Claro! Vamos analisar este código passo a passo, explicando cada parte e dando exemplos detalhados de como ele funciona, **com foco principal no JavaScript**, mas também explicando o HTML e o CSS que o suportam.

**🌐 1. HTML: Estrutura da Página**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="style.css">

<title>Arquira-3D</title>

</head>

<body>

<div class="card">

<img src="/2 Guerreiras.png" alt="img">

</div>

...

</body>

</html>

**Explicação:**

* **<!DOCTYPE html>**: define que o documento é um HTML5.
* **<html lang="pt-br">**: define o idioma da página como português do Brasil.
* **<link rel="stylesheet" href="style.css">**: importa o CSS externo para estilizar a página.
* **<div class="card">...</div>**: uma "cartão" com imagem que será interativo.
* **<img src="..."**: imagem posicionada sobre o card.

**🎨 2. CSS: Estilização Visual e 3D**

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

background: #121212;

}

.card {

width: 600px;

height: 500px;

background: url(/pexels-cornelius-8345087.jpg);

background-position: center;

background-size: 100%;

border-radius: 15px;

transform-style: preserve-3d;

cursor: pointer;

transition: transform 0.1s ease;

}

.card img {

position: absolute;

top: 65%;

left: 50%;

width: 800px;

transform: translate(-50%,-50%) translateZ(175px);

}

**Explicação:**

* O **body** centraliza tudo na tela com flexbox.
* A **.card** tem tamanho fixo, fundo com imagem e efeito 3D ativado com preserve-3d.
* A **imagem (img)** dentro do card está flutuando acima do plano principal, por isso translateZ(175px) — esse efeito faz parecer que a imagem “sai” da tela quando o usuário move o mouse.

**🧠 3. JavaScript: Interatividade 3D**

let card = document.querySelector('.card');

**Seleciona o elemento .card.**

**Evento de movimento do mouse:**

card.addEventListener('mousemove', (e) => {

let rect = card.getBoundingClientRect();

* getBoundingClientRect() pega o tamanho e posição do card na tela.

let x = e.clientX - rect.left - rect.width / 2;

let y = e.clientY - rect.top - rect.height / 2;

* Calcula a posição do mouse **em relação ao centro do card**.
* Exemplo: se o mouse está no centro exato do card, x e y = 0.

**Rotação com base na posição:**

let rotateX = (x / rect.height) \* -30;

let rotateY = (y / rect.width) \* -30;

* Define o ângulo de rotação baseado na posição do mouse.
* O valor 30 é o "limite" da rotação — quanto mais afastado do centro, mais o card gira.

card.style.transform = `perspective(1000px) rotate(${rotateX}deg) rotateY(${rotateY}deg)`;

* Aplica a rotação 3D no card.
* **perspective(1000px)** cria um efeito de profundidade realista.

**Ajuste do plano de fundo:**

let bgX = (x / rect.width) \* 50 + 50;

card.style.backgroundPositionX = `${bgX}%`;

* Move o **background** horizontalmente para dar um leve efeito de "parallax".

**Quando o mouse sai do card:**

card.addEventListener('mouseleave', () => {

card.style.transform = `perspective(1000px) rotateX(0deg) rotateY(0deg)`;

card.style.backgroundPositionX = ``;

})

* Ao sair com o mouse do card, ele **volta ao estado normal** (sem rotação).

**🎮 Exemplo Prático**

**Imagine:**

Você passa o mouse da esquerda para a direita no card.

* A imagem interna parece **sair da tela** (por causa do translateZ).
* O card se **inclina para os lados** conforme o movimento do mouse.
* O plano de fundo também **se move**, criando profundidade visual — como se fosse um efeito 3D.

**🧪 Sugestão de Experimento**

Você pode alterar esses valores para ver efeitos diferentes:

let rotateX = (x / rect.height) \* -10;

let rotateY = (y / rect.width) \* -10;

Isso deixaria o movimento mais **sutil**.

**✅ Resumo Final**

* **HTML**: estrutura da página com uma div.card e uma imagem.
* **CSS**: configura a aparência 3D com transform-style e translateZ.
* **JavaScript**: adiciona movimento 3D ao card com base na posição do mouse.

Esse tipo de efeito é muito usado em **portfólios**, **galerias interativas**, ou páginas modernas com foco em **UX/UI visual avançado**.

Se quiser, posso adaptar esse exemplo para que funcione em celulares também (com touchmove). Deseja isso?