

Introducción a la Programación

Proyecto “El juego de la Vida”

Martes, 20 de mayo de 2008

Introducción

El juego de la vida es un autómata celular diseñado por el matemático británico John Horton Conway en 1970. Es el mejor ejemplo de un autómata celular. Hizo su primera aparición pública en el número de octubre de 1970 de la revista Scientific American, en la columna de juegos matemáticos de Martin Gardner.

Desde un punto de vista teórico, es interesante porque es equivalente a una máquina universal de Turing, es decir, todo lo que se puede computar algorítmicamente se puede computar en el juego de la vida. Es interesante para los científicos observar cómo patrones complejos pueden provenir de la implementación de reglas muy sencillas. La vida tiene una variedad de patrones reconocidos que provienen de determinadas posiciones iniciales.

El juego de la vida para algunos programadores sólo es un desafío de programación y una manera divertida de usar ciclos de la CPU.

Especificaciones Teóricas

El juego de la vida se basa en una matriz de un tamaño determinado, que representa un cultivo de células, en la cual mueren y se crean nuevas células.

Dentro de la matriz se tienen dos valores para representar a las células, el valor de ‘1’ representa una célula viva, y una célula muerta se representa con la celda vacía (en algunos casos se representa con el valor de ‘0’). En tal defecto se visualizaran sólo las células vivas, como se muestra en el siguiente diagrama:

				1			1	
	1		1			1	1	
		1				1	1	
	1	1			1			1
		1						
	1	1						
	1			1	1			
				1	1	1		
					1	1	1	

Diagrama 1.- Representación de una generación de células.

Se comienza con un estado inicial en donde la matriz aparece con un conjunto de células vivas generadas aleatoriamente. En base a una serie de reglas (que se definen a continuación) algunas células se mantendrán vivas, otras morirán pero otras nuevas se crearan, dando como resultado a una nueva generación de células.

Mediante el transcurso del tiempo y siguiendo las reglas definidas para las células, se irán desarrollando nuevas generaciones de células creándose así nuevos patrones de comportamiento.

Vecinas de una célula: una célula que se encuentra representada mediante una celda con el valor de 1, tiene a lo más 8 vecinos que son las celdas que se encuentran ocupadas a su alrededor. Pero puede haber el caso que tenga solo 3 o 5 vecinos. Por ejemplo en la siguiente matriz:

Las células marcadas con * pueden tener a lo más 3 vecinos.

Las células marcadas con ** pueden tener a lo más 5 vecinos.

Las células marcadas con *** pueden tener a lo más 8 vecinos.

1*				1*
	1	1	1	
	1	1***	1	
	1	1	1	
	1**	1**	1**	

Reglas para cada célula

1. **Muerte por Soledad:** Si una celda esta ocupada por una célula ('1') y tiene una sola célula vecina o ninguna, en la próxima generación esta célula se muere por soledad.
2. **Muerte por Sobre población:** Si una celda esta ocupada por una célula y tiene 4 o mas vecinas, esta célula muere por sobre población y en la siguiente generación no aparece.
3. **Supervivencia:** Si una célula tiene 2 o 3 células vecinas, se mantiene viva en la siguiente generación.
4. **Origen de Nueva Célula:** Si una celda esta vacía y tiene 3 células vecinas, nace una nueva célula en dicho lugar para la siguiente generación.

Con las reglas que se definieron anteriormente se irá evolucionando generación tras generación del cultivo de células. Con esto se podrá ver el comportamiento que tienen las células a lo largo de un periodo de tiempo.

Especificaciones Técnicas

El programa a desarrollar deberá de considerar el manejo de menús, que permitan la interacción con el usuario.

- Menú principal:

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-I
Curso de Introducción a la Programación – Proyecto	

Deberá de contener una mensaje de bienvenida y describiendo el programa. El usuario tecleara la **dimensión de la matriz** o cultivo de células además tendrá la opción se seleccionar:

- **Generación aleatoria:** Si se genera de manera aleatoria la población de células de la primera generación.
- **Extracción de un archivo:** Si se desea obtener desde un archivo la información referente a la primera generación de la población.

Dependiendo de la selección del usuario se prosigue con la el siguiente submenú

➤ **Submenú Generación Aleatoria:**

Dentro de este submenú el usuario deberá de seleccionar el grado de aleatoriedad, es decir, si se desea que se generen un número determinado de células vivas que se colocaran en la matriz o cultivo de células.

Por ejemplo:

Teclee cuantas células deben de ser en la primera generación: 5

➤ **Submenú Extracción de un Archivo:**

Dentro de este submenú el usuario deberá de ingresar la ruta y nombre (con extensión) del archivo en donde se encuentra la información de la primera generación de células.

Independiente de la opción seleccionada, se deberá de desplegar generación tras generación de la evolución de la población de células.

Opcional:

Es posible que el resultado de cada una de las generaciones se vaya almacenando en un archivo, mismo que serviría como registro del comportamiento de las poblaciones.

Especificación del proyecto

Objetivo

El objetivo del proyecto del curso de Introducción a la Programación (IP), se enfoca en cubrir algunos de los temas cubiertos durante el curso.

Alcance

Los alumnos deben ser capaces de desarrollar un análisis y efectuar la carta estructural del proyecto, detallar cada una de las características y elementos para completar la aplicación a desarrollar.

El desarrollo del proyecto de IP, debe de seguir el modelo en cascada, el cual se divide en una serie de fases con la finalidad de tener una disciplina y control en el desarrollo del proyecto. A continuación se describen cada una de estas fases, con las características a desarrollar en el proyecto:

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-I
Curso de Introducción a la Programación – Proyecto	

➤ **Análisis del problema:**

Análisis del problema planteado, identificar los elementos de entrada y salida para cada uno de los módulos, con el objetivo de tener un planteamiento de la solución, que posteriormente será especificado en el diseño.

➤ **Diseño de la solución:**

Desarrollar la carta de estructura con los respectivos flujos de datos, y elaborar pseudocódigo para cada uno de los módulos de la aplicación, siguiendo los estándares propuestos a lo largo del curso.

➤ **Codificación:**

Efectuar la traducción del pseudocódigo al lenguaje C, siguiendo los estándares de codificación propuestos a lo largo del curso.

➤ **Compilación y prueba:**

Efectuar la compilación y considerar la corrección oportuna de errores encontrados.

Para efectuar las pruebas de manera adecuada y que permitan facilitar la evaluación y verificación de la resolución del problema o problemas, que fuesen implementos en el desarrollo del modulo; se efectuará una serie de **Casos de Pruebas** que permitan definir **Elementos de Entrada** y especifica los **Resultados Esperados** y al finalizar la prueba se consideraran los **Resultados Reales**.

➤ **Mantenimiento y documentación:**

Se debe de especificar la correcta *documentación interna* (a nivel de código); y la *documentación externa* (a nivel de manual de usuario).

Documentación a Entregar

Se deberá de efectuar la documentación requerida de acuerdo a cada uno de los siguientes apartados:

1. Crear las cartas de estructuras requeridas para el desarrollo del programa, deberá de almacenarse en un documento con el nombre de “**Carta de Estructura del Proyecto IP**”. Dentro de la carta se deberá detallar cada uno de los módulos a desarrollar en el programa.
2. Crear el pseudocódigo de cada uno de los módulos que contendrá el programa, se deberá de guardar en un documento con el nombre de “**Pseudocódigo del Proyecto IP**”.
3. Efectuar la codificación, compilación y ejecución del programa, guardando todos los archivos generados en una carpeta con el nombre de “**Código del Proyecto IP**”.
4. Desarrollar un documento de evidencias, en donde se mostraran los resultados obtenidos al ejecutar el programa en las distintas opciones que se ofrece. Nombre del documento “**Casos de Pruebas del Proyecto IP**”.

Universidad Autónoma Metropolitana – Iztapalapa	Trimestre: 08-I
Curso de Introducción a la Programación – Proyecto	

5. Desarrollar un manual de usuario que permita la ejecución correcta del programa, considerando las distintas configuraciones que puedan presentarse. Nombre del documento “**Manual de Usuario del Proyecto IP**”.

Observaciones

- La elaboración del proyecto deben de efectuarse siguiendo el modelo en cascada.
- **TODOS** los documentos deben de ser entregados de manera electrónica (incluyendo el pseudocódigo).
- El código se enviará la carpeta del proyecto: “**Codigo Proyecto IP**”, comprimidos en un solo archivo zip.
- La fecha de entrega es el **Miércoles 4 de junio de 2008**. Esta es la fecha limite, **NO hay prorroga de entrega bajo ninguna circunstancia y por ningún motivo se aceptara la entrega a la fecha indicada**, en caso contrario se tomara como proyecto no entregado.
- El equipo de desarrollo **debe contar con un mínimo de 2 personas y un máximo de 3**. No se aceptara el proyecto si no se cumple con lo anteriormente mencionado.
- La información del proyecto deberá ser enviado al correo del curso ip_uami@yahoo.com. Con el Subject: **IP – Proyecto I - <Nombres de los integrantes>**”. Recuerda que si no tiene el formato adecuado, es difícil distinguirlo de otros cursos. No olvide verificar que se haya enviado correctamente **TODOS** los archivos. Se tomará como valido el primer correo recibido y bajo este se efectuara la revisión.