



# Universidad de Guadalajara.

Centro Universitario de Ciencias Exactas e Ingenierías.

División de Electrónica y Computación.

Sem. de solución de problemas de algoritmia.

## **Etapa 6. Backtracking. Búsqueda en profundidad.**

### **Planteamiento.**

Diseñe un sistema computacional con la capacidad de representar escenarios utilizando grafos. Los escenarios son imágenes que cuentan con círculos negros. Para la construcción del grafo, los círculos representarán vértices, estos tienen la información del círculo (su coordenada y radio). El grafo es no dirigido (bidireccional). Considere que, un vértice puede tener una arista con otro vértice solamente si es posible trazar una línea recta imaginaria desde el centro de un vértice al otro sin pasar por alguna obstrucción en la imagen. Las obstrucciones pueden ser cualquier pixel que no pertenezca al círculo que representa a los vértices.

El sistema debe ofrecer la posibilidad de agregar hasta  $n-1$  agentes y un objetivo a un grafo de  $n$  vértices. Los agentes se diferencian entre presas y depredadores. Solamente se puede agregar agentes y objetivos en los vértices del grafo, y solo puede haber uno en cada vértice inicialmente. Si hay al menos un agente y un objetivo en el grafo, entonces puede comenzar una simulación, donde las presas recorrerán el grafo para llegar al objetivo, y el primero que llegue al objetivo, lo consume. Cuando una presa llega al objetivo, los demás agentes llegarán al vértice inmediato al que se dirigen y la simulación terminará. Después de finalizar una simulación, se puede añadir un nuevo objetivo y más agentes (si hay vértices disponibles).

### **Presas.**

Se pueden colocar presas en los vértices en los que no haya otra presa, un depredador ni un objetivo. Las presas buscan llegar al objetivo. Si una presa alcanza a llegar al vértice del objetivo, el objetivo desaparece.

Las presas no conocen el grafo, por lo que cada presa buscará al objetivo en el grafo, sin tener información acerca de la posición del objetivo ni del grafo en general. Las presas tienen memoria, por lo que podrán saber los vértices que han visitado, y las aristas que han utilizado; con esto, las presas podrán explorar los grafos con una técnica de búsqueda en profundidad.

Si una presa alcanza a llegar al vértice del objetivo, el objetivo desaparece.

### **Depredadores.**

Se pueden colocar depredadores en los vértices en los que no haya una presa, otro depredador ni un objetivo. El objetivo de los depredadores es alcanzar a las presas. Si un depredador alcanza a una presa, la presa desaparece.

Los depredadores tienen un “radar” que permite rastrear a una presa a la vez, el “radar” cubre un área circular de  $r_d$  diámetro, esta distancia se puede elegir antes de comenzar la simulación y es la misma para todos los depredadores.

Los depredadores conocen todo el grafo y se desplazan de una forma aleatoria y lenta, mientras no detecten presas.

Si una presa entra al radar de un depredador, el depredador estará enfocado en ella, incluso si se cruza una presa y ésta se encuentra más cerca, el depredador seguirá a la primera. Cuando un depredador tiene en su radar a una presa, éste conocerá el vértice hacia donde se dirige la presa y puede dirigirse de la forma más rápida posible (Dijkstra) al vértice destino de la presa. Además de aprovechar el conocimiento del destino de la presa y del grafo, cuando el depredador tiene en radar a una presa, su velocidad aumenta a 120% de la velocidad de la presa.

## Simulación.

Para que comience la simulación, deberán existir:

- 1 objetivo.
- 1 o más presas.
- 0 o más depredadores.

La simulación termina con todos los agentes en un vértice.

## Requerimientos funcionales.

- Paradigma de programación: P.O.O.
- El sistema debe contener una interfaz gráfica intuitiva.
- Mostrar la imagen a analizar.
- Mostrar una imagen que represente al grafo.
- Crear múltiples presas. (5 puntos)
- Crear múltiples depredadores. (5 puntos)
- Crear objetivo. (5 puntos)
- Búsqueda en profundidad para las presas (código y recorrido animado). (20 puntos)
- Desplazamiento aleatorio para los depredadores. (15 puntos)
- Desplazamiento basado en el algoritmo de Dijkstra para los depredadores que persiguen una presa. (15 puntos)
- Incremento de velocidad para los depredadores que persiguen una presa. (5 puntos)
- Eliminación de presas alcanzadas. (10 puntos)
- Representación visual de radar y enfoque a una presa. (15 puntos)
- 

## Reporte.

1. Planteamiento del problema.
2. Objetivos:
  1. Comprender e implementar algoritmos para el análisis de grafos y generación caminos mínimos
    1. Algoritmo de Dijkstra.
    2. Búsqueda en profundidad.
3. Marco teórico.
  1. Teoría de grafos.
    1. Grafo.
    2. Caminos.
      1. Recorrido o búsqueda en profundidad.
      2. Camino más corto (Dijkstra).
4. Desarrollo.
5. Pruebas y resultados.
6. Conclusiones.
7. Apendice(s).