

MISIONEROS Y CANÍBALES

Trabajo sobre el problema Teórico de “Misioneros y Caníbales”



Autor1: Diego-Édgar Gracia Peña

Correo: diego.gracia@edu.uah.es

Autor2: David Ariza Asperilla

Correo: david.ariza@edu.uah.es

Carrera: 3º de Ingeniería Informática UAH.

Asignatura: Inteligencia Artificial.

Contenido

MC1 – 3 Misioneros, 3 Caníbales y una barca de 2 plazas.....	2
MC2 – 4 Misioneros, 4 Caníbales y una barca de 2 plazas.....	2
MC3 – 5 Misioneros, 5 Caníbales y una barca de 3 plazas.....	2
Tutorial de Instalación de las librerías necesarias	3



MC1 – 3 Misioneros, 3 Caníbales y una barca de 2 plazas

Se consideran las operaciones de paso de una Orilla a otra:

- O1: va un caníbal
- O2: van un misionero
- O3: van 2 misioneros
- O4: van 2 caníbales
- O5: van 1 misionero y 1 caníbal

Contando con estas operaciones, se introduce en los abiertos el nodo inicial, se extrae sus posibles sucesores que cumplan con la condición de que no haya más Caníbales que misioneros en la misma orilla (si la barca está en una orilla, se considera también parte de esa orilla).

Con ello obtenemos los nodos abiertos, que serán ordenados respecto al coste calculado que tenga cada uno en cada iteración de la búsqueda nodal.

El coste de los nodos se obtiene tras obtener estos, y el cálculo sería tal que:

$$((N^{\circ} \text{ Misioneros Orilla Origen} + N^{\circ} \text{ Caníbales Orilla Origen}) * 2 - \text{Valor Barca}) + \text{Profundidad Nodo}$$

Se suma el cálculo heurístico y el coste (profundidad del nodo) en llegar a él.

Cuando encontramos el nodo meta, se terminará la búsqueda.

MC2 – 4 Misioneros, 4 Caníbales y una barca de 2 plazas

Se ha realizado el cálculo teórico del problema con 4 misioneros, 4 caníbales y una barca de 2 plazas, resultando en todas las búsquedas de nodos un estado cíclico, ya que se alcanza de forma irremediable nodos en los que hay más caníbales que misioneros en la misma orilla.

MC3 – 5 Misioneros, 5 Caníbales y una barca de 3 plazas

Al igual que en MC1, se determina el conjunto de Operaciones y se especifica las restricciones de estas:

- O1: va un caníbal
- O2: va un misionero
- O3: van 2 misioneros
- O4: van 2 caníbales
- O5: van 1 misionero y caníbal
- O6: van 3 caníbales
- O7: van 3 misioneros
- O8: van 1 misionero y 2 caníbales
- O9: van 2 misioneros y 1 caníbal




En cuanto a restricciones, se añade la restricción de que no puede haber más caníbales que misioneros en la barca, descartando automáticamente la operación O8.

Por el resto es igual en cálculo y método a MC1.

Tutorial de Instalación de las librerías necesarias

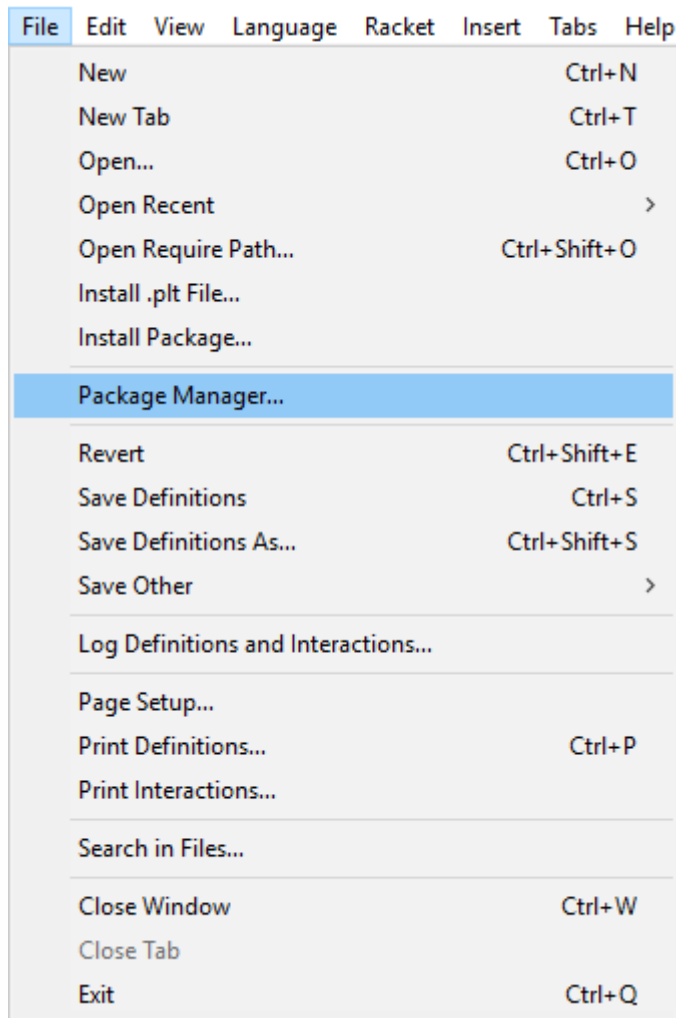
En caso de que le falten las librerías necesarias para la ejecución del código, el propio DrRacket proporciona las librerías usadas en el código.

Si recibe este mensaje al ejecutar el código:

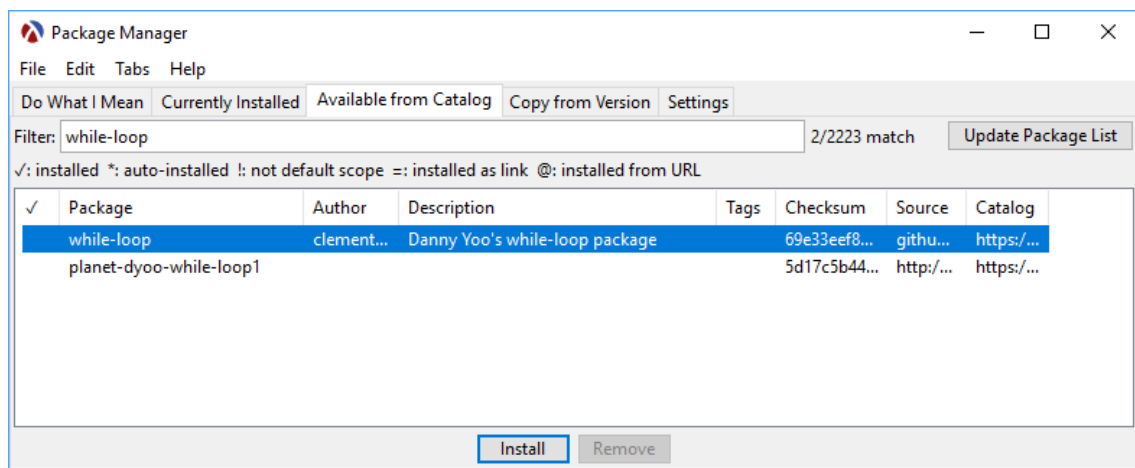
```
Welcome to DrRacket, version 6.12 [3m].  
Language: racket, with debugging; memory limit: 128 MB.  
 standard-module-name-resolver: collection not found  
for module path: dyoo-while-loop  
collection: "dyoo-while-loop"  
in collection directories:  
  C:\Users\diego\AppData\Roaming\Racket\6.12\collects  
  C:\Program Files\Racket\collects  
  ... [165 additional linked and package directories] in: dyoo-while-loop  
no package suggestions are available \[update catalog\]  
> |
```

necesita tener instalada la librería “while-loop”, para ello, vaya a “Archivo” o “File” y seleccione “Package Manager...”:

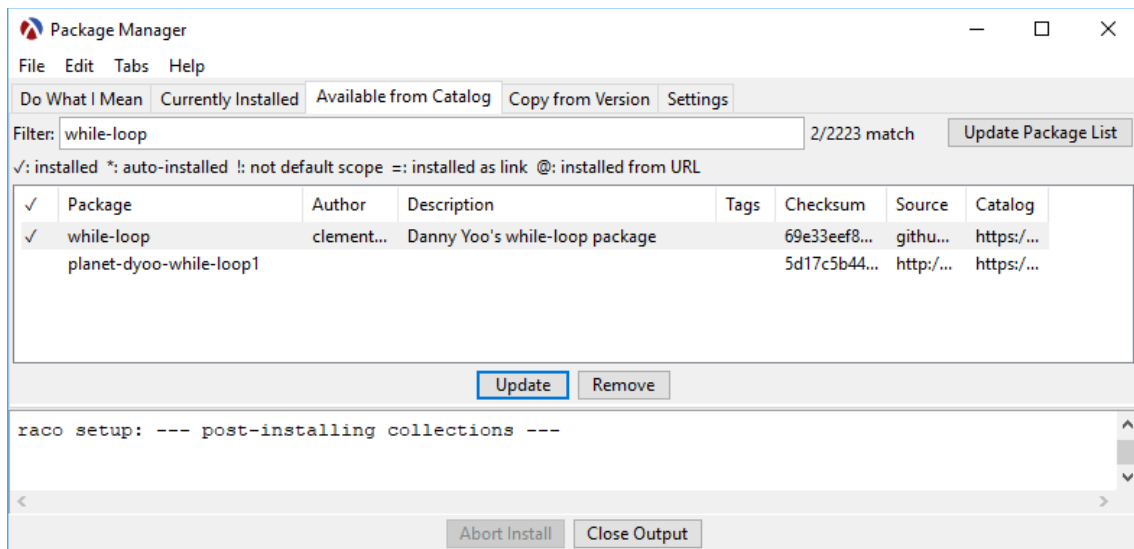




Tras esto, se abrirá esta ventana. Seleccione la pestaña “Available from Catalog” e introduzca el Filter “while-loop”. A continuación, seleccione la opción de la imagen y pulse “Install”:



Tras esto el gestor instalará la librería “While-Loop”, terminando de la siguiente forma:



Cierre la ventana y ya tendrá instalada la librería necesaria. Vuelva a ejecutar el código.

