

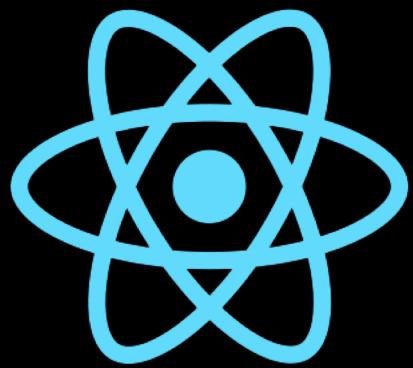
60 FPS и выше

Artem Kunets, Align Technology



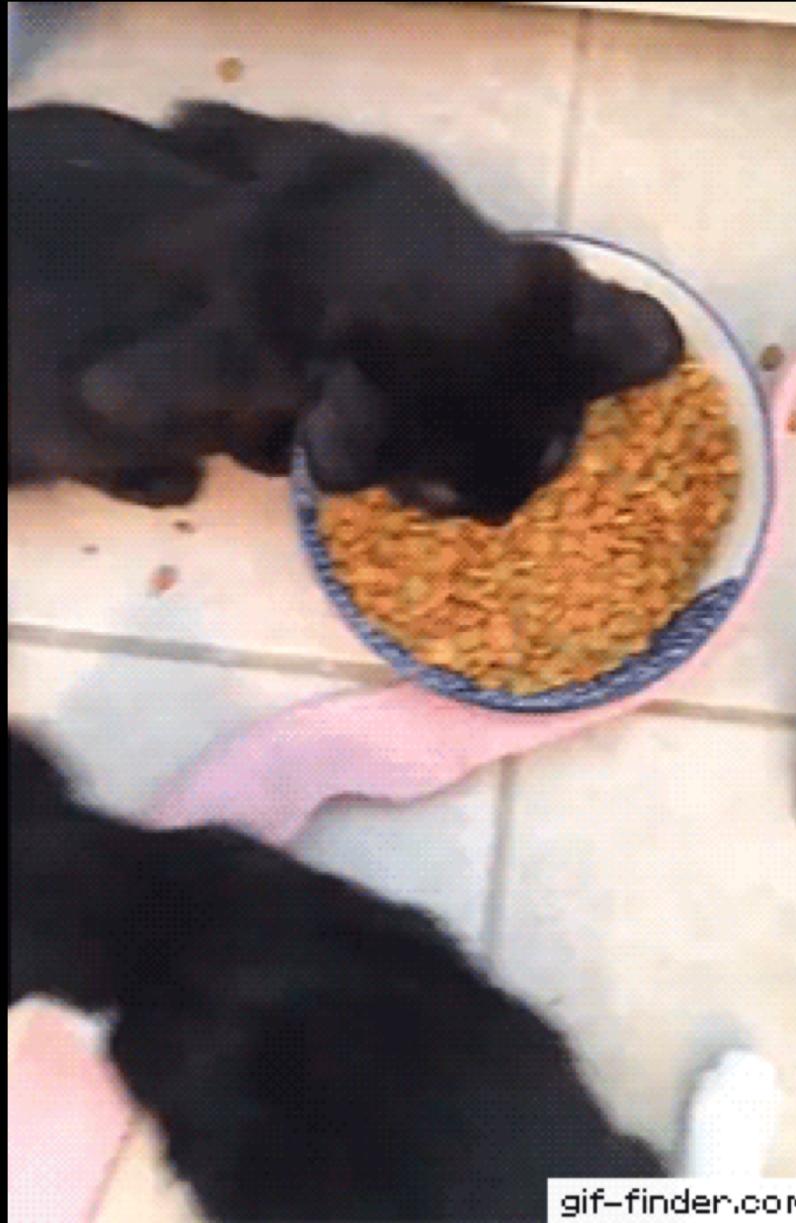
McAbbyFan | Tumblr

Стек

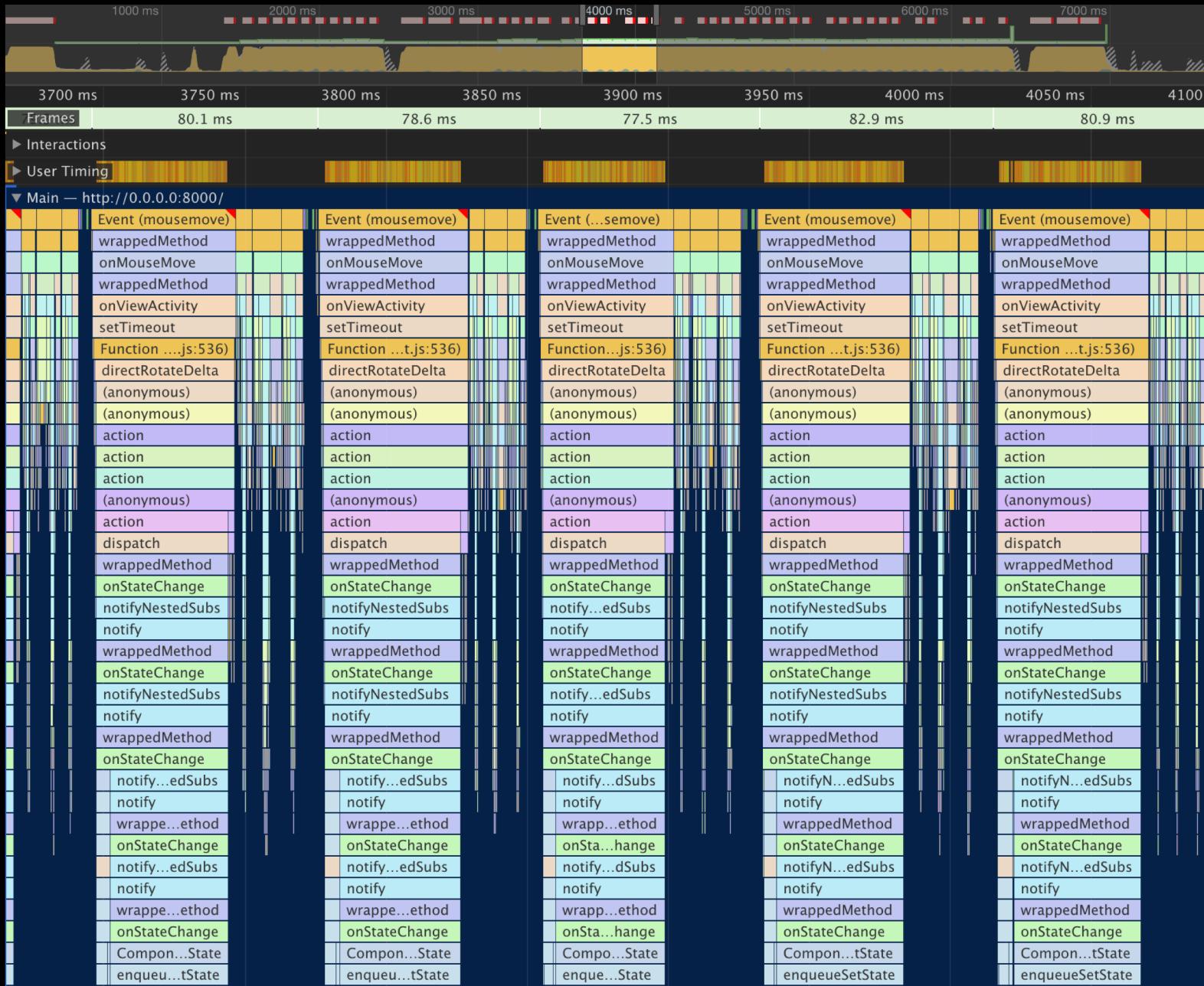


+





gif-finder.com

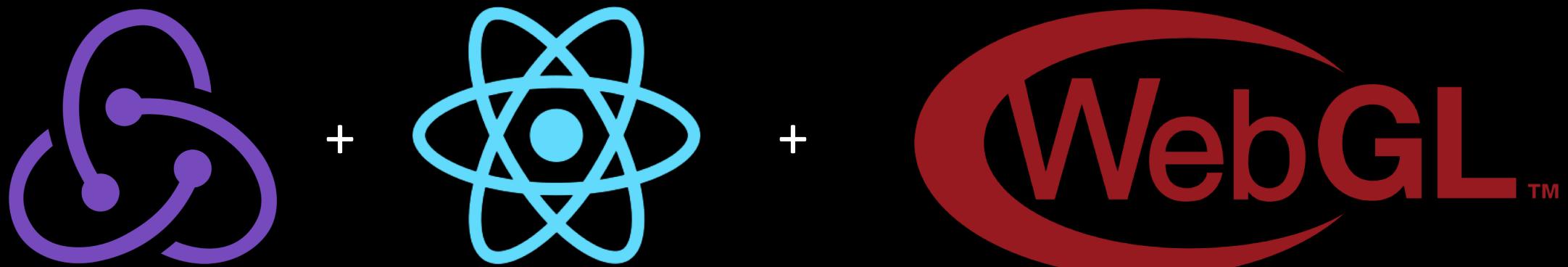


```
// Canvas component
updateCamera(camera: CameraState) {
    this.camera.update(camera);
}

// Root component
updateCamera(matrix: Matrix4) {
    this.camera.applyToCamera(matrix);
    this.canvas.updateCamera(this.getCameraState());
}

render() {
    return <EventObserver updateCamera={this.updateCamera} />;
}
```

Новый стек



Что оптимизировать



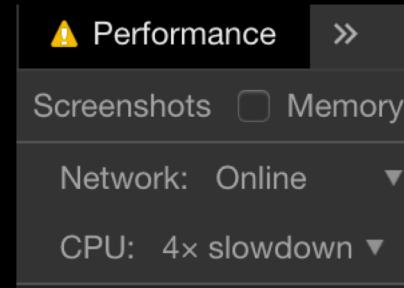
Что оптимизировать

```
render() {  
  const now = Date.now();  
  while((Date.now() - now) < 5000) {  
    console.log('WAITING');  
  }  
  
  return <div>Component</div>;  
}
```

Зачем оптимизировать?

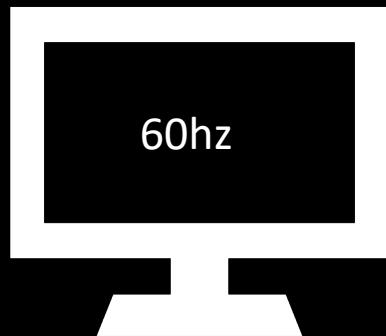


или



Что измерять

FPS – frame per second

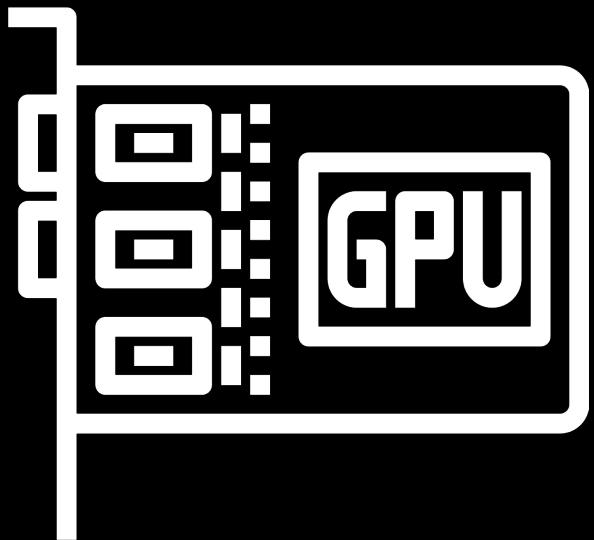


$$1000/60 = 16.66666 \text{ мс на кадр}$$



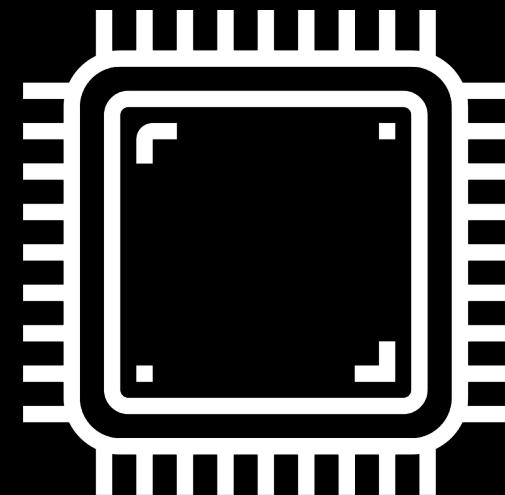
$$1000/144 = 6.94444 \text{ мс на кадр}$$

Что с ресурсами



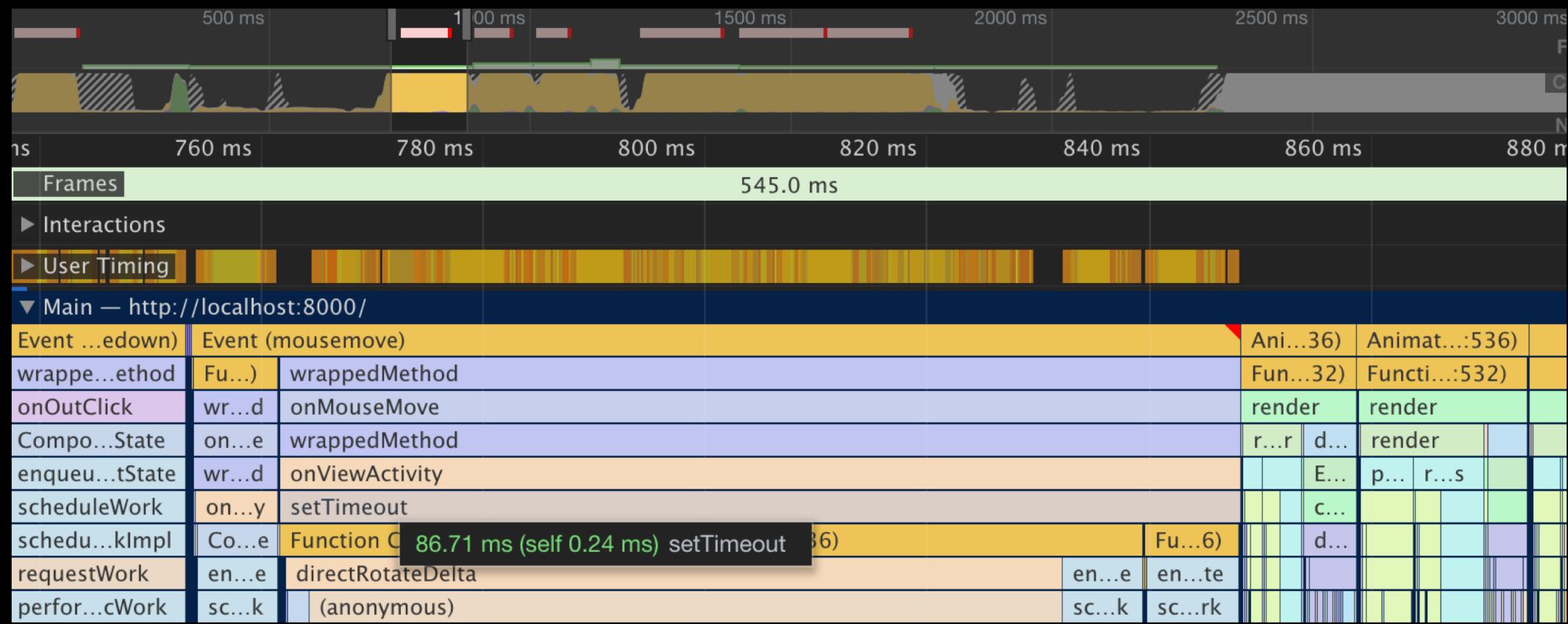
GLSL

и

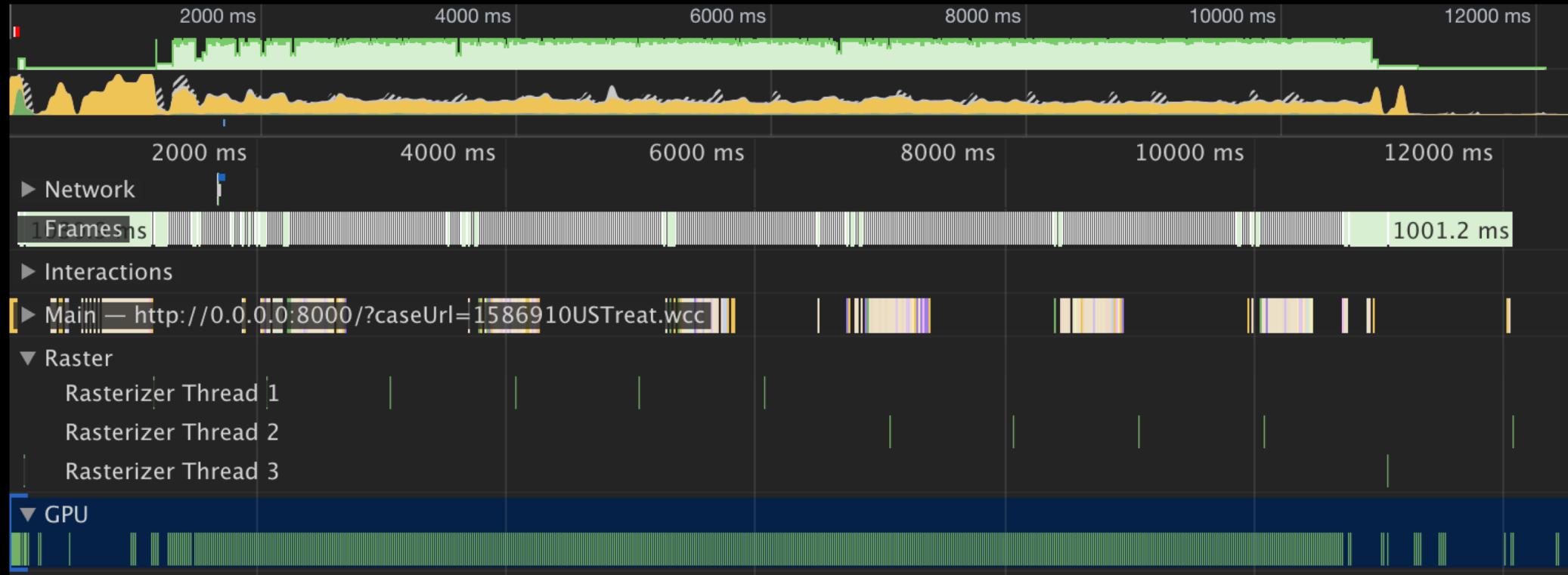


JS

Измеряем срн



Измеряем гру



Измеряем гри

```
// создаем запрос
var ext = gl.getExtension('EXT_disjoint_timer_query');
var query = ext.createQueryEXT();

// измеряем
ext.beginQueryEXT(ext.TIME_ELAPSED_EXT, query);
// То что будем измерять
gl.drawArrays(gl.TRIANGLES, 0, 3);
ext.endQueryEXT(ext.TIME_ELAPSED_EXT);

// получаем результаты
const timeElapsed = ext.getQueryObjectEXT(query, ext.QUERY_RESULT_EXT);
```

Куда GPU тратит ресурсы

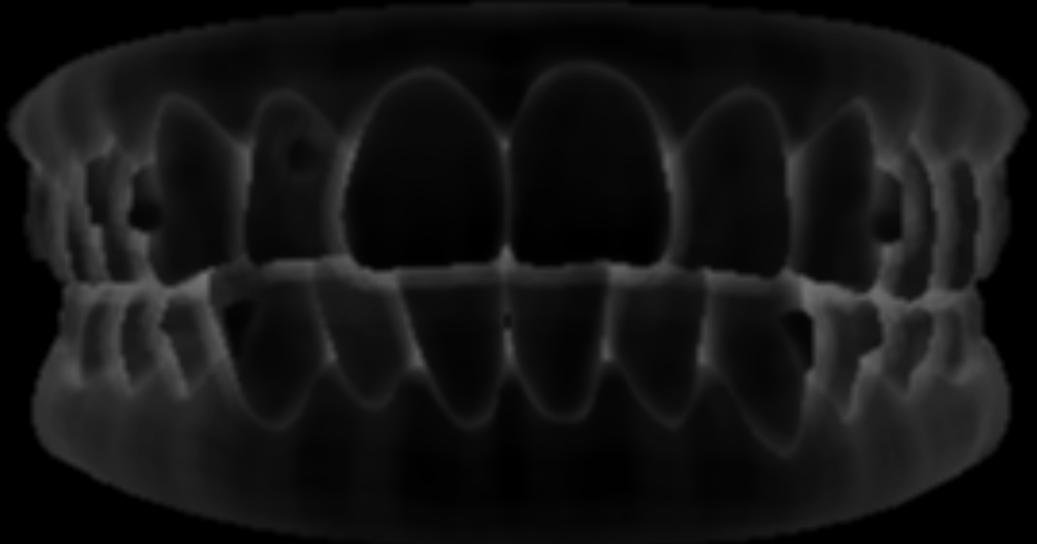


60 fps



30 fps

Пост обработка



SSAO



Тень

Логируем вызовы гру

```
function logGLCall(functionName, args) {  
    console.log(functionName)  
}  
  
function onError(err, functionName, args) {  
    console.log(err)  
}  
  
gl = WebGLDebugUtils.makeDebugContext(gl, onError, logGLCall);
```

Выводы

- Следите за внешним кодом
- Избегайте ререндера во время отрисовки сцены
- Overdraw – не рисуйте один пиксель несколько раз
- Уменьшите вызовы `gl.draw`
- Не синхронизируйте контекст GPU