
PROYECTO 1

202001574 – Josue Daniel Solis Osorio

Resumen

El siguiente proyecto se realizó con el fin de poder realizar una codificación a través del lenguaje de programación denominado como Python (en la versión 3.10.2 de 64bits).

Este se utilizó para Para poder completar el sistema de rescate y extracción, la empresa Chapín Warriors, S.A., se nos ha contratado para crear un sistema de control que recibirá la malla de celdas del dron ChapinEye a partir de esta información se nos entregó a través de un archivo xml por el cual utilizando una librería se puede acceder a dicha información

La matriz con la que se trabajo es denominada como matriz ortogonal o matriz dispersa gracias a que con este tipo de lista se hace más fácil el recorrer, reconocer y optimizar espacios ya que es por lo que se caracteriza este tipo de matriz, una vez los objetos que ya hacen para facilitar la creación del algoritmo que se utilizara posteriormente para encontrar la ruta que llevara a una ya sea a un rescate de un civil o extracción de un recurso tomando en cuenta las restricciones de movimiento para cada tipo distinto de operación a realizar tomando en cuenta al operador con el cual se realizara dicha misión

Palabras clave

Matriz ortogonal

Matriz Dispersa

Leer laberinto a través de matriz

Abstract

The following project was carried out in order to be able to encode through the programming language called Python (in version 3.10.2 of 64bits).

This was used to In order to complete the rescue and extraction system, the company Chapín Warriors, SA, has contracted us to create a control system that will receive the cell mesh of the ChapinEye drone based on this information that was delivered to us through of an xml file through which using a library you can access said information

The matrix with we work is called an orthogonal matrix or sparse matrix to the fact that with this type of list it is easier to traverse, recognize and optimize spaces, since this is what characterizes this type of matrix, once the objects that they already do to facilitate the creation of the algorithm that will be used later to find the route that will lead to either a rescue of a civilian or extraction of a resource taking into account the movement restrictions for each different type of operation to be carried out taking into account the operator with whom the mission will be carried out

Keywords

Orthogonal Matrix

Sparse Matrix

Read maze through Matrix

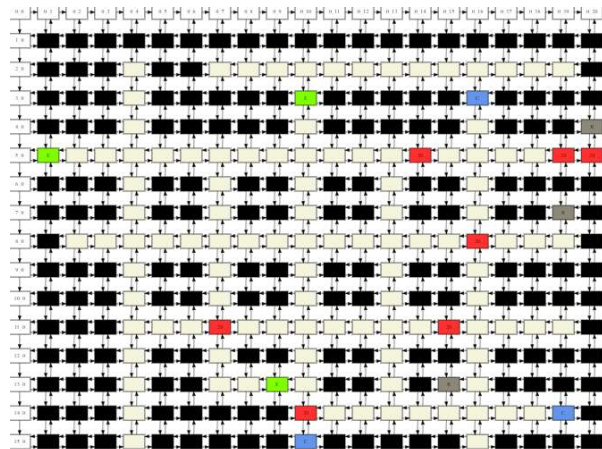
La información se manipulo a través de las matrices denominadas matrices ortogonales o matrices dispersas ya que esta es una estructura de datos que consiste en un conjuntode nodos enlazados secuencialmente y facilita el tratar con la información de los archivos xml de una manera mas sencilla sin tener que recurrir con las listas que ya trae el lenguaje de programación de Python

Con el fin de realizar las misiones de rescate y extracción, Chapín Warriors, S. A. ha construido drones autónomos e invisibles para los radares llamados ChapinEyes. Los ChapinEyes sobrevuelan las ciudades y construyen un mapa bidimensional de la misma, este mapa bidimensional consiste en una malla de celdas, donde cada celda es identificada como un camino, un punto de entrada, una unidad de defensa, una unidad civil, un recurso o una celda intransitable

Cada ciudad de compone de filas y en estas filas los ChapinEyes retorna la informacion de la siguiente manera: asterisco (*) - representa una celda intransitable), espacio (' ' - representa una celda transitable), letra E (E - representa un punto de entrada), letra C (C - representa una unidad civil), letra R (R - representa un recurso).

```
<?xml version="1.0"?>
<configuracion>
  <listaCiudades>
    <ciudad>
      <nombre fila="valorEntero" columnas="valorEntero" valorAlfanumerico </nombre>
      <fila numero="valorEntero" <../../../../../../../../2 </fila>
      <fila numero="valorEntero" > **** ** ** </fila>
      <fila numero="valorEntero" > *****E*****C***** </fila>
      <fila numero="valorEntero" > ***** ***** **R** </fila>
    ...
    <unidadMilitar fila="valorEntero" columna="valorEntero" valorEntero3 <unidadMilitar>
      ...
    </ciudad>
    ...
  </listaCiudades>
  <robots>
    <robot>
      <nombre tipo="ChapinFighter"4 capacidad="valorEntero"> valorAlfanumerico </nombre>
    </robot>
  </robots>
</configuracion>
```

Una vez obtenida la información de el archivo xml se procedió a elaborar una matriz ortogonal como se muestra a continuación



Una vez la ciudad ya sea graficada se procede a seleccionar el tipo de operación que se desea realizar cumpliendo con las condiciones de que siempre se debe de iniciar en los puntos verdes y si la misión seleccionada es de rescate los robots no pueden pelear o si se selecciona extracción de recursos los robots solo pueden pelear si únicamente superan la “capacidad” de su adversario

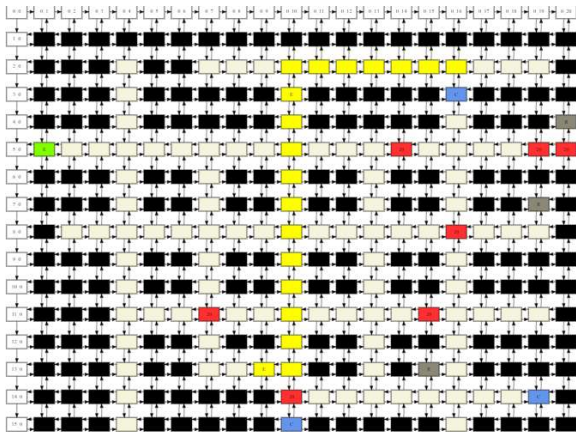


Figura 3. Misión de rescate
Fuente: elaboración propia 2022

Y se procedería a mostrar los resultados y el seguimiento de la misión en consola

```
se encontro camino posible
Se movio hacia la Derecha
Se movio hacia la Arriba
Se movio hacia la Arriba
Se movio hacia la Arriba
Se movio hacia la Arriba
Se movio hacia la Arriba
Se movio hacia la Arriba
Se movio hacia la Arriba
Se movio hacia la Arriba
Se movio hacia la Arriba
Se movio hacia la Derecha
Se movio hacia la Derecha
Se movio hacia la Derecha
Se movio hacia la Derecha
Se movio hacia la Derecha
Se movio hacia la Abajo
fin ----- 16 ----- 3 ----- original ----- 16 ----- 3
Tipo de mision: Rescate
Unidad civil rescatada: Y3 X16
Se utilizo a Ironman de tipo: ChapinRescure
```

Figura 4. Seguimiento de misión
Fuente: elaboración propia 2022

Para el caso de realizar una misión de extracción
Se deben de cumplir distintas condiciones como por ejemplo que los caminos rojos solo son transitables si

y solo si el operador que está realizando la misión tiene suficiente capacidad como para pasar

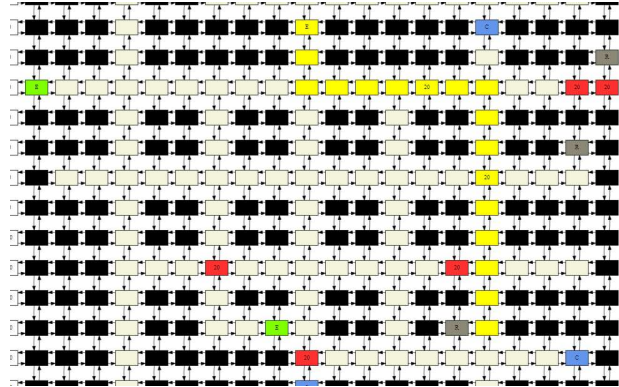


Figura 5. Misión Extracción de recurso
Fuente: elaboración propia 2022

Diagrama de clases

