

Documento de diseño para el juego Flow Free

Pablo Santander Álvarez, Juan Jose Bolaños

Departamento de Ingeniería de Sistemas, Pontificia Universidad Javeriana
Bogotá, Colombia

santander_p@javeriana.edu.co, bolanos.jj@javeriana.edu.co

3 de octubre de 2022

Resumen

En este documento se presenta las especificaciones del diseño del Juego de “Flow Free” construido en el lenguaje de programación de Python. A lo largo del documento se encontrarán diferentes diagramas que explican a detalle el funcionamiento del juego y los algoritmos que lo componen, generando un entendimiento interno del juego y como este fue construido. **Palabras clave:** Flow Free, Python, Juego, Matriz, Diagramas, Diseño.

Índice

1. Introducción	1
2. Explicación de Flow Free	1
3. Librerías usadas en Python	2
4. Vista Lógica del Sistema	2
5. Vista de Proceso	4
6. Comportamiento del Juego	4
7. Interfaz de usuario	5
7.1. Inicio del juego y generación de Tablero	5
7.2. Indicación de parámetros para iniciar movimientos	6
7.3. Completación de un color	6
7.4. Victoria	7

1. Introducción

El presente documento tiene como objetivo mostrar el proceso realizado para la construcción del famoso videojuego para celular llamado “Flow Free”.

2. Explicación de Flow Free

Es un juego relativamente simple de jugar, se tiene una matriz cuadrada la cual contiene 4 o 5 parejas de colores dependiendo del tamaño de la matriz. El objetivo es conectar todos los colores con su pareja respectiva de mismo color. La dificultad proviene de el hecho que en muchos casos conectar un color puede interferir

en el camino de otra pareja de colores que deben conectarse, entonces se deben conectar cuidadosamente los colores para tener un camino posible por el cual los colores restantes puedan conectarse. Cuando una casilla de la matriz es pintada por un color, supongamos que es rojo, un color azul no podrá pasar por esa casilla y pintarla debido a que se encuentra ocupada por un color que no concuerda con azul. Es posible que luego de pintar una casilla, o más de una casilla, el jugador decida borrar las casillas pintadas y devolverse hasta cierto punto, o, comenzar desde un punto dado. Si el usuario luego de pintar una línea decide volver a iniciar desde el origen de un color en el tablero, entonces toda la línea que ya estaba construida se borrará e iniciará desde el origen, o en dado caso, puede decidir solo borrar cierta cantidad de casillas y por ende puede iniciar a pintar desde alguna casilla de la línea ya construida, lo que efectivamente borraría los movimientos que ocurrieron luego de que esa casilla en donde se inicia el nuevo movimiento fuera pintada. El usuario gana y pasa de nivel en el momento en que todas las parejas de colores fueron conectadas satisfactoriamente, cada nivel tiene varias soluciones posibles, y el juego recompensa de 0 a 3 estrellas dependiendo de la cantidad de movimientos que fueron necesarios para conectar todos los colores.

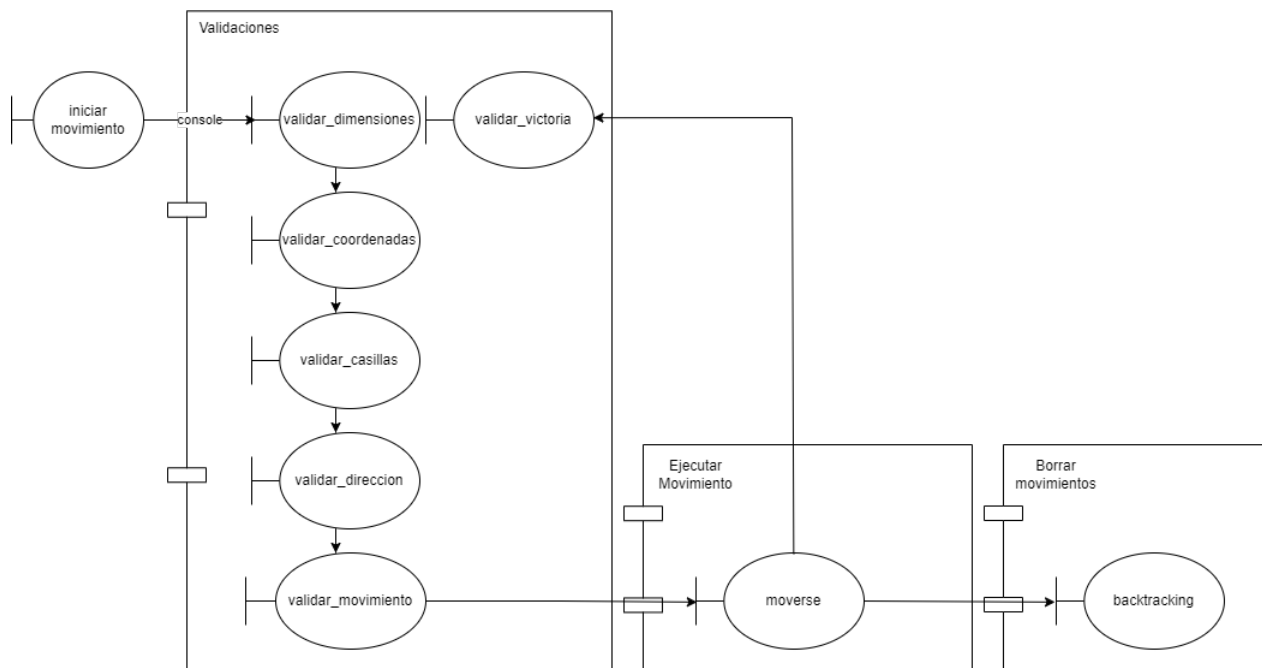
3. Librerías usadas en Python

numpy	NumPy es una biblioteca que da soporte para crear vectores y matrices grandes multidimensionales
random	Libreria para generar números pseudo-aleatorios
collections	Libreria que provee tipos de datos alternativos de Python

4. Vista Lógica del Sistema

En este apartado se abarcará el diagrama de componentes que representa el funcionamiento del sistema, especificando la comunicación de los objetos y su respectiva descripción.

En la ilustración 1 se visualizan la interacción de los componentes del software del sistema.

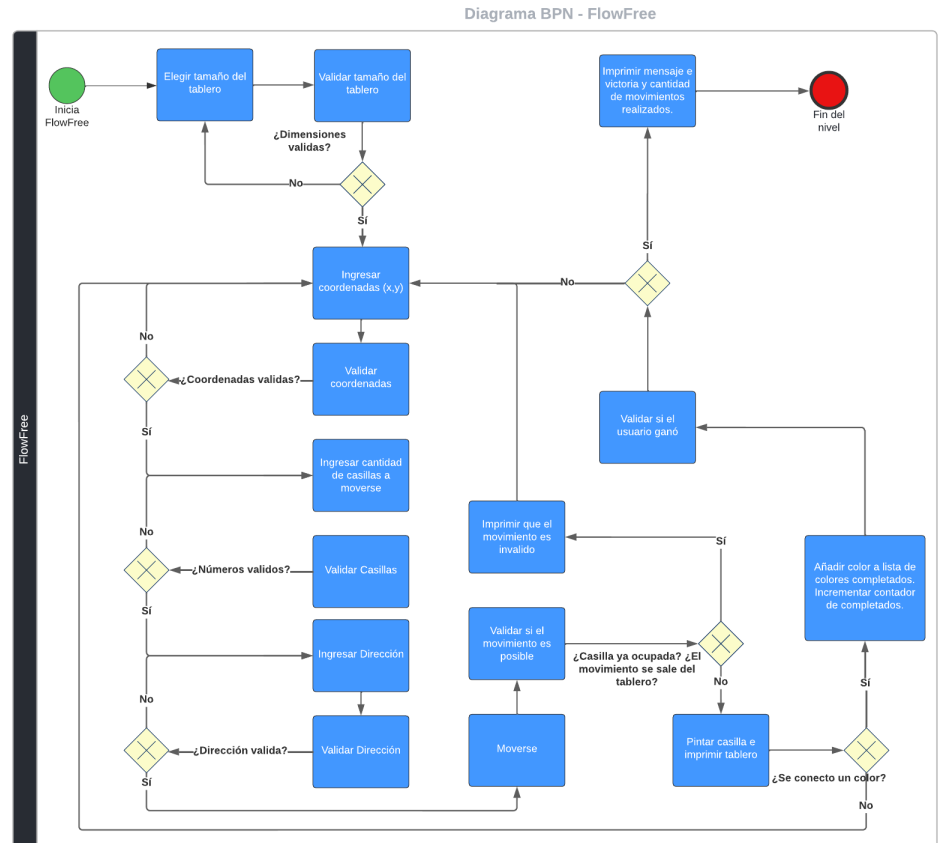


La explicación a detalle de los elementos existentes en los componentes y su interacción para el funcionamiento del sistema se describen en la tabla.

Nombre	Descripción
iniciar_movimiento	Este objeto es el encargado de iniciar el movimiento basado en los parámetros de movimiento ingresados por consola
validar_dimensiones	Este objeto es el encargado de validar que las dimensiones de la matriz cuadrada elegidas por el usuario sean validas
validar_casillas	Este objeto es el encargado de validar que la cantidad de casillas a moverse sean validas
validar_dirección	Este objeto es el encargado de validar que la dirección elegida para el movimiento sea una dirección posible.
validar_movimiento	Este objeto es el encargado de validar que el movimiento es posible al revisar casillas que estén ocupadas, al igual que movimientos que se salen del tablero.
moverse	Este objeto es el encargado de realizar el movimiento previamente validado, básicamente va pintando las casillas del color con el que se inicio el movimiento
backtracking	Este objeto es el encargado de realizar un proceso de backtracking en caso de que el usuario decida borrar movimientos y devolverse, ya sea a la posición inicial del color, o a alguna posición ya pintada, con el fin de iniciar un nuevo movimiento desde allí.
ganar	Este objeto es el encargado de validar si el usuario ha ganado y superado el nivel, validando entonces si todos los colores fueron completados.

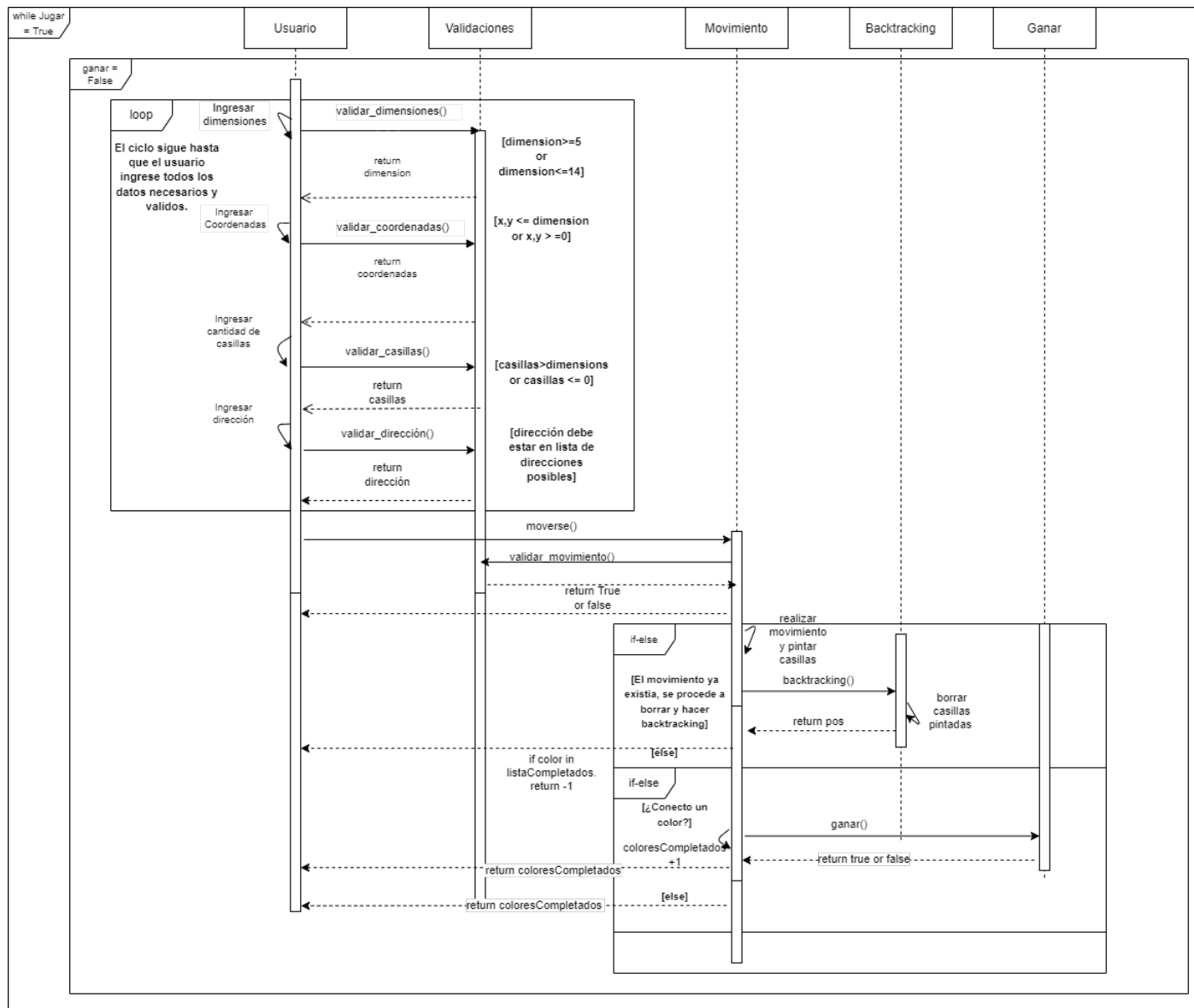
5. Vista de Proceso

A continuación, se desarrollara el proceso principal del juego Flow Free. La ilustración representa el flujo que se realiza para que el juego funcione. Este se basa, en los diferentes pasos y validaciones que ocurren mientras el jugador juega.



6. Comportamiento del Juego

En este apartado se abarcará el diagrama de componentes que representa el funcionamiento del sistema, especificando la comunicación de los objetos y su respectiva descripción.



7. Interfaz de usuario

Respecto a la interfaz de usuario mediante la cual el usuario podrá jugar Flow Free, se diseño mediante consola de comandos y no una interfaz gráfica. La interfaz se basa en información visual por consola que constantemente genera diferentes impresiones que sirven de guía para que el usuario pueda entender en todo momento que es lo que esta sucediendo en su sesión de juego.

7.1. Inicio del juego y generación de Tablero

De manera inicial, se imprime el titulo del juego y se le indica al usuario que ingrese la dimensión con la cual desea que se genere la matriz cuadrada del nivel. Una vez se selecciona un valor se imprime la matriz con las parejas de colores generadas aleatoriamente. Cabe resaltar que los colores están referenciados con números, donde dos números que sean iguales indica que son del mismo tipo o color.

```

C:\Users\Usuario\Desktop\ProyectoAlgoritmos\venv\Scripts\python.exe C:/Users/Usuario/Desktop/ProyectoAlgoritmos/main.py
=====
Bienvenido a Flow Free
=====
Ingresar la dimension del tablero deseado para jugar (5x5, 6x6, 7x7, 8x8, 9x9, 10x10, 11x11, 12x12, 13x13, 14x14): 6
[[0 0 0 0 0 0 0]
 [0 0 2 0 0 0 2]
 [0 0 0 0 0 0 0]
 [1 0 0 0 0 0 1]
 [0 0 0 0 0 4 0]
 [0 4 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [3 0 3 0 0 0 0]]

```

7.2. Indicación de parámetros para iniciar movimientos

En este momento, el usuario debera ingresar la coordenada en (x,y) donde desea iniciar el movimiento que pintara casillas. Aqui por supuesto se validan que las coordenadas sean validas, además de la validación de que el movimiento debe empezar desde un color de origen. Una vez se indican las coordenadas se pide ingresar la cantidad de casillas que desea moverse, finalmente la dirección en la que se quiere realizar el movimiento (arriba, abajo, izquierda, derecha) no se pueden movimientos diagonales. Cuando todo es validado correctamente, se imprime el primer movimiento (primera casilla recorrida y pintada), donde muestra la lista de colores completados hasta el momento, la lista de movimientos que ha realizado con un color dado, y finalmente el tablero con la casilla pintada. Por ejemplo en este caso el origen del color 2 esta en (1,2) y el siguiente movimiento ocurre en la casilla (1,3), por ende, en la lista de movimientos observamos las dos coordenadas consignadas, y por supuesto, vemos en el tablero que se pinto un 2 extra.

```

Ingresar coordenadas de origen (debe ser el origen de un color)
Coordenada x: 1
Coordenada y: 2
Cantidad de casillas que desea moverse: 3
--Direcciones posibles: arriba, abajo, izquierda, derecha--
Dirección (Solo puede moverse en una direccion por movimiento, en el siguiente movimiento puede cambiar direccion): derecha
Lista de colores completados: []
Movimiento: 1

Lista de movimientos del color 2
[[1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [2 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 2 2 0 0 2 0]
 [0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 1 0]
 [0 0 0 0 0 4 0 0]
 [0 4 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [3 0 3 0 0 0 0 0]]

```

7.3. Completación de un color

En esta interfaz, se indica por pantalla inmediatamente se conecte un color. Se indica el color que fue conectado y además este se añade a la lista de colores completados que sera impresa constantemente en movimientos posteriores.

```

Lista de movimientos del color 2
[[1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [2 3 4 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 --COLOR 2 CONECTADO--
[[0 0 0 0 0 0 0 0]
 [0 0 2 2 2 2 2 0]
 [0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 1 0]
 [0 0 0 0 0 4 0 0]
 [0 4 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [3 0 3 0 0 0 0 0]]

```

7.4. Victoria

Una vez que todos los colores son completados y validados, la función de victoria se ejecuta y se da por completado el nivel actual, imprimiendo por pantalla la cantidad de turnos que tomó ganar.

```
[2, 1, 4, 3]
Cantidad de colores completados: 4
[[0 0 0 0 0 0 0 0]
 [0 0 2 2 2 2 2 0]
 [0 0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1 0]
 [0 0 0 0 0 4 0 0]
 [0 4 4 4 4 4 0 0]
 [3 3 3 0 0 0 0 0]
 [3 0 3 0 0 0 0 0]]
=====
Felicitaciones, ha conectado todos los colores en 6 turnos.
=====
```