

PRÁCTICA 2: EL JUEGO DE LA VIDA

1. Objetivos de la práctica

Los objetivos de esta práctica son avanzar en el uso de objetos, eventos y gráficos, además de manejo muy básico de temporizadores.

2. El juego de la vida

Extraído de la Wikipedia, http://es.wikipedia.org/wiki/Juego_de_la_vida.

El juego de la vida es el mejor ejemplo de un autómata celular, diseñado por el matemático británico John Horton Conway en 1970.

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.

Desde su publicación, ha atraído mucho interés debido a la gran variabilidad de la evolución de los patrones. Se considera que la vida es un buen ejemplo de emergencia y autoorganización. Es interesante para los científicos, matemáticos, economistas y otros observar cómo patrones complejos pueden provenir de la implementación de reglas muy sencillas.

El juego de la vida tiene una variedad de patrones reconocidos que provienen de determinadas posiciones iniciales. Poco después de su publicación, se descubrieron el pentaminó R, el planeador o caminador (conjunto de células que se desplazan) y el explosionador (células que parecen formar la onda expansiva de una explosión), lo que atrajo un mayor interés hacia el juego. Lo que aquí se ha llamado “vida”, es una distribución de células por el mundo, y cuyo estado inicial es conocido. A continuación, se muestran algunas distribuciones cuyo funcionamiento es bien conocido y, por tanto, pueden utilizarse para probar el correcto funcionamiento del juego.



Parpadeador



Barco



Planeador

La popularidad del juego aumentó gracias al hecho de que se publicó justo cuando se estaba lanzando al mercado una nueva generación de microprocesadores de bajo coste.

Para muchos aficionados, el juego de la vida no sólo era un desafío de programación sino también una manera divertida de usar ciclos de la CPU. Para otros, sin embargo, el juego adquirió más connotaciones filosóficas. Desarrolló un seguimiento casi fanático a lo largo de los años 70 hasta mediados de los 80.

El juego de la vida es en realidad un juego de cero jugadores, lo que quiere decir que su evolución está determinada por el estado inicial y no necesita ninguna entrada de datos posterior, aunque también existe la posibilidad de interaccionar con el sistema. El "tablero de juego" es una malla formada por cuadrados ("células") que se extiende hasta el infinito en todas las direcciones.

3. Las células

Cada célula tiene 8 células vecinas, las 8 células que la rodean. Las células tienen dos estados, "vivas"/"muertas" (o "encendidas"/"apagadas"), y el estado de las mismas evoluciona a lo largo de unidades de tiempo discretas (se podría decir que por turnos).

El estado de todas las células se tiene en cuenta para calcular el estado de las mismas en el turno siguiente. Todas las células se actualizan simultáneamente.

Las transiciones viva→muerta y muerta→viva dependen del número de células vecinas vivas. En concreto, se tienen las siguientes reglas para actualizar el estado de cada célula:

- Una célula muerta con exactamente 3 células vecinas vivas "nace". Por tanto, al turno siguiente estará viva.
- Una célula viva con 2 ó 3 células vecinas vivas sigue viva. En otro caso muere o permanece muerta (por "soledad" o "superpoblación").

4. El mundo

Respecto al panel que forma el mundo en el que "viven" las células, se deben tener en cuenta las convenciones siguientes:

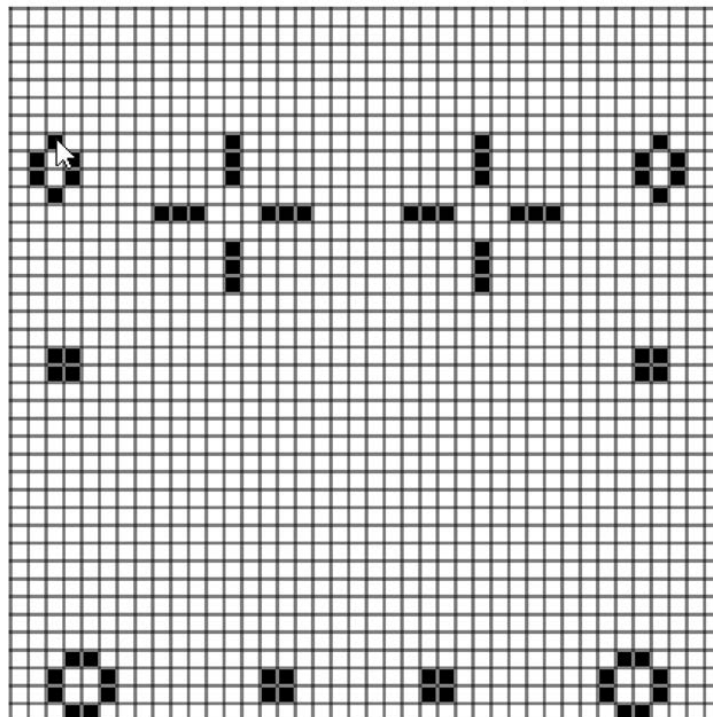
- El mundo disponible debe ser una matriz de ANCHO columnas por ALTO filas.
- Para simular un mundo "infinito", se supone que el borde superior continúa en el inferior, y viceversa. Lo mismo ocurre con los laterales.

5. Enunciado de la práctica

Crear un programa en JavaScript que, mediante los objetos adecuados sea capaz de modelar y mostrar el comportamiento del juego de la vida antes descrito.

- El programa debe permitir quitar y poner células en el mundo en cualquier momento mediante el ratón.
- El programa debe permitir empezar la simulación pulsando un botón al efecto.
- El programa debe permitir detener la simulación pulsando un botón al efecto.
- Al poner el ratón sobre una célula cualquiera debe aparecer información sobre cuánto tiempo lleva viva o muerta esa célula.
- El panel que representa el mundo debe mostrar el estado del mismo en cada paso de la simulación.

A continuación se muestra una posible vista del juego junto con los elementos antes descritos (panel, botones, información de la célula, etc...). Se trata simplemente de una sugerencia, el aspecto, colores y gráficos a usar se dejan a criterio de la creatividad del alumno.



Simulacion en marcha: Paso 331

La célula (3, 8) lleva viva 60 pasos.

6. Detalles de implementación

A continuación se muestran algunos detalles acerca de la implementación de la práctica:

- Aunque por su sencillez no es realmente necesario, el programa deberá implementar los siguientes elementos:
 - Un constructor para los objetos que modelan las células.
 - Un constructor para los objetos que modelan el mundo.
- Los objetos célula deben contener al menos dos atributos:
 - Estado (si está viva o muerta).
 - Tiempo que lleva viva.
- Los objetos mundo deben contener al menos:
 - Alto del mundo.
 - Ancho del mundo.
 - Una matriz bidimensional de “ancho x alto” células.
- El programa debe permitir trabajar con tamaños de lienzo o de mundo variables de manera trivial (modificando una constante o similar) sin necesidad de modificar ni re-escribir código (por defecto ANCHO = 80 y ALTO = 80).
- La evolución del mundo debe avanzar a un ritmo de N pasos por segundo (por defecto N = 10). Para implementar dicha evolución temporal se deben usar **temporizadores**. Aunque estos se verán más detenidamente en el Tema 5, de momento pueden utilizarse las funciones `setInterval()`, para activar el temporizador, y `clearInterval()` para detenerlo:
 - `var temporizador = setInterval(función, tiempo)`: Ejecuta la *función* indicada de forma periódica cada "*tiempo*" milisegundos. Devuelve una variable de tipo "Timer".
 - `clearInterval(temporizador)`: Detiene el temporizador asociado a la variable "temporizador".
- Para probar el programa se pueden usar las figuras mostradas en el segundo apartado de este enunciado.
 - *Parpadeador*. El resultado debe ser una evolución de células que alterna entre vertical y horizontal.
 - *Barco*. Cuya evolución es estática.
 - *Planeador*. Se desplaza por el mundo en diagonal girando sobre sí mismo.

7. Detalles de la entrega

A continuación se muestran los detalles sobre el formato de la entrega:

- 1) El programa debe mostrar los datos del autor o autores.

- 2) Al campus virtual debe subirse un único archivo comprimido con todos los recursos necesarios para ejecutar la práctica. El nombre de dicho archivo debe seguir el siguiente formato:

Ejemplo: Practica2_NombreApellidos1_NombreApellidos2.zip

- 3) Para aquellos alumnos que hagan la práctica por parejas, tan solo uno de los integrantes del grupo debe subir el archivo con la entrega al campus virtual.

8. Puntuación

La práctica se evaluará acorde al siguiente reparto. Los puntos de cada apartado son la puntuación máxima. Puede haber valoraciones intermedias en función de la solución aportada por el alumno.

Concepto	Puntos
Documentación del código fuente	1
Correcta definición e implementación de los objetos de tipo célula	1
Correcta definición e implementación de los objetos de tipo mundo	2
Gestión adecuada del ratón en su interacción con el mundo	1
Gestión adecuada de los botones y eventos asociados	1
Modularidad, usabilidad y diseño del código	1
Otros elementos destacables (funcionalidad extra, etc.)	3

9. Autores y plagio

Respecto a la autoría de cada uno de los componentes de la práctica, deben tenerse en cuenta los siguientes puntos:

- 1) En la pantalla ha de aparecer de forma clara y visible el autor o autores del juego, sus DNIs y la fecha.
- 2) Si se usan elementos adicionales (por ejemplo, sonidos, imágenes de fondo, etc.), hay que indicar claramente de dónde (web, libro, etc.) se han tomado. Esta indicación habrá de hacerse de forma visible en la propia página del juego debajo del nombre del autor de la práctica.
- 3) No se admite el uso de código fuente de terceros, sean de dominio público o no, salvo el código facilitado en el Campus Virtual de la asignatura (es decir, se puede usar parte del código de los ejercicios prácticos, pero no bajarse el programa de Internet sea todo o en parte).

10. Fecha de entrega

Antes de las 23:55 del 09/12/2017