

Primera Entrega Proyecto

Interfaz Gráfica

María José Niño Rodríguez¹

David Santiago Quintana Echavarria¹

¹Departamento de Ingeniería de Sistemas, Pontificia Universidad Javeriana

Bogotá, Colombia

{ma.nino, quintanae-david}@javeriana.edu.co

4 de octubre de 2022

Resumen

En este documento se presenta una pequeña explicación del juego Flow Free y el diseño de la interfaz gráfica para que se pueda jugar

Palabras clave: juego, puntos, flujo, tuberías, cuadrícula.

Índice

1. Juego	1
2. Diseño Interfaz Gráfica	2
2.1. Algoritmo	3
2.1.1. Jugador	5
2.1.2. Computador	6

1. Juego

Para poder explicar la interfaz del juego es necesario primero hablar de cómo funciona este. El juego tiene varios niveles y cada uno tiene una cuadrícula la cual puede variar entre tamaños de 5x5 hasta 15x15. Dentro de cada tablero hay pares de puntos de colores que se encuentran en algunos de los cuadrados de la cuadrícula, los puntos se deben unir sin que hagan una intersección con otro color para generar la tubería, completar el flujo y resolver el rompecabezas [1].

Para lograr el objetivo, el algoritmo recibe dos entradas de diferentes periféricos del computador, el primero es desde el puntero para ubicarse en un cuadrado de la cuadrícula; sin embargo, para que funcione, solo es posible ubicarse desde un punto de color o desde una parte de la tubería que ha sido iniciada desde un punto de color y no cuando el cuadrado de la cuadrícula este en negro. Después de ubicarse en un lugar es posible usar el siguiente periférico, el teclado, con el cual es posible mover el flujo hacia adelante, atrás, a la izquierda o a la derecha pero están limitados según la posición de la tubería y de los puntos de colores.

En la interfaz existen 5 tableros que van de tamaños desde 4x4 hasta 8x8 y entre los cuales el usuario se puede mover libremente sin necesidad de terminar, en el caso que termine alguno de los tableros se informara en pantalla que ganó, pero, en caso de cambiar de tablero y volver a uno anterior este tendrá que ser resultado otra vez si se quiere ganar.

2. Diseño Interfaz Gráfica

Para la interfaz gráfica se hizo uso de la extinción PyGame y para poder usarla se necesita que sea instalada [2]. La interfaz esta compuesta por el tablero, que cambia el tamaño de la cuadrícula a medida que se cambia de nivel, el título del juego en la parte superior y en la parte inferior dos flechas y un botón, una que apunta a la izquierda, para ir a un nivel anterior, una que apunta a la derecha para avanzar de nivel y el botón para que el computador resuelva el juego. En el caso que el computador complete el rompecabezas aparecerá una quinta parte en la interfaz para informar que se ganó el juego. A continuación se muestra la interfaz.

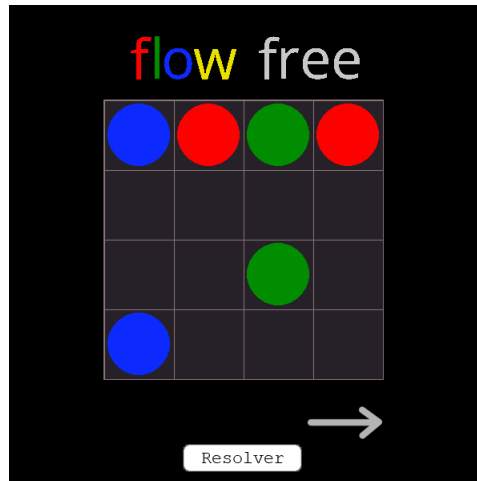


Figura 1: Primer nivel de la interfaz (sin flujos)

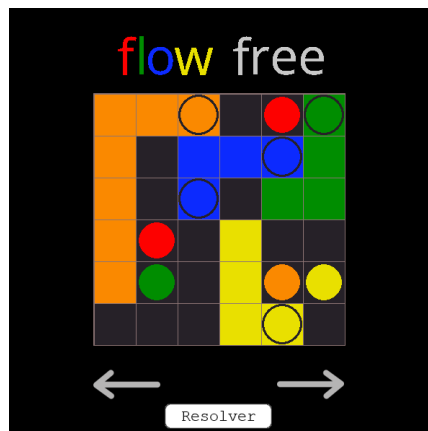


Figura 2: Nivel intermedio de la interfaz (algunos flujos)

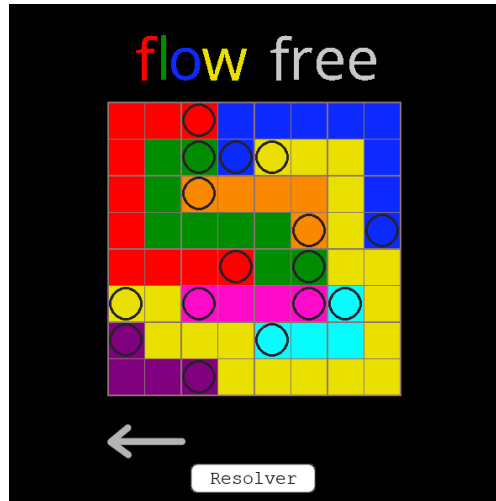


Figura 3: Último nivel de la interfaz (flujos completados)

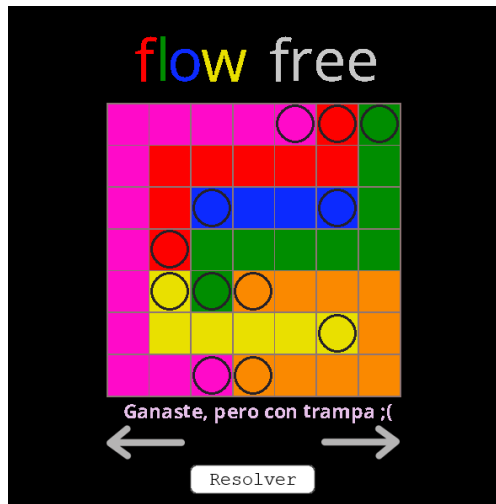


Figura 4: Juego solucionado por el computador

2.1. Algoritmo

Para el desarrollo del algoritmo y posterior despliegue en la interfaz se crearon tres clases, el diagrama que muestra sus relaciones se expone en la figura 5

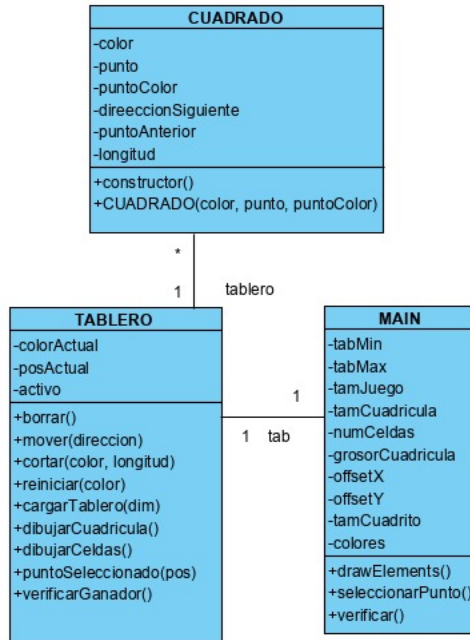


Figura 5: Diagrama de Clases

En el diagrama se puede observar la clase MAIN que esta encargada de la interfaz gráfica, y esta tiene una relación uno a uno con un tablero que contiene varios métodos tres de los más importantes son cargarTablero, que dado una dimensión diseña la cuadrícula, verificarGanador, la cual revisa si el jugador completó el tablero correctamente y mover, que dado una dirección continua el flujo de un color. Adicionalmente, el tablero esta compuesto de varios cuadrados que tienen la información de cada casilla dentro de la cuadrícula. Para poder entender mejor el proceso, a continuación se explica dos partes del algoritmo, la que permite al usuario jugar y la que resuelve el computador.

2.1.1. Jugador

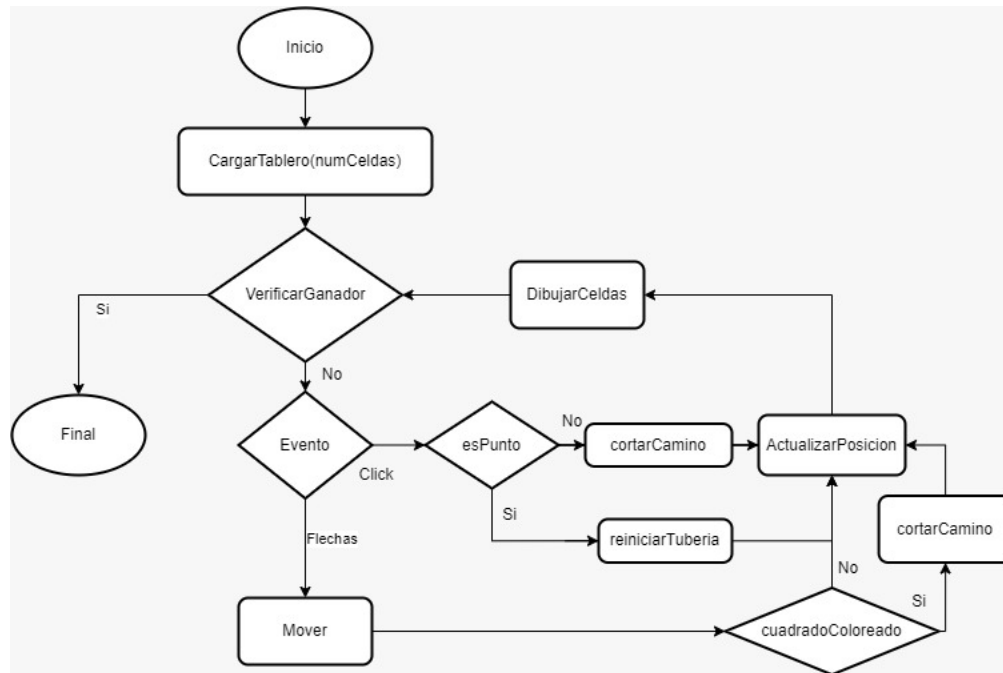


Figura 6: Diagrama de Flujo

En el flujo se crea un tablero según el tamaño de la cuadrícula, y se verifica si ya ganó, en el caso de haber ganado se termina, pero si no recibe dos posibles eventos.

1. Evento click

Cuando se hace uso de este evento se verifica si el cuadrado de la nueva posición tiene un punto o no.

■ El cuadrado tiene un punto

Cuando el click se hace sobre un cuadrado que tiene un punto se reinicia la tubería del color correspondiente y después se actualiza la posición en el tablero, se dibujan las celdas correspondientes y se vuelve a verificar el ganador.

■ El cuadrado no tiene un punto

Cuando el click se hace sobre un cuadrado que no tiene un punto pero esta dentro de una tubería se corta el camino desde ese cuadrado, se actualiza la posición en el tablero, se dibujan las celdas correspondientes y se vuelve a verificar el ganador.

2. Evento teclado

Cuando se hace uso de este evento se verifica que el cuadro pueda ser usado para la tubería, es decir, que no tenga color.

■ El cuadrado esta coloreado

Cuando el cuadrado tiene una tubería de otro color o de el mismo se corta el flujo del color que interrumpió, se actualiza la posición y se vuelve a verificar el ganador.

■ El cuadrado no esta coloreado

Cuando el cuadrado esta libre de flujos entonces se actualiza la posición en el tablero, se dibujan las celdas correspondientes y se vuelve a verificar el ganador.

2.1.2. Computador

Cuando el jugador selecciona la opción de Resolver” el flujo del juego cambia, primero se vuelve a cargar un archivo que contiene los colores del tablero, pero este se encuentra rodeado de un valor indicativo que es el número 1, el cual representa el borde del tablero, después de esto el tablero se limpia de cualquier movimiento previo desarrollado por el jugador

El juego se puede entender como una lista de nodos que deben ser conectados entre si, en el caso que estos nodos representados por colores estén conectados y todos los cuadros del tablero poseen un color, se gana el juego. Una vez limpiado el tablero y se haya cargado el nuevo, se recorre el tablero en busca de identificar los nodos. Se implementan dos arreglos para su almacenamiento, el primero que contiene los nodos iniciales y el segundo los nodos finales, se recorre el tablero al encontrar un nodo, si su color ya se encuentra en la lista de nodos iniciales este se agrega a la lista de nodos finales de lo contrario se agrega a la lista de nodos iniciales.

Se desarrollo un algoritmo de fuerza bruta, este funciona de manera recurrente, primero valida dos heurísticas, si algún nodo se encuentra rodeado por casillas con un color diferente al vacío y ninguna es de su color significa que el nodo se encuentra encerrado sin posible salida para conectarse, por lo que no es posible resolver el tablero por ese camino. También se valida si cada nodo no se encuentra rodeado por tres casillas de su mismo color, ya que de encontrarse el caso, el nodo se intercepta con si mismo por lo que tampoco seria una posición correcta.

En cada llamado a la función primero se valida que el estado actual no incumpla con alguna regla de heurística, si cumple se revisa si el tablero se encuentre resuelto, de ser así se finaliza pues la la solución del tablero fue encontrada. En caso de que sea valido y no este resuelto se recorren los nodos iniciales, en cada caso se revisa la distancia con su respectivo nodo final, si la distancia entre sus dos coordenadas es mayor a uno, significa que estos no se encuentran conectados, por lo que este nodo debe dirigirse a otra posición.

Se evalúa cuáles son los posibles movimientos unitarios que puede tomar el nodo inicial (arriba, abajo, izquierda o derecha), es posible moverse a una casilla si el color de dicha casilla no tiene algún color asignado. Para cada posible dirección que tiene el nodo inicial este se mueve en dicha dirección, es pintada de su color y evalúa recursivamente la función para analizar si con ese movimiento se llega a una solución para el tablero. En caso tal que el movimiento cumpla con la solución del tablero es juego es finalizado y se indica que fue solucionado el nivel, por otro lado, de no cumplir con la solución, el nodo inicial regresa a su posición anterior y libera la casilla en la que se encontraba.

De esta manera se recorre diversos movimientos en el tablero hasta que alguno de ellos cumpla como solución del del nivel en el que se encuentra el juego.

Referencias

- [1] *Flow Free*. URL: https://en.wikipedia.org/wiki/Flow_Free (visitado 01-10-2022).
- [2] *(1187) Learning pygame by creating Snake [python tutorial] - YouTube*. URL: <https://www.youtube.com/watch?v=QFvqStqPCRu> (visitado 01-10-2022).