WebGPU Particle System – Технички дел

Овој документ ги содржи најважните делови од кодот , проследени со слика , како би се создала представа како е направена самата симулација со партикулите.

Главниот екран за партикули изгледа вака:

Овде идејата е според математичка формула да се представи бројот на партикули кој може да се менува директно со слајд на панелот кој се наоѓа десно од самиот екран со партикули и кој е прикажан подолу во документов.

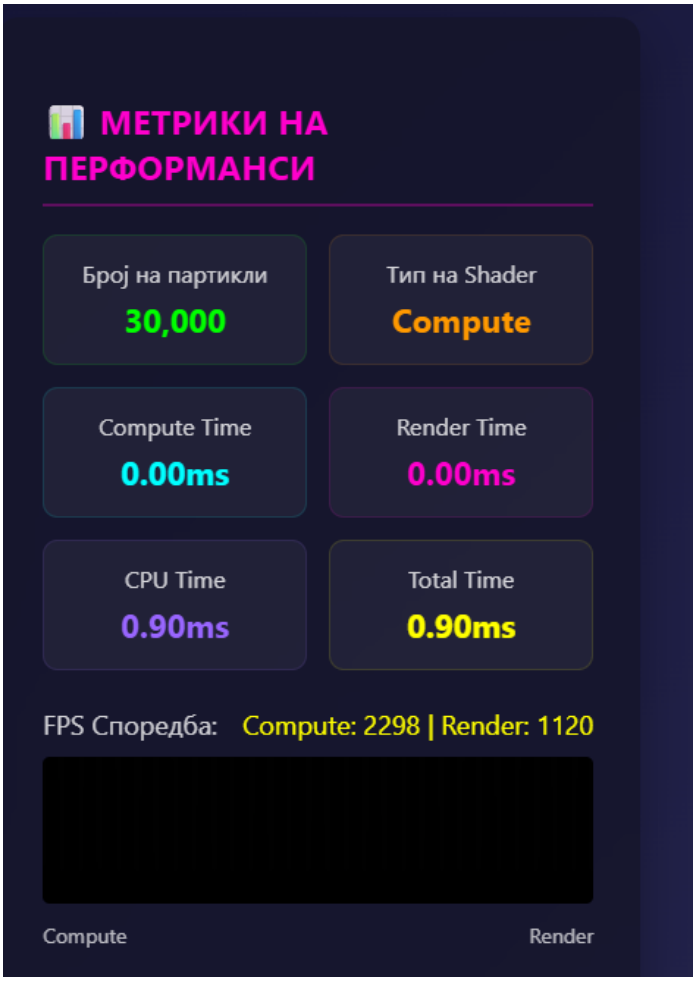
Кодот за создавање на овој екран е сложен , но математичката функција е следната

```
//inicijalizacija na partikli
function initializeParticles() {
    const positions = new Float32Array(particleCount * 2);
    const velocities = new Float32Array(particleCount * 2);
    const colors = new Float32Array(particleCount * 4);

    for (let i = 0; i < particleCount; i++) {
        positions[i * 2] = (Math.random() - 0.5) * 1.8;
        positions[i * 2 + 1] = (Math.random() - 0.5) * 1.8;

        velocities[i * 2] = (Math.random() - 0.5) * 0.02;
        velocities[i * 2 + 1] = (Math.random() - 0.5) * 0.02;

        const hue = (i / particleCount) * 360;
        colors[i * 4] = (Math.sin(hue * 0.01745) * 0.5 + 0.5);
        colors[i * 4 + 1] = (Math.sin((hue + 120) * 0.01745) * 0.5 + 0.5);
        colors[i * 4 + 2] = (Math.sin((hue + 240) * 0.01745) * 0.5 + 0.5);
        colors[i * 4 + 3] = 0.8;
    }

    device.queue.writeBuffer(particleBuffer, 0, positions);
    device.queue.writeBuffer(velocityBuffer, 0, velocities);
    device.queue.writeBuffer(colorBuffer, 0, colors);
}
```

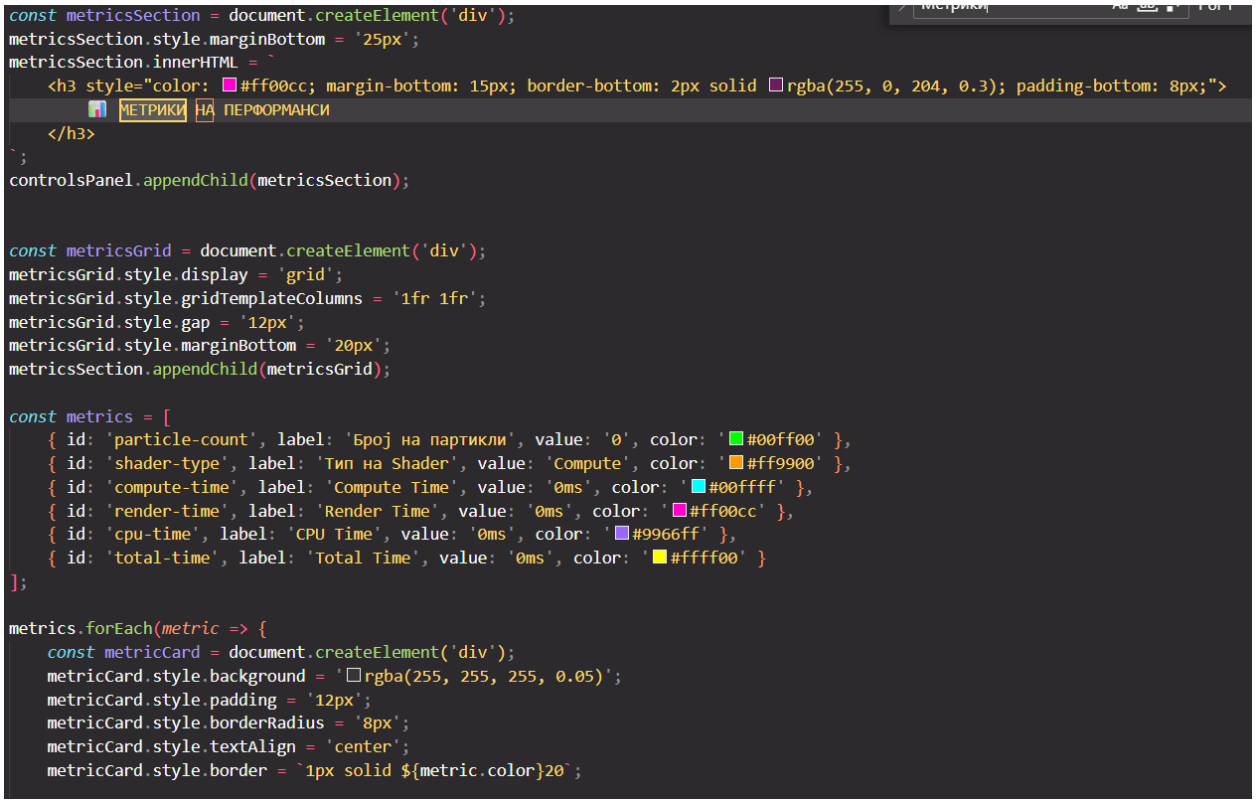
Овде гледаме дека се користат листи од флоат променливи во кои се зачувани самите партикули.

Десно од самиот екран со партикули се наоѓа и секцијата на ,,Метрики на перформанси’’

која изгледа вака



Создавањето на оваа секција во код изгледа вака:

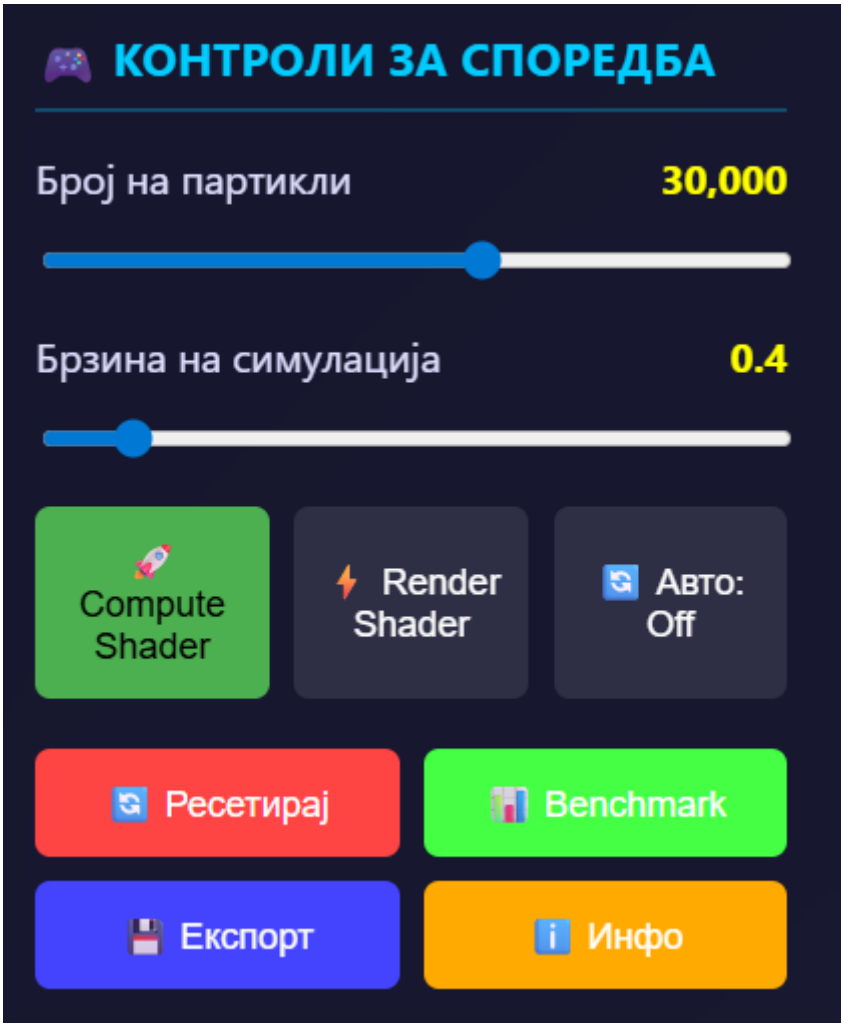


```
const metricsSection = document.createElement('div');
metricsSection.style.marginBottom = '25px';
metricsSection.innerHTML = `
    <h3 style="color: #ff00cc; margin-bottom: 15px; border-bottom: 2px solid rgba(255, 0, 204, 0.3); padding-bottom: 8px;">
        МЕТРИКИ НА ПЕРФОРМАНСИ
    </h3>
`;
controlsPanel.appendChild(metricsSection);


const metricsGrid = document.createElement('div');
metricsGrid.style.display = 'grid';
metricsGrid.style.gridTemplateColumns = '1fr 1fr';
metricsGrid.style.gap = '12px';
metricsGrid.style.marginBottom = '20px';
metricsSection.appendChild(metricsGrid);

const metrics = [
    { id: 'particle-count', label: 'Број на партикли', value: '0', color: '#00ff00' },
    { id: 'shader-type', label: 'Тип на Shader', value: 'Compute', color: '#ff9900' },
    { id: 'compute-time', label: 'Compute Time', value: '0ms', color: '#00ffff' },
    { id: 'render-time', label: 'Render Time', value: '0ms', color: '#ff00cc' },
    { id: 'cpu-time', label: 'CPU Time', value: '0ms', color: '#9966ff' },
    { id: 'total-time', label: 'Total Time', value: '0ms', color: '#ffff00' }
];

metrics.forEach(metric => {
    const metricCard = document.createElement('div');
    metricCard.style.background = 'rgba(255, 255, 255, 0.05)';
    metricCard.style.padding = '12px';
    metricCard.style.borderRadius = '8px';
    metricCard.style.textAlign = 'center';
    metricCard.style.border = `1px solid ${metric.color}20`;
```

може да забележиме инер хтмл за создавање и подобрување на изгледот на текстот , а потоа ги имаме променливите за грид на метриките , и ги имаме самите метрики во посебна константна променлива и исто така соодветните кодови за различни бои , а најдолу на сликата имаме една forEach јамка која слижи за изгледот на секцијата.

Најважниот дел е во секцијата ,,Контроли за споредба” , каде што корисникот е во можност со притискање на копчето ,,Авто” , да ја пушти во функција автоматската промена помеѓу двата различни Shaders , а нагласуваме дека е главен дел затоа што самата суштина на овој проект е споредба на двата типа Shaders и мерење на Fps-от кој го дава секој од нив , а исто така и користењето на меморија на процесор времето на извршување и слично.

тој дел изгледа вака:



Тука интересно е копчето ,,Benchmark” кое ни дава табела во која го пишува Fps-от и за Compute Shader и за Render Shader , исто така го пишува и времето на извршување и разликата во fps , па на крај ни дава предлог да го користиме shader-от кој има подобри перформанси.

за да ги добиеме копчињата кои ги гледаме на сликата , потребно беше да се напише следниов код :

```
const controlsSection = document.createElement('div');
controlsSection.innerHTML = `
    <h3 style="color: #00ccff; margin-bottom: 15px; border-bottom: 2px solid rgba(0, 204, 255, 0.3); padding-bottom: 8px;">
        КОНТРОЛИ ЗА СПОРЕДБА
    </h3>
`;
controlsPanel.appendChild(controlsSection);


const sliders = [
    { id: 'particle-slider', label: 'Број на партикли', min: 1000, max: 50000, step: 1000, value: 5000 },
    { id: 'speed-slider', label: 'Брзина на симулација', min: 0.1, max: 3, step: 0.1, value: 1.0 }
];

sliders.forEach(slider => {
    const sliderContainer = document.createElement('div');
    sliderContainer.style.marginBottom = '20px';

    const label = document.createElement('label');
    label.textContent = slider.label;
    label.style.display = 'block';
    label.style.marginBottom = '8px';
    label.style.color = '#e0e0ff';
    sliderContainer.appendChild(label);

    const valueDisplay = document.createElement('div');
    valueDisplay.id = `${slider.id}-value`;
    valueDisplay.style.float = 'right';
    valueDisplay.style.color = '#ffff00';
    valueDisplay.style.fontWeight = 'bold';
    label.appendChild(valueDisplay);
```

```
const shaderModes = [
    { id: 'btn-compute', text: '🚀 Compute Shader', active: true },
    { id: 'btn-render', text: '⚡ Render Shader', active: false },
    { id: 'btn-auto-switch', text: '🔄 Авто Прекинувач', active: false }
];

shaderModes.forEach(mode => {
    const button = document.createElement('button');
    button.id = mode.id;
    button.textContent = mode.text;
    button.style.padding = '12px 8px';
    button.style.border = 'none';
    button.style.borderRadius = '6px';
    button.style.background = mode.active ? '#4CAF50' : 'rgba(255, 255, 255, 0.1)';
    button.style.color = mode.active ? '#000' : '#fff';
    button.style.cursor = 'pointer';
    button.style.transition = 'all 0.3s ease';
    button.style.fontSize = '0.9rem';
    button.style.flex = '1';
```

```
const actions = [
    { id: 'btn-reset', text: '🔄 Ресетирај', color: '🟥#ff4444' },
    { id: 'btn-benchmark', text: '📊 Benchmark', color: '🟩#44ff44' },
    { id: 'btn-export', text: '💾 Експорт', color: '🟦#4444ff' },
    { id: 'btn-info', text: 'ℹ️ Инфо', color: '🟧#ffaa00' }
];

actions.forEach(action => {
    const button = document.createElement('button');
    button.id = action.id;
    button.textContent = action.text;
    button.style.padding = '12px 8px';
    button.style.border = 'none';
    button.style.borderRadius = '6px';
    button.style.background = action.color;
    button.style.color = '⬜#fff';
    button.style.cursor = 'pointer';
    button.style.transition = 'all 0.3s ease';
    button.style.fontSize = '0.9rem';
```

Најдолу имаме една секција ,,Информации” , во која излегуваат информации поврзани со WebGPU , со Compute Shader и Render Shader , како и со fps-от , кога се користат двата shaders.

таа секција изгледа вака: