

threejs canvas

threejs

```
// 
function createText(text, fontSize) {
  const scale = 0.01;
  let textLines = text.split("\n");
  if (textLines.length > 1) {
    let textLine2 = textLines[1];
    textLines.splice(1, 1); // 
    for (let i = 0; i < textLine2.length; i += 7) {
      textLines.push(textLine2.substr(i, 7));
    }
  }
}

currentCanvas.value = document.createElement("canvas", { antialias: true });
const canvas = currentCanvas.value;
const context = canvas.getContext("2d");
context.imageSmoothingEnabled = true;
context.imageSmoothingQuality = "high";

// 2x
const font = fontSize * 2 + "px Arial";
context.font = font;
// 
const textMetrics = context.measureText(text);
canvas.width = textMetrics.width + 20;
let textLineHeight = fontSize * 2 * 1.2; // 1.2
canvas.height = textLineHeight * (textLines.length + 1); // 

// 
context.clearRect(0, 0, canvas.width, canvas.height);

context.font = font;
context.textAlign = "center";
context.textBaseline = "middle";
```

```

context.fillStyle = "rgba(255,255, 255, 1)";

let textY = textLineHeight;
context.fillText(textLines[0], canvas.width / 2, textY);

context.fillStyle = "rgba(255,255, 255, 1)";
// 0000/000
for (let i = 1; i < textLines.length; i++) {
  textY += textLineHeight;
  context.fillText(textLines[i], canvas.width / 2, textY);
}

// 0000
const texture = new THREE.CanvasTexture(canvas);
texture.needsUpdate = true;

// 00000000
const geometry = new THREE.PlaneGeometry((canvas.width / 2) * scale, (canvas.height / 2) *
scale);
const material = new THREE.MeshBasicMaterial({ map: texture, opacity: 1, transparent: true
});

// 0000
const mesh = new THREE.Mesh(geometry, material);
return mesh;
}

```

000 #3

0 000 000 18 00 2025 14:39:27

0 000 000 18 00 2025 14:42:49