

Sustentación Laberinto

Alex Jose Alberto Barreto

Daniel Arturo Moreno Rincón



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Algoritmo de Búsqueda - Búsqueda Primero en Profundidad

- 1. HashMap en vez de matriz.*
- 2. Se implementó Pila para llevar el recorrido.*
- 3. Se acondicionó la lógica del procedimiento de la mano izquierda.*

Para retornar la posición anterior.

- 4. Se acondicionó SimpleTeseoAgentProgram.*

Implementación

Aumento de memoria para saber dónde estuvo y la ruta usada para llegar

```
protected HashMap<String, Boolean> mapa;  
protected Stack<Integer> recorrido;
```

Para actualizar la coordenada donde se encuentra y la orientación ya sea avanzando o retrocediendo se agregaron nuevos casos para actualizar la posición.

```
case 0:  
    switch(a){  
        case 0: y++; break;  
        case 1: x++; o=1; break;  
        case 2: y--; o=2; break;  
        case 3: x--; o=3; break;  
        case 4: y--; break;  
        case 5: y--; o=3; break;  
        case 6: y--; o=2; break;  
        case 7: y--; o=1; break;  
    }  
    break;
```

Implementación de las acciones que debe realizar

Se evalúa las acciones básicas y además si ya pasó por una de las posiciones aledañas, también se añadió la lógica para retroceder a una posición anterior.

```
public int accion(boolean PF, boolean PD, boolean PA, boolean PI, boolean MT, boolean FAIL) {
    int a;
    boolean devolvio = false;
    if (MT) return -1;
    if (!PI && !haPasado(3)) a = 3;
    else if (!PF && !haPasado(0)) a = 0;
    else if (!PD && !haPasado(1)) a = 1;
    else if (!haPasado(2)) a = 2;
    else {
        a = recorrido.pop();
        devolvio = true;
    }

    // Computing r, c, o
    actualizarCoordenada(a);
    if(!devolvio) recorrido.add(4+a);
    mapa.put(x+" "+y, true);
    return a;
}
```

Añadir la ejecución de las acciones de cuando retrocede

Se añade lógica para retroceder tanto en posición como en orientación.

```
int d = accion(PF, PD, PA, PI, MT, FAIL);
if (0 <= d && d < 4) {
    for (int i = 1; i <= d; i++) {
        cmd.add(language.getAction(3)); //rotate
    }
    cmd.add(language.getAction(2)); // advance
}
else if(d == 4 || d == 6 ) {
    cmd.add(language.getAction(3)); //rotate 2 veces
    cmd.add(language.getAction(3));
    cmd.add(language.getAction(2)); //advance
    cmd.add(language.getAction(3)); //rotate 2 veces
    cmd.add(language.getAction(3));
}
else if(d == 5 || d == 7) {
    cmd.add(language.getAction(3)); //rotate 2 veces
    cmd.add(language.getAction(3));
    cmd.add(language.getAction(2)); //advance
    for (int i = 1; i <= d-4; i++) {
        cmd.add(language.getAction(3)); //rotate
    }
}
else{
    cmd.add(language.getAction(0)); // die
}
```