



Proyecto 1: Flick Color

El juego

Consta de una grilla de 14 x 14, donde cada celda está pintada de uno de 6 posibles colores, y 6 botones, uno de cada color, como se muestra en la figura. Presionar un botón de color C causa que la celda superior izquierda 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. Notar que si $C = C'$ entonces no se produce cambio alguno, por lo tanto se asume que presionar el botón del mismo color que la celda superior izquierda no constituye una jugada válida.

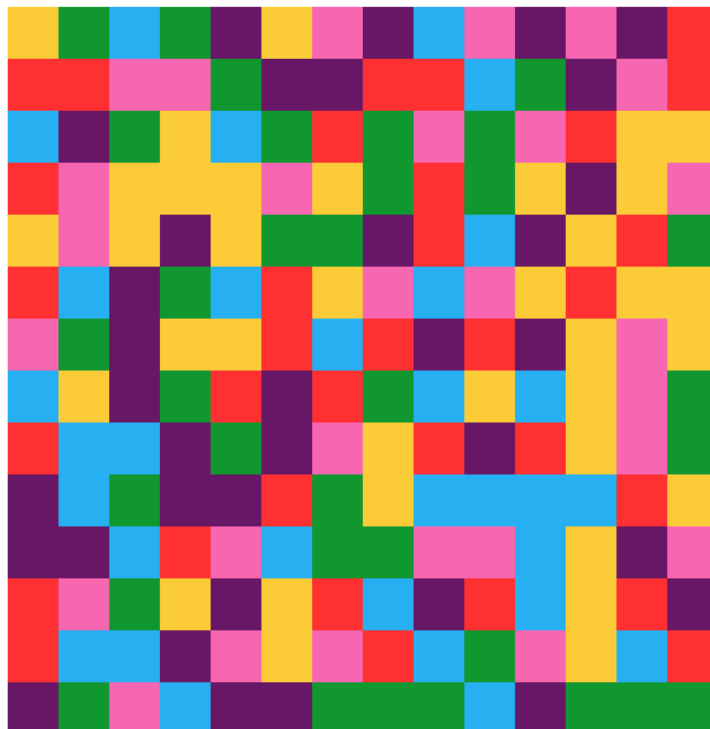
El **objetivo del juego** es lograr pintar todas las celdas de un mismo color, presionando los botones la menor cantidad de veces posible.

A continuación se muestra la secuencia de estados del juego que corresponden a presionar: rojo, rosa y amarillo.



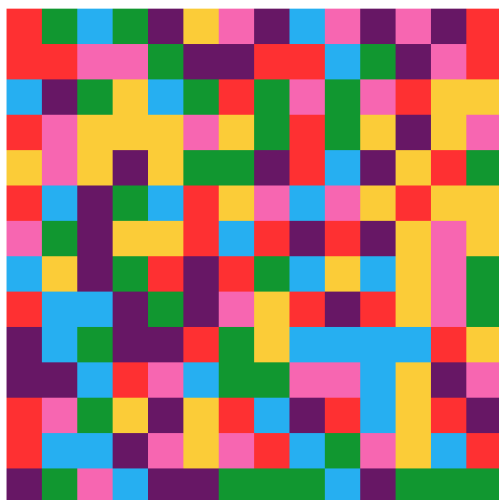
TURNOS

0

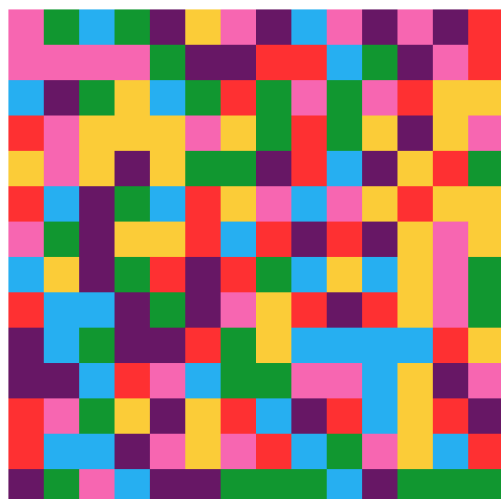




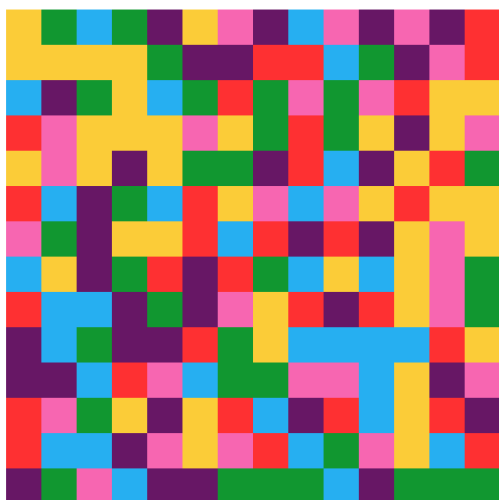
TURNOS
1



TURNOS
2



TURNOS
3



Requerimientos

Notación: usaremos adyacenteC^* 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 adyacenteC^* 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 adyacenteC^* a Y.

Funcionalidad

Se debe implementar una aplicación que permita a un usuario jugar *Flick Color*, con una interfaz del estilo de la mostrada en la captura de pantalla presentada anteriormente, además de satisfacer los siguientes requerimientos:

- Previo a comenzar el juego, permitir al usuario elegir cualquier celda de la grilla como celda *de origen*, esto es, la celda a partir de la cual se calculan las adyacenteC^* para determinar las celdas a ser pintadas como resultado de haber presionado un botón. Por defecto esta celda es la superior izquierda.
- Mostrar en todo momento la cantidad de celdas que se encuentran actualmente *capturadas*, esto es, que son adyacenteC^* a la celda de origen.
- Mostrar en cada instante del juego el historial de jugadas realizadas por el usuario (secuencia de colores presionados) desde el inicio de la partida.
- Detectar y manejar apropiadamente la terminación del juego.

Implementación

Debe extenderse la implementación molde (React + Prolog) provista en repositorio GitHub [tic-tact-toe](#), [branch flick](#), para cumplir con los requerimientos de funcionalidad mencionados anteriormente.

El archivo [init.pl](#) (módulo Prolog) en `pengines_server/apps/proylcc` permite especificar la configuración de la grilla del juego, que será mostrada por la interfaz. Es **importante** que no altere este esquema, y que **conservar la representación** de la grilla propuesta en el código molde, dado que asumiremos dicha representación para testear la implementación con diferentes grillas (casos de test) en la corrección.

Documentación

Se deberá realizar un informe que explique claramente la **implementación en Prolog** realizada.

Para la **implementación en React** simplemente pedimos que especifique los valores mantenidos en el estado del componente principal y su propósito, a alto nivel, además de las consultas realizadas a Prolog, y cómo se actualiza el estado, también a alto nivel, a partir de las respuestas obtenidas. Además, deberá escribirse una sección que explique brevemente los **pasos** requeridos **para interactuar con la interfaz**.

Se recomienda estructurar el informe de manera top-down, comenzando con una descripción a alto nivel de la implementación. Puede aprovechar el informe para destacar características positivas de la resolución, y documentar cualquier otra observación que considere pertinente.

Importante: en el desarrollo de software, la documentación de la implementación constituye un elemento fundamental. Es por esto que, para la evaluación del presente proyecto, se dará suma importancia a la calidad (claridad y completitud) del informe entregado. Aproveche las consultas para preguntar acerca del desarrollo del informe.

Comisiones y Entrega

1. Las comisiones pueden estar conformadas por hasta 2 integrantes, y deben ser registradas en la página de la materia. A cada comisión se le asignará un docente de la práctica, quien hará el seguimiento y corregirá el proyecto de la comisión.
2. La fecha límite de entrega del proyecto se encuentra publicada en la página de la materia. Los proyectos entregados fuera de término recibirán una penalización en su calificación, la cual será proporcional al retraso incurrido.
3. La entrega del proyecto consiste del envío por mail de la resolución del proyecto y versión electrónica del informe.
 - a. Enviar por mail directamente al docente asignado a la comisión, con copia al asistente (en caso de no ser el asignado). Mails:
 - Diego: dieorbe96+LCC@gmail.com
 - Nico: nicomleidi+LCC@gmail.com
 - Eliana: elikohon+LCC@gmail.com
 - Mauro (asistente): mgomezlucero+LCC@gmail.com
 - b. Asunto del mail: "Proyecto 1 LCC - Comisión <Ap.y Nom. Integrantes>"
 - c. Adjunto un.zip conteniendo las **carpetas public, src**, (implementación React) **y penguins-master/apps/proy1cc** (implementación Prolog) y un **pdf con el informe**. Alternativamente a enviar un adjunto puede compartir alguna carpeta en la nube (ejemplo: dropbox, google drive, etc.). Esta opción evita posibles 'rebotes' del mail por cuestiones de seguridad. Importante: en cualquier caso verifique que el mail no fue 'rebotado'.