



Proyecto 2: Estrategia 2248

Requerimientos

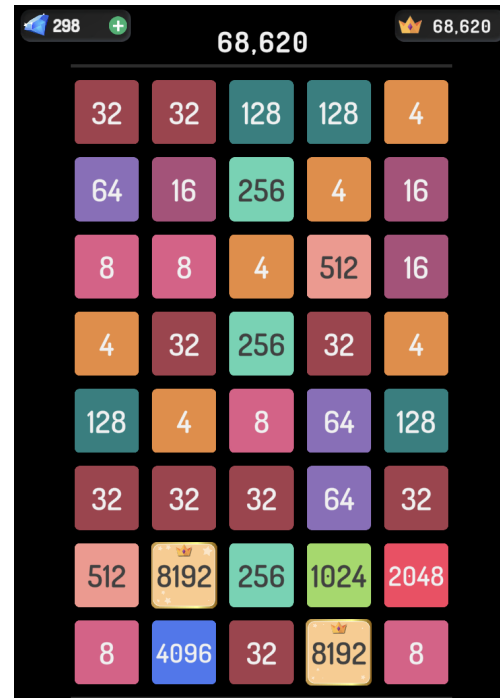
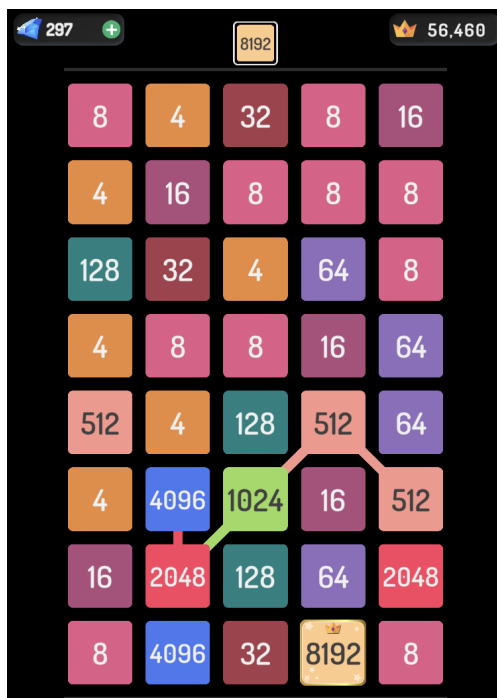
Funcionalidad

Extender la funcionalidad del proyecto 1 para permitir al usuario obtener ayuda en el juego.

Ayuda movida máxima

Agregar a la interfaz gráfica un botón que al ser clickeado calcule y muestre *el camino que consiga el mayor número* a partir de la configuración actual. Si hay más de uno que cumpla con esta condición, mostrar cualquiera de ellos.

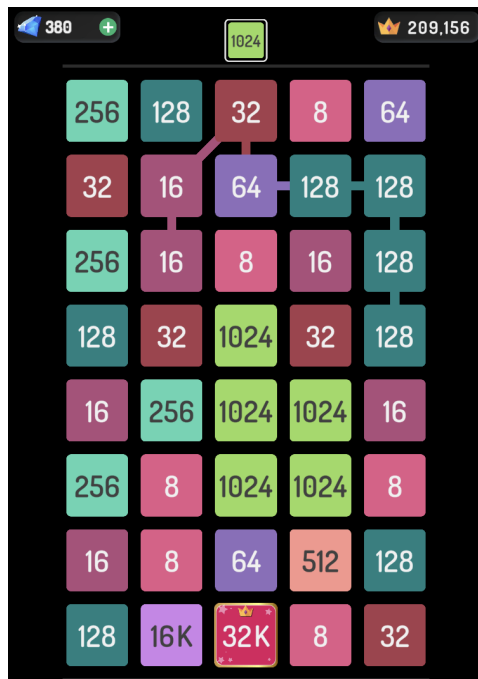
Ejemplo:



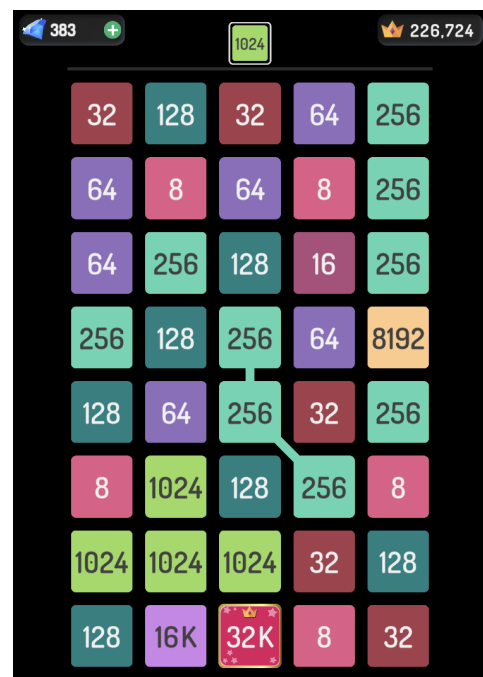
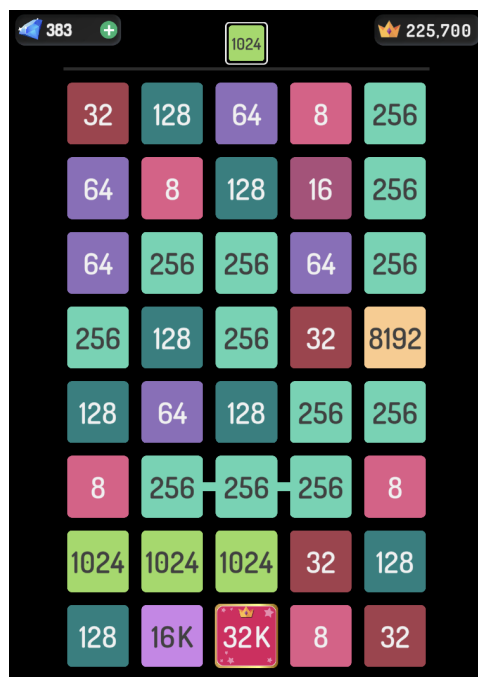
Ayuda máximos iguales adyacentes

Agregar a la interfaz gráfica un botón que al ser clickeado calcule y muestre *el camino que consiga generar el número más grande posible adyacente a otro igual* (preexistente). Si hay más de uno que cumpla con esta condición, mostrar cualquiera de ellos.

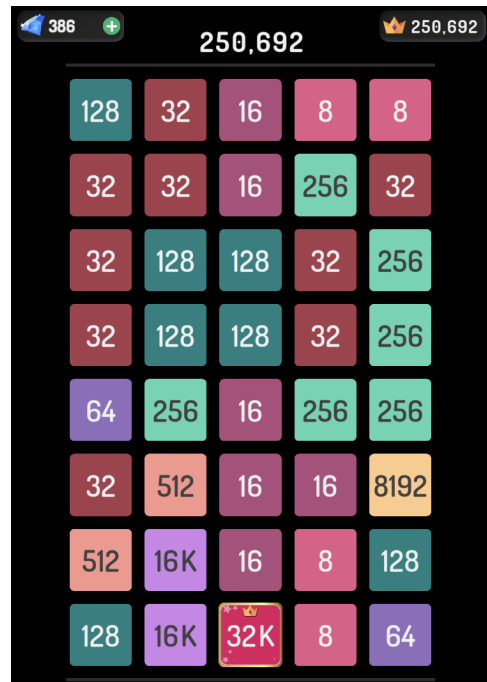
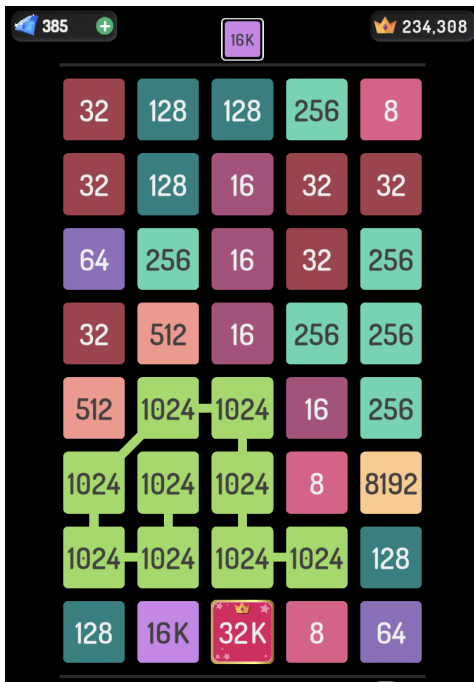
Ejemplo 1:



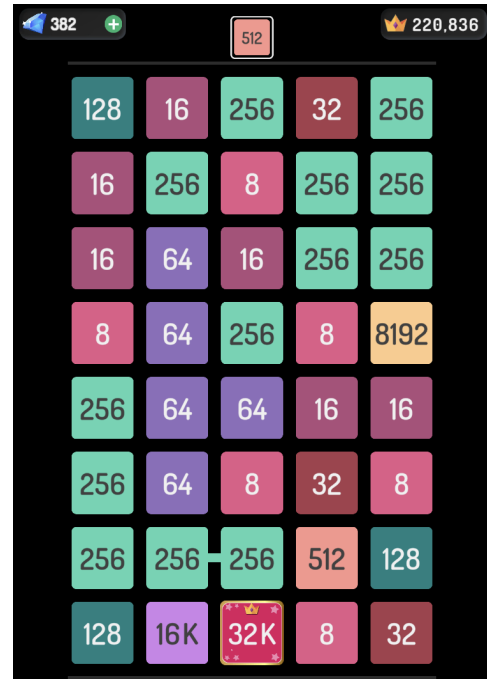
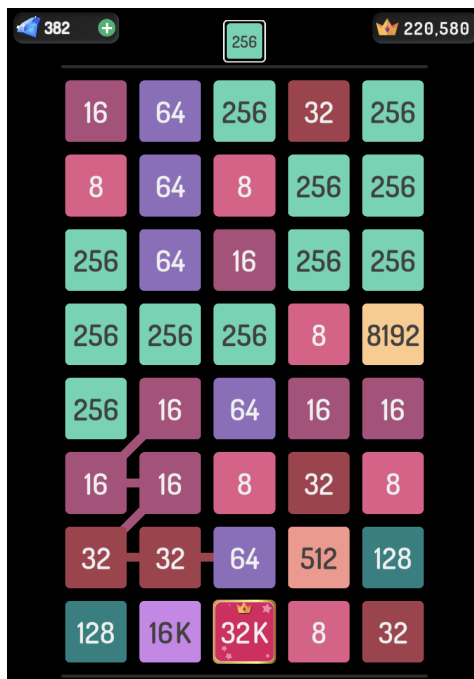
Ejemplo 2:



Ejemplo 3:



Ejemplo 4:



Implementación

Sugerencia: para mostrar el camino resultado de la ayuda se puede simplemente setear la variable de estado path con dicho camino, lo cual visualizará el camino tal y como si el jugador lo hubiera trazado, además de previsualizar el bloque resultado de la jugada. Luego basta con que el jugador haga click en el bloque final para aplicar la ayuda, o presione 'escape' para cancelarla.

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. 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.

4. La entrega del proyecto consiste del envío por mail de la resolución del proyecto y el informe.
 - a. Enviar por mail directamente al docente 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
 - Nico Komanski.: nico.komanski+LCC@gmail.com
 - Facundo Marín: facumarin1803+LCC@gmail.com
 - Mauro Gómez (asistente): mgomezlucero+LCC@gmail.com
 - b. Asunto del mail: "Proyecto 2 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.