| (7, 3) | 0 | 0 | 0 | 0 |
| (7, 4) | 0 | 0 | 0 | 0 |
| (7, 5) | 0 | 0.0 | 0.449707 | -0.5 |
| (7, 6) | 0.089824 | 0.356648 | 0.3375 | 1.0 |
| (7, 7) | 0.6 | 0.188817 | -0.405708 | 0.577734 |
| (7, 8) | 0.337153 | 0 | 0 | -0.455 |
| (7, 9) | -0.830232 | 0.009275 | 0.000997 | 0.0 |
| (7, 10) | 0.004694 | 0.0 | 7.8e-05 | 0.0 |
| (8, 1) | 0.0 | 0.0 | 0.0 | 7e-06 |
| (8, 2) | 0.0 | 0.010078 | 0.001061 | 0.001661 |
| (8, 3) | 0 | 0 | 0 | 0 |
| (8, 4) | 0 | 0 | 0 | 0 |
| (8, 5) | 0 | 0 | 0.328527 | 0 |
| (8, 6) | 0.132675 | 0.107043 | 0.162 | 0.6 |
| (8, 7) | 0.264375 | 0.121499 | 0.002976 | 0.36 |
| (8, 8) | 0 | 0 | 0.009027 | 0 |
| (8, 9) | 0.003016 | 0.016463 | 0 | 0.002458 |
| (8, 10) | 0.000283 | 0 | 0 | 0.0 |
| (9, 1) | 0.0 | 0.00562 | 0.0 | 0.0 |
| (9, 2) | 0.001855 | 0.016796 | 0.007983 | 0.003563 |
| (9, 3) | 0 | 0 | 0 | 0 |
| (9, 4) | 0 | 0 | 0 | 0 |
| (9, 5) | 0 | 0 | 0.120706 | 0.0 |
| (9, 6) | 0 | 0.046397 | 0.0 | 0.36 |
| (9, 7) | 0.059694 | 0.069345 | 0.06375 | 0.216 |
| (9, 8) | 0.12574 | 0 | 0 | 0 |
| (9, 9) | 0.0 | 0.027857 | 0.0 | 0 |
| (9, 10) | 0.0 | 0.0 | 0.0 | 0.0 |
| (10, 1) | 0.0 | 0.0 | 0.014933 | 0.0 |
| (10, 2) | 0.002813 | 0.01147 | 0.027994 | 0.000132 |
| (10, 3) | 0.011786 | 0.023085 | 0.046656 | 0.017775 |
| (10, 4) | 0 | 0.023328 | 0.07776 | 0 |
| (10, 5) | 0.033054 | 0.020995 | 0.1296 | 0.013259 |
| (10, 6) | 0.013989 | 0.082295 | 0.07776 | 0.216 |
| (10, 7) | 0.040631 | 0.075303 | 0.023293 | 0.1296 |
| (10, 8) | 0.077758 | 0.0 | 0.018401 | 0 |
| (10, 9) | 0.04664 | 0.0 | 0.0 | 0.0 |
| (10, 10) | 0.013968 | 0.0 | 0.0 | 0.0 |

### 3. Análisis de resultados:

Se implementó el algoritmo de Q-learning y se evaluó sobre dos escenarios de Gridworld. Para ambos se utilizaron los mismos parámetros de aprendizaje: alpha = 0.5, gamma = 0.6 y épsilon = 0.1. Se evaluaron episodios del algoritmo hasta su convergencia, la cual fue medida como la falta de cambios en la política durante 30 episodios consecutivos; en cada gráfica, el título de la figura indica el número de episodios que se evaluaron hasta alcanzar este criterio de convergencia. Como se puede ver en la visualización gráfica de los resultados de la política y de la q-tabla, los resultados son los esperados y son consistentes con los resultados obtenidos con otros algoritmos. La política es la óptima principalmente para los estados más cercanos a los estados terminales, que son los únicos que tienen una recompensa.

Escenario 2: Laberinto de cuartos



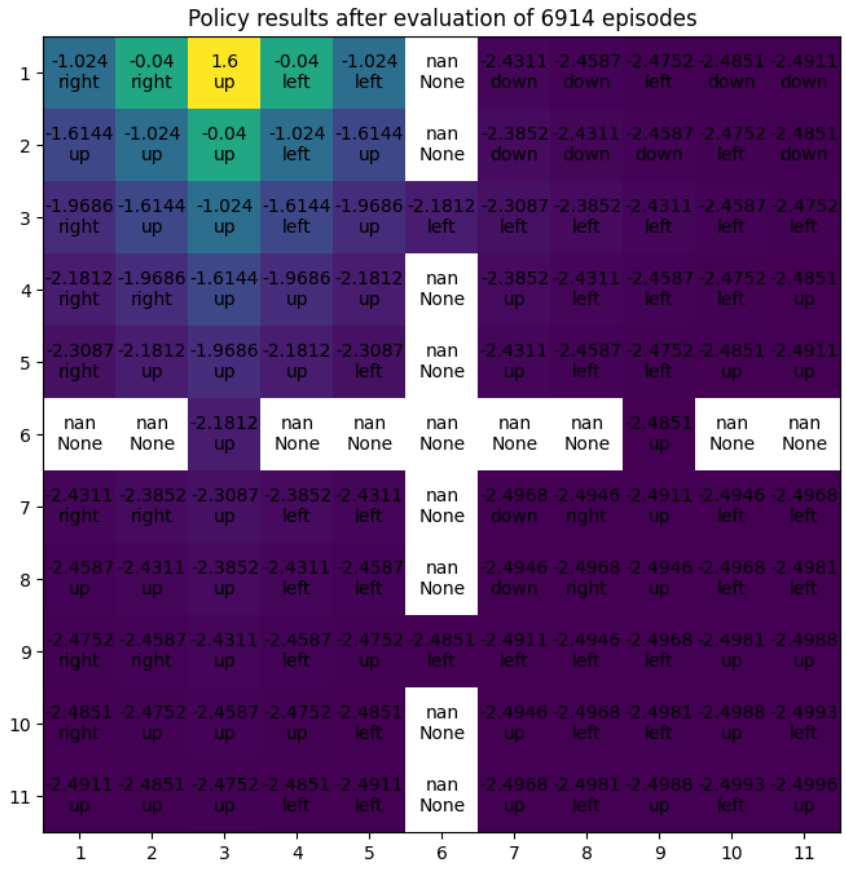
1. **Explicación de la ejecución:**

Se implementó el algoritmo de Q-learning para el escenario del laberinto de cuartos, en donde independientemente de la posición inicial del agente (aleatoria), su objetivo es llegar lo más rápido posible a la salida en el cuadro superior izquierdo. Para esto, se agregaron casillas de tipo obstáculo ('o') a la grilla para representar las paredes del laberinto, de forma que la representación de esta matriz 10x10 en realidad se implementó como una matriz 11x11 con una fila (o columna, respectivamente) de casillas no pisables.

Se utilizaron los siguientes parámetros de aprendizaje: alpha = 0.5, gamma = 0.6, épsilon = 0.1. Y se decidió darle una recompensa de -1 al agente por cada movimiento en el tablero a excepción de la acción que le permite salir del tablero ('up' desde la casilla de la entrada, que tiene una recompensa de 1, en cuyo caso el máximo q-valor del siguiente estado también toma un valor de 1. Este sistema de recompensas le ayuda al agente a siempre buscar la ruta más corta hacia la salida desde cada posición. El algoritmo se evaluó hasta su convergencia, que se tomó como el no cambio de la política durante 30 episodios consecutivos.

Los resultados a continuación muestran los resultados de la política y los valores finales en la q-tabla. Allí se puede ver que exitosamente se encontró una política que lleva al agente a la salida lo más rápido posible (i.e. en la menor cantidad de movimientos).

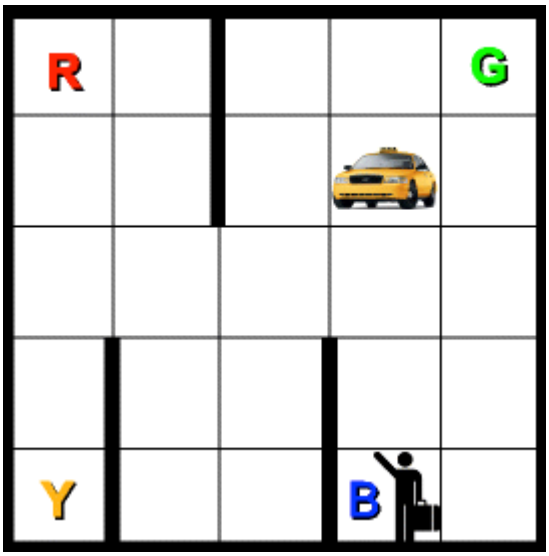
## 2. Resultados de la política:



## 3. Q-tabla:

| State | up | right | down | left |
|---|---|---|---|---|
| (1, 1) | -1.531425 | -1.024 | -1.904141 | -1.589548 |
| (1, 2) | -1.6144 | -1.6144 | -2.117942 | -1.82472 |
| (1, 3) | -1.96864 | -1.96864 | -2.295175 | -2.147559 |
| (1, 4) | -2.181184 | -2.181184 | -2.383098 | -2.268852 |
| (1, 5) | -2.30871 | -2.30871 | -2.370585 | -2.367635 |
| (1, 6) | 0 | 0 | 0 | 0 |
| (1, 7) | -2.457855 | -2.431136 | -2.474933 | -2.458113 |
| (1, 8) | -2.458681 | -2.458681 | -2.483511 | -2.47498 |
| (1, 9) | -2.475209 | -2.475209 | -2.490353 | -2.48301 |
| (1, 10) | -2.485125 | -2.485125 | -2.492988 | -2.485197 |
| (1, 11) | -2.491075 | -2.491075 | -2.493223 | -2.493007 |
| (2, 1) | -1.023968 | -0.04 | -1.606866 | -1.614332 |
| (2, 2) | -1.024 | -1.024 | -1.968639 | -1.968599 |
| (2, 3) | -1.6144 | -1.6144 | -2.180401 | -2.176962 |
| (2, 4) | -1.96864 | -1.96864 | -2.305854 | -2.308457 |
| (2, 5) | -2.181184 | -2.181184 | -2.303863 | -2.344818 |
| (2, 6) | 0 | 0 | 0 | 0 |
| (2, 7) | -2.430702 | -2.385226 | -2.458615 | -2.449644 |
| (2, 8) | -2.431136 | -2.431136 | -2.475166 | -2.475146 |
| (2, 9) | -2.458681 | -2.458681 | -2.485119 | -2.484412 |
| (2, 10) | -2.475209 | -2.475209 | -2.489659 | -2.48853 |
| (2, 11) | -2.485125 | -2.485125 | -2.489564 | -2.492155 |
| (3, 1) | 1.6 | -1.024 | -1.024 | -1.024 |
| (3, 2) | -0.04 | -1.6144 | -1.6144 | -1.6144 |
| (3, 3) | -1.024 | -1.96864 | -1.96864 | -1.96864 |
| (3, 4) | -1.6144 | -2.181184 | -2.181184 | -2.181184 |
| (3, 5) | -1.96864 | -2.30871 | -2.30871 | -2.30871 |
| (3, 6) | -2.181184 | -2.30871 | -2.385226 | -2.30871 |
| (3, 7) | -2.30871 | -2.431136 | -2.431136 | -2.431136 |
| (3, 8) | -2.385226 | -2.458681 | -2.458681 | -2.458681 |
| (3, 9) | -2.431136 | -2.475089 | -2.475208 | -2.475209 |
| (3, 10) | -2.458681 | -2.485028 | -2.484901 | -2.485094 |
| (3, 11) | -2.475209 | -2.490996 | -2.482985 | -2.487688 |
| (4, 1) | -1.023992 | -1.609171 | -1.612223 | -0.04 |
| (4, 2) | -1.024 | -1.96864 | -1.96864 | -1.024 |
| (4, 3) | -1.6144 | -2.181152 | -2.181175 | -1.6144 |
| (4, 4) | -1.96864 | -2.30692 | -2.306958 | -1.96864 |
| (4, 5) | -2.181184 | -2.3511 | -2.308107 | -2.181184 |
| (4, 6) | 0 | 0 | 0 | 0 |
| (4, 7) | -2.431136 | -2.458681 | -2.458681 | -2.385226 |
| (4, 8) | -2.431136 | -2.475209 | -2.475209 | -2.431136 |
| (4, 9) | -2.458681 | -2.485125 | -2.485125 | -2.458681 |
| (4, 10) | -2.475209 | -2.491026 | -2.490986 | -2.475209 |
| (4, 11) | -2.485125 | -2.494311 | -2.490117 | -2.485125 |
| (5, 1) | -1.602075 | -1.49705 | -1.929819 | -1.024 |
| (5, 2) | -1.6144 | -1.96864 | -2.181184 | -1.6144 |
| (5, 3) | -1.96864 | -2.30871 | -2.30871 | -1.96864 |
| (5, 4) | -2.181184 | -2.299067 | -2.350815 | -2.181184 |
| (5, 5) | -2.30871 | -2.349031 | -2.379056 | -2.30871 |
| (5, 6) | 0 | 0 | 0 | 0 |
| (5, 7) | -2.458072 | -2.458374 | -2.471255 | -2.431136 |
| (5, 8) | -2.458681 | -2.473804 | -2.482858 | -2.458681 |
| (5, 9) | -2.475209 | -2.491075 | -2.491075 | -2.475209 |
| (5, 10) | -2.485125 | -2.490551 | -2.494014 | -2.485125 |
| (5, 11) | -2.491075 | -2.494608 | -2.49409 | -2.491075 |
| (6, 1) | 0 | 0 | 0 | 0 |
| (6, 2) | 0 | 0 | 0 | 0 |
| (6, 3) | -2.30871 | -2.385226 | -2.30871 | -2.181184 |
| (6, 4) | 0 | 0 | 0 | 0 |
| (6, 5) | 0 | 0 | 0 | 0 |
| (6, 6) | 0 | 0 | 0 | 0 |
| (6, 7) | 0 | 0 | 0 | 0 |
| (6, 8) | 0 | 0 | 0 | 0 |
| (6, 9) | -2.491075 | -2.494645 | -2.491075 | -2.485125 |
| (6, 10) | 0 | 0 | 0 | 0 |
| (6, 11) | 0 | 0 | 0 | 0 |
| (7, 1) | -2.456809 | -2.466076 | -2.431136 | -2.457745 |
| (7, 2) | -2.458676 | -2.450861 | -2.385226 | -2.42972 |
| (7, 3) | -2.431136 | -2.431136 | -2.431136 | -2.30871 |
| (7, 4) | -2.385226 | -2.458669 | -2.458675 | -2.431133 |
| (7, 5) | -2.431136 | -2.475124 | -2.458613 | -2.458183 |
| (7, 6) | 0 | 0 | 0 | 0 |
| (7, 7) | -2.49745 | -2.496787 | -2.496787 | -2.497439 |
| (7, 8) | -2.495387 | -2.497875 | -2.494645 | -2.496724 |
| (7, 9) | -2.496787 | -2.496786 | -2.496787 | -2.491075 |
| (7, 10) | -2.494645 | -2.498071 | -2.498067 | -2.496785 |
| (7, 11) | -2.496787 | -2.498754 | -2.497828 | -2.498005 |
| (8, 1) | -2.46959 | -2.472556 | -2.458681 | -2.458681 |
| (8, 2) | -2.475056 | -2.475124 | -2.431136 | -2.431136 |
| (8, 3) | -2.458681 | -2.458681 | -2.458681 | -2.385226 |
| (8, 4) | -2.431136 | -2.475209 | -2.475209 | -2.431136 |
| (8, 5) | -2.458681 | -2.480409 | -2.474974 | -2.458681 |
| (8, 6) | 0 | 0 | 0 | 0 |
| (8, 7) | -2.496638 | -2.494645 | -2.497201 | -2.498046 |
| (8, 8) | -2.496787 | -2.496787 | -2.496787 | -2.496787 |
| (8, 9) | -2.498013 | -2.497686 | -2.49785 | -2.494645 |
| (8, 10) | -2.496787 | -2.498735 | -2.498797 | -2.496787 |
| (8, 11) | -2.498072 | -2.499133 | -2.498817 | -2.498072 |
| (9, 1) | -2.485018 | -2.489948 | -2.475209 | -2.475209 |
| (9, 2) | -2.485018 | -2.48512 | -2.458681 | -2.458681 |
| (9, 3) | -2.475209 | -2.475209 | -2.475209 | -2.431136 |
| (9, 4) | -2.458681 | -2.485125 | -2.485125 | -2.458681 |
| (9, 5) | -2.475209 | -2.491075 | -2.491075 | -2.475209 |
| (9, 6) | -2.485125 | -2.491075 | -2.494645 | -2.491075 |
| (9, 7) | -2.491075 | -2.496787 | -2.496787 | -2.496787 |
| (9, 8) | -2.494645 | -2.498072 | -2.498072 | -2.498071 |
| (9, 9) | -2.496787 | -2.498843 | -2.498827 | -2.496787 |
| (9, 10) | -2.498072 | -2.499299 | -2.499271 | -2.498072 |
| (9, 11) | -2.498843 | -2.499324 | -2.49928 | -2.498843 |
| (10, 1) | -2.490592 | -2.494481 | -2.485125 | -2.485125 |
| (10, 2) | -2.491021 | -2.490979 | -2.475209 | -2.475209 |
| (10, 3) | -2.485123 | -2.485115 | -2.485125 | -2.458681 |
| (10, 4) | -2.475209 | -2.490762 | -2.489284 | -2.475209 |
| (10, 5) | -2.485125 | -2.494618 | -2.490838 | -2.485125 |
| (10, 6) | 0 | 0 | 0 | 0 |
| (10, 7) | -2.496783 | -2.498071 | -2.49807 | -2.494645 |
| (10, 8) | -2.496787 | -2.498816 | -2.498842 | -2.496787 |
| (10, 9) | -2.498072 | -2.499304 | -2.499286 | -2.498072 |
| (10, 10) | -2.498843 | -2.499215 | -2.499562 | -2.498843 |
| (10, 11) | -2.499306 | -2.499562 | -2.499574 | -2.499306 |
| (11, 1) | -2.492821 | -2.494302 | -2.491075 | -2.491075 |
| (11, 2) | -2.489417 | -2.490711 | -2.485125 | -2.485125 |
| (11, 3) | -2.490154 | -2.485055 | -2.489268 | -2.475209 |
| (11, 4) | -2.485125 | -2.490946 | -2.493322 | -2.485125 |
| (11, 5) | -2.491075 | -2.494151 | -2.493922 | -2.491075 |
| (11, 6) | 0 | 0 | 0 | 0 |
| (11, 7) | -2.497909 | -2.498054 | -2.498749 | -2.496787 |
| (11, 8) | -2.498072 | -2.498837 | -2.499262 | -2.498072 |
| (11, 9) | -2.498843 | -2.499284 | -2.499257 | -2.498843 |
| (11, 10) | -2.499306 | -2.499446 | -2.499744 | -2.499306 |
| (11, 11) | -2.499584 | -2.499634 | -2.499738 | -2.499584 |

Escenario 3: Taxi



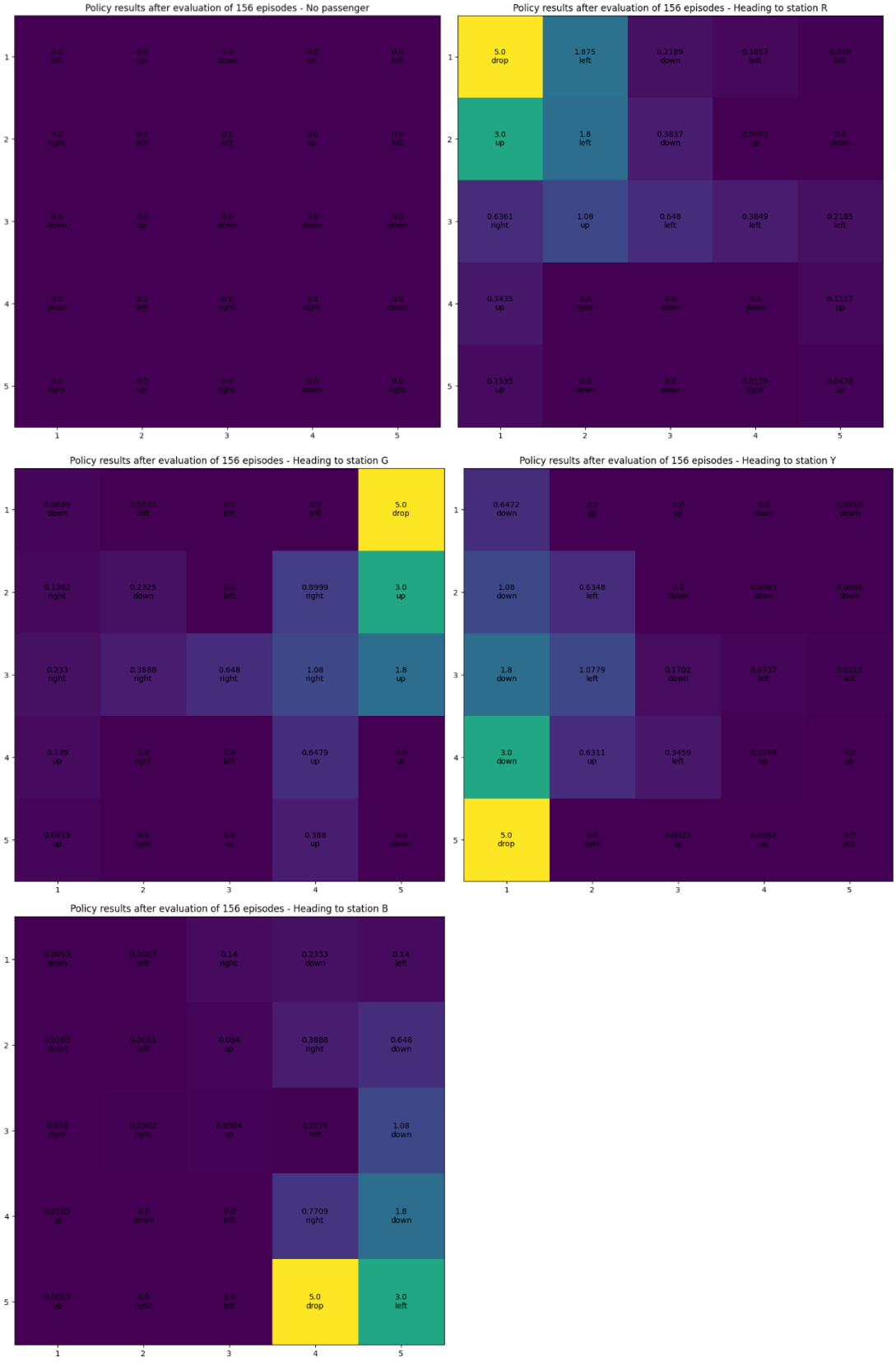

1. **Explicación de la ejecución:**

Para la implementación del escenario del taxi los estados del agente se tomaron como una 3-tupla: (posición en x, posición en y, estación de destino). El tercer valor se tomó como el destino del pasajero recogido por el taxi, que era vacío en el caso de que el taxi no hubiera recogido aún a su pasajero. Cada episodio se tomó como el desplazamiento del taxi desde su posición inicial (aleatoria) en el tablero hasta recoger a un pasajero en una estación desconocida para el agente y concluye cuando deja al pasajero en su estación de destino.

Nuevamente se evaluaron episodios hasta la convergencia de la política, que en este caso se tomaron como no cambios en la política durante 50 episodios consecutivos. Sin embargo, dado que cuando el taxi no lleva pasajero tampoco conoce la estación donde se encuentra el pasajero que debe recoger (la cual es aleatoria en cada iteración) y recibe recompensas negativas por intentar recoger un pasajero donde no hay recibe recompensa negativa, pero al recogerlo efectivamente recibe una recompensa positiva, no se toma en cuenta la política de los estados sin estación de destino para la evaluación de la convergencia (en caso de tener estos estados en cuenta la política no convergería).

Para la representación de las paredes en el código no se agregaron casillas extra como se hizo en el caso del laberinto, sino que se tomaron estados 'w' que indican que hay una pared (wall) a la derecha, de manera que se restringe el movimiento hacia la derecha desde estas posiciones y tampoco se permite llegar allí desde la izquierda. Los parámetros de aprendizaje utilizados fueron: alpha = 0.5, gamma=0.6 y épsilon=0.1. Los resultados de la política y de la q-tabla se muestran a continuación.

## 2. Resultados de la política:



## 3. Q-tabla:

| State | up | right | down | left | pick | drop |
|---|---|---|---|---|---|---|
| (1, 1, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -9.827925 | -10.0 |
| (1, 1, 'R') 0 | 0.0 | 0.0 | 0.0 | 0.0 | -7.484375 | 5.0 |
| (1, 1, 'G') 0.0 | 0.0 | 0.0 | 0.069918 | 0.0 | -8.75 | -9.375 |
| (1, 1, 'Y') 0.0 | 0.0 | 0.0 | 0.647212 | 0.0 | -10.0 | -10.0 |
| (1, 1, 'B') 0.0 | 0.0 | 0.0 | 0.005327 | 0.0 | -9.999268 | -9.997559 |
| (1, 2, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (1, 2, 'R') 3.0 | 0.536111 | 0.0 | 0.0 | 0.0 | -5.0 | -8.75 |
| (1, 2, 'G') 0.0 | 0.136152 | 0.0 | 0.0 | 0.0 | -10.0 | -9.999924 |
| (1, 2, 'Y') 0.14011 | 0.0 | 1.079985 | 0.444345 | 0.0 | -10.0 | -10.0 |
| (1, 2, 'B') 0.000562 | 0.0 | 0.010335 | 0.0 | 0.0 | -9.997559 | -9.999847 |
| (1, 3, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (1, 3, 'R') 0.0 | 0.636069 | 0.0 | 0.0 | 0.0 | -7.5 | -5.0 |
| (1, 3, 'G') 0.039975 | 0.232994 | 0 | 0.061197 | 0 | 0 | 0 |
| (1, 3, 'Y') 0.0 | 0.318937 | 1.8 | 0.54 | 0.0 | -9.99939 | -9.99939 |
| (1, 3, 'B') 0.0 | 0.018047 | 0.0 | 1e-06 | 0.0 | -10.0 | -10.0 |
| (1, 4, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (1, 4, 'R') 0.343499 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (1, 4, 'G') 0.138962 | 0.0 | 0.023307 | 0.035532 | 0.0 | -7.5 | -9.646011 |
| (1, 4, 'Y') 0.809111 | 0.0 | 3.0 | 0.0 | 0.0 | -9.375 | -8.75 |
| (1, 4, 'B') 0.010458 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (1, 5, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -8.608175 | -10.0 |
| (1, 5, 'R') 0.153513 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (1, 5, 'G') 0.081948 | 0.0 | 0 | 0 | 0 | 0 | 0 |
| (1, 5, 'Y') 0.0 | 0.0 | 0.0 | 0.0 | 2.15625 | -7.239922 | 5.0 |
| (1, 5, 'B') 0.005492 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (2, 1, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (2, 1, 'R') 0 | 0 | 0 | 0 | 1.875 | 0 | 0 |
| (2, 1, 'G') 0.0 | 0.0 | 0.0 | 0.0 | 0.003233 | -5.0 | -7.499775 |
| (2, 1, 'Y') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -9.921875 | -9.995117 |
| (2, 1, 'B') 0.0 | 0.0 | 0.0 | 0.0 | 0.00071 | -9.995117 | -9.84375 |
| (2, 2, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (2, 2, 'R') 0.225 | 0.0 | 0 | 1.799999 | -5.0 | -8.210053 | -8.210053 |
| (2, 2, 'G') 0.000225 | 0.069074 | 0.232515 | 0.0 | 0.0 | -9.308615 | -7.5 |
| (2, 2, 'Y') 0.0 | 0.0 | 0.0 | 0.634828 | -9.921875 | -9.921875 | -9.921875 |
| (2, 2, 'B') 0.0 | 0.0 | 0.0 | 0.003074 | -7.5 | -7.5 | -7.5 |
| (2, 3, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (2, 3, 'R') 1.079997 | 0.0 | 0.0 | 0.0 | 0.0 | -9.608475 | -8.719625 |
| (2, 3, 'G') 0.0 | 0.388799 | 0.0 | 0.0 | 0.0 | -9.84375 | -9.375 |
| (2, 3, 'Y') 0.357112 | 0.0 | 0.112167 | 1.07789 | 0.0 | -9.678532 | -9.905419 |
| (2, 3, 'B') 0.000322 | 0.030191 | 0.0 | 0.0 | 0.0 | -9.999924 | -10.0 |
| (2, 4, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (2, 4, 'R') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (2, 4, 'G') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -8.75 | -7.5 |
| (2, 4, 'Y') 0.631059 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (2, 4, 'B') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (2, 5, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (2, 5, 'R') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (2, 5, 'G') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -9.6875 | -9.921875 |
| (2, 5, 'Y') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (2, 5, 'B') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (3, 1, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (3, 1, 'R') 0.0 | 0.0 | 0.0 | 0.218928 | 0.0 | -9.990234 | -9.999995 |
| (3, 1, 'G') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -8.75 | -9.84375 |
| (3, 1, 'Y') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -9.997559 | -9.999981 |
| (3, 1, 'B') 0.04188 | 0.139967 | 0.0 | 0.0 | 0.0 | -9.99939 | -9.999695 |
| (3, 2, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (3, 2, 'R') 0.059858 | 0.0 | 0.383729 | 0.0 | 0.0 | -9.990234 | -9.738614 |
| (3, 2, 'G') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -9.84375 | -9.980469 |
| (3, 2, 'Y') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (3, 2, 'B') 0.083976 | 0.0 | 0 | 0.024556 | 0.0 | -4.986808 | -8.713228 |
| (3, 3, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (3, 3, 'R') 0 | 0 | 0 | 0.647956 | 0 | 0 | 0 |
| (3, 3, 'G') 0.0 | 0.648 | 0.0 | 0.0 | 0.0 | -7.5 | -9.182912 |
| (3, 3, 'Y') 0.0 | 0.0 | 0.170177 | 0.0 | 0.0 | -10.0 | -9.979321 |
| (3, 3, 'B') 0.050373 | 0.001075 | 0.0 | 0.0 | 0.0 | -9.999847 | -9.973689 |
| (3, 4, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (3, 4, 'R') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -9.6875 | -9.995117 |
| (3, 4, 'G') 0.0 | 0.0 | 0 | 0 | 0.0 | -5.0 | 0 |
| (3, 4, 'Y') 0.003361 | 0.0 | 0.0 | 0.0 | 0.345932 | -10.0 | -9.91314 |
| (3, 4, 'B') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -9.980469 | -9.6875 |
| (3, 5, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (3, 5, 'R') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (3, 5, 'G') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -5.0 | -5.0 |
| (3, 5, 'Y') 0.002296 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (3, 5, 'B') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -9.6875 | -9.375 |
| (4, 1, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (4, 1, 'R') 0.0 | 0.0 | 0.0 | 0.0 | 0.105671 | -9.99939 | -9.999981 |
| (4, 1, 'G') 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 | 0 |
| (4, 1, 'Y') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -9.99939 | -9.999847 |
| (4, 1, 'B') 0.006288 | 0.041294 | 0.23328 | 0.041989 | -7.396528 | -4.999453 | -4.999453 |
| (4, 2, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (4, 2, 'R') 0.000246 | 0.0 | 0.0 | 0.0 | 0.0 | -7.5 | -7.5 |
| (4, 2, 'G') 0.0 | 0.8999 | 0.0 | 0.0 | 0.0 | -8.75 | -8.75 |
| (4, 2, 'Y') 0.0 | 0.0 | 0.006286 | 0.0 | 0.0 | -10.0 | -10.0 |
| (4, 2, 'B') 0.0 | 0.3888 | 0.004698 | 0.030691 | 0.0 | -5.0 | -7.39362 |
| (4, 3, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (4, 3, 'R') 0.0 | 0 | 0.0 | 0.38487 | 0.0 | -7.5 | -7.5 |
| (4, 3, 'G') 0.0 | 1.08 | 0.192298 | 0.340198 | 0.0 | -9.999981 | -9.99939 |
| (4, 3, 'Y') 0.0 | 2.7e-05 | 0.0 | 0.073739 | 0.0 | -10.0 | -10.0 |
| (4, 3, 'B') 0.0 | 0 | 0.0 | 0.022602 | 0.0 | -5.0 | -8.75 |
| (4, 4, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (4, 4, 'R') 0 | 0.0 | 0 | 0 | 0.0 | -8.75 | -7.5 |
| (4, 4, 'G') 0.64789 | 0.0 | 0.108501 | 0.0 | 0.0 | -10.0 | -9.999999 |
| (4, 4, 'Y') 0.024788 | 0.0 | 0.0 | 0.001385 | 0.0 | -9.998539 | -9.995117 |
| (4, 4, 'B') 0.0 | 0.770923 | 0 | 0 | 0.0 | -7.361414 | 0 |
| (4, 5, '') 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | -7.160423 | -10.0 |
| (4, 5, 'R') 0.0 | 0.015937 | 0.0 | 0.0 | 0.0 | -9.6875 | -7.5 |
| (4, 5, 'G') 0.38798 | 0.0 | 0.0 | 0.0 | 0.0 | -5.0 | -5.0 |
| (4, 5, 'Y') 0.00617 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (4, 5, 'B') 0.138586 | 0.899935 | 1.499999 | 0.0 | 0.0 | 0 | 5.0 |

| State | up | right | down | left | pick | drop |
|---|---|---|---|---|---|---|
| (5, 1, '') | 0.0 | 0.0 | 0.0 | 0.0 | -4.45799 | -10.0 |
| (5, 1, 'R') | 0.0 | 0.0 | 0.0 | 0.03895 | -9.998779 | -9.960938 |
| (5, 1, 'G') | 0 | 1.5 | 0.868359 | 0 | -6.0 | 5.0 |
| (5, 1, 'Y') | 0.0 | 0.0 | 0.00158 | 0.0 | -9.84375 | -5.0 |
| (5, 1, 'B') | 0.0 | 0.0 | 0.0 | 0.139966 | -5.0 | -9.84375 |
| (5, 2, '') | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (5, 2, 'R') | 0.0 | 0 | 0.0 | 0.0 | -7.5 | -5.0 |
| (5, 2, 'G') | 3.0 | 1.35 | 0.0 | 0 | -8.13125 | -4.100004 |
| (5, 2, 'Y') | 0.0 | 0.0 | 0.006598 | 0.0 | -9.6875 | -8.75 |
| (5, 2, 'B') | 0.062969 | 0.0 | 0.648 | 0.116591 | -9.804382 | -9.202207 |
| (5, 3, '') | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (5, 3, 'R') | 0.0 | 0.0 | 0 | 0.218492 | -5.0 | -7.5 |
| (5, 3, 'G') | 1.8 | 0.0 | 0.0 | 0.468479 | -9.729999 | -9.999995 |
| (5, 3, 'Y') | 0.0 | 0.0 | 0.0 | 0.021468 | -9.99939 | -9.995117 |
| (5, 3, 'B') | 0.0 | 0.324 | 1.08 | 0.0 | -9.442366 | -9.6875 |
| (5, 4, '') | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (5, 4, 'R') | 0.111668 | 0 | 0.0 | 0.0 | -7.47051 | -5.0 |
| (5, 4, 'G') | 0.0 | 0.0 | 0.0 | 0.0 | -5.0 | -9.375 |
| (5, 4, 'Y') | 0.0 | 0.0 | 0.0 | 0.0 | -8.75 | -9.6875 |
| (5, 4, 'B') | 0.0 | 0.0 | 1.8 | 0.0 | -8.75 | -6.960004 |
| (5, 5, '') | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (5, 5, 'R') | 0.047811 | 0.0 | 0.0 | 0.0 | -5.0 | -4.992579 |
| (5, 5, 'G') | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | -10.0 |
| (5, 5, 'Y') | 0.0 | 0.0 | 0.0 | 0.0 | -9.84375 | -9.999695 |
| (5, 5, 'B') | 0.0 | 0.0 | 0 | 3.0 | -6.6 | -8.421875 |

## 4. Análisis de los resultados

Dado que en este caso el estado tiene tres dimensiones, para la visualización de los resultados de la política se muestra una grilla (con la política para cada posición del tablero) para cada uno de los valores que puede tomar la estación de destino (i.e. no destino o cada una de las estaciones en el tablero).

En la política de los estados sin destino se puede ver que para cada estado el mejor q-valor es 0 y que todas las acciones de la política son movimientos ('up', 'down', 'left' o 'right'), pero en ningún caso la mejor acción es recoger ('pick') o dejar ('drop') un pasajero. Este es el resultado esperado dado que el agente es penalizado por intentar dejar o recoger un pasajero en una casilla equivocada (vacía, ya sea estación o no), y antes de recoger un pasajero no tiene forma de saber en dónde aparecerá, dado que esta es una posición aleatoria que cambia en cada episodio. Este comportamiento tiene sentido puesto que eso es lo que ocurre en la realidad, dado que un taxi no sabe en donde aparecerá un pasajero y debe desplazarse hasta encontrar uno.

En el caso de los estados con estación de destino, la política es efectiva puesto que se trata de los movimientos que llevan al agente a la estación y a la ejecución de la acción 'drop' allí. Es importante notar que en ninguno de estos estados la acción indicada por la política es 'pick', lo cual es consecuente con el modelo porque no tendría sentido que intentara recoger un pasajero ya llevando uno.