React Three Fiber

Web Application

Suriya Natsupakpong, PhD

Institute of Field Robotics (FIBO)

King Mongkut's University of Technology Thonburi (KMUTT)

React Three Fiber

- React three fiber is React renderer.
- It takes care of a lot of default setting and make good use of React tools.
- Created in 2019 by Paul Henschel (0xca0a), and stable since version 8 (May 2022)

First Vite Project

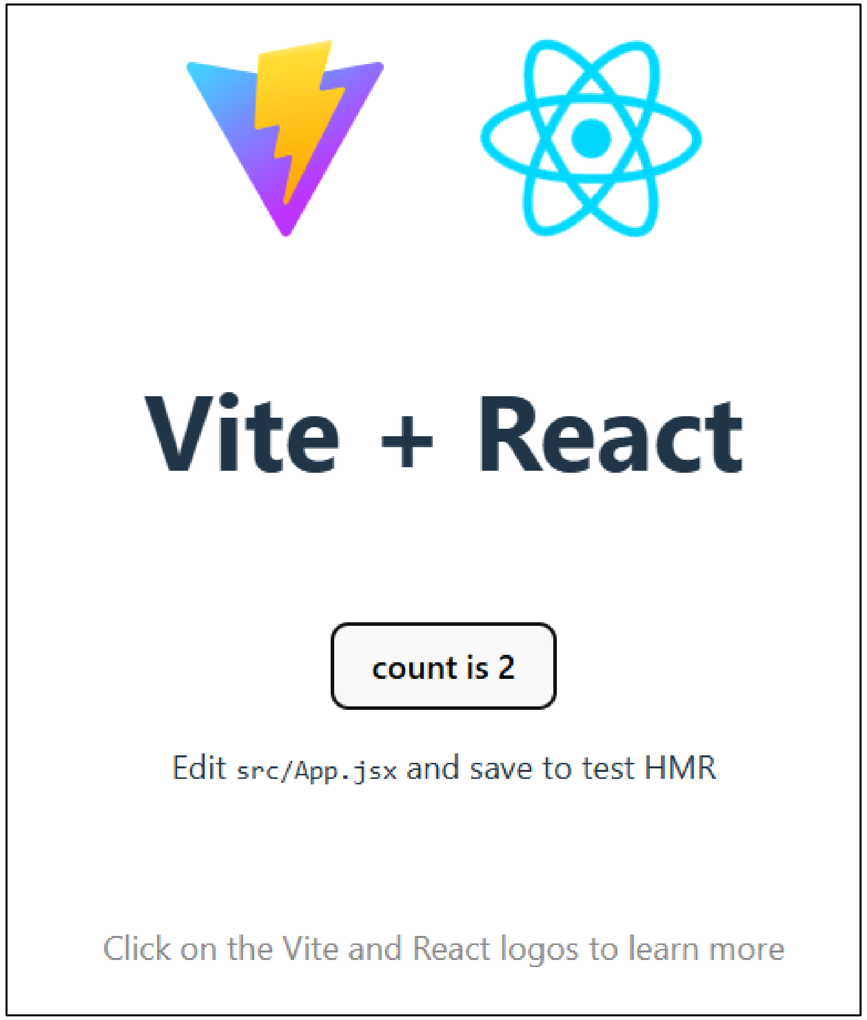
- To create Vite project type:

```
npm create vite@latest
```

```
> npm create vite@latest
Need to install the following packages:
create-vite@6.1.1
Ok to proceed? (y) y
```

- Change the project name as you want,
  select a framework: `React`
  and then select a variant: `Javascript`
- Go to project folder and type:

  `npm install` and `npm run dev`


Vite + React

count is 2

Edit `src/App.jsx` and save to test HMR

Click on the Vite and React logos to learn more

React Three Fiber Application

- Install:

```
npm install three @react-three/fiber
npm install three@0.166 @react-three/fiber@8.16
```

- R3F is a React renderer. We write JSX and it gets rendered into Three.js.

```
const mesh = new THREE.Mesh()
mesh.geometry = new THREE.BoxGeometry(1, 1, 1)
mesh.material = new THREE.MeshBasicMaterial({ color: 'red' })

scene.add(mesh)
```

→

```
<mesh>
    <boxGeometry />
    <meshBasicMaterial color="red" />
</mesh>
```

```
const group = new THREE.Group()
scene.add(group)

const mesh1 = new THREE.Mesh()
mesh1.geometry = new THREE.BoxGeometry(1, 1, 1)
mesh1.material = new THREE.MeshBasicMaterial({ color: 'red' })

const mesh2 = new THREE.Mesh()
mesh2.geometry = new THREE.SphereGeometry(0.5)
mesh2.material = new THREE.MeshBasicMaterial({ color: 'orange' })

group.add(mesh1, mesh2)
```

→

```
<group>
    <mesh>
        <boxGeometry />
        <meshBasicMaterial color="red" />
    </mesh>
    <mesh>
        <sphereGeometry />
        <meshBasicMaterial color="orange" />
    </mesh>
</group>
```

APP.jsx

~~index.jsx~~

```
import { createRoot } from 'react-dom/client'
import * as THREE from 'three'
import { Canvas } from '@react-three/fiber'
import Experience from './Experience.jsx'
import './style.css'

const root = createRoot(document.querySelector('#root'))
root.render(
    <>
        <Canvas gl = { {
            antialias: true,
            toneMapping: THREE.ACESFilmicToneMapping } }
                camera={ {
            fov: 45, near: 0.1, far: 200,
            position: [3, 2, 6] } }
        >
            <Experience/>
        </Canvas>
    </>
)
```

style.css

```
html,
body,
#root
{
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    overflow: hidden;
    background: lightskyblue;
}
```

Experience.jsx

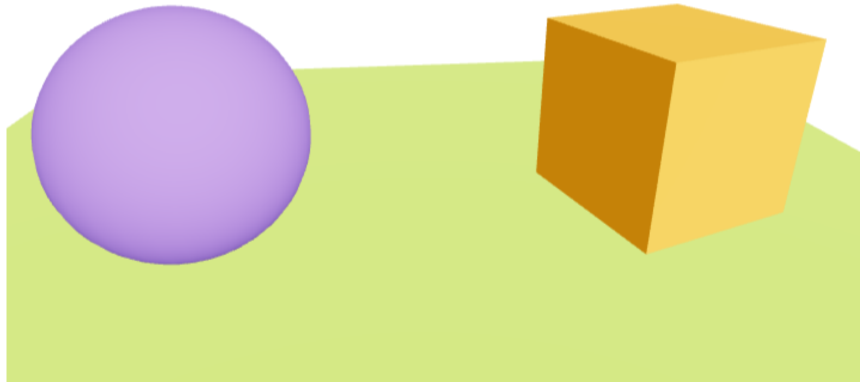
```
import { useFrame } from "@react-three/fiber"
import { useRef } from 'react'

export default function Experience()
{
    const cubeRef = useRef()
    const groupRef = useRef()

    useFrame((state, delta)=>
    {
        cubeRef.current.rotation.y += delta
        groupRef.current.rotation.y += delta
    })

    return <>
        <group ref={groupRef}>
            <mesh ref={cubeRef} position-x={2} scale = {1.5}>
                <boxGeometry/>
                <meshBasicMaterial color="orange"/>
            </mesh>
            <mesh position-x = {-2} >
                <sphereGeometry args={[ 1, 32, 32 ]} />
                <meshBasicMaterial color="mediumpurple" />
            </mesh>
            <mesh position-y={-1} rotation-x={-Math.PI * 0.5} scale={10}>
                <planeGeometry/>
                <meshBasicMaterial color="greenyellow"/>
            </mesh>
        </group>
    </>
}
```

$\nearrow$ group rotate

Controls

Experience.jsx

```
import { extend, useThree, useFrame } from "@react-three/fiber"
import { useRef } from 'react'
import { OrbitControls } from "three/examples/jsm/controls/OrbitControls.js"

extend({ OrbitControls })

export default function Experience()
{
    const {camera, gl} = useThree()

    const cubeRef = useRef()
    const groupRef = useRef()

    useFrame((state, delta)=>
    {
        cubeRef.current.rotation.y += delta
        groupRef.current.rotation.y += delta
    })

    return <>
        <orbitControls args={[camera, gl.domElement]}/>


        <group ref={groupRef}>

           {/* ...  */}

        </group>
    </>
}
```



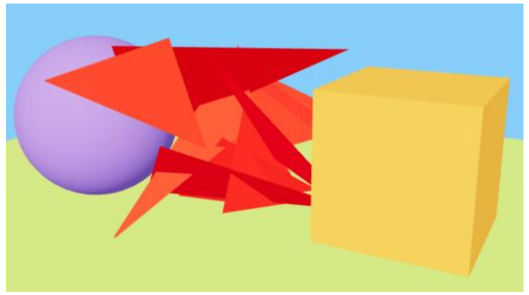
Extend will try to automatically convert a Three.js class into a declarative version and make it available in JSX.

Lights

```
        <directionalLight position={[1, 2, 3]} intensity={4.5}/>
        <ambientLight intensity={1.5}/>
```

And replace all <meshBasicMaterial> by <meshStandardMaterial>

Custom Geometry



Experience.jsx

```jsx
import { extend, useThree, useFrame } from "@react-three/fiber"
{/* ...  */}
import CustomObject from "./CustomObject.jsx"

extend({ OrbitControls })

export default function Experience()
{
    {/* ...  */}

    return <>
        <orbitControls args={[camera, gl.domElement]}/>
        <directionalLight position={[1, 2, 3]} intensity={4.5}/>
        <ambientLight intensity={1.5}/>

        <group ref={groupRef}>
           {/* ...  */}

           <CustomObject/>
        </group>

    </>
}
```

CustomObject.jsx

```jsx
import * as THREE from 'three'
import { useEffect, useRef, useMemo } from 'react'

export default function CustomObject()
{
    const geometryRef = useRef()
    const verticesCount = 10 * 3
    const positions = useMemo(() =>
    {
        const positions = new Float32Array(verticesCount * 3)

        for(let i=0; i<verticesCount*3; i++)
            positions[i] = (Math.random() - 0.5) * 3

        return positions
    }, [])

    useEffect(() =>
    {
        geometryRef.current.computeVertexNormals()
    }, [])

    return <mesh>
        <bufferGeometry ref={geometryRef}>
            <bufferAttribute
                attach="attributes-position"
                count={ verticesCount }
                itemSize ={ 3 }
                array={ positions }
            />
        </bufferGeometry>
        <meshStandardMaterial color="red" side={ THREE.DoubleSide }/>
    </mesh>
}
```

drei

https://github.com/pmndrs/drei

- A growing collection of useful helpers and fully functional, ready-made abstractions for @react-three/fiber.
- Install: `npm install @react-three/drei@9.108`
- One of the advantages of React is the ability to make things reusable. R3F took advantage of that and developers are creating many components (called helpers) ready to be used in your R3F application.
- Some examples: Camera Controls, Complex geometries, Post-processing, HTML implementation, Loaders, Environment settings, Complex calculations, Etc.
- Try these:
    - OrbitControls
    - TransformControls
    - PivotControls
    - Html
    - Text

Experience.jsx

```
import { Html, PivotControls, TransformControls, OrbitControls } from '@react-three/drei'
import { useRef } from 'react'

export default function Experience()
{
    const cube = useRef()
    const sphere = useRef()

    return <>
        <OrbitControls makeDefault />

        <directionalLight
            position={ [ 1, 2, 3 ] }
            intensity={ 4.5 } />
        <ambientLight intensity={ 1.5 } />


            <mesh ref={ sphere } position-x={ - 2 }>
                <sphereGeometry />
                <meshStandardMaterial color="orange" />

            </mesh>


        <mesh ref={cube} position-x={ 2 } scale={ 1.5 }>
            <boxGeometry />
            <meshStandardMaterial color="mediumpurple" />
        </mesh>
        <TransformControls object={ cube } mode="translate" />

        <mesh position-y={ - 1 } rotation-x={ - Math.PI * 0.5 } scale={ 10 }>
            <planeGeometry />
            <meshStandardMaterial color="greenyellow" />
        </mesh>
    </>
}
```

```
<PivotControls
            anchor={ [0, 0, 0] }
            depthTest={ false }
            lineWidth={ 4 }
            axisColors={ ['red', 'green', 'blue'] }
            scale={ 100 }
            fixed={ true }
>
</PivotControls>
```

```
<Html
    position={ [ 1, 1, 0 ] }
    wrapperClass="label"
    center
    distanceFactor={ 6 }
    occlude={ [ sphere, cube ] }
>That's a sphere 👍</Html>
```

Modified style.css

```
.label > div
{
    font-family: Helvetica, Arial;
    position: absolute;
    background: #00000088;
    color: white;
    padding: 15px;
    white-space: nowrap;
    overflow: hidden;
    border-radius: 30px;
    user-select: none;
}
```

Text

- Import Text from @react-three/drei: `import { Text, ... } from '@react-three/drei'`
- And a <Text> anywhere in the scene

```
        <Text
            font="./bangers-v20-latin-regular.woff"
            fontSize={ 1 }
            color="salmon"
            position-y={ 2 }
            maxWidth={ 2 }
            textAlign='center'
        >
            Web Programming
        </Text>
```

SDF fonts

SDF stands for Signed Distance Field and is usually used in fragment shaders to draw shapes.

https://fonts.google.com/

https://transfonter.org/

Float

- Import Float from @react-three/drei: `import { Float, ... } from '@react-three/drei'`
- Add it around <Text>

```
<Float
    speed={ 5 }
    floatIntensity={ 2 }
>
   <Text>
       {/* ...  */}
   </Text>
</Float>
```

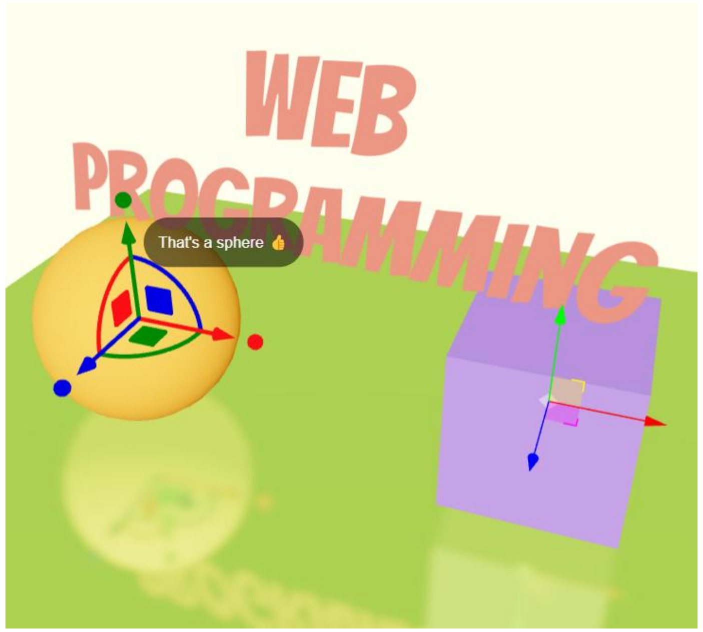
MeshReflectorMaterial

- Import MeshReflectorMaterial from @react-three/drei:

```
import { MeshReflectorMaterial, ... } from '@react-three/drei'
```

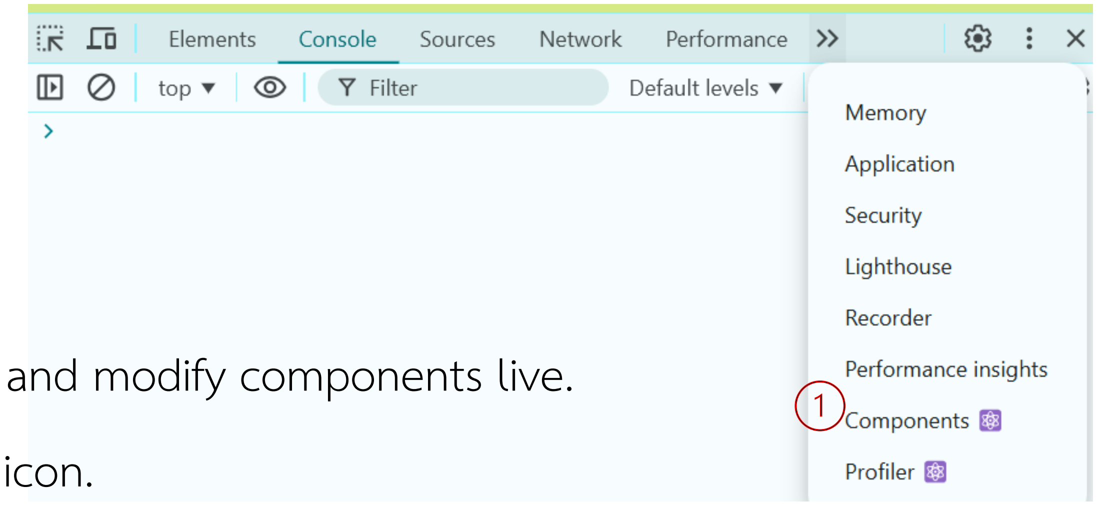
- Replace the <meshStandardMaterial> of the floor with <MeshReflectorMaterial> (in PascalCase) without the color attribute:

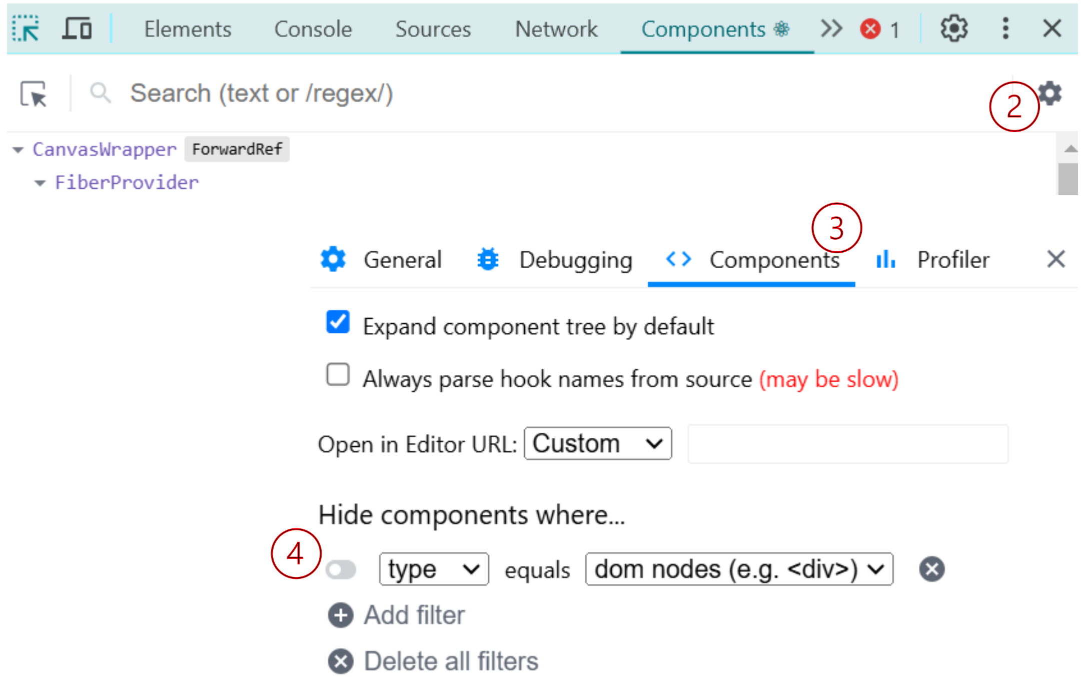
```
        <mesh position-y={ - 1 } rotation-x={ - Math.PI * 0.5 } scale={ 10 }>
             <planeGeometry />
             {/* <meshStandardMaterial color="greenyellow" /> */}
             <MeshReflectorMaterial
                 resolution={ 512 }
                 blur={ [ 1000, 1000] }
                 mixBlur={ 1 }
                 mirror={ 0.5 }
                 color="greenyellow"
             />
         </mesh>
```



Debug in R3F

- Install an extension named [React Developer Tools]() for check and modify components live.
- Click at Components tab, select canvas tag and click at gear icon.
- Select Components tab and disable type





Expand component tree by default [✓]

Always parse hook names from source (may be slow) [ ]

Open in Editor URL: Custom

Hide components where...

type equals dom nodes (e.g. <div>)

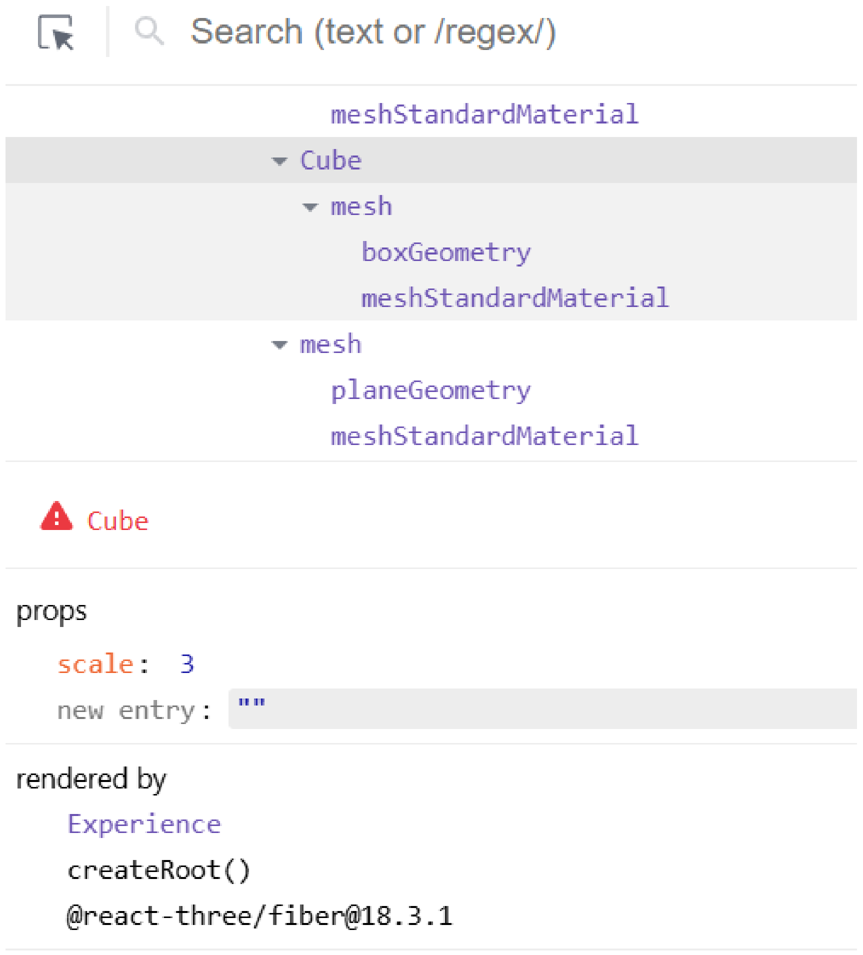
Debug Example

Modified `Experience.jsx`

```
import { OrbitControls } from '@react-three/drei'
import Cube from './Cube'

export default function Experience()
{
    return <>
        {/* ...  */}

        <Cube scale={ 2 }/>

        {/* ...  */}
    </>
}
```

`Cube.jsx`

```
export default function Cube( { scale = 1})
{
    return <mesh position-x={ 2 } scale={ scale }>
        <boxGeometry />
        <meshStandardMaterial color="mediumpurple" />
    </mesh>
}
```



```
Search (text or /regex/)
        meshStandardMaterial
      ▾ Cube
        ▾ mesh
            boxGeometry
            meshStandardMaterial
      ▾ mesh
          planeGeometry
          meshStandardMaterial

⚠ Cube

props
  scale: 3
  new entry: ""

rendered by
  Experience
  createRoot()
  @react-three/fiber@18.3.1
```

Debug UI with Leva

- Install: `npm install leva@0.9.34`

Check the documentation for more attributes https://github.com/pmndrs/leva/blob/main/docs/configuration.md

And if you want more examples https://codesandbox.io/examples/package/leva

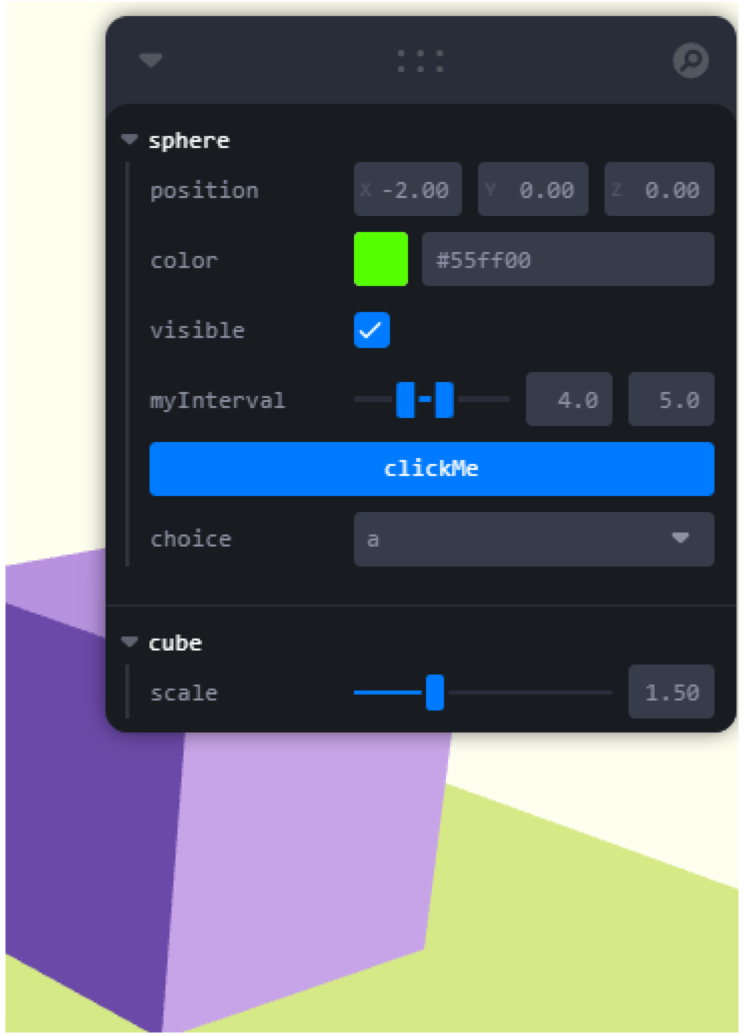
Modified `Experience.jsx`

```jsx
import { OrbitControls } from '@react-three/drei'
import { button, useControls } from 'leva'

export default function Experience()
{
    const { position, color, visible } = useControls('sphere', {
        position: {
            value: { x: -2, y: 0, z: 0 },
            step: 0.01
        },
        color: 'hsl(100deg, 100%, 50%)',
        visible: true,
        myInterval: { min:0, max:10, value: [ 4, 5 ] },
        clickMe: button(()=>{ console.log('ok')}),
        choice: { options: [ 'a', 'b', 'c' ] }
    })

    const { scale } = useControls('cube', {
        scale: {
            value: 1.5,
            step: 0.01,
            min: 0,
            max: 5
        }
    })
```

```jsx
    return <>
        <OrbitControls makeDefault />

        <directionalLight position={ [ 1, 2, 3 ] } intensity={ 4.5 } />
        <ambientLight intensity={ 1.5 } />

        <mesh position={ [position.x, position.y, position.z] }  visible={ visible }>
            <sphereGeometry />
            <meshStandardMaterial color={ color } />
        </mesh>

        <mesh position-x={ 2 } scale={ scale }>
            <boxGeometry />
            <meshStandardMaterial color="mediumpurple" />
        </mesh>

        <mesh position-y={ - 1 } rotation-x={ - Math.PI * 0.5 } scale={ 10 }>
            <planeGeometry />
            <meshStandardMaterial color="greenyellow" />
        </mesh>
    </>
}
```

index.jsx

```
import './style.css'
import ReactDOM from 'react-dom/client'
import { Canvas } from '@react-three/fiber'
import Experience from './Experience.jsx'
import { StrictMode } from 'react'
import { Leva } from 'leva'

const root = ReactDOM.createRoot(document.querySelector('#root'))

root.render(
    <StrictMode>
    <Leva collapsed />
    <Canvas
        camera={ {
            fov: 45,
            near: 0.1,
            far: 200,
            position: [ - 4, 3, 6 ]
        } }
    >
        <Experience />
    </Canvas>
    </StrictMode>
)
```



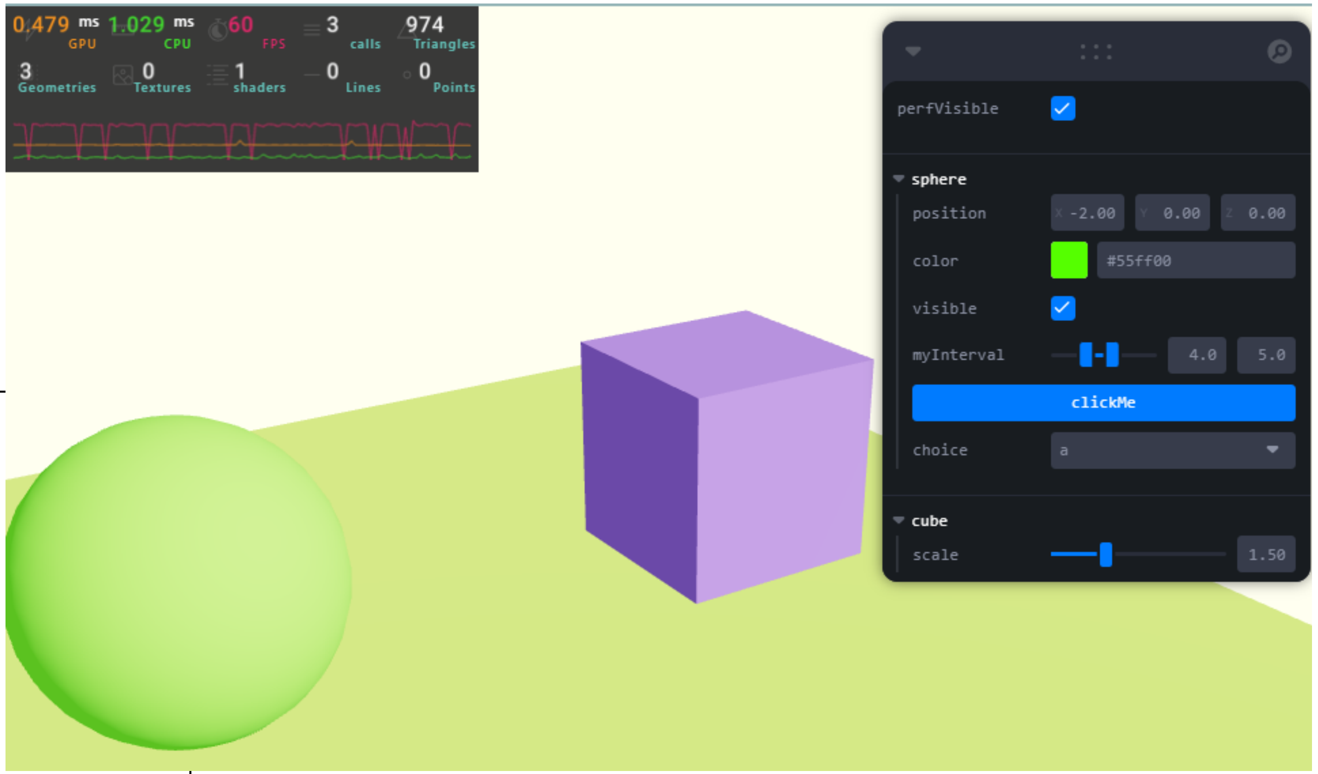
```
sphere
position     X -2.00   Y 0.00   Z 0.00
color        #55ff00
visible      ✓
myInterval   4.0   5.0
clickMe
choice       a
cube
scale        1.50
```

R3F Stat

- Install: `npm install r3f-perf@7.2`

Modified `Experience.jsx`

```jsx
import { OrbitControls } from '@react-three/drei'
import { button, useControls } from 'leva'
import { Perf } from 'r3f-perf'

export default function Experience()
{
    const { perfVisible }= useControls({
        perfVisible: true
    })

   {/* ...  */}

    return <>
        { perfVisible ? <Perf position='top-left'/> : null }

        {/* ...  */}

    </>
}
```



```
0.479 ms GPU   1.029 ms CPU   60 FPS   3 calls   974 Triangles
3 Geometries   0 Textures   1 shaders   0 Lines   0 Points
```

perfVisible ✓

sphere
- position: x -2.00, y 0.00, z 0.00
- color: #55ff00
- visible ✓
- myInterval: 4.0 5.0
- clickMe
- choice: a

cube
- scale: 1.50

Check the documentation for more info about R3F-Perf https://github.com/utsuboco/r3f-perf

Find and Load Models

- https://market.pmnd.rs/

```
import {ContactShadows, PresentationControls, Float, Environment, useGLTF, OrbitControls } from '@react-three/drei’

export default function Experience()
{
    const computer = useGLTF('https://vazxmixjsiawhamofees.supabase.co/storage/v1/object/public/models/macbook/model.gltf')

    // ...
    return <>

        {/* ... */}

        <Environment preset="city" />
        <PresentationControls
           global rotation={ [ 0.13, 0.1, 0 ] } polar={ [ - 0.4, 0.2 ] } azimuth={ [ - 1, 0.75 ] }
           config={ { mass: 2, tension: 400 } } snap={ { mass: 4, tension: 400 } }
        >
           <Float rotationIntensity={ 0.4 } >
              <primitive object={ computer.scene } position-y={ - 1.2 } />
           </Float>
        </PresentationControls>
        <ContactShadows position-y={ - 1.4 } opacity={ 0.4 } scale={ 5 } blur={ 2.4 } />
    </>

}
```

iframe

```
import { Html, Text, ContactShadows, PresentationControls, Float, Environment, useGLTF } from '@react-three/drei'
```

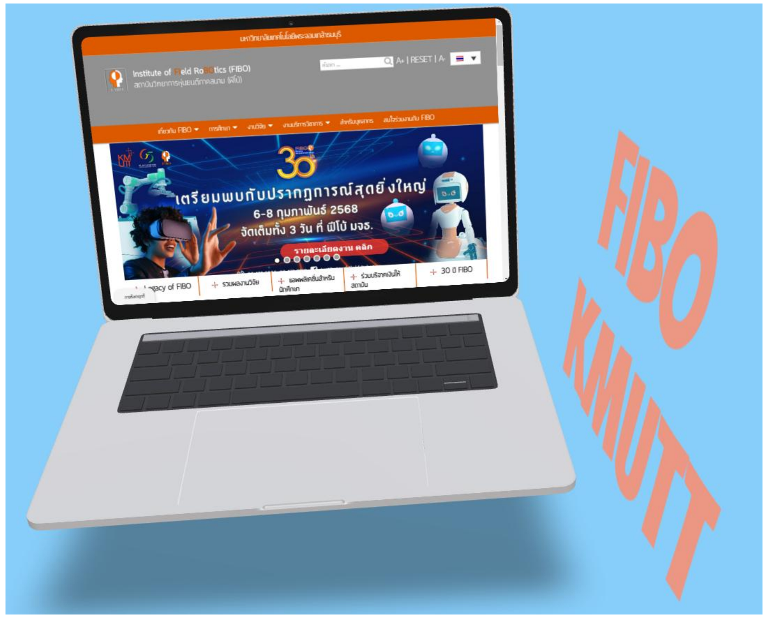
Modified `Experience.jsx`

```
<primitive
    object={ computer.scene }
    position-y={ - 1.2 }
    rotation-x={ 0.13 }
>
    <Html
       transform
       wrapperClass="htmlScreen"
       distanceFactor={ 1.17 }
       position={ [ 0, 1.56, - 1.4 ] }
       rotation-x={ - 0.256 }
    >
       <iframe src="https://fibo.kmutt.ac.th" />
    </Html>

</primitive>
```

```
<Text
  font="./Bangers-Regular.woff"
  fontSize={ 1 }
  color="salmon"
  position={ [ 2, 0.75, 0.75 ] }
  rotation-y={ - 1.25 }
  maxWidth={ 2 }
  textAlign='center'
>
  FIBO KMUTT
</Text>
```

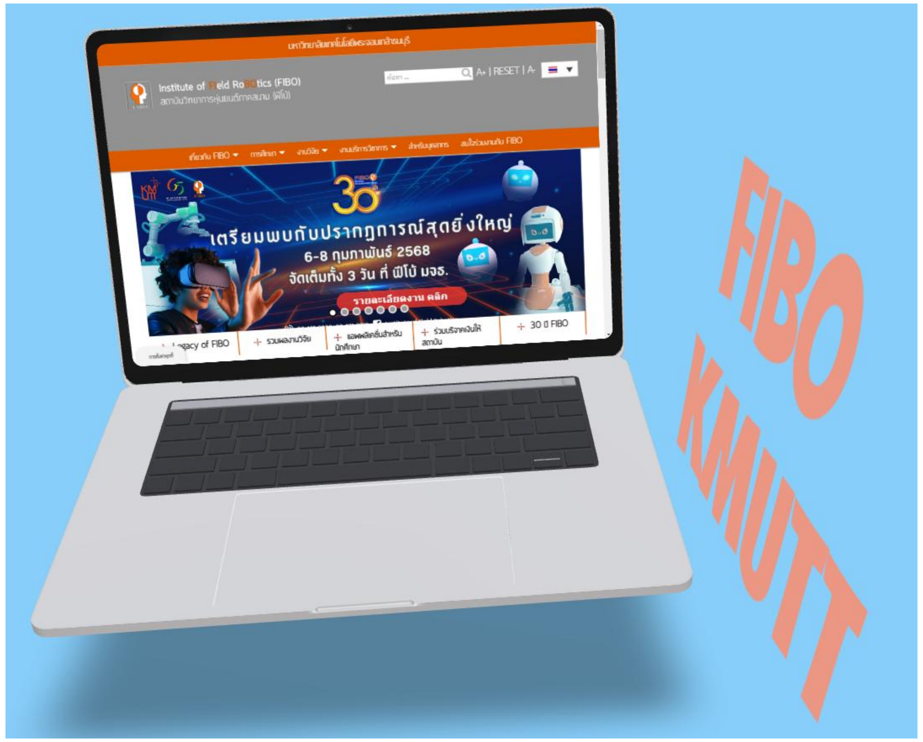
Modified `style.css`

```
.htmlScreen iframe
{
    width: 1024px;
    height: 670px;
    border: none;
    border-radius: 20px;
    background: #000000;

}
```



Fun 06: Three.js

- สร้าง Scene 3 มิติ ตามจินตนาการ โดยใช้ Three-React-Fiber
- มีการแสดงหน้าเว็บไซต์โดยใช้ iframe
- มี Model 3 มิติ อย่างน้อย 3 ชิ้น
- มี Text แสดง อย่างน้อยชื่อของตนเอง


เตรียมพบกับปรากฏการณ์สุดยิ่งใหญ่
6-8 กุมภาพันธ์ 2568
จัดเต็มทั้ง 3 วัน ที่ ฟีโบ้ มจธ.
FIBO
KMUTT