

Documentación Tarea 4 IRB1001 - Path Planning

Índice

1. Clase maptile	2
1.1. solidity	2
1.2. visual	2
1.3. value	2
2. Clase map	2
2.1. map (<i>width,height,filepath</i>)	3
2.2. reset ()	3
2.3. redrawmap ()	3
2.4. drawsmall (<i>bool</i>)	3
2.5. getsize ()	4
2.6. getfinish ()	4
2.7. getstart ()	4
2.8. gettile (<i>x,y</i>)	4
2.9. getmatrix ()	4
2.10. moverobot (<i>direction</i>)	4
2.11. teleport (<i>x,y</i>)	4
2.12. setfulldraw (<i>drawfullrobot</i>)	4
2.13. collisioncheck ()	4
2.14. finishcheck ()	5
2.15. drawrobotborder	5
2.16. hascrashed	5
2.17. hasfinished	5
2.18. width	5
2.19. height	5
2.20. path	5
2.21. maptable	5
2.22. robotposx,robotposy	5
2.23. finishx,finishy	5
2.24. startx,starty	5

1. Clase `maptile`

La clase `maptile` corresponde al elemento que representa cada cuadrado dentro del mapa; no tiene funciones pero tiene 3 parámetros

1.1. `solidity`

Permite saber si el robot tiene permitido ocupar ese espacio, los valores posibles son:

- **empty**: Espacio vacío ocupable
- **solid**: Espacio que el robot no puede tocar (como una pared)
- **virtual**: Espacio que el centro del robot no puede tocar, pero si el resto de sus dimensiones

1.2. `visual`

Corresponde al diseño gráfico que se mostrará en el mapa, se cargará el archivo del nombre igual a este parámetro pero con extensión PNG para ser dibujado en el espacio que corresponda.

Los valores *tilestart* y *tilefinish* corresponden a posiciones especiales y si los reemplaza puede eliminar la meta o la partida, no es recomendable que modifique estos valores si es que no es su propósito explícito.

1.3. `value`

Inicializada en 0, no cumple ninguna función y puede modificarla a voluntad para usarla dentro de su programa.

2. Clase `map`

Esta clase contiene todas las funciones y parámetros que representan al mapa y al estado del robot. Posee varios parámetros que son útiles y funciones para generar movimientos, debe usarla como base para su tarea.

Cabe destacar que las funciones que reciben como argumento una coordenada las reciben como (X, Y) partiendo desde 1, pero son transformadas a un par ordenado $(X - 1, Y - 1)$ partiendo desde 0. No olvide esta diferencia cuando use las funciones vs cuando modifique los parámetros directamente.

2.1. `map (width,height,filepath)`

Constructor de la clase toma el archivo ubicado en `filepath`, el cual debe ser de extensión CSV y la extensión debe estar incluida en el `filepath`; además de tener que representar uno de los mapas a usar (que le serán proporcionados para la tarea). Es necesario además incluir el alto (`height`) y el ancho (`width`) del mapa.

Esta clase genera la matriz de mapeles a usar, coloca al robot en la partida del mapa y genera la interfaz gráfica a usar. Las dimensiones a usar para cada mapa se muestran en las siguientes líneas de código, cada línea inicia uno de los mapas una vez importado el simulador con las líneas.

```
import pygame
import irbt4
```

Mapa 1

```
mp = irbt4.map(30,48,'irb1001t4-map01.csv')
```

Mapa 2

```
mp = irbt4.map(47,57,'irb1001t4-map02.csv')
```

Mapa 3

```
mp = irbt4.map(54,61,'irb1001t4-map03.csv')
```

Mapa 4

```
mp = irbt4.map(54,61,'irb1001t4-map04.csv')
```

2.2. `reset ()`

Limpia todos los valores y vuelve a generar el mapa, volviendo al estado inicial. Es prácticamente equivalente a generar una instancia nueva de `map`.

2.3. `redrawmap ()`

Dibuja la información correspondiente a los datos actuales del mapa, junto con la posición actual del robot. Es llamada automáticamente con las funciones `map`, `reset`, `moverobot` y `teleport`.

2.4. `drawsmall (bool)`

Si `bool` es `true` dibuja el mapa utilizando cuadros pequeños de 6x6 píxeles en vez de 25x25.

2.5. `getsize ()`

Entrega (ancho, largo) del mapa.

2.6. `getfinish ()`

Entrega las coordenadas de la meta en (X,Y).

2.7. `getstart ()`

Entrega las coordenadas de la partida en (X,Y).

2.8. `gettile (x,y)`

Entrega el maptile correspondiente a las coordendas (x,y). Equivale a buscar dentro del arreglo del mapa las posiciones [x-1][y-1].

2.9. `getmatrix ()`

Entrega la matriz de maptiles correspondiente al mapa.

2.10. `moverobot (direction)`

Mueve el robot en la direccion especificada, las direcciones pueden ser [north, south, east, west].

Además revisa si ha colisionado o si ha llegado a la meta, luego dibuja el mapa con la nueva información.

2.11. `teleport (x,y)`

Mueve el robot a las coordenadas (X,Y), revisa colisiones y arribo a la meta al igual que moverobot. No puede usar esta función dentro de su código final.

2.12. `setfulldraw (drawfullrobot)`

Valor boolean que indica si el robot será dibujado en su totalidad (5x5) o puntualmente. Las funciones de colisión no alteran su funcionamiento.

2.13. `collisioncheck ()`

Indica si es que el robot ha colisionado desde el último reset. Esta función es ejecutada dentro de moverobot y teleport.

2.14. finishcheck ()

Indica si es que el robot ha llegado a la meta desde el último reset. Esta función es ejecutada dentro de moverobot y teleport.

2.15. qtime()

Entrega el tiempo en milisegundos desde la creación del mapa o el último reset.

2.16. qmovements()

Entrega la cantidad de movimientos realizados por el robot desde la creación del mapa o el último reset.

2.17. drawrobotborder

Boolean que indica si se debe dibujar el robot entero.

2.18. hascrashed

Boolean que indica si el robot ha chocado desde el último reset.

2.19. hasfinished

Boolean que indica si el robot ha llegado a la meta desde el último reset.

2.20. width

Ancho del mapa en tiles (dimensión Y).

2.21. height

Alto del mapa en tiles (dimensión X).

2.22. path

Dirección del archivo del mapa cargado.

2.23. maptable

Lista de listas que contiene la información del mapa, los indices son [X][Y] y el cuadro [0][0] corresponde a la esquina superior izquierda.

2.24. robotposx,robotposy

Posición actual del robot.

2.25. finishx,finishy

Posición de la meta.

2.26. startx,starty

Posición de la partida.