
Especificación primer proyecto

tema

Resolviendo laberintos

Versión 1.0

Valor 10 %

Preparado por Eddy Ramírez

Índice general

1. Introducción	5
1.1. Propósito	5
1.2. Alcance	5
1.3. Referencias	5
2. Descripción general	6
2.0.1. Creación del laberinto	6
2.1. Simulación	6
2.1.1. Entrada del programa	7
2.2. Aspectos generales	7
2.2.1. Evaluación	7
2.2.2. Aspectos técnicos	8
2.2.3. Aspectos administrativos	8

1 Introducción

Este documento contiene la especificación del primer proyecto programado para el curso de Lenguajes de Programación, correspondiente al primer semestre 2014.

1.1. Propósito

El proyecto tiene como propósito dar al estudiante familiaridad con un lenguaje estructurado como C y también permitir que el estudiante profundice en un tema de programación paralela y algoritmos de control y búsqueda.

1.2. Alcance

Se espera que el estudiante desarrolle un sistema capaz de encontrar una solución satisfactoria a un laberinto.

1.3. Referencias

1. http://en.wikipedia.org/wiki/Maze_generation_algorithm
2. <http://www.youtube.com/watch?v=ucWX34Vrel8>
3. http://profstewart.org/pm1/talks_09/MazeCreating.pdf
4. <http://www.algosome.com/articles/maze-generation-depth-first.html>

2 Descripción general

Se utilizará Prim o Kruskal para generar un laberinto de $n \times m$ filas y columnas. La generación de este laberinto queda a criterio de cada grupo de trabajo, pero debe cumplir con conectividad y una única ruta entre dos puntos cualesquiera del laberinto. Una vez generado el laberinto, deben de generarse tres hilos, los cuales van a recorrer el laberinto. Para representar gráficamente estos recorridos se utilizarán ratones animados, que se desplacen en el laberinto. Cada uno de los hilos debe de buscar la mayor cantidad de quesos (los cuales son colocados aleatoriamente) y buscarán evitar caer en el veneno (el cual es colocado también aleatoriamente). Los algoritmos de cada uno de los hilos deben de variar y serán: Profundidad primero, anchura primero y random.

2.0.1. Creación del laberinto

El laberinto debe de ser creado a partir de alguno de los algoritmos mencionados en la sección de referencias, pero debe garantizar lo siguiente:

1. Conectividad

Para cualquiera dos puntos en la matriz, debe de existir una ruta entre ellos

2. Ruta única

Entre dos puntos cualquiera debe de existir una ruta única

3. $O(n) = n^2$

El máximo $O(n)$ para calcular el laberinto debe de ser n^2

Quesos y venenos

En un archivo de entrada se indicarán la cantidad de quesos y venenos que se deben de colocar de manera aleatoria en el laberinto. Las posiciones de los ratones también serán aleatorias.

2.1. Simulación

La idea es que los ratones comiencen con una velocidad constante, pero cada vez que come un queso, el ratón debe de aumentar ligeramente su velocidad ¹. Si el ratón cae

¹Se debe de decidir a priori la velocidad máxima que puede tener un ratón, siempre y cuando sea perceptible para un ser humano y sobre esa velocidad máxima se calcula la velocidad actual con base en la cantidad de quesos comidos y la cantidad máxima de quesos en el tablero

en un veneno, se muere, inmediatamente. Si dos ratoncitos comieran veneno, entonces el tercero sería declarado inmediatamente el ganador de la ronda.

El programa debe de contar con GUI de usuario para poder interpretar correctamente lo que ocurre.

2.1.1. Entrada del programa

El programa recibe de consola las condiciones iniciales del laberinto:

1. n: Número de filas del laberinto
2. m: Número de columnas del laberinto
3. q: Número de quesos que habrá
4. v: Número de venenos que habrá (valor entre 0 y 3)

Ejemplo de entrada

`./maze 40 30 25 2`

Este generaría un laberinto de 40x35 donde se colocarían 25 quesos y 2 venenos. Nótese que cada vez que se ejecuta ese mismo comando, generará un laberinto diferente.

Luego deberá haber algún botón de inicio, para que se coloque los ratones aleatoriamente, cada ratón debe tener un color diferente decidido por los equipos de trabajo, de modo que se pueda diferenciar claramente cuál sigue qué algoritmo.

2.2. Aspectos generales

2.2.1. Evaluación

A continuación se muestra una tabla de evaluación sobre los diferentes puntos a tomar en cuenta

Rubro	Valor		
Creación del laberinto	15 %	5 % cada requerimiento	
Búsqueda en anchura	15 %	5 % Uso correcto de la cola	10 % Recorrido
Búsqueda en profundidad	15 %	5 % Uso correcto de la pila	10 % Recorrido
Búsqueda aleatoria	10 %	10 % Recorrido	
Uso de los hilos	15 %	5 % por controlar quesos	-10 % concurrencia
Escritura de código de a cuerdo con los parámetros de C	10 %	-5 % por no utilizar estructuras propias de C	-5 % por no separar archivos en .h y .c
GUI	20 %	15 % por mostrar ratones animados	5 % Mostrar creación de laberinto

Nota: Sin GUI se pueden ver afectados los demás rubros de la evaluación

2.2.2. Aspectos técnicos

1. Toda la programación debe de realizarse sobre C en GNU/Linux
2. El programa debe de tener un makefile
3. Al incorporar GUI las bibliotecas que se utilicen deben ser entregadas junto a la tarea.

2.2.3. Aspectos administrativos

1. Se debe entregar sólo el código fuente, el makefile y las bibliotecas de GUI.
2. Cualquier sospecha de fraude implica una nota de cero y se aplicarán las sanciones respectivas
3. La tarea se entrega a través del aula virtual del grupo en el periodo de tiempo estipulado en ella