

Búsqueda en Pacman

Introducción

En la primera parte de la asignatura usaremos en clásico juego Pacman, siguiendo un proyecto desarrollado por la Universidad de California en Berkeley.

El proyecto será desarrollado en Python y corregido en Linux. Sin embargo, puede usar Windows o Mac para el desarrollo, si instala el entorno de trabajo adecuado.

Puede usar IDLE, un editor para Python desarrollado en Python. Sino,

- En Linux puede usar Gedit o Emacs.
- En Windows puede usar Notepad++. Existe un plugin para Python
- En Mac puede usar Aquamacs.

La más críticos es que el editor no confunda espacios y tabulaciones (pesadilla garantizada al trabajar en Python). Idealmente debería suportar tabulador y shift-tabulador para indentar o desindentar líneas o bloques de código.

Calentamiento

Siga la práctica de calentamiento siguiendo en enlace **Unix/Python tutorial** que se encuentra en el aula global. Esta práctica no será entregada.

Search in Pacman

Siga el enlace **Search in Pacman** que se encuentra en el aula global. Para obtener toda la nota debe responder las preguntas 1 a la 6.

Para repasar los algoritmos vistos en clase, puede ver el enlace **Search summary (PDF)**, que contiene un resumen de los algoritmos DFS, BFS, BestFirst. Este último tiene como casos particulares Uniform Cost y A^* .

Extra crédito - 15%

Si completa las preguntas 7 y 8 podría obtener hasta 15% adicional.

Requisitos y consejos

- Implemente sus algoritmos de búsqueda usando siempre la verificación de nodos cerrados (closed nodes) usando la estructura de datos Python `set`.
 - Si un ha sido expandido (es decir, se calcularon sus sucesores), debe colocarse en closed.
 - Al expandir nodos, todos los que estén en closed, serán ignorados (es decir, nunca agregados a la lista de nodos por procesar).

- Entienda pronto la diferencia entre nodos y estado.
- Los nodos son, en teoría, caminos, pero es mejor representarlos con un estado, su cost, la última acción ejecutada, más una referencia al nodo padre. Así podrá calcular el camino desde el estado inicial.
- En algunas distribuciones de Linux, puede que necesite instalar el paquete python-tk, o algunos otros. La línea siguiente debería instalar todo lo requerido para ejecutarla:

```
sudo apt-get install python python-tk idle python-pmw python-imaging
```

 incluido el editor IDLE.
- Esté siempre consciente de la estructura de datos que está utilizando; su costo; el tiempo que demora haciendo las operaciones.

Condiciones de la entrega

1. Las prácticas se harán en grupos de máximo dos personas.
2. También en mínimo de dos personas, a menos que haya un número impar de personas. Es decir, máximo una persona trabajará solo.
3. En la sesión de práctica siguiente se realizará una micro-entrevista, dirigiendo preguntas a cada miembro del equipo. Son preguntas sencillas que quien hizo la práctica podrá responder con facilidad. De no responder correctamente, podrían perder la nota de la práctica.
4. La entrega consistirá de tres archivos, a ser entregados en tres puntos diferentes en el Aula Global:
 - `search.py`: Modificado por ustedes.
 - `searchAgents.py`: Modificado por ustedes.
 - `informe.pdf`, indicando
 - Nombres, NIA
 - Estado de la práctica:
 - * ¿Qué funciona?
 - * ¿Qué no funciona?
 - * ¿Qué problemas encontraron?
 - * ¿Qué aprendieron?

Recuerde no modificar los archivos en Python incorrectamente o fallará la autocorrección.
5. Respete los nombre de los archivos
6. Se puede entregar hasta el miércoles 28 de Enero, a las 22h.

Otros comentarios

- El plagio, tanto entre estudiantes como desde internet, y la colaboración de código entre equipos no será tolerada. Se puede discutir sobre la práctica, pero no muestren su código a otros equipos.
- Desarrolle incrementalmente y vaya probando tras cada cambio. Puede ser útil ir desarrollando un script o una función que pruebe el programa con diversas entradas y en diversos escenarios.

Tips en Python

Python usa referencias, como Java. Por lo tanto, tras hacer las siguientes instrucciones

```
a=[1, 2, 3]
b=a
b[0]=10
print a
```

Veremos que se imprime

```
[10, 2, 3]
```

Esto es uno de los errores más frecuentes al usar Python en esta asignatura. Para copiar la lista a tendría que usarse un `slice`:

```
a=[1, 2, 3]
b=a[:]
b[0]=10
print a
que imprimiría
[1, 2, 3]
```