

Paso 1: Seleccioná un color de la paleta y mostralo en el indicador de color

Primero, vamos a hacer que nuestro programa permita al usuario seleccionar un color de la paleta. Para eso, necesitamos lograr que, al hacer clic en algún color, el `indicador-de-color` cambie y refleje la selección.

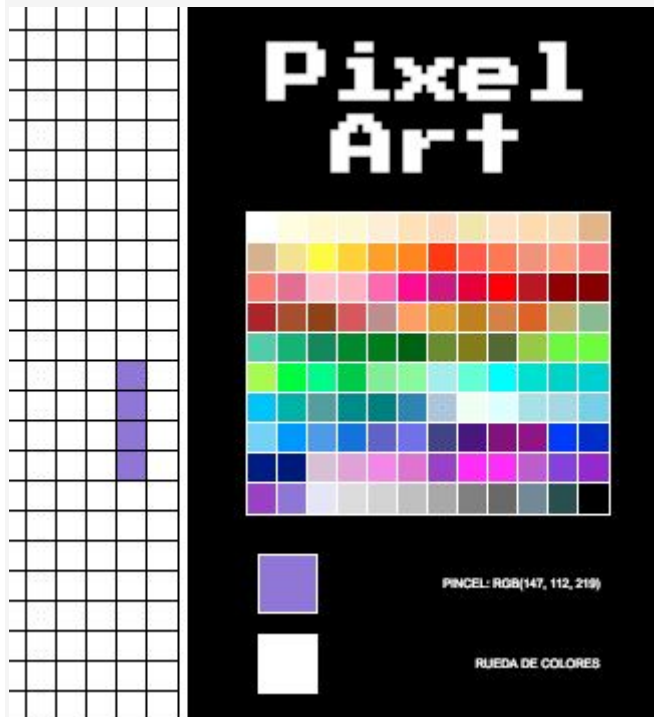
Por ejemplo, si clickeamos en el color violeta, el `indicador-de-color` deberá cambiar a violeta.



Pista: Pensá qué evento vas a necesitar usar.

Paso 2: Pintá un pixel de la grilla

Un vez que tenemos un color seleccionado en el `indicador-de-color` se deberá programar la funcionalidad para que el usuario pueda pintar un pixel al hacer clic en un cuadrado de la grilla.



Pista: Pensá qué atributo de css deberías cambiar en el pixel para simular el pintado.

Paso 3: Terminá la funcionalidad de Rueda de Colores

En el proyecto ya viene implementada una rueda para elegir un color personalizado. El `id` de este elemento es `color-personalizado`. Cuando cliqueemos este elemento se abrirá una ventana donde podremos elegir un color.

Ahora bien, una vez que elegimos el color, este tendrá que verse reflejado en el `indicador-de-color` de la Rueda de Colores. Esta funcionalidad está hecha por la mitad con el siguiente código:

```

colorPersonalizado.addEventListener('change',
(function() {
    // Se guarda el color de la rueda en colorActual
    colorActual = colorPersonalizado.value;
    // Completar para que cambie el indicador-de-color al colorActual

}))
);

```

Completá la función anterior para hacer que el indicador de color de la rueda de colores tenga el color seleccionado.



Buenas prácticas: Pensá en reutilizar la funcionalidad para cambiar el `indicador-de-color` que ya hiciste

Paso 4: Detectá si el Mouse está Apretado o no

Para poder pintar en movimiento, vamos a necesitar una variable que nos diga si el mouse está o no apretado. El valor de esta variable deberá cambiar cada vez que se apriete el mouse y cada vez que se suelte. Tené en cuenta que vas a necesitar más de un evento para detectar cuando se aprieta el mouse y cuando se suelta.

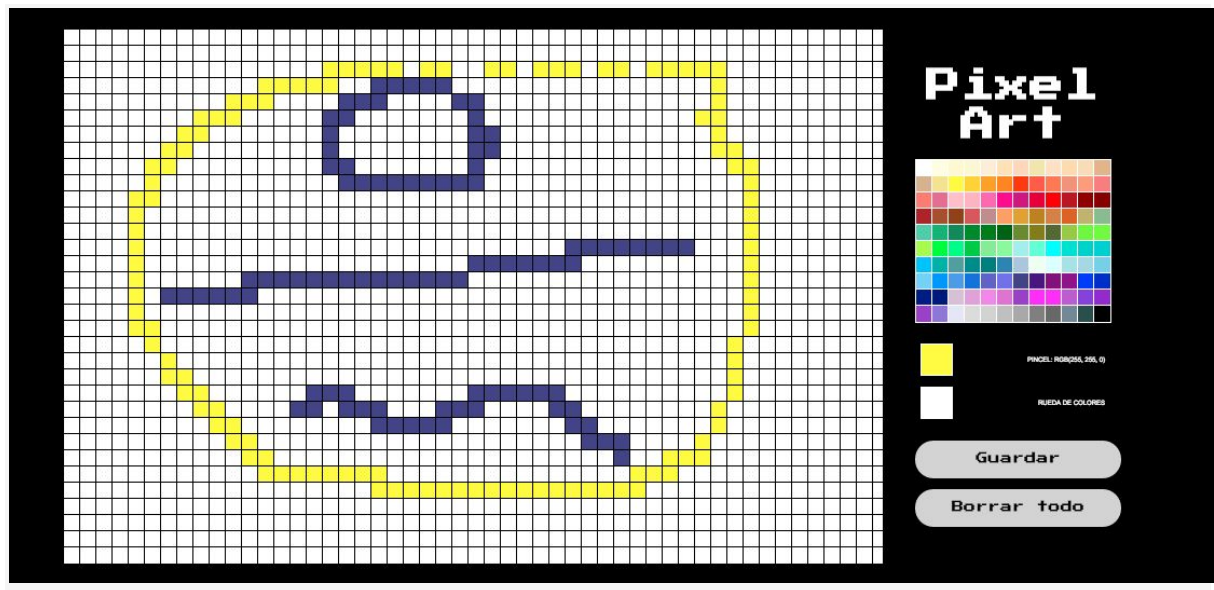
Esta acción no va a tener representación visual por ahora, de eso nos vamos a encargar en el paso siguiente.

Nota: para probar esta funcionalidad, recordá que podés usar `console.log()` para mostrar la variable.

Paso 5: Implementá la acción de Pintar en Movimiento

En este paso queremos programar la funcionalidad para poder pintar con el mouse muchos pixeles a la vez con sólo mantenerlo apretado y moviéndolo por la grilla. Para poder realizar trazos como un pincel, vas a utilizar algo que ya hicimos antes: pintar un pixel. Sólo que esta vez se necesita captar el movimiento del mouse. Si está apretado pintamos y si no, no.

De esta forma el usuario podrá pintar de forma continuada, como si tuviera un pincel y un lienzo, pudiendo hacer obras de arte como esta:



Pista: Acordate de usar la variable que creaste en el Paso 4 para detectar si está apretado o no el mouse.