Proyecto Final

Luis Fernando Yang Fong Baeza
fernandofong@ciencias.unam.mx

Valeria García Landa
vale.garcia.landa@gmail.com

Francisco Valdés Souto
fvaldes@ciencias.unam.mx

17 de Octubre 2024

1. Introducción

Para este proyecto se dividirá en dos etapas para que el alumno tenga una transición mucho más tranquila y pueda realizarse con éxito. El alumno deberá de realizar el juego conocido como `Nonogram`, explicado a continuación. El proyecto se dividirá en dos fases, la creación de la lógica detrás del juego (back-end) y la conexión con la interfaz del juego y ajustarla para cumplir con lo que se pide.

2. Explicación del juego

El juego en realidad es muy sencillo, sin embargo, implementarlo no es trivial puesto que tiene muchos escenarios que se pueden volver muy complejos, para evitar estos escenarios, se han simplificado y ajustado las reglas del juego para que sea más sencillo de programar.

`Nonogram` tiene como propósito llenar una matriz de $n \times n$ con dos posibles estados, *lleno* o *vacío*, esto simplemente comunicándole al usuario cuántos elementos de una columna están llenos y cuántos elementos de un renglón están llenos. El usuario a través de deducciones lógicas tiene que determinar si una casilla cualquiera está vacía o está llena.

Este juego consta de 3 niveles, principiante, intermedio y avanzado. La lógica es exactamente la misma, lo único que varía entre un nivel y otro es el tamaño de la matriz.

1. Fácil: 5×5
2. Medio: 10×10
3. Difícil: 15×15

A continuación, se muestra un ejemplo de un juego de Nonogram con matriz de tamaño 6×6 en la Figura 1

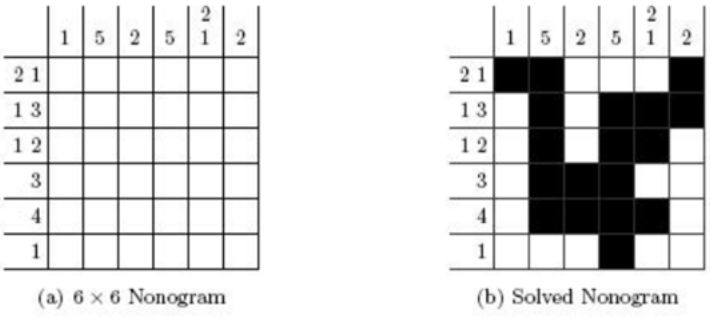

Figura 1: Un ejemplo de cómo se debe de ver una instancia del juego aunque no se hayan marcado los espacios *vacíos*.

2.1. Reglas del juego

El usuario solo consta de 3 vidas, es decir, el usuario puede equivocarse en la elección que haya escogido para una casilla a lo más 3 veces, al cuarto error, el usuario pierde automáticamente. Si el usuario completa toda la matriz, entonces el usuario gana.

En esta versión del juego, el usuario no puede contar con `hints` adicionales mas que los números que se presentan en la parte superior de la matriz y los números que se presentan del lado izquierdo de la misma.

Para determinar el estado de una casilla, el usuario seleccionar entre marcar vacío o marcar lleno, esto debe ser especificado antes de marcar una casilla y por default empieza en estado para marcar. Si el usuario quiere empezar a marcar en el otro estado, tiene que espcificarlo antes de empezar a marcar la(s) casilla(s).

Inicialmente todas las casillas tienen que empezar como *no marcadas*, este estado inicial es distinto al de *vacío*.

3. Primera parte del proyecto

La primera parte del proyecto en realidad es bastante específica, programar el juego aunque sea en terminal, imprimiéndolo en pantalla y especificando claramente el estado del juego después de una operación dictada por el usuario a través del teclado. El formato de la entrada en esta parte es a libre criterio, así como la impresión de la matriz de juego, sin embargo, el usuario tiene que ser capaz de ver el estatus del juego después de haber hecho una acción, ya sea cambiar de estado para marcar o marcar una casilla o en su defecto, notificarle al usuario que ha ganado o perdido el juego.

Una responsabilidad del alumno es el idear un algoritmo para programar la creación de las matrices en todos los niveles, con la restricción de que los números presentados en las columnas, tienen que encajar con los números presentados en los renglones, es decir, que no sea posible que una columna indique que una casilla es vacía y que un renglón especifique que esa misma casilla está llena.

En esta primer parte es donde el alumno debe de considerar implementar los 3 patrones de diseño expuestos por los compañeros en clase, no necesariamente tienen que verse claramente en la implementación del proyecto, aunque se espera al menos un código limpio (empaquetado correctamente).

De igual manera y para conectarlo con la teoría, se le ha pedido al alumno que haga el diagrama de clases del proyecto antes de empezar a programarlo, aquí sí se debe de incluir de alguna manera (ya sea en el mismo diagrama con un código de colores o en distintos archivos), los patrones que serán utilizados en el código fuente del proyecto.

3.1. Consideraciones para el algoritmo de creación

Si el alumno utiliza un algoritmo completamente aleatorio, es decir, que para los n renglones (o columnas) se obtengan números al azar, esto llevará a un nonogram no necesariamente resolvible, lo cual repercutirá en su calificación de la primera parte.

Para evitar esto, se ha hecho un análisis en clase en el cual se demostró que para que una columna o renglón con $pistas = \{p_1, p_2, \dots p_k\}$ sea resolvible de manera inmediata se tiene que cumplir la siguiente fórmula:

$$n = (\sum_{i=0}^{k} h_i) + |pistas| - 1$$

Un renglón o una columna se consideran como resolvible de manera inmediata si y solo si se puede determinar exactamente cuantas casillas son vacías y cuantas son llenas únicamente con la pista correspondiente presentada, por ejemplo, si en un nonogram de 5×5 un renglón contiene como pista $\{1, 3\}$. Tomando como ejemplo la Figura 1, no cuenta con ninguna columna o renglón resolvible de manera inmediata puesto que todas requieren de al menos otra columna o renglón para resolver el juego.

Para que un juego de nonogram sea fácilmente resolvible tiene que contar al menos con una de estas columnas, de lo contrario, el juego se puede volver tedioso y muy largo, un escenario que puede pasar si simplementemente se toman únicamente números al azar.

3.2. Criterios de evaluación

Para tener una rúbrica clara y que el alumno entienda lo que se espera que se aprenda del curso, se muestra a continuación una rúbrica de qué es lo que se va a evaluar:

1. Creación y representación correcta de los diagramas de clase y diagramas de patrones implementados → 30 %
2. Robustez del proyecto → 20 %. Por robustez del proyecto se refiere a que no tenga fallas en tiempo de ejecución si el usuario ingresa algo mal y que el juego pueda continuar su ejecución.
3. Implementación correcta de los patrones de software → 30 %. En este punto se incluye la creación de matrices de juego, así como la determinación de si el usuario ya ganó o no, etc...
4. Generación de instancias de juego mediante algoritmo no-aleatorio → 20 %.

La fecha de entrega para esta primer parte del proyecto es para el día 19 de Noviembre de 2024 antes de las 23:59:59.

4. Segunda parte del proyecto

El desarrollo de la segunda parte consta de conectar la lógica del proyecto ya implementado para que funcione mediante una interfaz para interacciones con el usuario utilizando el framework conocido com Swing (Para Java 8+).

El esqueleto para esta interfaz es provisto por el ayudante de laboratorio, sin embargo, el alumno tiene la obligación de modificar la interfaz para que esta reaccione ante los botones que en ella se presentan, así como mostrar un mensaje de confirmación en la pantalla cuando el usuario ha perdido o cuando el juego se ha completado.

Como el nivel es determinado por el usuario, puede ser implementado de varias formas, ya sea mediante dos interfaces, una que se muestre primero dando al usuario la posibilidad de escoger su dificultad o que reciba una cadena de entrada desde la entrada estándar o vía línea de comandos, esto es a criterio del alumno y no sumará ni restará puntos.

Lo anterior es con el afán de tener una aplicación completa y profesional de desarrollo de software. Como se está hablando de una aplicación formal, también se incluyen los criterios de evaluación para esta segunda parte del proyecto.

4.1. Criterios de evaluación

Para esta segunda parte se evaluarán los siguientes puntos:

1. Responsividad para variar el tamaño de la interfaz dependiendo el nivel. 30 %
2. El funcionamiento correcto de la interfaz con respecto al juego. 50 %
3. Implementación de algún indicador en la interfaz para las vidas restantes del usuario. 20 %

La fecha de entrega para esta segunda parte del proyecto es para el día 29 de Noviembre de 2024 antes de las 23:59:59.

5. Cálculo de la calificación final

Ambas partes del proyecto son importantes pero no cuentan de la misma manera ya que el propósito del curso es aprender respecto al Modelado y Programación de soluciones, lo más importante es la primer parte del proyecto y a la cual se le destinará la mayor cantidad de la calificación final del proyecto. (65 %)

Como profesionales, también es importante que se entreguen soluciones de calidad como producto final pero no es tan estricto en este curso, de manera que la segunda parte del proyecto consta del 35 % restante.

5.1. Puntos extra

Debido a que el propósito del curso en realidad es de utilización de patrones de diseño y modelado de soluciones y no tanto la creación y el uso de interfaces, se ha decidido incluir en esta parte del proyecto puntos extra sobre la calificación total del proyecto, en particular 3 puntos extra escritos a continuación.

- Implementación de dos niveles adicionales Principiante, el cual consta de crear una instancia del juego de tamaño 5×5 pero indicándole al usuario al menos una casilla vacía por columna o por renglón, así como la implementación del nivel Intermedio, el cual consta de lo mismo que lo anterior pero con una instancia de juego de 10×10. 1pt.
- Implementación de un sistema de pistas, en el cual se le otorguen a lo más 3 pistas el cual le digan al usuario el estado real de una casilla de su elección. 1 pt.
- Cuando el usuario logre completar todas las casillas llenas de una columna o de un renglón, el sistema deberá de completar todas las demás casillas que sean vacías pero que no hayan sido marcadas por el usuario, si esto ocurre para el último renglón o columna por llenar, entonces el programa debe de marcar automáticamente todas las casillas restantes y declarar que ha ganado el juego. 1.5pts.

Para los puntos extra, no existirán medios puntos, simplemente se juzgará de manera binario (se obtuvo o no se obtuvo), de manera que solo existen estos posibles resultados, 0 puntos extra, 1 punto extra, 1,5 puntos extra, 2 puntos extra, 2,5 puntos extra o 3,5 puntos extra sobre calificación final del proyecto

6. Cálculo total de la calificación

La calificación final del proyecto se obtendrá mediante la siguiente fórmula: Sean A la calificación final obtenida en la primer parte del proyecto, B la obtenida en la segunda parte del proyecto y p_e el número de puntos extras realizados, entonces:

$$Cal_{proyecto} = \frac{6{,}5 \times A}{10} + \frac{3{,}5 \times B}{10} + p_e$$