

# Nonogram Backtracking Algorithm

Tarea Programada 1 de Análisis de Algoritmos

1<sup>st</sup> Ignacio Álvarez Barrantes  
2019039643

2<sup>nd</sup> Sebastián Gamboa Bolaños  
2019044679

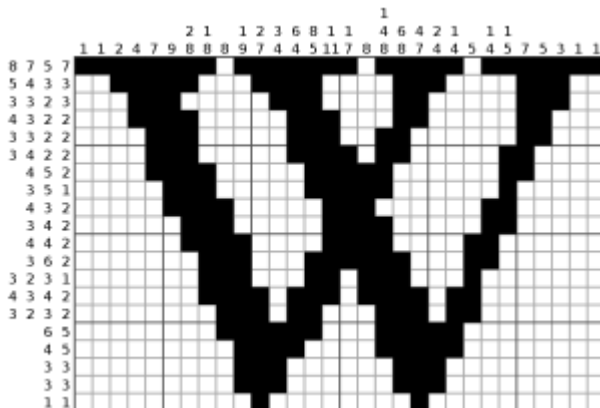
**Abstract**—This project consists in the elaboration of a computer application capable of solving and displaying in real time the solution to a NxM nonogram using backtracking. The project is developed using Unity and C-Sharp. The program can load a specific nonogram and it's giving hints from a txt file, represent it graphically and solve it.

encontrar una solución es el mismo que usaría una persona: línea por línea (o por columnas), usando lógica y matemática para descartar o verificar los espacios (poda). Sin embargo, hay situaciones en las que se tiene que adivinar, ya que las pistas no son suficientes para continuar con certeza. En estos es posible llegar a una contradicción, en cuyo caso hay que retroceder y seguir por la otra opción (backtracking).

## I. INTRODUCCIÓN

Nonogram, o "Paint by numbers", es un juego de lógica japonés que consiste en rellenar las celdas de una cuadrícula siguiendo algunas pistas numéricas. Se dan pistas por cada fila y cada columna de la cuadrícula; estas consisten en una secuencia de números enteros que indican la cantidad de celdas contiguas que se deben rellenar en esa fila o columna. Así por ejemplo, la secuencia (3 4) indica que en esa fila hay un grupo de 3 celdas rellenas seguido de un grupo de 4. El tamaño del espacio en blanco que está antes y después de cada grupo es desconocido.

Ejemplo:



En la imagen anterior se puede observar un ejemplo de un nonograma de (20x30) con sus respectivas pistas a lo largo de los ejes X y Y. En muchos casos las soluciones de estos nonogramas representan una imagen aunque no es obligatorio que cumplan con este requisito.

El problema de resolver un nonogram es NP-duro, y en general, el algoritmo computacional más eficiente para

Mediante el uso de Unity Engine se debe realizar un programa capaz de cargar un nonograma, junto con sus respectivas pistas, mediante un archivo de texto. El nonograma puede ser de un tamaño NxM a su vez el programa debe graficarlo y ser capaz de resolverlo por medio del uso de "Backtracking". El algoritmo de resolución debe ser lo más eficiente posible.

El formato del archivo txt es el siguiente:

```
Línea 1: x, y      //número de filas y número de columnas
Línea 2: FILAS
Línea 3 hasta x+2: a, b, c ... //pistas por fila
Línea x+3: COLUMNAS
Línea x+4 hasta y+4: a, b, c ... //pistas por columna
```

Un ejemplo del formato aplicado:

```
2, 3
FILAS
2
1, 1
COLUMNAS
2
1
1
```

Además se debe de incluir un menu principal donde poder cargar el archivo de texto y una vez cargado poder proceder a la pantalla de resolución del nonograma.

## II. TRABAJO RELACIONADO

### A. Creación de la interfaz gráfica

Para la temática de la interfaz se optó por un estilo oriental, basado en el "Sakura", árbol japonés. Se hace uso de colores rosados claros y morados oscuros característicos del árbol. La aplicación constará de 2 escenas principales.

La primera escena será el menú principal, esta contará con 3 botones:

- Botón de Play: para hacer uso de este botón primero se tuvo que haber cargado nonograma, de no ser así el programa no imprimirá nada en pantalla. Este botón cargará la siguiente escena donde se llevará a cabo la resolución del nonograma.
- Botón de Load Nonogram: al presionar este botón se abrirá el explorador de archivos para poder elegir el archivo txt con el nonograma en el formato especificado en la introducción de este documento.
- Botón Exit: este botón no tiene uso alguno si el programa es corrido desde el mismo Unity, pero una vez compilado y hecho un ejecutable al presionar este botón la aplicación procederá a cerrarse.

Imagen de la primera escena:

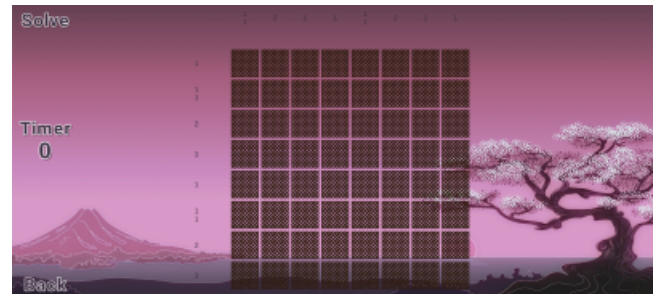


La segunda escena es donde se lleva a cabo la resolución del nonograma y cuenta con 2 botones, un temporizador y el nonograma.

- Botón de Solve: Este botón iniciará el algoritmo de backtracking que resolverá el nonograma en caso de haber solución.
- Botón de Back: Este botón regresará al usuario al menú principal donde podrá volver a cargar otro nonograma y repetir el proceso n veces.

- Temporizador: Al presionar el botón de solve este iniciará a aumentar cada segundo hasta que el nonograma sea resuelto, de esta forma se facilitará el análisis de resultados.
- Nonograma: Este se generará automáticamente y se modificará en tiempo real para representar las casillas marcadas por el algoritmo.

Imagen de la segunda escena:



## III. ALGORITMO BACKTRACKING

### A. Backtracking y Nonogram

Backtracking es un algoritmo para encontrar todas (o algunas) soluciones a un problema que genera candidatos para las soluciones de forma incremental. Tan pronto como determina que un candidato no puede conducir a una solución válida, lo abandona. El retroceso se trata de elecciones y consecuencias.

Abandonar a un candidato generalmente resulta en visitar una etapa previa del proceso de resolución de problemas. Esto es lo que significa "retroceder": visitar una etapa anterior y explorar nuevas posibilidades a partir de allí.

Por lo general, aparte del problema original y el objetivo final, también tenemos un conjunto de restricciones que la solución debe satisfacer.

Las implementaciones más simples (léase "las más tontas") a menudo usan poca o ninguna "lógica" o "percepción" del problema. En cambio, intentan frenéticamente encontrar una solución mediante conjeturas.

Un algoritmo de retroceso puede considerarse como un árbol de posibilidades. En este árbol, el nodo raíz es el problema original, cada nodo es un candidato y los nodos hoja son posibles candidatos a la solución. Atravesamos el

árbol primero en profundidad desde la raíz hasta un nodo hoja para resolver el problema.

Para lograr resolver este problema, y lograr entender el algoritmo implementado, primero hay que entender a cabalidad lo que es y como funciona un nonogram.

Los nonogramas, también conocidos como Picross o Griddlers, son rompecabezas de lógica de imágenes en los que las celdas de una cuadrícula deben colorearse o dejarse en blanco de acuerdo con los números al costado de la cuadrícula para revelar una imagen oculta. En este tipo de rompecabezas, los números son una forma de tomografía discreta que mide cuántas líneas continuas de cuadrados rellenos hay en una fila o columna dada. Por ejemplo, una pista de "4 8 3" significaría que hay conjuntos de cuatro, ocho y tres cuadrados rellenos, en ese orden, con al menos un cuadrado en blanco entre conjuntos sucesivos.

Para resolver un rompecabezas, es necesario determinar qué celdas serán cajas y cuáles estarán vacías. Los solucionadores a menudo usan un punto o una cruz para marcar las celdas que están seguros son espacios. Las celdas que pueden determinarse por lógica deben llenarse. Si se utiliza la suposición, un solo error puede extenderse por todo el campo y arruinar completamente la solución. A veces, un error sale a la superficie solo después de un tiempo, cuando es muy difícil corregir el rompecabezas. La imagen oculta juega poco o nada en el proceso de resolución, ya que puede inducir a error. La imagen puede ayudar a encontrar y eliminar un error.

Los acertijos más simples generalmente se pueden resolver mediante un razonamiento en una sola fila (o una sola columna) en cada momento dado, para determinar la mayor cantidad posible de casillas y espacios en esa fila. Luego intente con otra fila (o columna), hasta que no haya filas que contengan celdas indeterminadas. Los rompecabezas más difíciles también pueden requerir varios tipos de "¿y si?" razonamiento que incluye más de una fila (o columna). Esto funciona en la búsqueda de contradicciones: cuando una celda no puede ser una caja, porque alguna otra celda produciría un error, definitivamente será un espacio. Y viceversa. Los solucionadores avanzados a veces pueden buscar incluso más profundamente que en el primer "¿y si?" razonamiento.

A veces es imposible evitar adivinar. Por ejemplo, al final de un rompecabezas si dos filas y dos columnas contienen cada una una casilla no contabilizada, no hay información para determinar si las casillas residen en las celdas superior izquierda e inferior derecha o en la casilla superior derecha e inferior izquierda.

## B. Algoritmo

Con la lógica del backtracking, y el nonogram claros, se puede empezar a implementar el algoritmo. Este se plantea de la misma manera que un backtracking tradicional. Con algunas funciones características que hacen el algoritmo general más distinto.

Logicamente, en la maquina, el nonogram se tiene representado como una matriz de ceros, la cual se va llenando con un uno, si ese espacio se marca, o un dos si ese espacio se queda vacío.

Se empieza evaluando en columnas, se asume que el espacio va marcado. Luego, se llama a una función auxiliar, la cual revisa que no haya ningún problema en el nonogram. Esto quiere decir que esa casilla marcada no haga que la solución sea imposible, o cause algún problema en general.

Si no hay ningún problema, la casilla se queda marcada con un uno, significando que está marcada, y se prosigue con la siguiente casilla. Se repiten los mismos pasos, hasta que cuando en uno de esos, se encuentre un error.

Al encontrarse un error, se borra ese uno, y se reemplaza con un 2, se revisa que no haya ningún problema. Si no hay problemas, se prosigue. Si si hay problema, entonces el algoritmo corta su ciclo de recursion, y se devuelve a la casilla pasada a encontrar una solución diferente a la que ya estaba, ya que si en la casilla actual no hay solución, el error tiene que estar en las casillas pasadas.

Si al devolverse a la casilla pasada, tampoco hay una solución diferente, hace lo mismo y se devuelve otra vez a la casilla pasada, para encontrar una casilla en la que haya una solución diferente a la actual. Y así repetir todo este proceso hasta encontrar la solución de todo el nonogram.

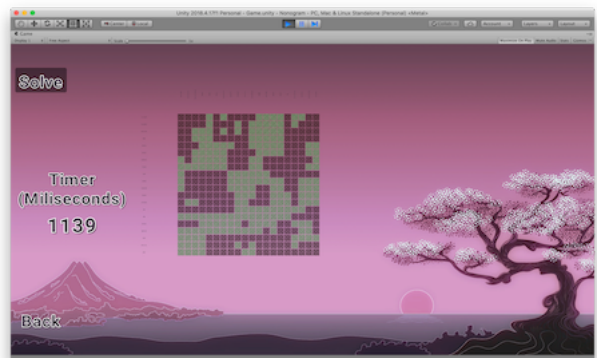
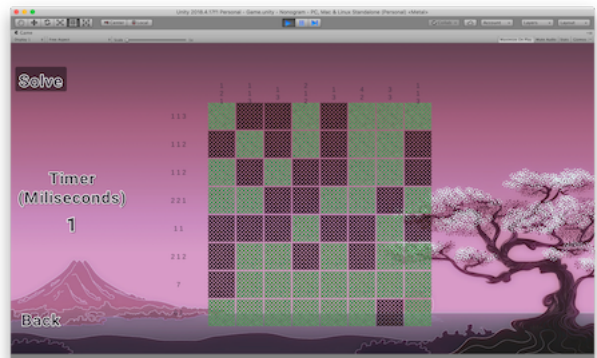
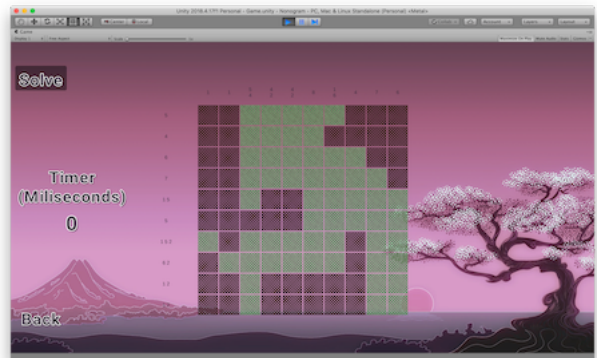
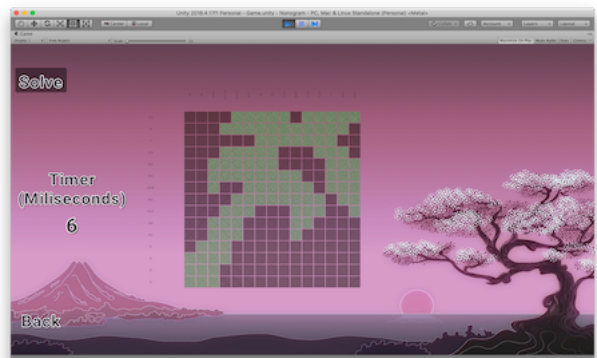
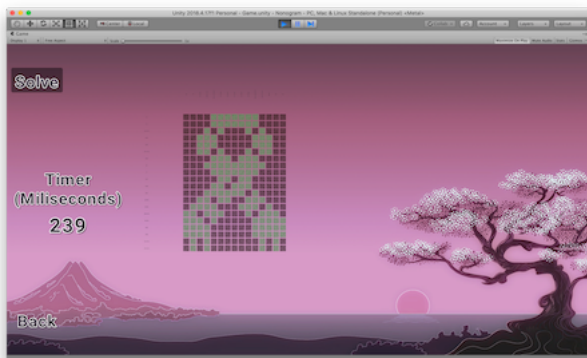
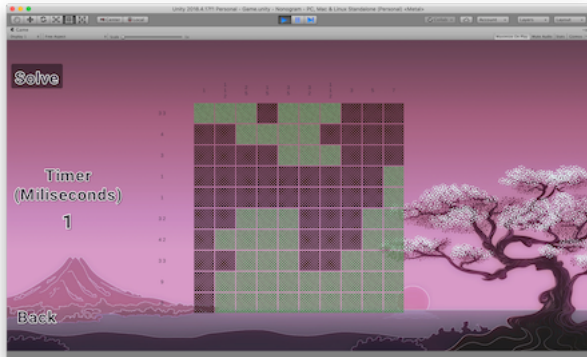
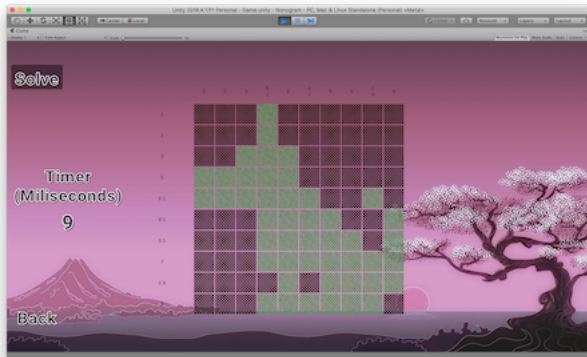
## IV. ANÁLISIS DE RESULTADOS

### A. Velocidad del Algoritmo

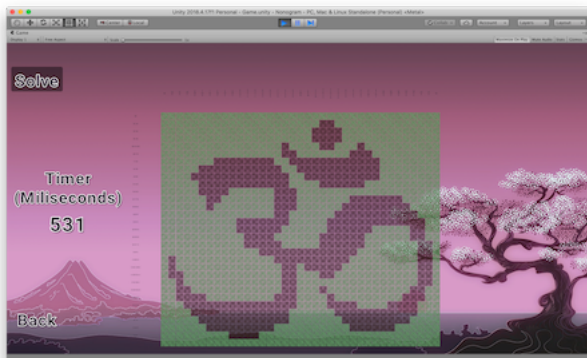
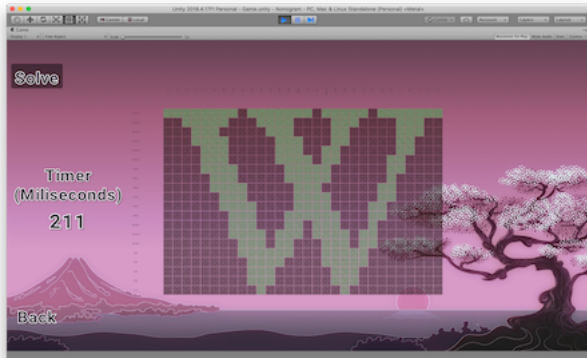
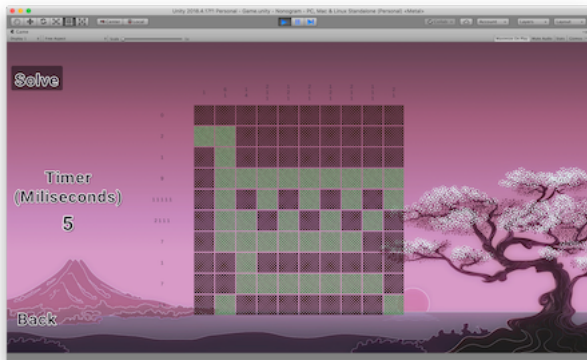
El algoritmo al ser de backtracking, es de fuerza bruta. Tiene que hacer muchas suposiciones. Por lo tanto calcular la velocidad en la que el algoritmo va a producir la respuesta es muy difícil. Este tiempo varía mucho de acuerdo a cada nonogram. Puede que la solución se encuentre muy arriba en la rama izquierda del árbol, y se encuentre muy rápido la respuesta. Mientras que también es una posibilidad que la solución esté en la rama derecha en el ultimo nivel. En ese caso el algoritmo durará mucho en encontrar la respuesta.

Dicho esto, la mayoría de nonograms que se probaron, tienen un tiempo de ejecución menor a 1000 milisegundos. Con excepciones, ya que probamos algunos nonograms complejos que duraban hasta 10 segundos en resolverse, claro que casi ningún nonogram duraba más de un segundo.

Se agregan a continuación unas fotos de algunos tiempos de ejecución del programa.







Brackeys. (2017). PAUSE MENU in Unity [Video]. Retrieved 8 March 2020, from <https://www.youtube.com/watch?v=JivuXdrIHK0>.

Technologies, U. (2011). Unity - Manual: The Project window. Docs.unity3d.com. Retrieved 8 March 2020, from <https://docs.unity3d.com/Manual/ProjectView.html>.

Zolran. (2014). Creating a TextMeshPro object via Script [Video]. Retrieved 8 March 2020, from <https://www.youtube.com/watch?v=7leJy3axiRM>.

Scripting the creation of Text Meshes ?. Unity Forum. (2007). Retrieved 8 March 2020, from <https://forum.unity.com/threads/scripting-the-creation-of-text-meshes.6487/>.

## V. BIBLIOGRAFÍA

Link al repositorio de git :

<https://github.com/Nacho-Alvarez0412/Nonogram-Analysis-de-Algoritmos->

Youtube (2018). Unity Tutorial - Create a Graph. [video] Available at: <https://www.youtube.com/watch?v=CmU5-v-v1Qo> [Accessed 29 Feb. 2020].

Brackeys. (2017). START MENU in Unity [Video]. Retrieved 8 March 2020, from <https://www.youtube.com/watch?v=zc8ac-qUXQY>. [Accessed 29 Feb. 2020].