

Algoritmos y Programación I

(75.40/95.14)

Trabajo práctico N°1

Segundo cuatrimestre 2016

Introducción

La compañía eléctrica “Eh?-DelSur” está contratando nuevos empleados para la sección de emergencia eléctrica del área del conurbano. A todas estas personas hay que capacitarlas, pero el presupuesto no da para tanto. La mejor alternativa es conseguir un simulador con el que puedan capacitarlos a todos con una mínima inversión.

Es por esto que el co-secretario adjunto de la compañía, Sr. Lamparita, solicitó ayuda a la FIUBA. El sistema que solicitan debe poder simular el comportamiento del tendido eléctrico del conurbano, donde cambiar de fase un transformador altera la energía de la zona del mismo y las circundantes. La Facultad, sin pensarlo demasiado, consideró que era tarea de una prestigiosa cátedra de Algoritmos y Programación resolver el problema. Luego de horas, días y semanas, de analizar el problema llegó a la conclusión de que el sistema es bastante similar al juego Lights Out, pero por razones de tiempo no se pudo implementar. Por esta razón, dicha cátedra encomendó la ardua tarea a sus prometedores estudiantes.

Consigna

Se pide implementar un juego interactivo *Luces fuera*¹.

Reglas del juego

Se trata de un juego individual y se desarrolla en un tablero de 5×5 . El objetivo es lograr apagar todas las luces encendidas en el tablero. Cuando todas las luces han sido apagadas el jugador ha ganado.

El juego comienza con un patrón de luces encendidas en el tablero. En cada turno se debe seleccionar algún casillero y si la luz estaba encendida se apagará y viceversa. Esta acción también afectará a sus vecinas. (Horizontal y vertical, pero **no** diagonal).

Ejemplo 1:

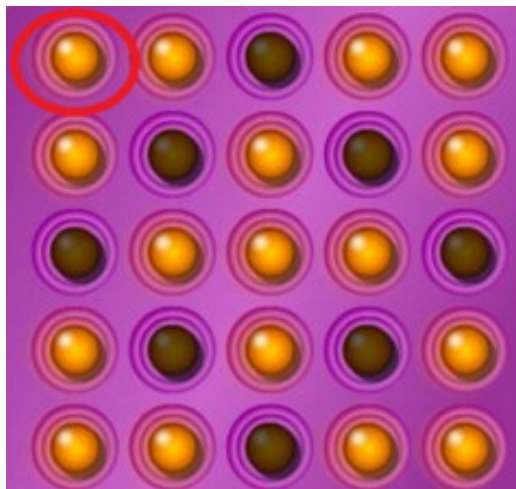


Figura 1: Luz a presionar

¹ <https://help.gnome.org/users/lightsoff/stable/index.html.es>

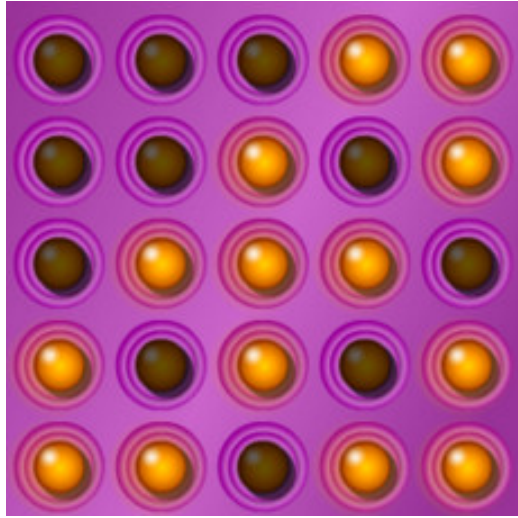


Figura 2: Resultado de presionar la luz marcada

Ejemplo 2:

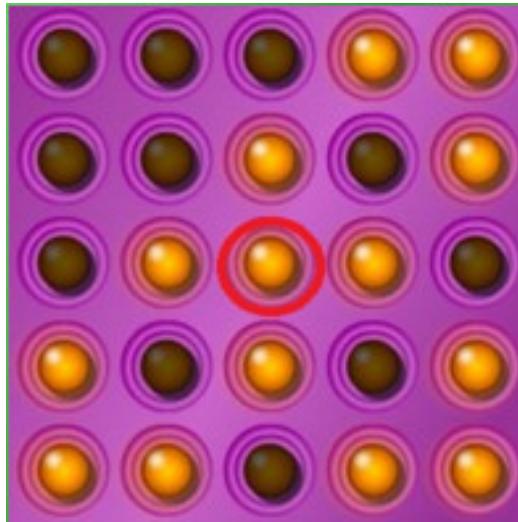


Figura 3: Luz a presionar

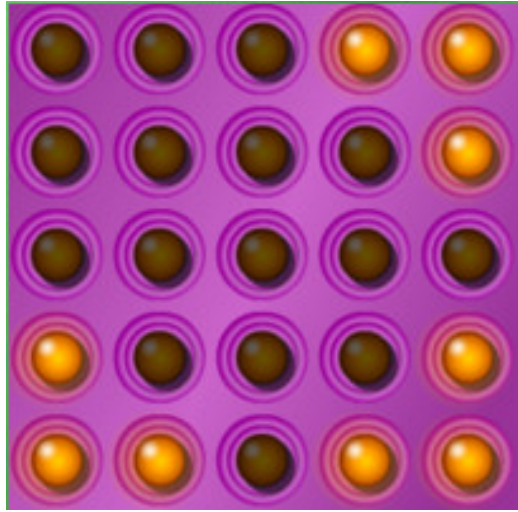


Figura 4: Resultado de presionar la luz marcada

Implementación

Al iniciar el juego se debe dar un mensaje de bienvenida y mostrar un menú para que el jugador elija en qué modo desea jugar, o salir del juego.

El usuario podrá elegir entre un modo aleatorio y uno predeterminado. En ambos modos se juegan 5 niveles; una vez que el jugador ganó el nivel se debe pasar automáticamente al siguiente hasta terminar con los niveles. Si no hay más niveles (o si el jugador perdió) se debe mostrar el menú nuevamente.

En el modo aleatorio, el jugador podrá elegir la dimensión del tablero, entre 5×5 hasta un máximo de 10×10 , y la disposición inicial de cada uno de los 5 niveles será aleatoria.

En el modo predeterminado, se pide implementar 5 niveles del juego donde las luces van a estar iniciadas como se indica en las figuras 5, 6, 7, 8 y 9.

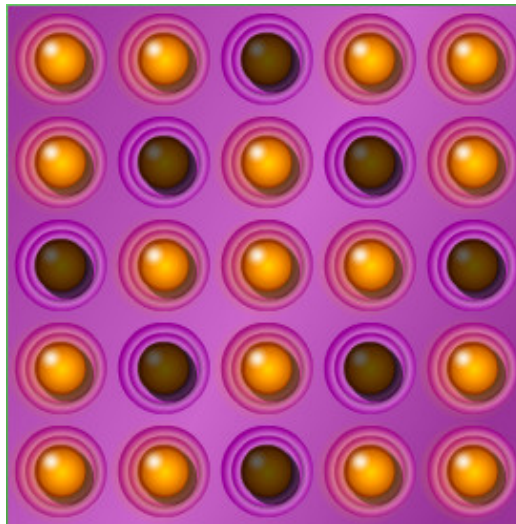


Figura 5: Nivel 1

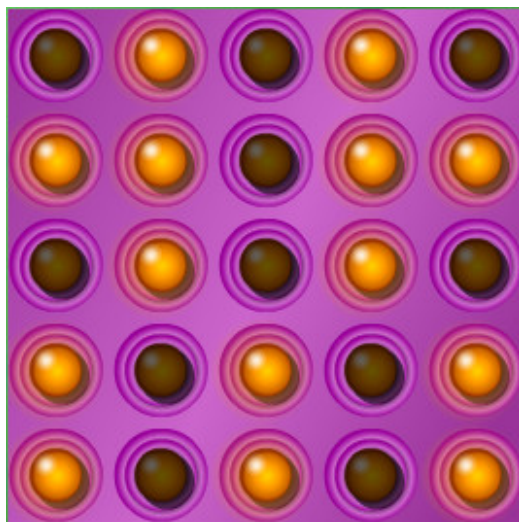


Figura 6: Nivel 2

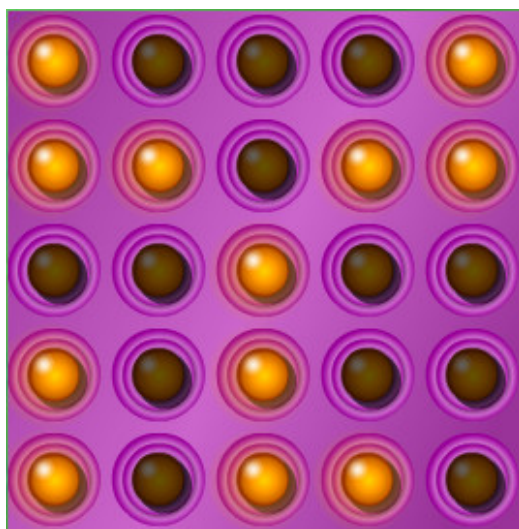


Figura 7: Nivel 3

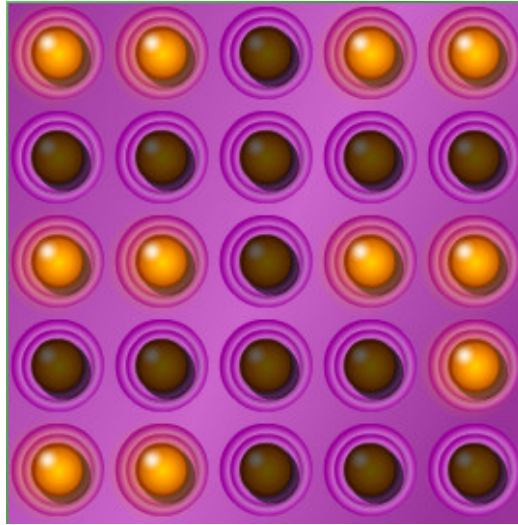


Figura 8: Nivel 4

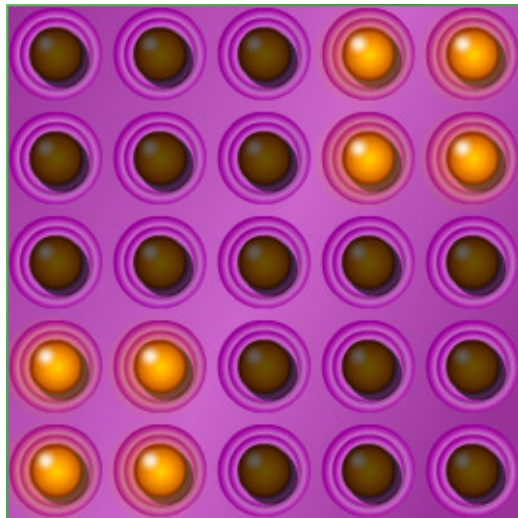


Figura 9: Nivel 5

En cualquier momento, antes de elegir cualquier casillero el jugador debe poder reiniciar el estado del juego, volviendo todas las luces a su posición original.

Además existe un máximo de turnos de 3 veces la dimensión del tablero (15 cuando la dimensión es 5); en caso de sobrepasar ese límite se debe dar por perdido el juego.

Se debe mantener a través de los niveles, el puntaje por nivel y el puntaje total. Para ello se tendrán en cuenta:

- Si todas las luces se han apagado: +500
- Si se quedó sin movimientos: -300
- Resetear el juego: -50 por cada luz que sigue encendida.

Al finalizar el nivel o resetear el juego, se le debe informar al usuario el puntaje obtenido en ese nivel y el total.

Interacción con el usuario

Para mostrar el tablero, se sugiere etiquetar las filas con números enteros de 1 a N , y las columnas con letras A, B, C En cada celda, mostrar las luces encendidas con un caracter y las apagadas con otro. Por ejemplo:

	A	B	C	D	E
1	o	.	.	o	o
2	.	.	.	o	o
3	o	o	.	.	.
4	.	o	.	.	o
5	.	.	.	o	o

Tras mostrar el tablero se solicitará al usuario una letra y un número para seleccionar la celda ($A1..E5$). Si el usuario ingresa un valor incorrecto se le debe informar y solicitar de nuevo.

Criterios de aprobación

A continuación se describen los criterios y lineamientos que deben respetarse en el desarrollo del trabajo.

Informe

El informe debe consistir en una descripción del **diseño** del programa.

Debe recordarse que la etapa de diseño es *anterior a la implementación*, por lo tanto debe describirse, utilizando texto y/o diagramas, cómo se va a estructurar el código para cumplir con las especificaciones de la consigna.

Algunas preguntas que deberían responderse:

- A grandes rasgos, ¿cómo será el flujo del programa?
- ¿Cómo se va a guardar en memoria el estado del juego?
- ¿Qué operaciones se efectuarán durante el juego?

Código

Además de satisfacer las especificaciones de la consigna, el código entregado debe cumplir los siguientes requerimientos:

- El código debe ser claro y legible.
- El código debe estructurarse en funciones y, cuando corresponda, módulos. Las funciones deben definirse de la manera más genérica posible.
- Todas las funciones deben estar adecuadamente documentadas, y donde sea necesario el código debe estar acompañado de comentarios.

Entrega

La entrega del trabajo consiste en:

- El informe y código fuente impresos. Para el código fuente utilizar una tipografía `monoespacio`.
- El informe en formato *PDF*.
- Una versión digital de todos archivos `.py` de código, separados del informe. En el caso de ser más de un archivo, comprimidos en un `.zip`.

El informe impreso debe entregarse en clase. Los dos últimos (PDF y código fuente) deben enviarse a la dirección electrónica `tps.7540rw@gmail.com` con el asunto “TP1 - *<Padrón>*”.

Este trabajo práctico se desarrolla en forma **individual**. El plazo de entrega vence el **lunes 26 de septiembre de 2016**.