Tarea 4 - Problema 2 - Programación avanzada

Nicolás Aylwin

Buscamos crear un programa que simule 'El juego de la vida' de John Conway, para un sistema inicial específico (input.dat) representado por una matriz de ceros (célula muerta) y unos (célula viva) de 32 x 32. Luego crear un video de 700 generaciones con el script subido a u-cursos.

Tres archivos importantes permiten lo anterior; matriz.h, problema2.cc y peli.bash. El primero es la clase de templates de matrices usada. Esta se hizo previo al conocimiento de la tarea, por lo que incluye varias funciones que no se usan realmente. Dado esto, se separa explícitamente lo usado en el desafío y lo que no (para no sobrecargarte al revisar claramente).

Sobre la clase, destaco solo dos cosas (todo lo demás es estándar en clases de c++); la matriz trabaja con un solo vector, en contra de la idea intuitiva de usar vector de vectores, y se llama al elemento como (i, j) en vez de $[i][j]$ que también es lo más común.

Por otro lado, problema2.cc contiene el programa en cuestión. Se comienza con la definición de la funcion 'mod' que recibe dos enteros y entrega uno. La idea de esta es cumplir con la condición de periodicidad, de manera que si los índices se escapan inicialmente de donde exista la matriz, se devuelva el índice correspondiente. El primer entero es el que se quiere hacer pasar por el módulo del segundo.

Luego se define 'next_gen' que recibe la matriz del estado anterior (const para no cambiarla y por referencia para disminuir el tiempo de cómputo), dos valores enteros de 'sobrevivencia' y dos valores enteros de 'nacimiento'. En esta se crean primero tres valores enteros; I, J y acum. Los dos primeros serán los índices (respectivamente fila y columna) luego de hacerlos pasar por la función mod. El tercero es donde acumulamos la cantidad de 'vecinos' de cada célula. También se crea con el constructor de copia la matriz 'res' (por resultado), copiada de la entregada claramente.

El trabajo real de la función consta de tres ciclos for. Los dos primeros se van moviendo en cada elemento de la matriz entregada, el tercero suma los vecinos en acum usando I y J luego de usar mod. Al terminar el tercer ciclo pero antes de empezar a ver un nuevo elemento (i, j), se evalua si el elemento nace (de estar muerto inicialmente) o muere (de estar vivo inicialmente), ambas condiciones respecto los enteros mencionados como argumentos de la función. De no cumplir ninguna de estas dos condiciones, se mantiene el elemento tal cual en la matriz 'res' pues como se mencionó, es una copia de la original, asi que si no cambia, se deja como estaba. Finalmente, luego de recorrer toda la matriz entregada se retorna la matriz res como la nueva generación.

Luego en el programa está la función main. Esta recibe un argumento de la linea de comando, y lo usa como la condición inicial del sistema. Se almacenan los datos del archivo en un vector y se usa este para crear la matriz con que se trabajará. También se crea una carpeta donde se almacenarán los resultados, llamada 'juego_de_la_vida'.

Se entra a un ciclo for de 0 a 700, en este primero se crea un string que será parte del nombre del archivo de salida en cada iteración. Está hecho de manera que al listar los archivos se ordenen cronológicamente, pues esto es vital para la creación del video. Luego se crea el archivo en la carpeta mencionada, se guarda en este la matriz (por lo que el primer archivo es idéntico a input.dat) se cierra el archivo. Al final del bucle se usa next_gen sobre la matriz, y se asigna la nueva matriz a la

misma variable que contenía la entregada inicialmente (así solo trabajamos con una variable tipo matriz). En este último paso se asignan los valores estipulados para sobrevivencia y nacimiento.

Por último, el archivo peli.bash es casi idéntico al subido en u-cursos, pero con la diferencia que actua en la carpeta 'juego_de_la_vida' y no en la que uno se encuentra. Se encarga de crear archivos .png con gnuplot de cada archivo y luego codificarlos en dos videos, uno tipo avi y otro tipo ogv, ambos guardados en la carpeta donde uno lo corra.

En la carpeta solo se incluye makefile, input.dat, este informe y los tres archivos ya mencionados, al hacer correr make, compilará, correrá con input.dat como argumento, y correra peli.bash. Dado esto, basta correr make, y luego de que se procese todo aparecerán ambos videos en la misma carpeta para visualizarlos.