Assignment 6

Jesús David Barrios (j.barrios) - 201921887

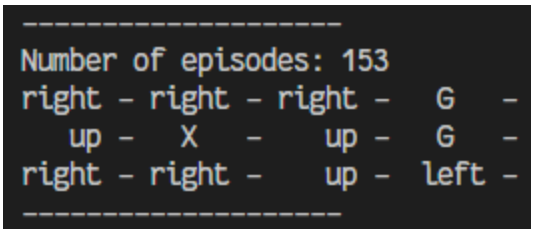
Para cumplir con el objetivo de esta actividad, se crearon dos archivos de python: *q_learning.py* y *grid_environment.py*. En el primer archivo, se puede encontrar la implementación del agente de Q-Learnig. Para probarlo, se debe ejecutar este archivo, y en consola se podrá escoger que ambiente utilizar.

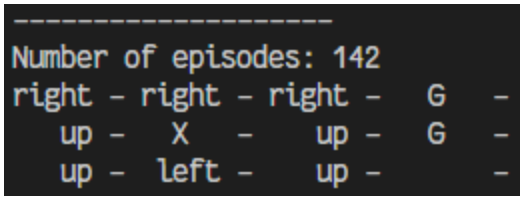
En el segundo archivo, se encuentran 4 clases. La primera es GridEnvironment, la cual funciona como clase abstracta para todos los ambientes. En este, se definen atributos comunes como el nombre, el ancho y alto, el inicio y la meta. Además, se tienen métodos relacionados con el movimiento en el ambiente, la impresión del ambiente, y la revisión de si un estado es final y de las posibles acciones en un método determinado. Las otras 3 clases hacen referencia a los 3 ambientes solicitados. Cada una de las clases extiende a GridEnvironment para adaptarse a las condiciones y requerimientos de sus ambientes correspondientes.

A continuación, se presentan resultados de las ejecuciones de cada uno de los 3 ambientes:

GridWorld

En esta implementación se obtuvieron resultados similares a los obtenidos en las tareas anteriores. En general, se observa que se obtiene una convergencia (estabilización) luego de ejecutar entre 130 y 160 episodios. En cuanto a la efectividad, se observa que las politicas abstraídas del resultado final de la tabla de q-valores siempre permiten obtener una ruta optima al estado final y evitan llegar al estado negativo. En general, luego de multiples ejecuciones, se obtiene una política igual para los diferentes estados, con la excepción de los estados más lejanos del estado final. A continuación, se presentan dos ejemplos de resultados de ejecución para este ambiente:

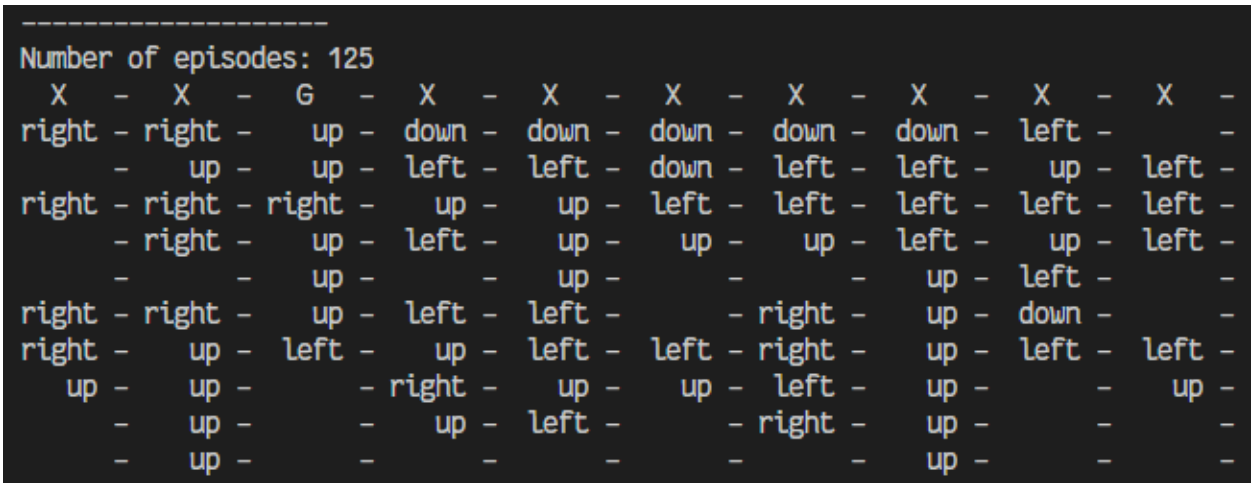




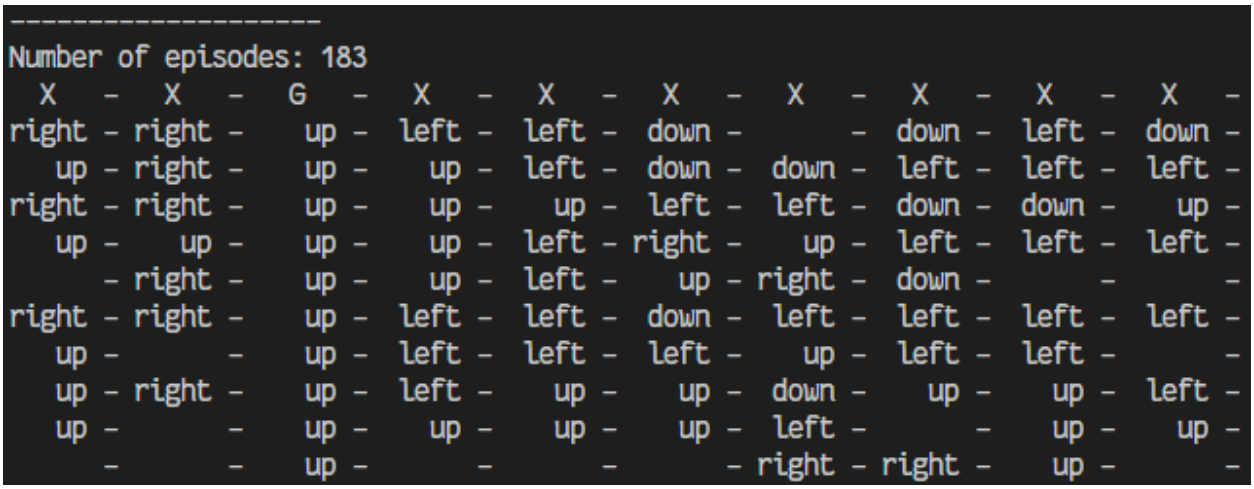
Laberinto de cuartos

En esta implementación se observa una mayor distribución del número de episodios necesarios para que se de una estabilización de los q-valores. En este caso, se observa una oscilación entre 90 y 180 episodios para obtener un resultado. En cuanto a la efectividad, en terminos generales se observa que se obtienen políticas que permiten llegar de manera correcta al estado final, y que en general tienden a ser optimas. En general, desde cualquier estado que se escoja, si se sigue la política obtenida a partir de la q-tabla, se llega al estado que permite salir del laberinto. Además, en general, se obtienen políticas que redireccionan hacia las puertas y evitan las esquinas de los cuartos.

Dada la amplitud del ambiente, en este caso se tiende a obtener estados sin política, dado que los q-valores asociados a ese estado luego de todas las implementaciones fueron todos iguales a cero. Esto, dado que se dio una estabilización antes de tener una exploración completa de estados. A continuación, se presentan dos ejemplos de resultados de ejecución para este ambiente:



```
--------------------
Number of episodes: 125
  X   -   X   -   G   -   X   -   X   -   X   -   X   -   X   -   X   -   X   -
right - right -    up -  down -  down -  down -  down -  down -  left -       -
      -    up -    up -  left -  left -  down -  left -  left -    up -  left -
right - right - right -    up -    up -  left -  left -  left -  left -  left -
      - right -    up -  left -    up -    up -    up -  left -    up -  left -
      -       -    up -       -    up -       -       -    up -  left -       -
right - right -    up -  left -  left -       - right -    up -  down -       -
right -    up -  left -    up -  left -  left - right -    up -  left -  left -
   up -    up -       - right -    up -    up -  left -    up -       -    up -
      -    up -       -    up -  left -       - right -    up -       -       -
      -    up -       -       -       -       -       -    up -       -       -
```



```
--------------------
Number of episodes: 183
  X   -   X   -   G   -   X   -   X   -   X   -   X   -   X   -   X   -   X   -
right - right -    up -  left -  left -  down -       -  down -  left -  down -
   up - right -    up -    up -  left -  down -  down -  left -  left -  left -
right - right -    up -    up -    up -  left -  left -  down -  down -    up -
   up -    up -    up -    up -  left - right -    up -  left -  left -  left -
      - right -    up -    up -  left -    up - right -  down -       -       -
right - right -    up -  left -  left -  down -  left -  left -  left -  left -
   up -       -    up -  left -  left -  left -    up -  left -  left -       -
   up - right -    up -  left -    up -    up -  down -    up -    up -  left -
   up -       -    up -    up -    up -    up -  left -       -    up -    up -
      -       -    up -       -       -       - right - right -    up -       -
```

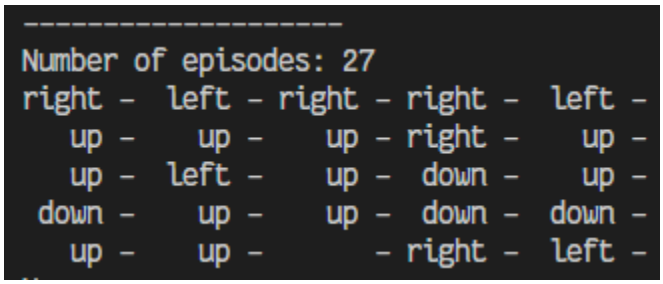
Para este ambiente, la primera fila está llena de espacios a los que no se puede acceder. Esta se incluye solo para poder incluir el estado final, el cual queda fuera de la cuadrícula del ambiente.

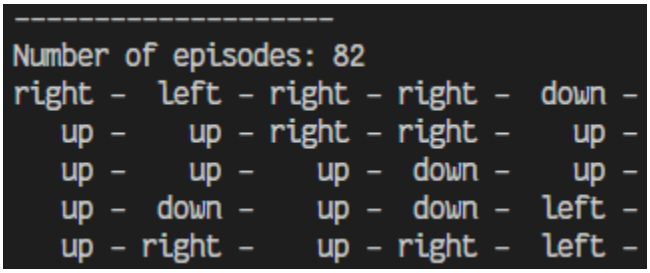
Taxi

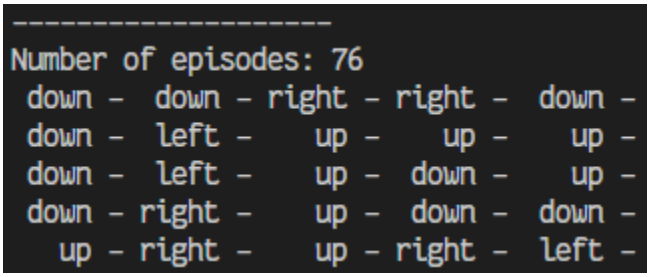
Esta implementación es la que más tiempo toma en estabilizarse, por lo que se incluyó un límite de 3 minutos para que realice episodios (en caso de tomar más tiempo la ejecución se termina). Esto se debe a que en contraste con los ambientes anteriores, cada episodio implica más acciones y es más dinámico. Dado que se tienen 4 paradas de taxi y que en cualquiera puede aparecer el pasajero del episodio, el taxista no puede determinar una ruta a seguir siempre, sino que debe estar explorando las diferentes paradas. Además, en contraste con los otros ambientes, en este caso también aumenta

el rango de acciones, teniendo las acciones de recoger y de dejar a un pasajero, por lo que la tabla q y la exploración crece.

Aunque no hay una ruta fija que se pueda seguir en todos los episodios, si se observa en la política final obtenida de la q-tabla del agente, que desde los diferentes estados, se busca siemrpe dirigirse hacia un paradero. En este orden de ideas, se considera una buena señal de eficiencia, dado que el taxista siempre buscará dirigirse a un paradero, y estando en un paradero, siempre buscará salir de este. A continuación, se presentan 3 ejemplos de resultados de ejecución para este ambiente:







El riesgo de los resultados obtenidos, es que a partir de las posiciones de los pasajeros que hayan salido en los episodios anteriores, se puede generar sesgo frente a ciertos paraderos.