

FLICKCOLOR

Informe Proyecto 1

Autores:

Cano, Micaela 114024

Moyano, Yohana Karina 105536

Materia: Lógica para Ciencias de la
Computación

FlickColor

¿Qué es FlickColor?

Consta de una grilla de 14×14 , donde cada celda está pintada de uno de 6 posibles colores, y 6 botones, uno de cada color. Para jugar hay que presionar un botón de color C causa que la celda seleccionada actualmente de la grilla, pintada de color C' , al igual que todas las del mismo color C' adyacentes a esta, y las de color C' adyacentes a éstas últimas, y así siguiendo, se pinten de color C . Si $C = C'$ entonces no se produce cambio alguno.

Objetivo del Juego

El objetivo del juego es lograr pintar todas las celdas de un mismo color, presionando los botones la menor cantidad de veces posible.

Requerimientos

Notación: usaremos $\text{adyacente}C^*$ para referirnos a la clausura transitiva de la relación de adyacencia entre dos celdas del mismo color. Es decir, diremos que una celda X es $\text{adyacente}C^*$ a una celda Y si ambas son del mismo color y además X es o bien adyacente a Y , o existe una celda Z del mismo color que X e Y tal que X es adyacente a Z y recursivamente Z es $\text{adyacente}C^*$ a Y .

Funcionalidad

En el juego presenta una grilla de 14×14 en la cual podremos elegir en qué casilla de esta queremos empezar a jugar. En el panel izquierdo hay una botonera con 6 colores para poder seleccionar.

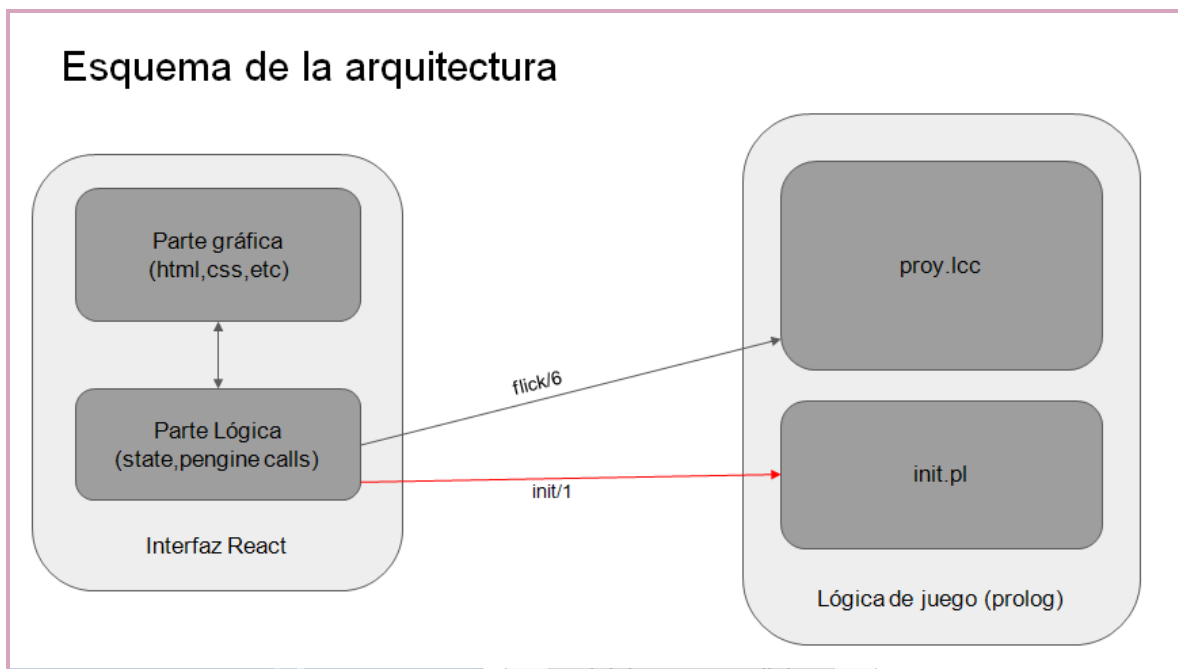
El juego se desarrolla al ir seleccionando alguno de estos 6 colores sin olvidar que se aplicará la regla `adyacenteC*` para el nuevo color seleccionado.

El concepto de esta regla es que el nuevo color se esparcirá hacia todas las casillas del mismo color que se toquen entre sí.

Se mostrará en todo momento la cantidad de celdas que se encuentran capturadas actualmente, es decir, las celdas que son `adyacenteC*` a la celda de origen. También se mostrará en todo momento, el historial de jugadas realizadas por el usuario (secuencia de colores presionados) desde el inicio de la partida.

El juego terminará cuando el tablero quede completado de un solo color.

Arquitectura de FlickColor



La primer llamada realizada del parte de react a prolog es la de `init/1` para generar el tablero

Luego La segunda llamada realizada es `flick/6` la cual me devolverá la cantidad de capturados y la Grilla actualizada

React:

Podemos encontrar 3 clases principales que extienden de React Component: Game, Board y Square.

En el constructor de la clase Game encontramos:

```
this.state = {
  turns: 0,
  grid: null,
  complete: false, // true if game is complete, false otherwise
  waiting: false,
  origin: undefined,
  clicks: [],
  capturadas: 0
};
```

Con estos datos, inicializamos el estado de la grilla.

```
this.handleClick = this.handleClick.bind(this);
this.handlePenguinCreate = this.handlePenguinCreate.bind(this);
this.onOriginSelected = this.onOriginSelected.bind(this);
this.penguin = new PenguinClient(this.handlePenguinCreate);
```

Y enlazamos estos manejadores de eventos a una instancia.

Llamadas a Prolog desde React

flick/6

Desde React en la clase Game llamamos al predicado flick/6 de prolog con los siguientes parámetros:

```
"flick(" + gridS + "," + color + ", Grid, "+fila+", "+col+", Capturadas)";
```

Este me devuelve:

- **Grid:** Una nueva grilla con los movimientos actualizados que vamos realizando.
- **Capturadas:** Cantidad de elementos adyacentes que están capturados actualmente.

Prolog:

Estrategia de Resolución:

El flujo hacia la grafica es a través del predicado principal **flick/6**, dentro de este predicado:

Llamo al método "**obtenerElemGrilla**" con el cual obtenemos el color del elemento seleccionado actualmente.

Realizamos una comparación entre el color seleccionado y el nuevo color a pintar.

Llamamos al método **cambiarC**, en este método se compara el elemento actual con el anterior, si son iguales cambió su color en la grilla y busco sus **adyacentes**(arriba, abajo, izquierda y derecha) y si son del mismo color aplicó la regla de **adyacentesC***.

Obtendremos la grilla actualizada y la cantidad de adyacentes.

Predicados principales utilizados:

- **flick/6**: Dada una Grilla, color Nuevo, posición X, posición Y, devuelvo la cantidad de capturados actuales y la Grilla con los colores actualizados.
- **obtenerElemGrilla/4**: Dada una posición X, posición Y, una grilla, obtengo el elemento E de la grilla en la posición (X,Y).
- **cambiarC/7**: Dada una Grilla, color actual, color nuevo, una posición X, posición Y, devuelvo una Grilla con los colores actualizados y la cantidad de adyacentes.
- **replace/4**: Dada una posición X, una lista y un elemento nuevo, devuelvo una lista con el elemento nuevo reemplazado en la posición (X).
- **cambiarElemG/5**: Dada una posicion X, posicion Y, una grilla y un color nuevo, devuelvo una nueva grilla con el elemento que se encuentra en la posicion (X,Y) cambiado de color.

- **adyacentes/7:** Dada una grilla, color actual, color nuevo, y dos posiciones devuelvo una grilla actualizada y la cantidad de adyacentes que cumplen con la condición AdyacentesC*.
- **sum_num/3:** Dado dos números lo sumo y doy un resultado

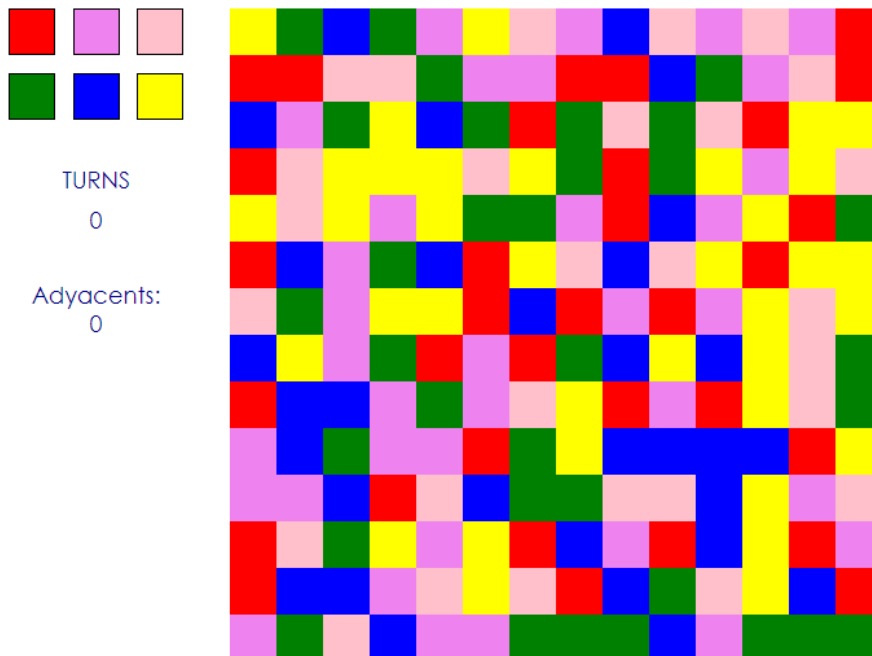
Predicados predefinidos:

Los predicados predefinidos que utilizamos son los siguientes:

- **nth0/3:** Buscamos un elemento dada una posición.
- **nth0/4:** Reemplazamos un elemento dada una posición.

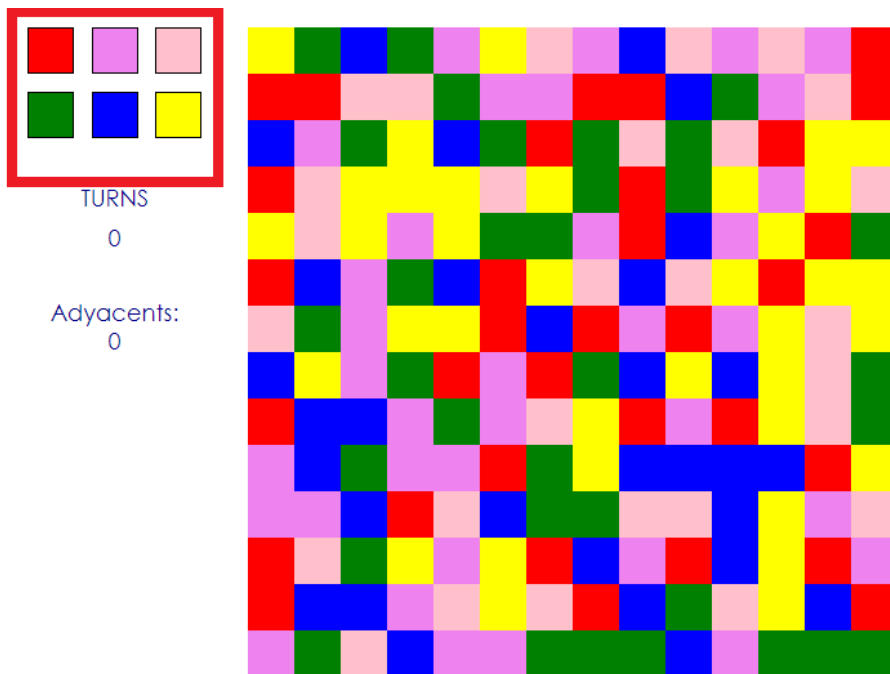
Manual de Usuario:

Al comenzar el juego tendremos una grilla sin resolver.



History:

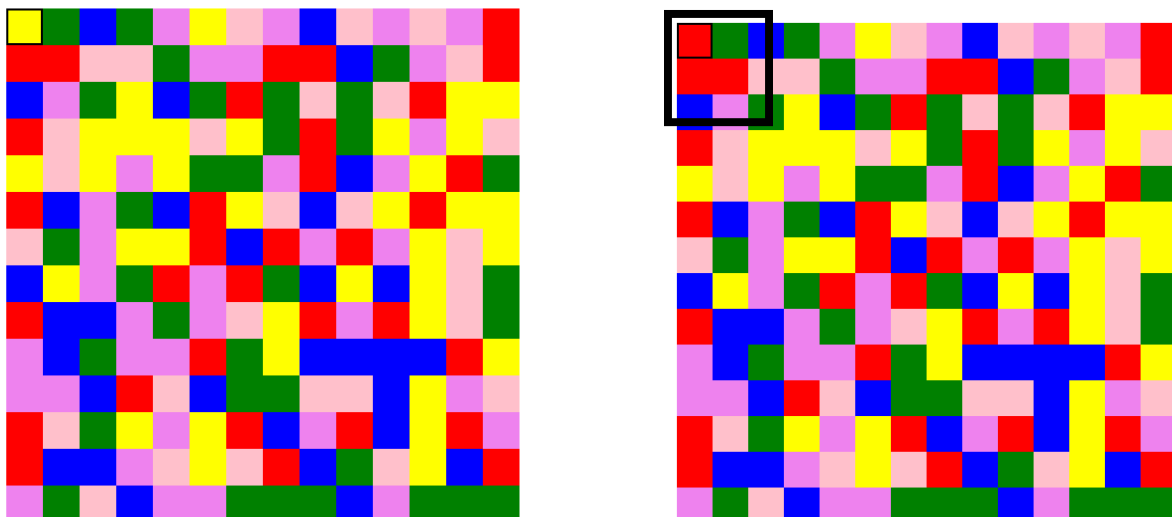
En el panel izquierdo se muestra 6 botones con colores distintos para seleccionar



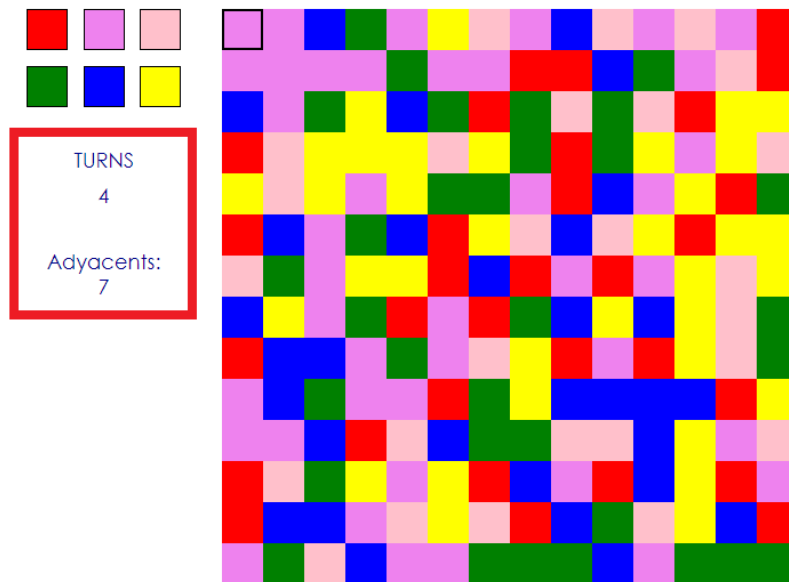
History:

El juego inicia al hacer clic en una celda, si esto no ocurre se iniciará con la celda superior izquierda.

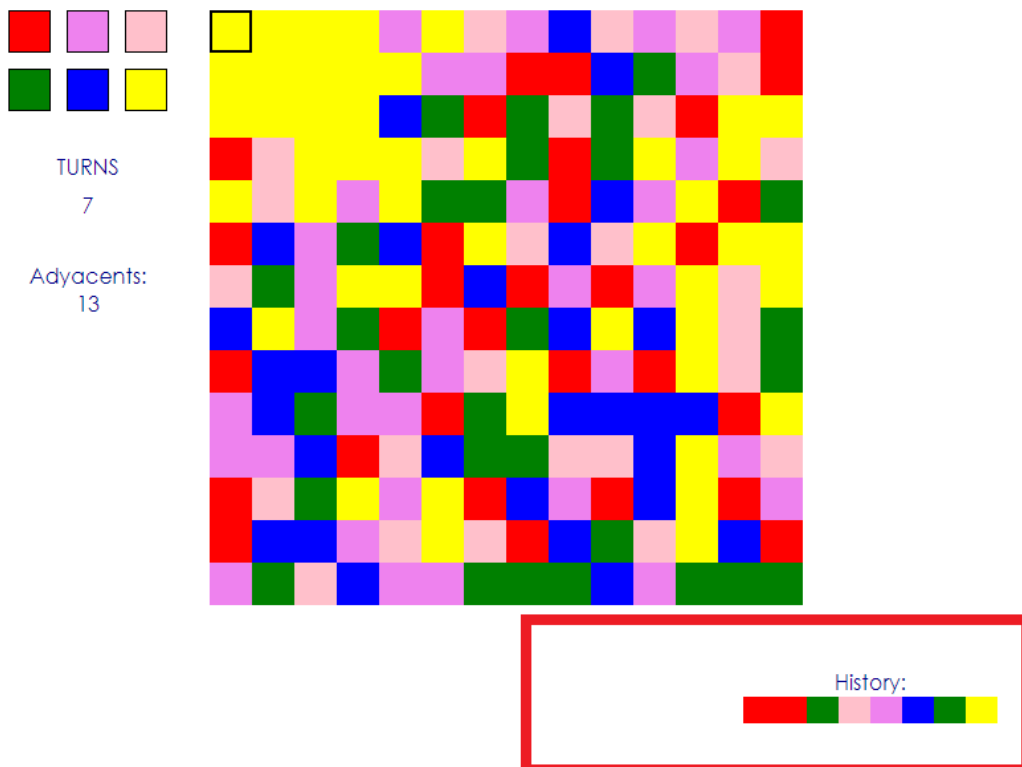
Al elegir una celda o utilizar la celda por defecto se podrá visualizar que sus adyacentes del mismo color se van reemplazando, al no haber más adyacentes para pintar se espera un nuevo clic.



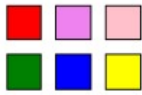
En el panel izquierdo tambien se podra visualizar en todo momento la cantidad de jugadas realizadas y la cantidad de adyacentes capturados



Tendremos un historial en la parte inferior para visualizar las secuencia de colores presionados

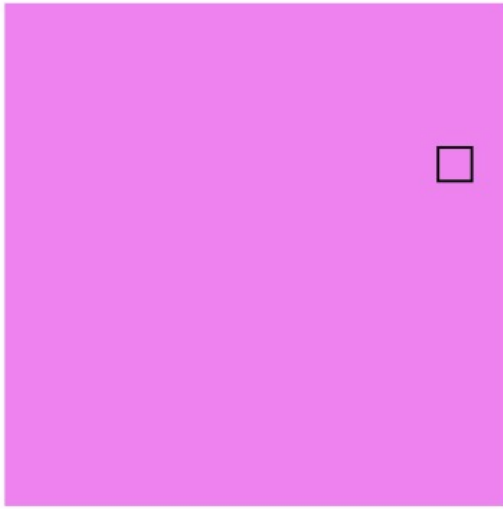


El juego finaliza cuando la grilla queda completa de un solo color.



TURNS
28

Adjacents:
196



History:

