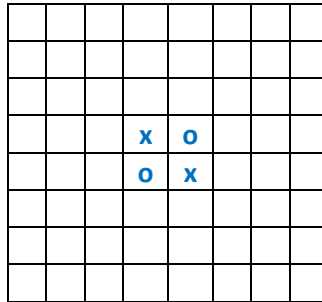


SALÓN DE JUEGOS



REVERSI



- **Movimientos**

- sobrepasar ficha contraria.
- diagonal, horizontal o verticalmente.

- **Comer fichas**

- toda ficha enemiga que se encuentre entre 2 o más fichas propias.

- **Gana**

- quien tenga más fichas de su color de las 64 posibles.
- cuando no haya movimientos posibles.

- **Extras**

- añadir páginas de otros juegos: damas, 4 en raya, etc
- presentar publicidad real y propia, que lleve a sitios conocidos
 - (tradedoubler, afiliación a tiendas online, etc)
- adjuntar enlaces a vídeos explicativos de reglas de los juegos.

- **Lógica**

1. recorrer tablero 8x8
2. cuando se encuentra una ficha propia, valorar las 8 posibles direcciones:
 - x-1
 - x+1
 - y-1
 - y+1
 - x+1, y+1
 - x-1, y-1
 - x+1, y-1
 - x-1, y+1
3. cuando encontramos una ficha enemiga, seguir en esa dirección
4. espacio libre -> marcar posible movimiento (class='free')

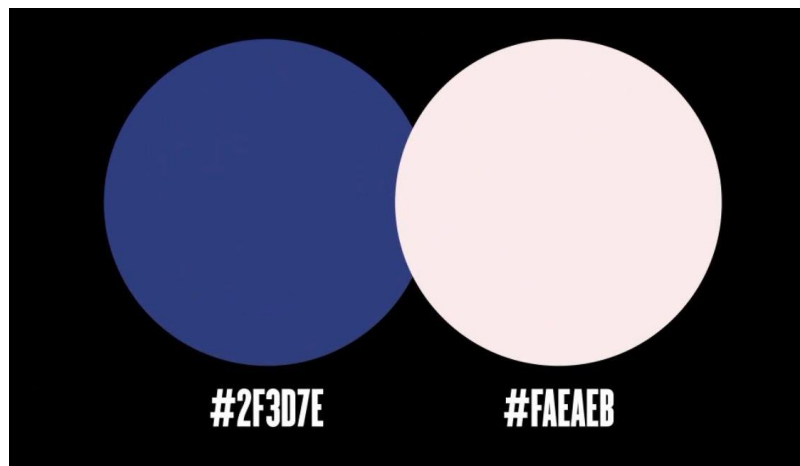
- **Mover ficha**

1. en esa posición, valorar las 8 posibles direcciones.
2. guardar fichas enemigas en json.
3. si encontramos ficha propia, comer fichas guardadas, si no no.

- **Ejemplo (movimiento inicial)**

1. posición = {x:4, y:4}
2. bucle hasta llegar a x=4, y=4
3. recorrer las 8 direcciones sobre (4,4)
4. encontramos fichas enemigas en (4,3) y (5,4)
5. posibles movimientos: (4,2) y (6,4)
6. añadimos la clase free sobre ambas posiciones
7. clicamos en (6,4)
8. movemos las 8 direcciones sobre (6,4)
9. encontramos ficha enemiga en (5,4) y la guardamos
10. como encontramos ficha propia en (4,4), comemos (5,4)

Paleta de colores



Fuente

POPPINS
POPPINS
POPPINS

A B C D E F G H I J K L M N Ñ O
P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9

Uso de Framework React

En mi proyecto, me encontré con desafíos relacionados con la funcionalidad de la aplicación, ya que la lógica subyacente no estaba bien estructurada. Para abordar este problema, decidí integrar React como añadido. Aquí hay algunas razones clave para considerar el uso de React en proyectos front-end:

Virtual DOM y Eficiencia: React utiliza un Virtual DOM (Documento de Objeto Modelo) para gestionar eficientemente los cambios en la interfaz de usuario. En lugar de actualizar directamente el DOM cada vez que cambia el estado, React realiza comparaciones en un DOM virtual, optimizando así la manipulación del DOM real y mejorando el rendimiento de la aplicación.

Componentización: React fomenta la creación de aplicaciones modulares mediante el uso de componentes. Los componentes son bloques de construcción reutilizables que pueden encapsular lógica y presentación. Esto simplifica el desarrollo, facilita el mantenimiento y mejora la legibilidad del código.

Declaratividad: React utiliza un enfoque declarativo para la creación de interfaces de usuario. En lugar de describir cómo manipular el DOM para lograr un estado deseado, los desarrolladores declaran cómo debería lucir la interfaz en diferentes estados. Esto hace que el código sea más predecible, fácil de entender y mantener.

React Native: React también se puede utilizar para el desarrollo de aplicaciones móviles a través de React Native. Compartir código entre aplicaciones web y móviles puede ser más sencillo, ya que React Native utiliza una sintaxis similar a React para construir aplicaciones nativas.

Gran Ecosistema y Comunidad: React forma parte de un ecosistema robusto con una amplia gama de bibliotecas y herramientas complementarias, como Redux para la gestión del estado. Además, cuenta con una comunidad activa que proporciona soporte, tutoriales y recursos valiosos.

Flexibilidad: React es flexible y se puede integrar fácilmente con otras bibliotecas o frameworks. Además, se puede utilizar junto con diversas herramientas y tecnologías para el desarrollo de aplicaciones complejas.

React Hooks: La introducción de Hooks en React permite el uso de estado y otras características de React en componentes de función, simplificando la lógica del componente y eliminando la necesidad de clases en muchos casos.

En resumen, React se ha vuelto popular en proyectos front-end debido a su eficiencia, modularidad, declaratividad y el sólido ecosistema que lo respalda. Estas características hacen que sea una elección sólida para el desarrollo de interfaces de usuario interactivas y escalables.

Funciones estéticas, comunicativas y complementarias.

En el **diseño visual** de mi plataforma de juegos, me he enfocado en crear una experiencia atractiva y moderna para nuestros usuarios. He elegido una paleta de colores vivos y modernos, destacando especialmente el morado, para transmitir energía, creatividad y un toque de misterio que añade intriga a la experiencia de juego. Creo que estos colores no solo son llamativos, sino que también crean una atmósfera vibrante que invita a los usuarios a sumergirse en nuestro mundo de entretenimiento.

Además, he integrado **imágenes atractivas** que complementan la temática de cada juego, brindando a los usuarios una vista previa visualmente impactante de lo que pueden esperar al participar. Pienso que las imágenes juegan un papel crucial para despertar la curiosidad y el interés del usuario, y me he esforzado por seleccionar gráficos que no solo sean estéticamente agradables, sino que también **sugieran la diversión** y la emoción que encontrarán en los juegos.

Para mejorar la interactividad, he **incorporado efectos de aumento con transiciones suaves**. Estos efectos no solo añaden un toque dinámico al diseño, sino que también destacan elementos importantes, guiando sutilmente la atención del usuario y mejorando la usabilidad de la plataforma.

En cuanto a la **publicidad**, comparto la visión sobre la importancia de evitar la invasión y respetar la experiencia del usuario. Por ello, he decidido limitar la publicidad a enlaces que redirigen a páginas externas, permitiendo a los usuarios explorar contenido adicional de manera opcional. Creo que esta estrategia respeta la privacidad y preferencias del usuario, ofreciendo un espacio enfocado en la experiencia de juego sin distracciones innecesarias. Además, este proyecto ya forma parte de mi portfolio profesional, por lo que la impresión y el diseño es muy importante.

En conjunto, **mi objetivo es crear un entorno visualmente atractivo**, amigable y centrado en el usuario, donde los colores, imágenes y efectos trabajen juntos para generar un ambiente emocionante que inspire a nuestros usuarios a disfrutar de nuestros juegos de manera única y memorable.

Deployment

VERCEL -> <https://game-room-5uul.vercel.app/>

LOCAL -> npm run dev (Comando para consola)

Vercel es una plataforma de desarrollo web que se ha vuelto extremadamente popular entre los desarrolladores gracias a su enfoque en la creación e implementación sencilla de aplicaciones web. Fundada en 2015 por Guillermo Rauch, Vercel se destaca por su enfoque en la experiencia del desarrollador y su capacidad para facilitar el despliegue rápido y eficiente de proyectos.

Una de las características clave de Vercel es su **integración con Git**, lo que permite a los desarrolladores implementar fácilmente sus proyectos directamente desde sus repositorios de código. Esto simplifica el proceso de implementación y reduce significativamente el tiempo necesario para llevar una aplicación desde el desarrollo hasta la producción.

Además, Vercel **ofrece soporte para una amplia variedad de tecnologías** y marcos de trabajo, lo que la hace versátil y adaptable a diferentes tipos de proyectos. Desde aplicaciones estáticas hasta aplicaciones de página única (SPA) y aplicaciones de servidor completo, Vercel puede manejar una amplia gama de casos de uso.

La plataforma también se destaca por sus capacidades de pre-renderización y su enfoque en la entrega de contenido estático de manera eficiente. Esto contribuye a tiempos de **carga rápidos y una experiencia de usuario mejorada**.

Otra ventaja significativa de Vercel es su **red global de entrega de contenido (CDN)**, que ayuda a mejorar la velocidad de carga de las aplicaciones al distribuir el contenido a través de servidores ubicados estratégicamente en todo el mundo.

En resumen, Vercel se ha ganado la confianza de la comunidad de desarrollo por su enfoque en la simplicidad, la integración con Git, la versatilidad tecnológica y la eficiencia en la entrega de contenido, lo que la convierte en una opción popular para desarrolladores que buscan una solución robusta para el desarrollo y despliegue de aplicaciones web.