**Lien :** [**https://vu.fr/JDTBW**](https://vu.fr/JDTBW)

[**https://vu.fr/xwSZF**](https://vu.fr/xwSZF) lien 1 er semestre transmission info

**Capter plusieurs données afin d’avoir l’état de la personne**

**Si c’est en fin de journée moins d’énergie**

**Si y a des répétitions**

**Démonstrateur : processing**

🡪Poids

🡪 Serrage de main : Jauge de contrainte

On relève la pression

Plus :

🡪 Cardiaque

🡪Température

On récupère données ensuite affichage sur le logiciel.

**Démonstrateur IA : page WEB**

Plateforme : Teachable machine

🡪Squelette : Activité physique

🡪Imagerie : Activité cognitive

Idées :

-Les Cubes.

-Le tangram

-Dessin par recopie

-Écrire etc

Figure correcte ou non.

**Interface**

**Frameworks**

**🡪Appli mobile**

**🡪page web**

**ENTRAINEMENT JEU**

Codesandbox

**🡪App.js :**

Le code dans App.js crée un composant**.** Dans React, un composant est un bout de code réutilisable qui représente une partie de l’interface utilisateur (UI, pour User Interface).

export default function Square() {

return <button className="square">X</button>;

}

**«**Square » : fonction

**«**export » : rend cette fonction accessible à l’extérieur de ce fichier

**«**default » : indique aux autres fichiers utilisant votre code qu’il s’agit là de la fonction principale de votre fichier.

« return » : indique que tout ce qui le suit est renvoyé comme valeur à l’appelant de la fonction.

« <button> » : élément JSX

**«**className="square" » : propriété du bouton qui indique à CSS comment styler le bouton

« X » : texte affiché à l’intérieur du bouton

« </button> » : ferme l’élément JSX en indiquant que tout ce qui suit ne devrait pas figurer dans le bouton.

« <div className="board-row"> » : permet separer les carré et ne pas les avoir sur la meme ligne. ( quand on en mets plusieurs )

« Board » : Votre composant nommé Square n’est plus vraiment un carré tout seul. Corrigeons ça en changeant son nom pour Board.

**Rendre le composant interactif :**

Faisons en sorte que le composant Square se remplisse d’un X lorsqu’on clique dessus.

Déclarez une fonction appelée handleClick au sein du composant Square.

Ensuite, ajoutez la prop onClick à l’élément JSX de bouton renvoyé par Square

function Square({ value }) {

function handleClick() {

console.log('cliqué !');

}

return (

<button

className="square"

onClick={handleClick}

>

{value}

</button>

);

}

Désormais, si vous cliquez sur un carré, vous devriez voir un message disant "cliqué !" dans l’onglet Console en bas de la section Browser de CodeSandbox

Vous souhaitez maintenant que le composant Square « se souvienne » qu’on lui a cliqué dessus, et se remplisse alors avec la marque « X »

« useState » : React fournit une fonction spéciale appelée useState que vous pouvez appeler depuis votre composant pour qu’il « se souvienne » de choses.

Stockons donc la valeur actuelle de Square dans un état, et modifions-la quand on clique sur le Square.

Importez useState au début du fichier. Retirez la prop value du composant Square. Remplacez-la par une nouvelle ligne au début de la fonction Square qui appelle useState. Faites-lui renvoyer une variable d’état appelée value

import { useState } from 'react';

function Square() {

const [value, setValue] = useState(null);

function handleClick() {

//...

« value » stocke la valeur

« setValue » est une fonction qu’on peut utiliser pour modifier la valeur

« null » : utilisé comme valeur initiale de la variable d’état

**🡪index.js**

il est la passerelle entre le composant que vous avez créé dans le fichier App.js et le navigateur web.

import React, { StrictMode } from "react";

import { createRoot } from "react-dom/client";

import "./styles.css";

import App from "./App";

* React
* La bibliothèque React qui parle aux navigateurs web (React DOM)
* Les styles de vos composants
* Le composant que vous avez créé dans App.js.

**🡪style.css**

Ce fichier définit les styles de votre appli React

« \* »  et « body » : définissent le style de larges pans de votre appli

« .square » : définit le style de tout composant dont la propriété className vaudra square.

PROJET

Ce qu’il faut faire

* **Choisir pour chaque page l’esthétique qu’on veut, les cases, textes, couleurs, boutons etc. : CANVA**
* **Transférer résultat de Teachable machine dans ma page web et réussir à traiter les resultats**
* **Trouver code pour créer clique bouton et pouvoir envoyer les résultats dans notre base de données**

**TO DO LIST MOI**

* **Arrivée a integrer une to-do-list**
* **Trouver un nom pour notre société**
* **Faire un logo pour notre société**
* **Mettre le fond en blanc**
* **Changer la police d’ecriture**
* **Rajouter un ecritot tableau de bord**
* **Mettre une marge coloré avec tableau de bord**

**Police ecriture code : https://fonts.google.com/selection/embed**

**Logo code :** [**https://fontawesome.com/kits/4188c0ee28/setup**](https://fontawesome.com/kits/4188c0ee28/setup)

**Pour moi chaque modele correspond a un dessin donc y a bon et pas bon**

**On fait un modele pour chaque dessin**

**Ensuite dans le code on fait un truc aleatoire pour avoir un modele donc le dessin qu’il doit faire et la webcam**