



Proyecto 2: Resolución de Nonogramas

Requerimientos

Funcionalidad

Extender la funcionalidad del proyecto 1 para permitir al usuario obtener ayuda en el juego. Concretamente, deberá extender la interfaz gráfica con:

- Un botón "revelar celda" (similar al botón con la lamparita del Nonogram.com), tal que al clicar este botón y luego una celda vacía, se muestre el contenido de la celda de acuerdo a la solución. A diferencia del Nonogram.com, no debe haber límite a la cantidad de veces que puede usarse esta ayuda.
- Un botón "mostrar solución" (toggle), que permite mostrar / ocultar la solución (completa) al nonograma. Cuando se activa esta opción se muestra la solución, ya sea en el propio tablero en el que se está jugando (reemplazando la configuración actual, pero deshabilitando la posibilidad de interactuar con el tablero), o en un tablero aparte. Cuando se desactiva se vuelve a la situación normal de juego, en el estado en el que estaba antes de mostrar la solución.

Implementación

- La implementación de ambas funcionalidades requiere resolver previamente el nonograma.
- La resolución del nonograma debe realizarse íntegramente en Prolog.
- La resolución (consulta a Prolog) se realizará una única vez, al inicio (apenas se cargue la web-app), y la solución obtenida se almacenará en React para ser usada tantas veces como sea necesario para implementar los pedidos de ayuda del jugador.
- La interfaz no debe permitir comenzar a jugar (o debería al menos deshabilitar los botones de ayuda) hasta haber computado la solución al nonograma.

Documentación

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

Además, deberá incluirse una sección con casos de test significativos (capturas de pantalla).

El informe debe ser:

- **Claro:** información bien estructurada y presentada
- **Completo:** explicando cómo resolvieron **cada requerimiento funcional** (la parte de Prolog, y a nivel de estrategia, no a nivel de código), funcionalidades extra implementadas (si es que alguna), aspectos positivos de la resolución, desafíos que encontraron y cómo los enfrentaron, **casos de test** (capturas de pantalla).
- **Sintético y relevante:** no repetir información que está en el enunciado, como reglas del juego, no documentar funcionalidad de muy bajo nivel o auxiliar, que no hace al entendimiento de la estrategia principal.

Consejo: darle algunas pasadas (lectura y modificaciones) hasta conseguir todo esto.

Comisiones y Entrega

1. Las comisiones deben ser las conformadas para realizar el proyecto 1.
1. Los **errores y sugerencias** marcados por el docente asignado en la corrección **del proyecto 1 deben ser contemplados (corregidos)** en la entrega de este proyecto.
2. La fecha límite de entrega del presente 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 integrante de cátedra asignado a la comisión, con copia al asistente (en caso de no ser el asignado). Mails:
 - Nico Leidi: nicomleidi+lcc@gmail.com
 - Ivan Sandiumenge: iks8001+lcc@gmail.com
 - Matías Gandolfo: matiasgandolfo17+lcc@gmail.com
 - Laureano De Luca: laureanondeluca+lcc@gmail.com
 - Mauro Gómez (asistente): mgomezlucero+lcc@gmail.com
 - b. Asunto del mail: "Proyecto 1 LCC - Comisión <Ap.y Nom. Integrantes>"
 - c. Link a una carpeta en la nube (ejemplo: dropbox, google drive, etc.) conteniendo un .zip con:
 - **public** (carpeta)
 - **src** (carpeta - impl. React)
 - **package.json** (archivo - paquetes instalados)
 - **pengines-master/apps/proylcc** (carpeta - impl. Prolog)
 - **.pdf** con el informe.