PROYECTO LEAP MOTION

## 1-Gestos implementados.

La interfaz creada para esta aplicación usa varios gestos para su uso. Se han implementado los gestos más intuitivos a nuestro parecer, buscando siempre la simplicidad y facilidad de uso de la interfaz. A continuación vamos a detallar los gestos, y su funcionalidad.

1. Clicar con el dedo índice.

Este gesto se realiza teniendo la mano extendida con los dedos extendidos y clicando con el dedo índice. Este gesto se usa para clicar en los diferentes elementos de la pantalla.

1. Disparo.

El gesto de disparo se basa en la imitación de una pistola, en la que el dedo índice es el cañón y el dedo gordo el gatillo. Se usa para clicar los diferentes elementos de la pantalla, es más efectivo que el gesto clicar dado que este genera continuamente eventos de click, por lo que evita las imprecisiones que pueda tener el leap o la interfaz.



1. Encoger los dedos.

Encoger los dedos es un gesto que se usa para ampliar la imagen o reducirla, solo funciona cuando el botón “Ampliar cuadro esta activo”. La mano derecha se usa para ampliar y la mano izquierda para reducir.

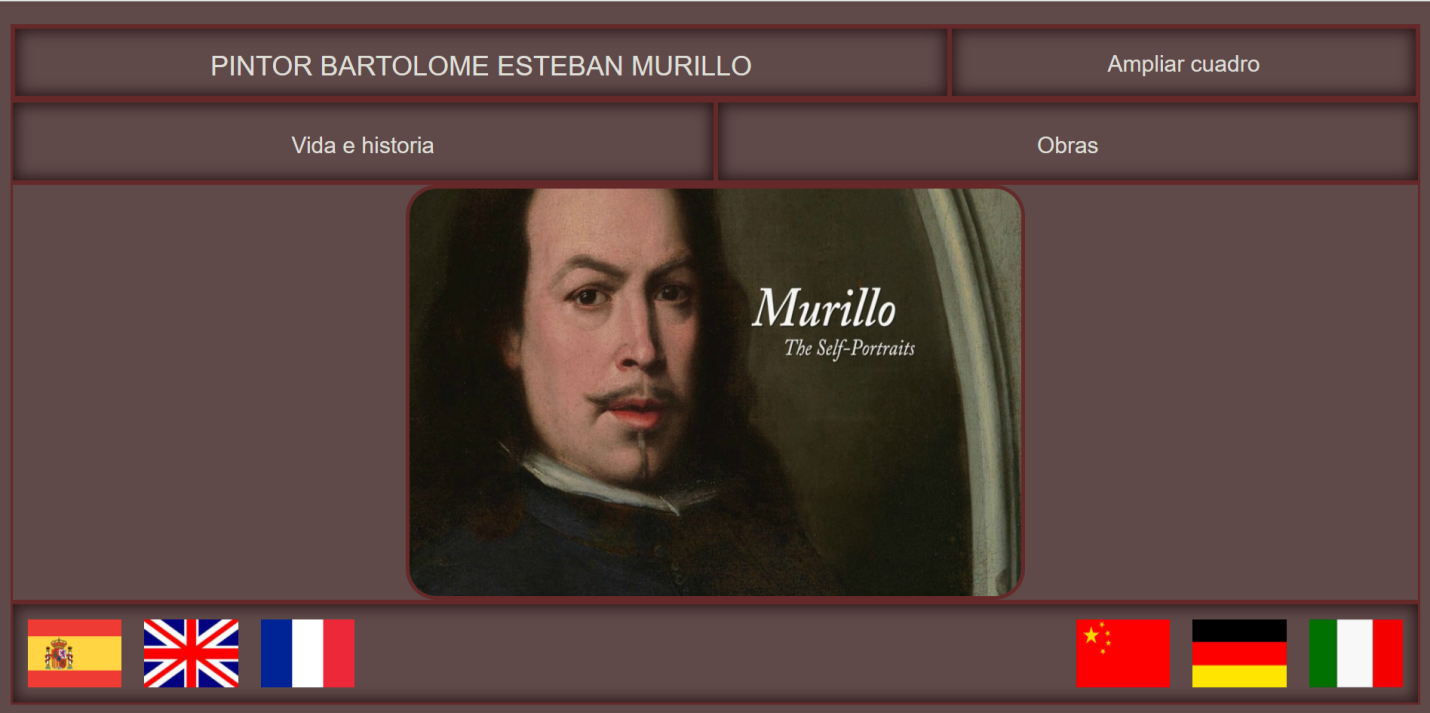
1. Deslizar la mano.

La mano se puede deslizar de derecha a izquierda o de izquierda a derecha para pasar a la siguiente imagen del expositor de imágenes. También puede hacerse el gesto de arriba abajo o de abajo a arriba para hacer scroll en la pantalla.

Este gesto debe hacerse claramente, para que la interfaz lo considere valido, marcando con la mano un momento de arrancada, cierto desplazamiento, y una parada.

## 2-Funcionalidad de la interfaz.

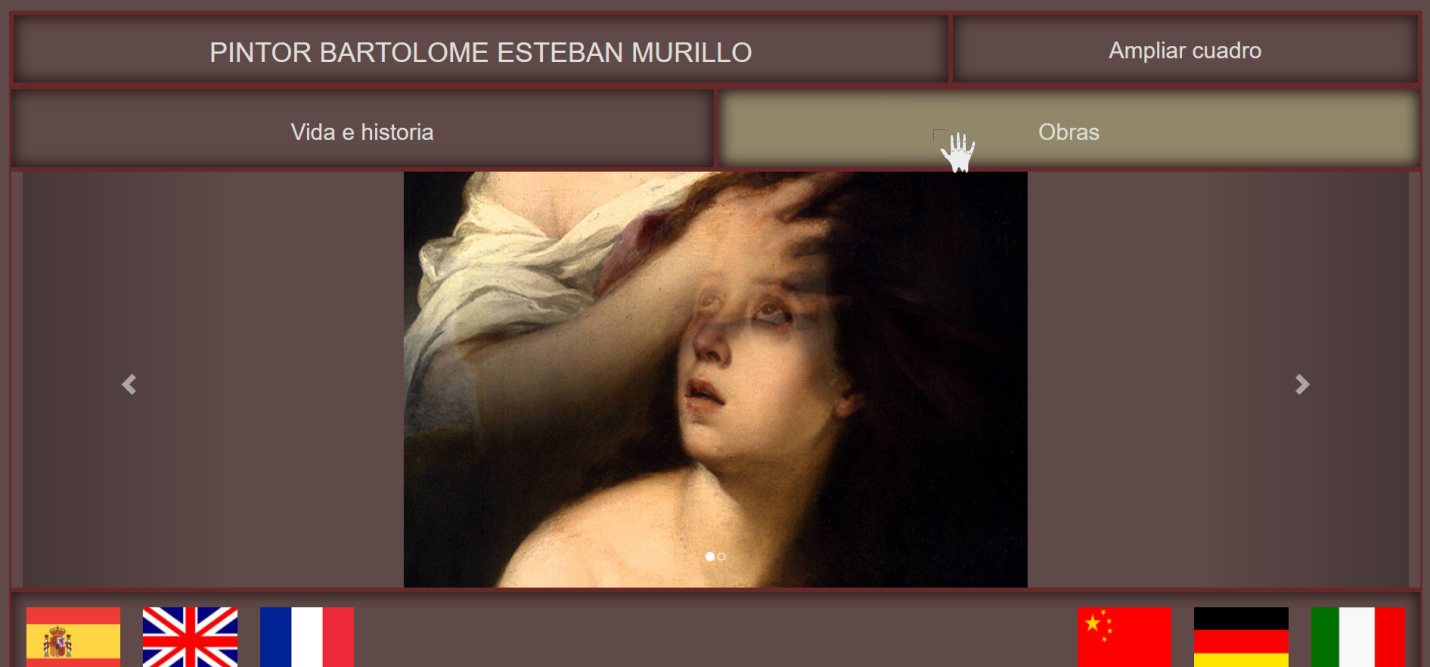
La interfaz consiste en una cabecera en la cual encontramos tres botones, “vida e historia”, “obras” y “Ampliar elemento. Una parte centrar que se una de mostrador de información. Y finalmente un pie de pagina que usamos para mostrar los diferentes idiomas que puede tener la interfaz y otra información.

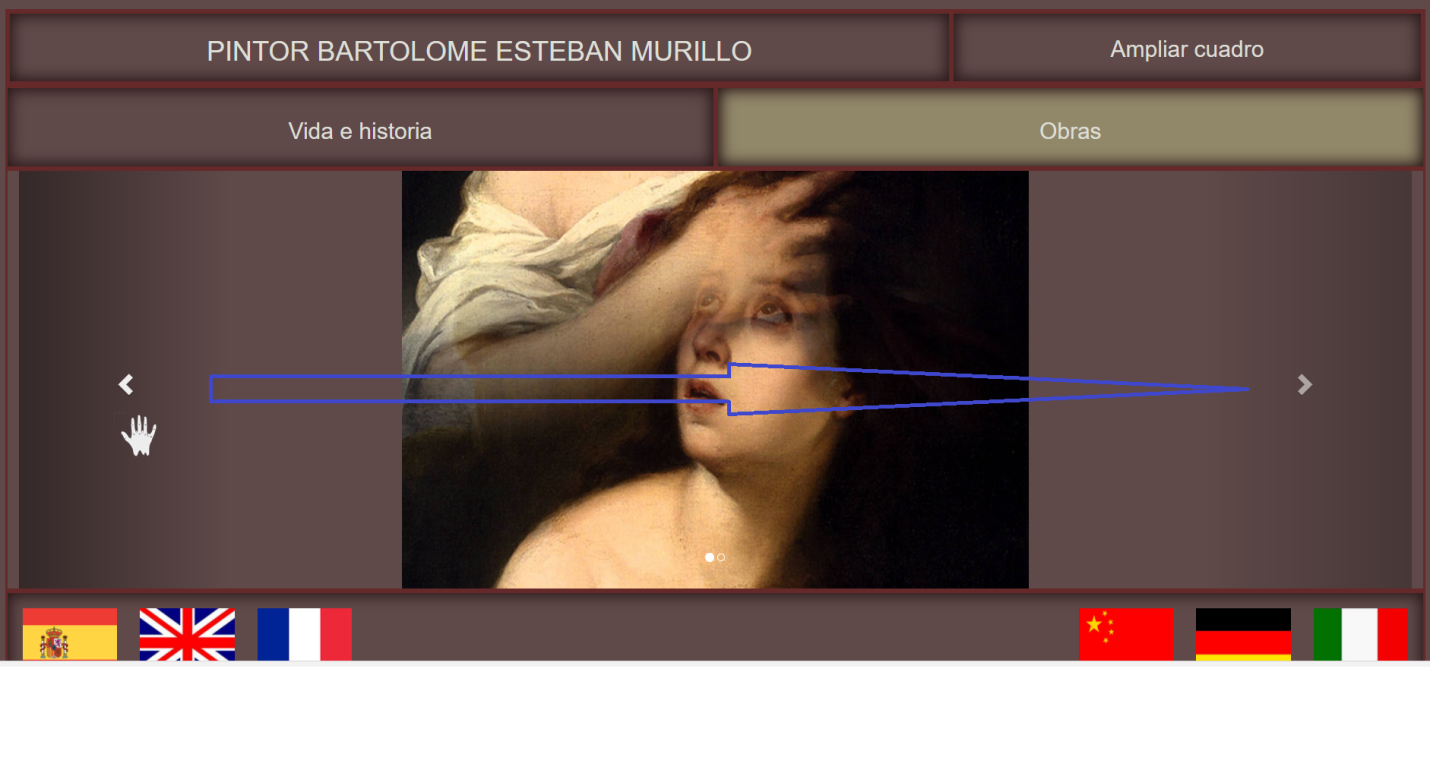


Los botones del menú se activa mendiante el evento click, que es generado por los gestos clicar y disparo. El botón de “vida e historia” muestra una breve descripción de la biografía del autor.



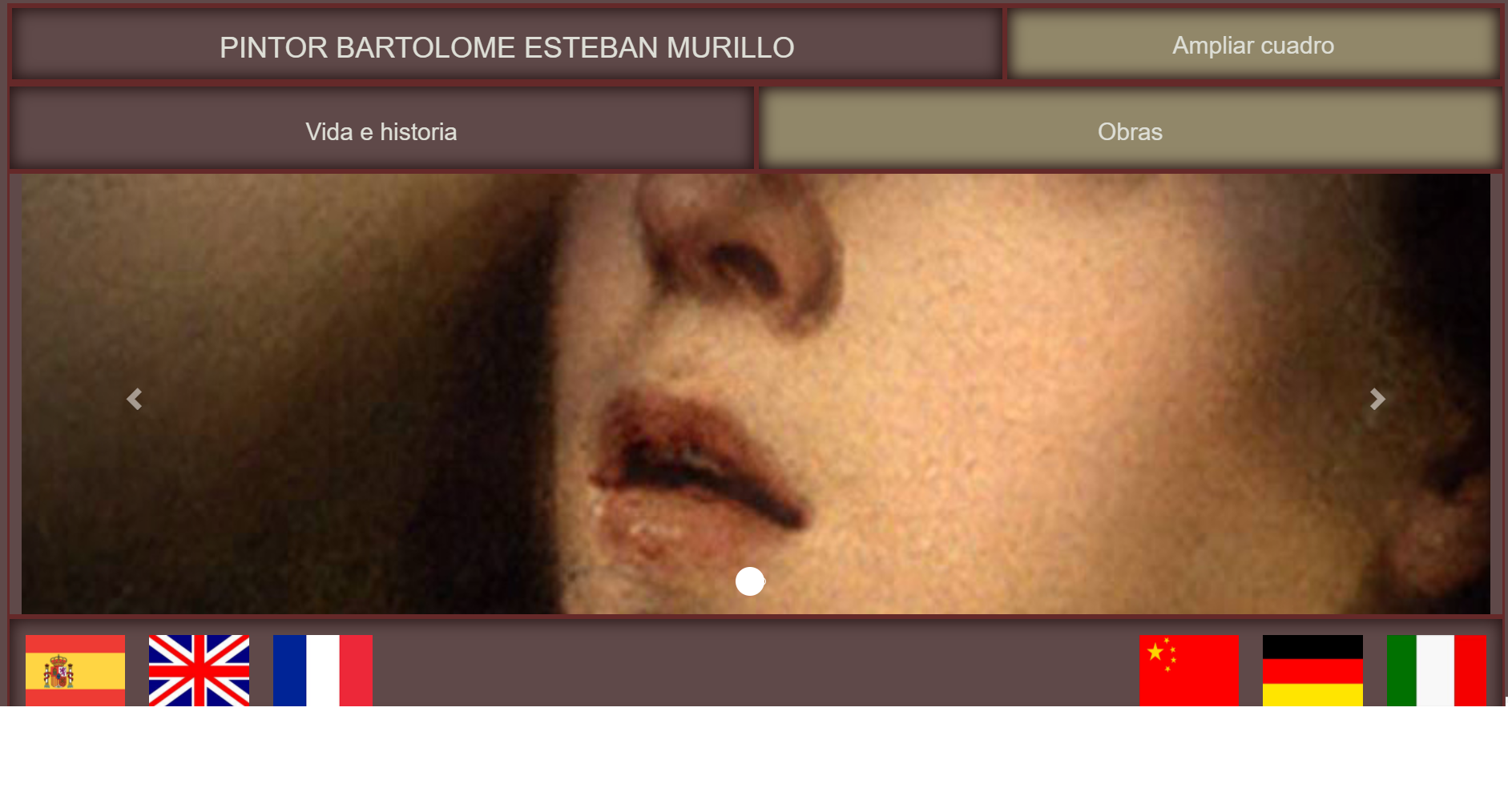
El boton obras activa un mostrador de obras del autor. Despues de activar este boton se puede pasar las imágenes con el gesto deslizar la mano de izquierda derecha o derecha izquierda.



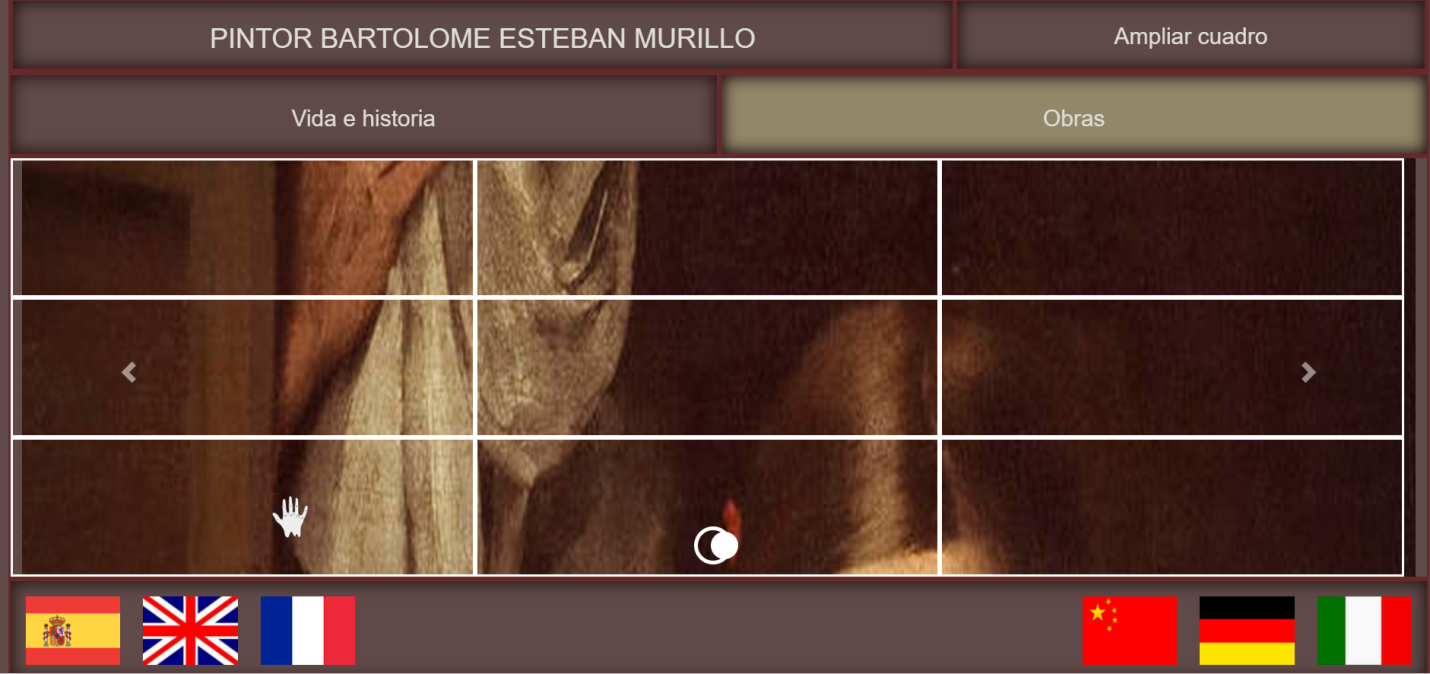


El botón ampliar cuadro es clicable, la mejor forma de clicarlo es mediante el gesto clicar. Una vez activo, habilita el zoom en la imagen que aparece en el mostrador. Mediante el gesto encoger los dedos con la mano derecha se amplía esta imagen, y mediante este gesto con la mano izquierda se reduce el tamaño. Este zoom está limitado para que no se reduzca o amplié la imagen exageradamente.

Cuando el botón “ampliar imagen esta activo” todos los demás gestos de desactivas, permitiendo solo ampliar o reducir la imagen, y desactivar el botón “ampliar imagen” mediante el gesto clicar o disparo.



Una vez ampliada la imagen tenemos la opción de movernos a través de ella. Existe un grid imaginario sobre la imagen que nos permite elegir la dirección a la que queremos movernos clicando en esa parte de la imagen. En la captura de pantalla de abajo hemos representado ese grid. Este movimiento está limitado a los border de la imagen.



## 3-Descripción de código e implementación.

El programa se ha implementado usando el lenguaje JavaScript. Este usa la SDK proporcionada en la web <https://developer.leapmotion.com/get-started/> . El código no lo hemos desarrollado desde cero, si no que hemos partido de la base de un proyecto de github “git clone [git@github.com:leapmotion-examples/javascript](mailto:git@github.com:leapmotion-examples/javascript)” el cual no ha ayudado ha empezar, dado que implementa varios gestos y proporciona una lógica de implementación que ha sido muy útil para nosotros. El proyecto final no tiene mucho que ver con el de partida, dado que hemos añadido muchísima funcionalidad.

El programa consta de la interfaz html mostrada anteriormente y un motor en javascript. Este consta de 4 diferentes documentos.

1-Vector.js. Implementa algunas funciones para tratar las posiciones que nos proporciona la SDK de leap.

2-Code\_pinch.js. En este documento tratamos los datos que nos proporciona el leap, calculando la position, y detectando los posibles gestos. Contiene parámetros de configuración, como los gestos que estamos detectando. También interpreta los datos enviados por el leap, calculando la posición que representara en la pantalla así como la sensibilidad de estos en la interfaz. Una vez recogidos los gestos y posiciones los envía al InputEngine.js.

|  |
| --- |
| const PALM\_TRACKING = true;  const HAND\_DIRECTION = false;  const TRIGGER\_CLICK = true; // gesture gun trigger  const PINCH\_CLICK = false; // pinch with the index and thumb fingers  const PINCH\_DOUBLE = false;  const ALT\_PINCH = false;  const DOUBLE\_TAP = false;  const KEY\_CLICK = true; // tap one time  const SWIPE = true; // swipe in any direction  const SHRINKFINGERS = true; // gesture of shrink the fingers  var LEAP\_MIN = { 'x':-15.0, 'y':15.0, 'z':-20.0 };  var LEAP\_MAX = { 'x': 15.0, 'y':26.0, 'z': 20.0 };  const PINCH\_MIN = 0.8;  const SWIPE\_SENSIBILITY = 1000;  const MAX\_HANDS = 2;  const TRIGGER\_BOUND = 0.8;  const ALT\_PINCH\_BOUND = 27.0; |

3-imputEngine.js. Esta parte del código recibe los gestos y posiciones detectados en code\_pinch, los interpreta creando eventos que son enviados a UIEngine.js

4-UIEngine.js. Esta es la vista de la aplicación, trabaja sobre la interfaz, recibe los eventos de inputEngine.js y realiza las acciones necesarias modificando lo que sea necesario de la interfaz.