← Tutorials

Olivier Larose

February 17, 2024 / Beginner / Short

3D Glass Effect

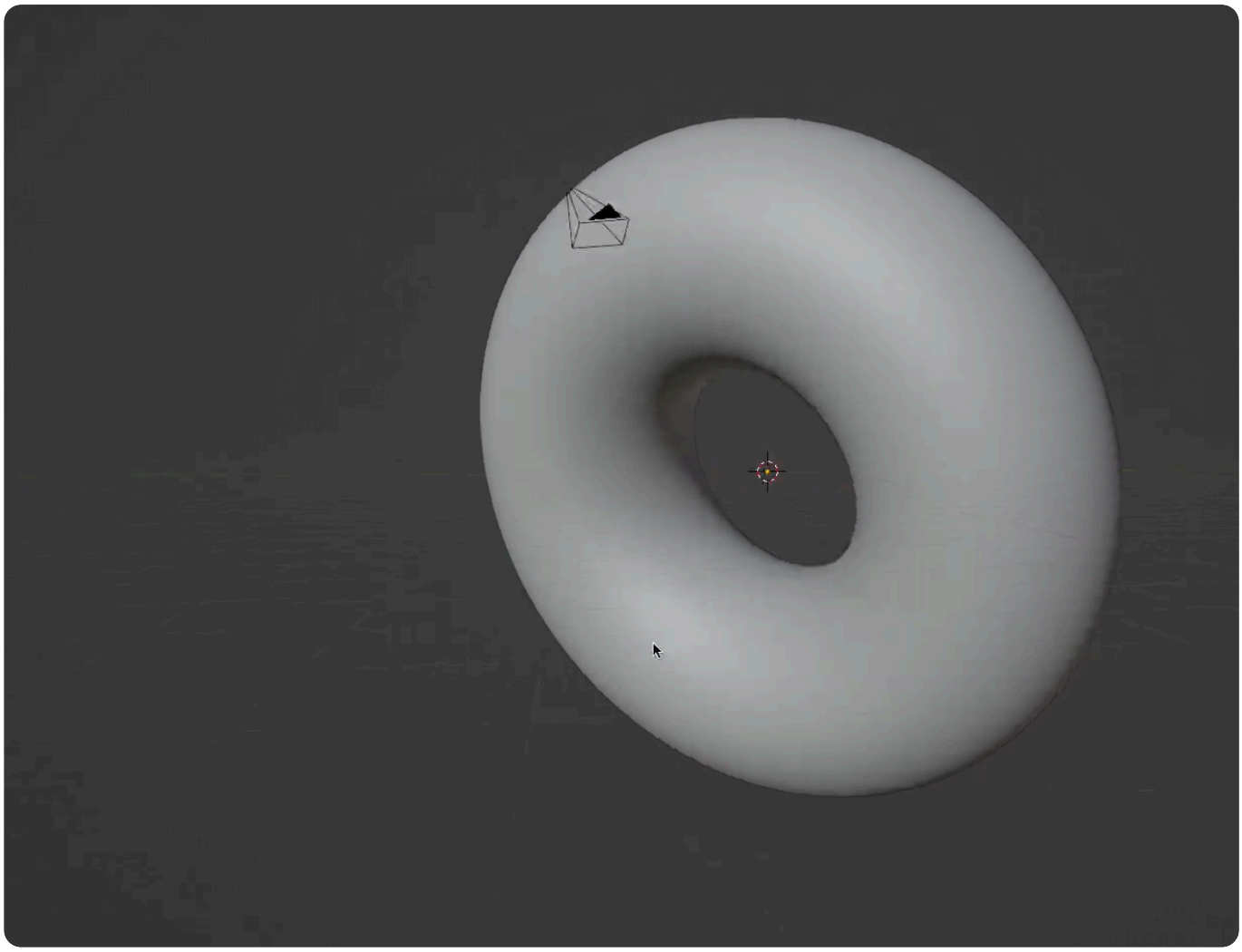
How to Make a 3D Glass Effect using Three.js and React

A website tutorial on how to create a Glass looking Material with distortion by using the MeshTransmissionMaterial with Three.js, React, React Three Fiber and Next.js.

Source code ↗

Modeling the Torus

Inside Blender I create a 3D model of a Taurus, the shape that I want for this animation.



When the 3D scene is done, we can export it inside a .GLTF or .GLB file that we can then import inside a Three.js scene.

→ You can also use this tool to visualize your file inside a web browser.

Initializing the project

Let's start the project by creating a Next.js application. We can do that by running `npx create-next-app@latest client` inside of a terminal.

We can delete everything in the `page.js` , `global.css` and `page.module.css` and add our own HTML and CSS, to start with a nice blank application.

→ We will use React Three Fiber for the 3D, so we can run `npm i three @react-three/fiber`.

→ We will use React Three Drei for utility functions, so we can run `npm i @react-three/drei`.

Setting up a Scene

For the scene, I'm creating an external component that I'm lazy loading using the Dynamic import, which is a Next.js built in function.

page.js

```
import styles from './page.module.css'
import dynamic from 'next/dynamic'

const Scene = dynamic(() => import('@/components/Scene'), {
    ssr: false,
})

export default function Home() {
  return (
     <main className={styles.main}>
         <Scene />
     </main>
   )
 }
```

Note: Here I use the `ssr: false` option to force the component to strictly be rendered client-side.

→ The upside is I could eventually render a placeholder while the 3D scene is loading.

Creating a Canvas

Here I'm creating a React Three Fiber Canvas and inside of it I'm adding an external Model Component where I'll render the 3D models.

I also add a basic light with an Environment to add lighting and colors to the overall scene.

Scene.jsx

```
'use client';
import { Canvas } from '@react-three/fiber'
import Model from './Model';
import { Environment } from '@react-three/drei'

export default function Index() {
  return (
    <Canvas style={{background: '#000000'}}>
        <Model />
         <directionalLight intensity={2} position={[0, 2, 3]}/>
         <Environment preset="city" />
    </Canvas>
  )
}
```

Creating the Model

For the model, I simply import the .GLB file and select the Torus mesh inside of it. Inside the node is a bunch

Model.jsx

```
import React, { useRef } from 'react'
import { useGLTF, Text } from "@react-three/drei";
import { useFrame, useThree } from '@react-three/fiber'

export default function Model() {
    const { nodes } = useGLTF("/medias/torrus.glb");
    const { viewport } = useThree()
    const torus = useRef(null);

     useFrame( () => {
         torus.current.rotation.x += 0.02
     })

     return (
         <group scale={viewport.width / 3.75} >
             <Text font={'/fonts/PPNeueMontreal-Bold.otf'} position={[0, 0, -1]} fontSize={0.
                 hello world!
             </Text>
             <mesh ref={torus} {...nodes.Torus002}>
                 <meshBasicMaterial/>
             </mesh>
         </group>
     )
```

```
24   }
```

- Here I also use the viewport width to scale up and down the scene depending on the size of the window to make everything responsive.
- I also use the `useFramer` to continually rotate the mesh.
- For the Text I use the **Text** component from React Three Drei. It's important to have an actual webGL text and not a simple DOM text for the distortion to be applied on it.

We should have something like this:

Transmission Material

For the transmission material, if you want transparency as well as distortion, you would have to do a custom shader. Thankfully, there is one made by the community inside the React Three Drei package called **MeshTransmissionMaterial** which gives us everything we need out of the box.

It's basically an extension of the Three.js MeshPhysicalMaterial, which already support transmission but adds on top of it multiple other properties like thickness, distortion and chromaticAberration. These properties really help give a realistic glass effect.

So what I'll do is apply this material to my Taurus:

Model.jsx

```jsx
import { MeshTransmissionMaterial, ... } from "@react-three/drei";
import { useControls } from 'leva';

...

export default function Model() {
    ...
    const materialProps = useControls({
        thickness: { value: 0.2, min: 0, max: 3, step: 0.05 },
         roughness: { value: 0, min: 0, max: 1, step: 0.1 },
         transmission: {value: 1, min: 0, max: 1, step: 0.1},
```

```
        ior: { value: 1.2, min: 0, max: 3, step: 0.1 },
        chromaticAberration: { value: 0.02, min: 0, max: 1},
        backside: { value: true},
    })

    return (
        <group scale={viewport.width / 3} >
            ...
            <mesh ref={torus} {...nodes.Torus002}>
                <MeshTransmissionMaterial {...materialProps}/>
            </mesh>
        </group>
    )
}
```

We should have something like this:

Wrapping up

That's it for this tutorial!

I'm just taken away by how powerful and easy to use this material is. I'm super grateful of being part of such a great community as well. Hope you learned something!

-Oli

Related Animations

April 21, 2024

3D

Ripple Shader

A website animation tutorial featuring a ripple shader effect using React Three Fiber, Next.js and React. Inspired by https://homunculus.jp/ and Yuri Artiukh.

April 21, 2024

3D

Bulge Effect

A website tutorial featuring a bulge distortion animation, made with a shader in GLSL, using React Three Fiber, Next.js and React.

April 18, 2024

3D

3D Wave on Scroll

A website animation tutorial featuring a vertex shader with a wave animation applied on a plane. Made with React-three-fiber, Framer Motion and Next.js

Subscribe to the newsletters to stay in touch with the latest.

example@gmail.com

SUBSCRIBE

Tutorials

Demos

Design system inspired by Maxime Heckel