

EIF204 - Programación 2

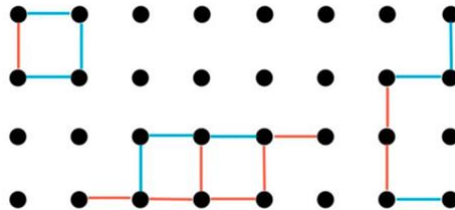
Proyecto de programación #2

Porcentaje evaluación: 15%

Prof. Máster. Georges Alfaro S.
Prof. Máster Santiago Caamaño P. (coordinador)
Prof. Dr. Juan de Dios Murillo M.

PLANTEAMIENTO DEL PROBLEMA

Timbiriche, colchón, cuadrito, cajas o puntitos es un juego de lápiz y papel. Fue publicado por primera vez en 1889 por Édouard Lucas (con el nombre de **pipopipette**, conocido como '*Dots and Boxes*' en inglés), y se puede jugar con diferentes niveles de dificultad. Las reglas son muy simples: el objetivo del juego es dibujar líneas para conectar puntos adyacentes en una cuadrícula, con el objetivo de formar cuadrados. Cada vez que un jugador completa un cuadrado, obtiene un punto y puede hacer otro movimiento. El jugador con más puntos al final del juego gana.



OBJETIVOS

El objetivo del proyecto es completar una aplicación donde se puedan poner en práctica las técnicas de programación estudiadas en el curso, en especial el empleo de patrones de diseño para lograr un diseño simple y robusto, respetando los principios y técnicas de diseño más importantes.

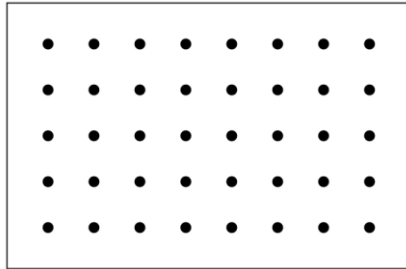
DESCRIPCIÓN DE PROBLEMA Y FUNCIONALIDAD SOLICITADA

Se juega sobre una hoja de papel, a modo de tablero, por dos o más jugadores, en una matriz en forma de cuadrículado; dibujando un punto por cada esquina de lo que será un cuadro. Los puntos en forma de cuadrícula deben ser, proporcional a la cantidad de participantes. En cada jugada, de forma alternada, un jugador unirá dos puntos consecutivos horizontal o verticalmente; los cuadritos se van formando lado por lado, línea por línea, y las líneas diagonales no son permitidas en el juego.

Cuando un jugador forma con estas líneas un cuadrado, se anota este, escribiendo una inicial en el centro o marcándola con un color distintivo. Después de formar un cuadrado deberá dibujar una línea más, cerrando todos los cuadros que el tablero permita. El jugador que se haga con más cuadrados gana la partida.

Para este proyecto, el cuadrículado se mostrará en la ventana de la consola del computador, usando texto simple. No se requiere de ningún tipo de interfaz gráfica.

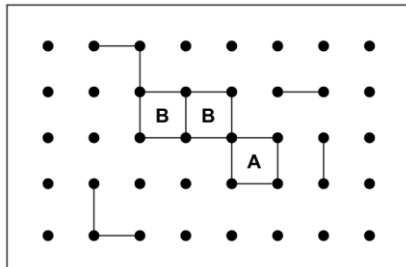
Por ejemplo, el tablero inicial de un juego con 4 filas y 7 columnas como este:



se puede mostrar de la siguiente manera:

	0	1	2	3	4	5	6	7
0	+	+	+	+	+	+	+	+
1	+	+	+	+	+	+	+	+
2	+	+	+	+	+	+	+	+
3	+	+	+	+	+	+	+	+
4	+	+	+	+	+	+	+	+

A medida que los jugadores van completando sus jugadas, el tablero podría verse así:



	0	1	2	3	4	5	6	7
0	+	+	-	+	+	+	+	+
1	+	+		-	-	+	+	-
2	+	+	B	B		+	+	+
3	+	+	-	+	+	A		+
4	+	+		-	+	+	+	+

Aquí se muestran varias líneas que han sido trazadas por los jugadores y algunas cajas que ya se han completado (en este caso usando las letras 'A' y 'B' para identificar a cada jugador).

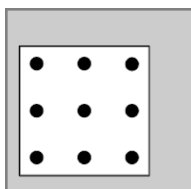
Un juego ya completo podría ser como el que se observa abajo:

	0	1	2	3	4	5	6	7
0	+	-	+	-	+	-	+	-
1		A		A		B		B
2	+	-	+	-	+	-	+	-
3		B		A		B		B
4	+	-	+	-	+	-	+	-
5		B		B		A		A
6	+	-	+	-	+	-	+	-
7		B		B		A		B

Aquí el jugador 'A' ha conseguido cerrar 9 cajas y el jugador 'B' 19, por lo que es el ganador.

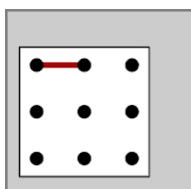
Desarrollo del juego

Primero se construye el área de juego, que no necesariamente es rectangular o cuadrada. Podemos, por ejemplo, dibujar una matriz de 3 puntos horizontales y verticales (3×3), así:



Tenemos dos oponentes y cada uno, en su turno, va dibujando con un lápiz y uniendo con una línea dos puntos cercanos o aledaños, verticales u horizontales (nunca en diagonal), comenzando por donde desee.

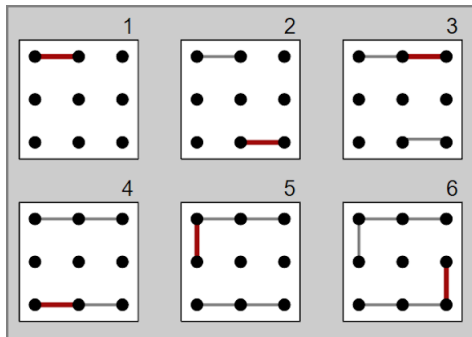
El primer jugador puede iniciar en la parte superior izquierda:



Luego le corresponde al siguiente jugador hacer un nuevo trazo y así sucesivamente en cada turno, tratando de completar un cuadrado o caja, evitando al mismo tiempo la posibilidad de que su oponente pueda con su jugada, cerrar uno o varios cuadros. Si el jugador logra cerrar una caja, puede continuar jugando hasta que no pueda completar ninguna, en cuyo caso, cederá al turno a su oponente.

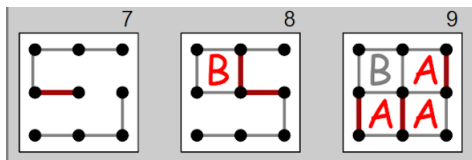
La finalidad del juego es terminar con el mayor número mayor de cajas capturadas. Dentro de cada caja o cuadro capturado se deberá poner un número o letra que identifique al jugador ('A' o 'B', como en este ejemplo).

Aquí se tiene un posible desarrollo del juego por dos personas (se muestra en un color diferente última línea trazada):



El primer jugador ha realizado las jugadas impares (1, 3, 5) y el segundo las jugadas pares (2, 4, 6).

En el desarrollo de la partida llegará un momento donde un jugador puede cerrar o alguna caja o cuadro:



En la octava jugada, el segundo jugador ('B') puede cerrar una caja. Debe volver a jugar, pero no puede cerrar otra caja. Cuando traza la línea y cede el turno, el jugador 'A' puede cerrar una caja y completar el resto cerrándolas en secuencia, ganando así la partida.

REQUERIMIENTOS DE IMPLEMENTACIÓN Y FUNCIONALIDAD

Modalidades de juego

Implemente el juego para dos jugadores solamente. Cuando se inicia el juego, el usuario puede seleccionar jugar contra otro oponente humano o hacerlo contra el computador. El juego podrá ser:

- Jugador contra jugador
- Jugador contra computador

Estrategia de juego

Si se juega contra la máquina, la persona podrá escoger la estrategia de juego que será utilizada por el programa. La persona puede indicar que se usará la misma estrategia en todo momento o cambiar a otra estrategia en cualquier momento que lo desee.

Se proponen **5 estrategias distintas** que serán implementadas por el programa:

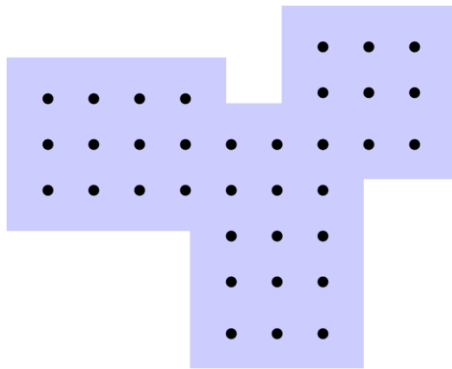
- **Juego aleatorio**, donde se colocan las líneas en cualquier posición factible.
- **Juego cercano**, donde se juega en un lugar cercano al lugar donde juega su oponente.
- **Juego periférico**, en el que primero trata de cerrar todo el borde del área de juego para ir formando casillas desde ahí.
- **Juego central**, para el que intenta colocar líneas iniciando desde el centro del área de juego.
- **Juego por "islas"**, donde procura completar pequeñas "islas" o áreas de tamaño reducido en cualquier zona del área de juego.

En todos los casos, el programa debe intentar cerrar alguna caja y evitar la captura por el oponente humano al completar la jugada. Es decir, debe evitar, hasta donde sea posible, colocar una línea que permita al jugador completar una casilla en la siguiente jugada.

Puede implementar, además, cualquier otra estrategia que considere adecuada. La implementación de una nueva estrategia se considerará para otorgar puntaje adicional, y será acompañada con una documentación breve que explique su escogencia.

Configuración

El juego puede realizarse sobre un área rectangular o sobre un área irregular, que se puede formar uniendo áreas contiguas más pequeñas. En este caso, es importante definir una manera de construir el área y verificar las posiciones de juego. Puede combinar áreas simples (rectangulares) para crear otras más complejas y luego volver a combinarlas a su vez para crear áreas de mayor tamaño.



Tenga en cuenta que, al utilizar una interfaz de consola, existen limitaciones que debe considerar respecto al tamaño del área máxima de juego.

Persistencia de la configuración y estado del juego

La configuración del juego (definición del área de juego) y el estado de las partidas se guardarán en uno o varios archivos. El usuario debe poder guardar y recuperar el estado de una partida en cualquier momento y reproducir luego la secuencia de jugadas de una partida anterior.

CONSIDERACIONES DE IMPLEMENTACIÓN Y OBSERVACIONES GENERALES

- Todas las funciones y métodos del programa deben validar adecuadamente los datos de entrada. Las condiciones de error deben gestionarse apropiadamente y se hará un manejo adecuado de excepciones.
- Las clases deben categorizarse y encapsularse correctamente como clases de entidad, control y frontera (interfaz). No se considera correcto hacer procesos de entrada/salida en las clases de entidad y control. Es posible, sin embargo, incluir mensajes de verificación controlados por una variable de compilación condicional.
- Se espera que la aplicación esté construida considerando los principios de diseño que se han estudiado y que se empleen correctamente los patrones de diseño que corresponden. **No es obligatorio el uso de ningún patrón de diseño en particular, pero se considerará un error, y se rebajarán los puntos correspondientes si no se utilizan patrones en situaciones típicas de uso, por ejemplo, si no se utilizan iteradores para examinar el contenido de una colección o si no se usa el patrón Estrategia (*Strategy*) al implementar las estrategias de juego.**

- La interfaz debe diseñarse de manera que sea fácil de entender y usar, sin necesidad de un manual o referencia de usuario.
- Deberá utilizar el IDE indicado por su profesor para completar el proyecto. En todo caso, no deberán existir errores de sintaxis en el código. Si el programa no compila correctamente, se calificará con una nota de 0 (cero). Si al ejecutar el proyecto se presentan errores que impidan su ejecución (como errores en la ruta de los archivos o bibliotecas mal enlazadas), también se perderá la calificación.
- El proyecto debe acompañarse con el diagrama de clases UML (completo) en formato PDF. Si es necesario para la correcta comprensión del diseño de la solución, pueden incluirse también diagramas de objetos o de secuencia.
- En caso de detectarse un plagio en alguno de los métodos o cualquier otra parte del programa, también se asignará un (cero) en la calificación y, según criterio del profesor, se elevará el caso a las autoridades de la escuela para aplicar las sanciones que correspondan.
- Cada profesor podrá solicitar la defensa del proyecto dentro o fuera del tiempo de clase, en un horario convenientemente acordado. De la misma manera, el profesor definirá la rúbrica o tabla de evaluación que usará.
- En caso de trabajar en parejas, solo uno de los dos deberá subir los archivos del proyecto. No se incluirá el código objeto o ejecutable del proyecto, y cada archivo de cabecera (.h) incluirá la información de las personas que han colaborado con el trabajo, usando un formato como el siguiente:

```
/* -----+
 *
 * (c) 2024
 * 1-2345-6789 - Carlos Martínez Rodríguez
 * 9-8888-7777 - Carolina Pérez Benavides
 * version 1.0.0 2024-06-01
 *
 * -----+
 */
```

- Se debe adjuntar un documento de texto con la cédula y nombre de los estudiantes, incluso cuando trabajen de manera individual.
- Sólo se acepta la entrega del proyecto por medio del aula virtual, que es el medio oficial establecido por la universidad. No se aceptarán ningún proyecto luego de la fecha límite de entrega indicada.

REFERENCIAS

Ferguson, T. S. (8 de mayo de 2024). *Dots and Boxes*. Obtenido de UCLA Mathematics:
<https://www.math.ucla.edu/~tom/Games/dots&boxes.html>

Timbiriche (juego). (8 de mayo de 2024). Obtenido de Wikipedia: [https://es.wikipedia.org/wiki/Timbiriche_\(juego\)](https://es.wikipedia.org/wiki/Timbiriche_(juego))