



Proyecto 1: 2248

El juego

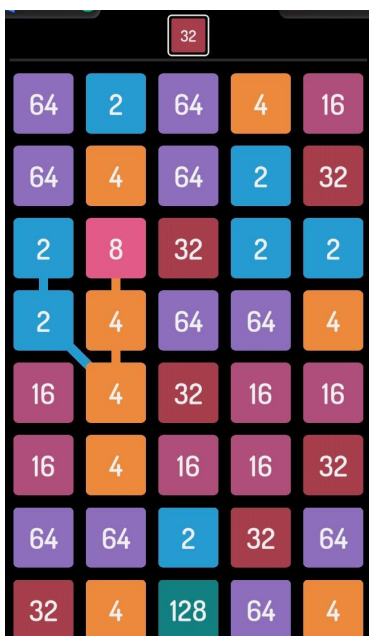
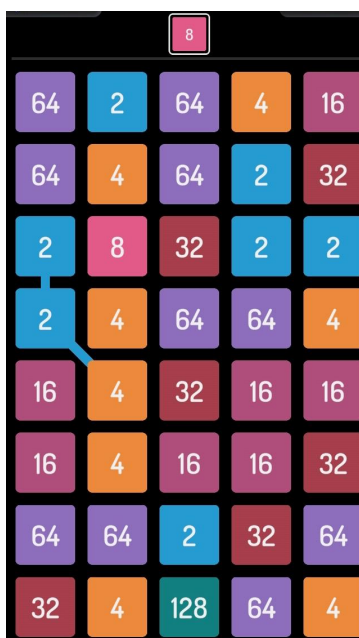
El 2248 es un juego de puzzle y estrategia cuyo **objetivo** es combinar bloques numerados para crear bloques de mayor valor y ganar puntos. Se encuentra disponible para [Android](#) y [iOS](#).

El juego se desarrolla sobre una grilla de números potencia de 2, como la siguiente.

152				
64	4	64	2	16
64	8	32	2	32
2	4	64	64	2
2	4	32	16	4
16	4	16	16	16
16	64	2	32	32
64	2	64	32	64
32	2	64	32	4

La **movida básica** del juego consiste en trazar un camino conectando 2 o más números, donde:

- Cada número del camino se conecta con uno siguiente que es adyacente horizontal, vertical o diagonal en la grilla.
- El camino arranca conectando 2 números iguales, y luego cada número del camino es igual al anterior o es la potencia de 2 siguiente a dicho número anterior, esto es, el camino va saltando de cada número a otro igual o a la potencia de 2 siguiente.



Luego de trazado el camino, el **efecto** es el siguiente:

- Todos los números involucrados desaparecen y en el lugar del último aparece un bloque cuyo valor V es la menor potencia de 2 mayor o igual que la sumatoria de los números del camino.
- Los espacios vacíos son ocupados por los números que están por encima, que caen por *gravedad* junto a los números por encima de ellos, y así siguiendo, y dónde números nuevos, generados al azar, ingresarán por la parte superior de la grilla para finalmente rellenar los espacios vacíos.
- El valor V se suma al puntaje actual.

380				
64	2	64	4	16
64	4	64	2	32
	32	32	2	2
		64	64	4
16		32	16	16
16	4	16	16	32
64	64	2	32	64
32	4	128	64	4

380				
64	32	64	4	16
32	8	64	2	32
64	2	32	2	2
64	4	64	64	4
16	32	32	16	16
16	4	16	16	32
64	64	2	32	64
32	4	128	64	4

Requerimientos

Funcionalidad

Se debe implementar una aplicación web que permita a un usuario jugar 2248, con una interfaz del estilo de las apps para [Android](#) y [iOS](#). Además de implementar las reglas y dinámica básica del juego (movida básica y su efecto, especificados anteriormente), debe cumplirse con los siguientes requerimientos:

- El efecto de combinar los bloques del camino trazado debe mostrarse progresivamente, en etapas, mínimamente las siguientes: 1) grilla donde se eliminaron los bloques del camino, y apareció el bloque resultante, 2) grilla donde se aplicó el efecto de gravedad, e ingreso de nuevos bloques aleatorios. Ver detalles de implementación en la sección siguiente. Si se desea mostrar una evolución más progresiva, pueden incluirse otras grillas intermedias.
- Al estilo de la app para dispositivos móviles, mientras el usuario va trazando un camino, ir mostrando el valor que se va "*generando*" al ir conectando bloques progresivamente, esto es, en cada instante mostrar el bloque numérico que se generaría si se "cerrara" en ese momento el camino que se está construyendo.
- *Booster Colapsar Iguales*, cuyo efecto consiste en colapsar todos los grupos de bloques adyacentes y de igual valor reemplazándolos por un bloque numérico calculado como la menor potencia de 2 mayor o igual a sumatoria de los bloques del grupo, y a ser ubicado en la posición más abajo y a la derecha del grupo. El booster no suma puntos. El booster debe presentarse como un botón, y podrá ser usado en cualquier momento del juego, y cuantas veces se quiera.

Implementación

Debe extenderse la implementación molde (React + Prolog) provista por la cátedra (ver [en Moodle](#)), para cumplir con los requerimientos de funcionalidad mencionados anteriormente.

El archivo [init.pl](#) (módulo Prolog) en `penguins_server/apps/proylcc` permite especificar la configuración de la grilla inicial 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.

El archivo proylcc.pl contiene el predicado `join/4`, recibiendo una grilla, el número de columnas, y un camino, y retornando una lista de grillas representando el efecto, en etapas, de la movida (ver primer requerimiento de la sección Funcionalidad). Si bien basta con devolver 2 grillas, tienen la libertad de devolver más grillas si se quiere reflejar una evolución más progresiva.

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 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 Komański.: nico.komanski+LCC@gmail.com
 - Facundo Marín: facumarin1803+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/proy1cc** (carpeta - impl. Prolog)
 - **.pdf** con el informe.