

TOWELJS: ENGINE 3D EM JAVASCRIPT USANDO ARQUITETURA BASEADA EM COMPONENTES

Aluno: Gabriel Zanluca

Orientador: Dalton S. dos Reis

Roteiro

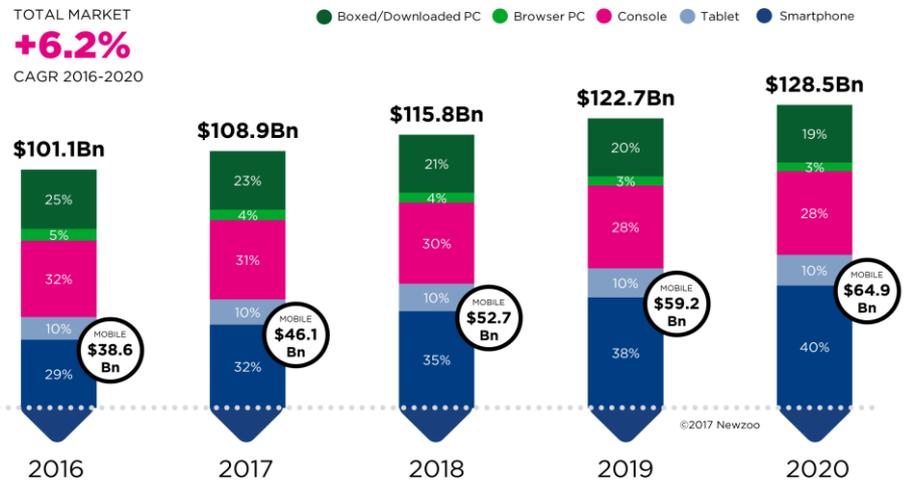
- Introdução
- Objetivos
- Fundamentação teórica
- Trabalho Correlatos
- Requisitos
- Implementação
- Operacionalidade da Implementação
- Resultados e Discussões
- Conclusões e Sugestões

Introdução

- Indústria de jogos em crescimento;
- Uso de motores de jogos;
- Arquitetura baseada em componentes.

2016-2020 GLOBAL GAMES MARKET

FORECAST PER SEGMENT TOWARD 2020



Source: ©Newzoo | Q2 2017 Update | Global Games Market Report
newzoo.com/globalgamesreport

newzoo

Objetivos

Desenvolver um motor de jogos 3D utilizando arquitetura baseada em componentes para facilitar o desenvolvimento de jogos em JavaScript.

Objetivos Específicos

- a) desenvolver componentes dedicados para análise da performance;
- b) analisar a performance do motor desenvolvido comparado com outro motor disponível para uso.

Fundamentação Teórica

- Motor de jogos;
- Arquitetura baseada em componentes;
- WebGL.

Motor de jogos

- Coleção de outros motores;
- Reutilização;
- Ferramentas importantes na construção de jogos.

Arquitetura Baseada em Componentes

- Componentes;
- Princípio da divisão e conquista;
- Desenvolvimento:
 - de componentes x com componentes.

WebGL

- API de gráficos 3D;
- OpenGL ES;
- vertex shader;
- fragment shader.



WebGL(vertex shader)

```
1. attribute vec4 aVertexPosition;
2. uniform mat4 uModelViewMatrix;
3.
4. void main() {
5.     gl_Position = uModelViewMatrix *
6.                 aVertexPosition
7. }
```

WebGL(fragment shader)

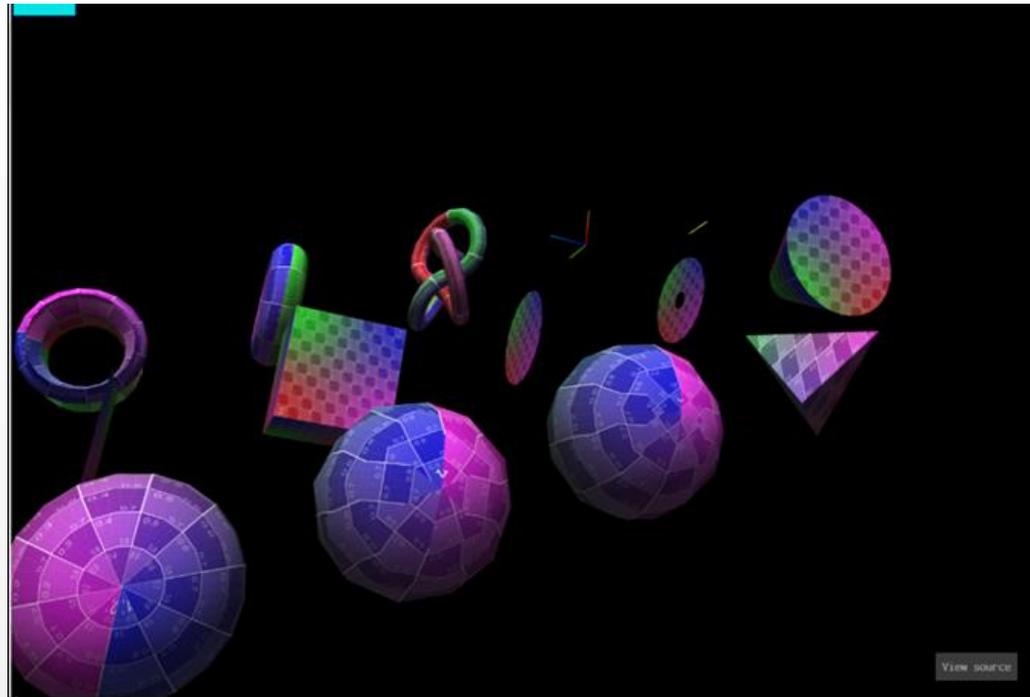
```
1. varying vec4 v_color;  
2.  
3. void main() {  
4.     gl_FragColor = v_color;  
5. }
```

Trabalhos Correlatos

- Three.js
- WebGLStudio.js
- VisEdu-Engine.

THREE.JS

- Biblioteca 3D leve e fácil de usar;
- Possui objetos: cubos, quadrados, círculos, cilindros, esferas, linhas, etc;
- Conta também com recursos para ajudar na criação de animação de objetos.



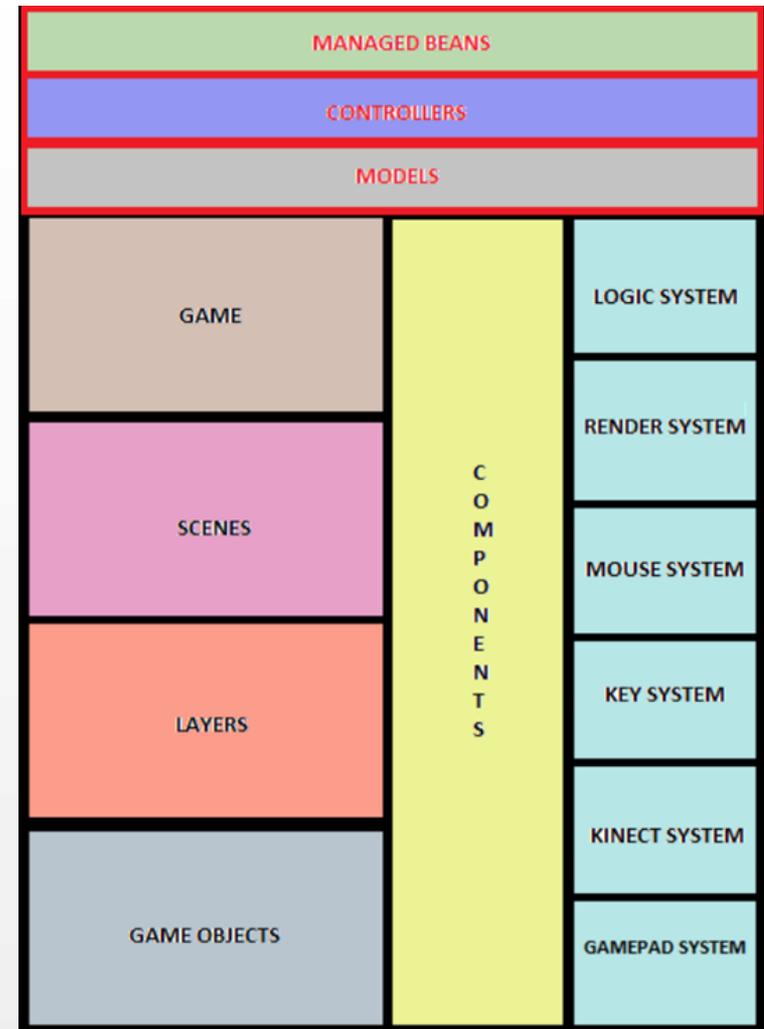
WEBGLSTUDIO.JS

- É um editor de gráficos 3D;
- Utiliza a biblioteca gráfica LiteScene;
- Possui um sistema baseado em componentes.



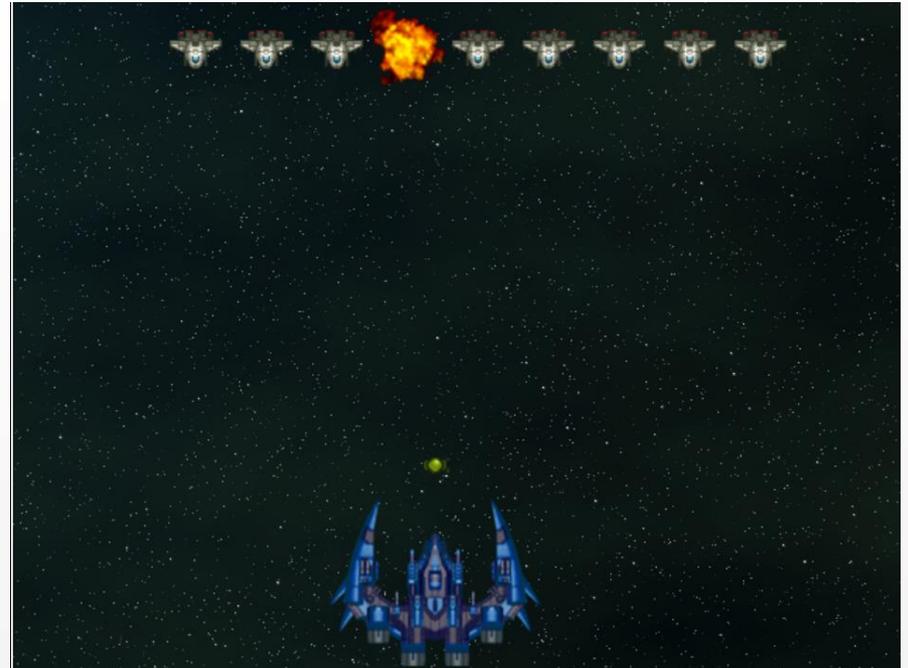
VISEDU-ENGINE

- Arquitetura do motor de jogos é baseada em componentes;
- Trabalha somente com 2D;



VISEDU-ENGINE

- Conta com controle para periféricos comuns como *mouse* e teclado e também suporte para o Kinect e *joystick*.



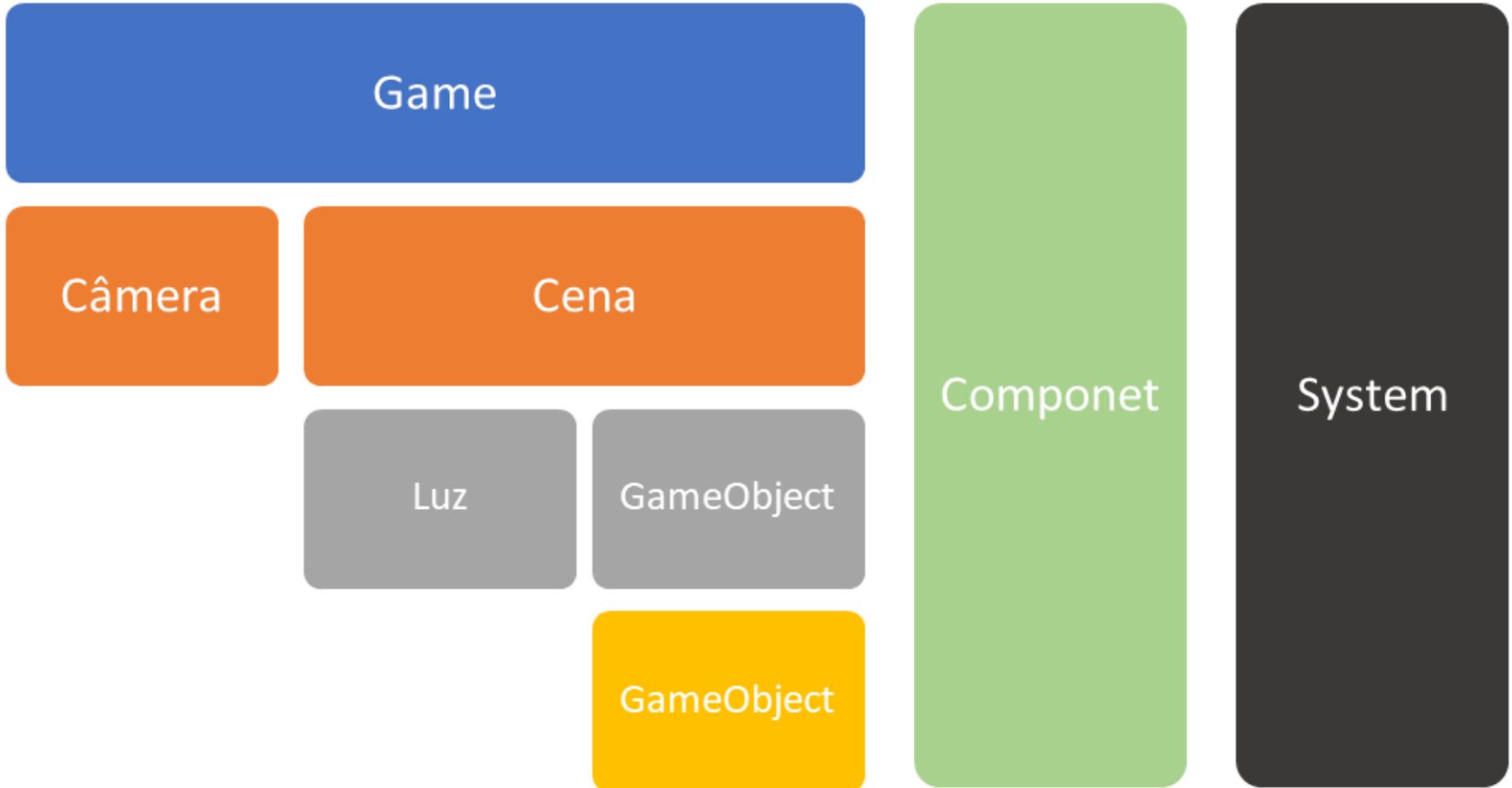
Requisitos Funcionais

- permitir a criação/renderização de objetos gráficos;
- possuir objetos gráficos como cubo e esferas já implementados;
- permitir a criação ou extensão de objetos gráficos já criados;
- permitir a criação de novos componentes;
- implementar a estrutura de grafo de cena;
- possuir componentes para analisar a performance e consumo de recursos;
- contar com ao menos uma opção de câmera sintética;
- possuir diferentes tipos de iluminação disponível para colocar-se em cena;

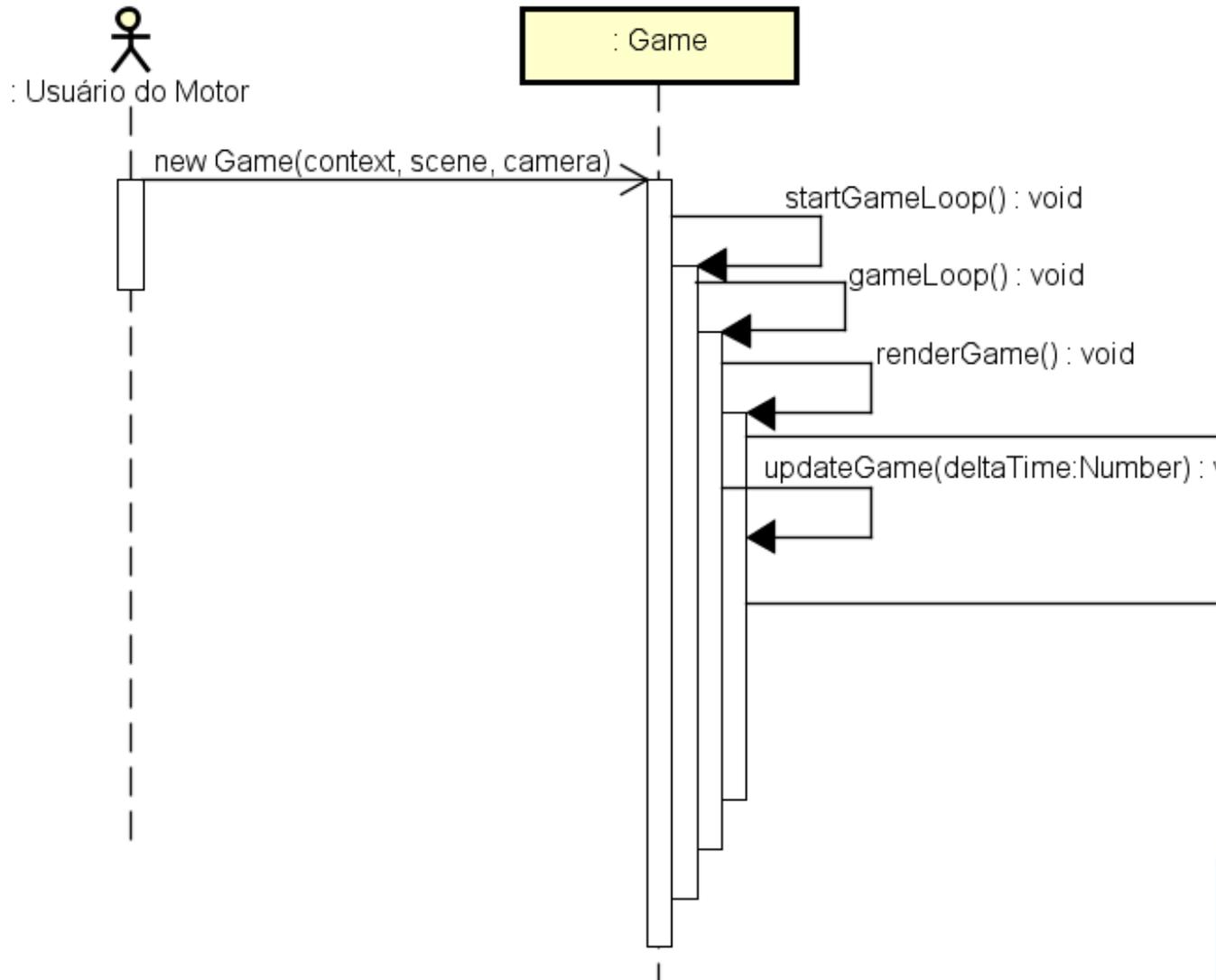
Requisitos não funcionais

- ser implementado utilizando JavaScript;
- utilizar componentes para realizar a renderização dos objetos gráficos;
- ser implementado utilizando a arquitetura baseada em componentes.

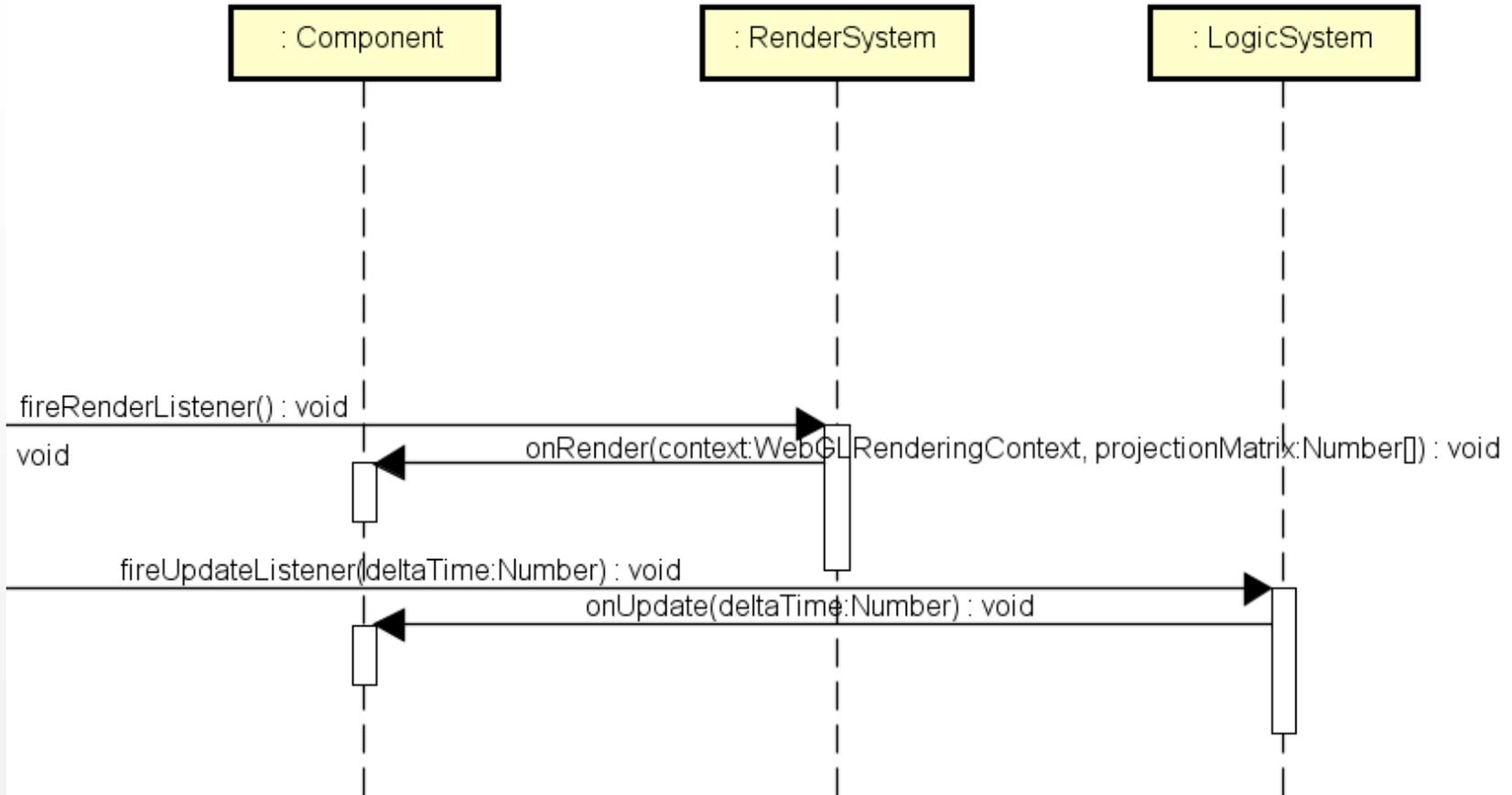
Arquitetura do Motor de Jogos



DS: Loop principal



DS: Loop principal



Implementação (Component)

```
1. class Component {
2.     constructor({ owner }) {
3.         this.__id = JSUtils.generateUUID();
4.         this.__enabled = true;
5.         this.__owner = owner;
6.     }
7.     onKeyDown(keyCode) { }
8.     onKeyUp(keyCode) { }
9.     onKeyPress(keyCode) { }
10.    onBeforeRender(context) { }
11.    onRender(context, projectionMatrix) { }
12.    onUpdate(delta) { }
13.    onLoad() { }
14.    onDestroy() { }
15. }
```

Implementação (GameObject)

```
1.  constructor({ oringin = new Point3D(0, 0, 0), color =
    new Color(), }) {
2.      this.__id = JSUtils.generateUUID();
3.      this.__matrix = mat4.create();
4.      this.__oringin = oringin;
5.      this.__color = color;
6.      this.__listComponents = new ComponentList();
7.      this.__listComponents.addComponent(new
        TranslateComponent({ owner: this }));
8.      this.__listComponents.addComponent(new
        RotateComponent({ owner: this }));
9.      this.__listComponents.addComponent(new
        ScaleComponent({ owner: this }));
10.     this.translate.translation = [oringin.x,
        oringin.y, oringin.z];
11.     this.__children = [];
12. }
```

Implementação (CubeGameObject)

```
1. constructor({ oringin = new Point3D(0, 0, 0), color =  
new Color(), }) {  
2.     super({oringin : point, color});  
3.     this.__listComponents.addComponent(new  
         CubeRenderComponent({owner : this}));  
4.     this.render.onLoad();  
5.     this.render.color = color;  
6. }
```

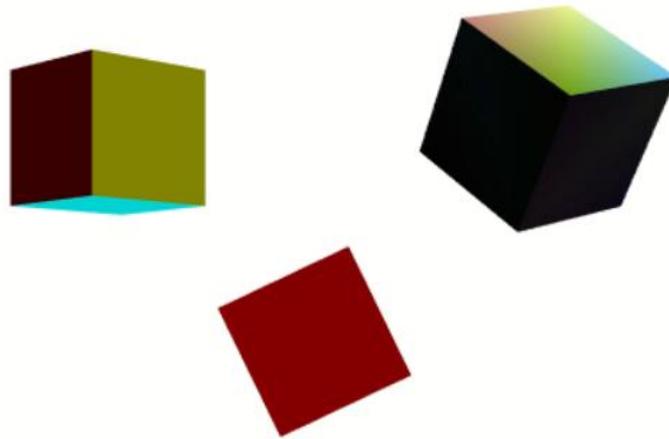
Implementação (Game)

```
1.  startGameLoop() {
2.      let Loop = () => {
3.          this.__requestAnimFrame =
4.              window.requestAnimationFrame(Loop);
5.          this.gameLoop();
6.      };
7.      Loop();
8.      window.addEventListener("keypress",
9.          KeySystem.fireKeyPressListener);
10.     window.addEventListener("keydown",
11.         KeySystem.fireKeyDownListener);
12.     window.addEventListener("keyup",
13.         KeySystem.fireKeyUpListner);
14. }
```

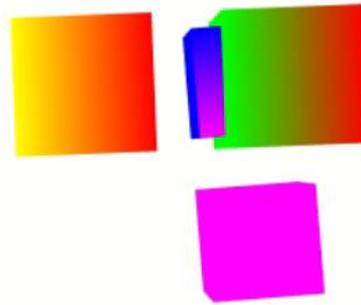
Implementação (Game)

```
1. gameLoop() {  
2.   this.renderGame();  
3.   let now = Date.now();  
4.   let deltaTime = (now - this.__lastUpdateTime) / 1000;  
5.   this.updateGame(deltaTime);  
6.   this.__lastUpdateTime = Date.now();  
7. }
```

Operacionalidade da Implementação



Operacionalidade da Implementação



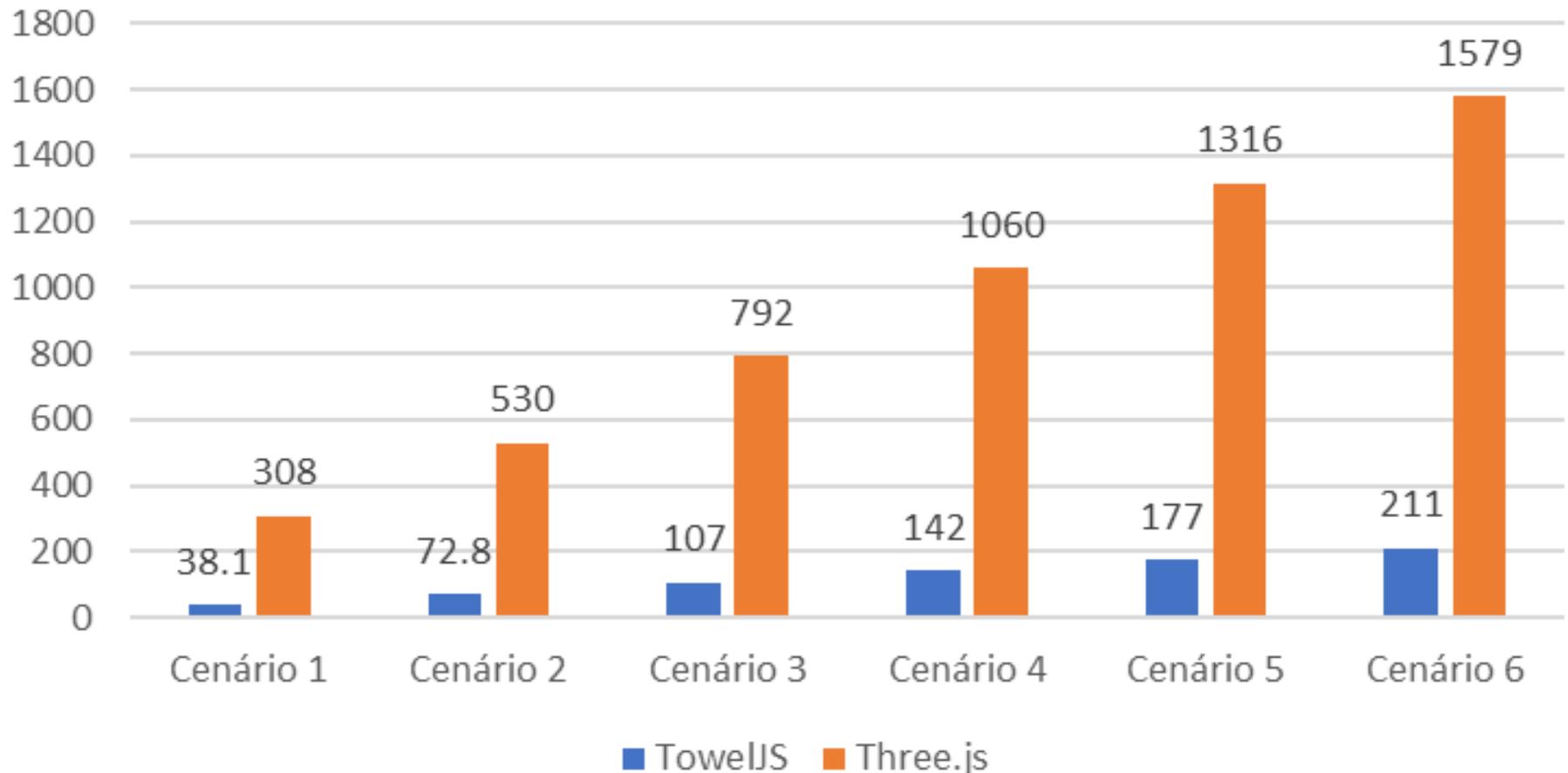
Resultados e Discussões

Cenários	TowelJS	Three.js
Cenário 1	60	60
Cenário 2	60	60
Cenário 3	59	60
Cenário 4	57	60
Cenário 5	50	60
Cenário 6	46	60

valores em FPS

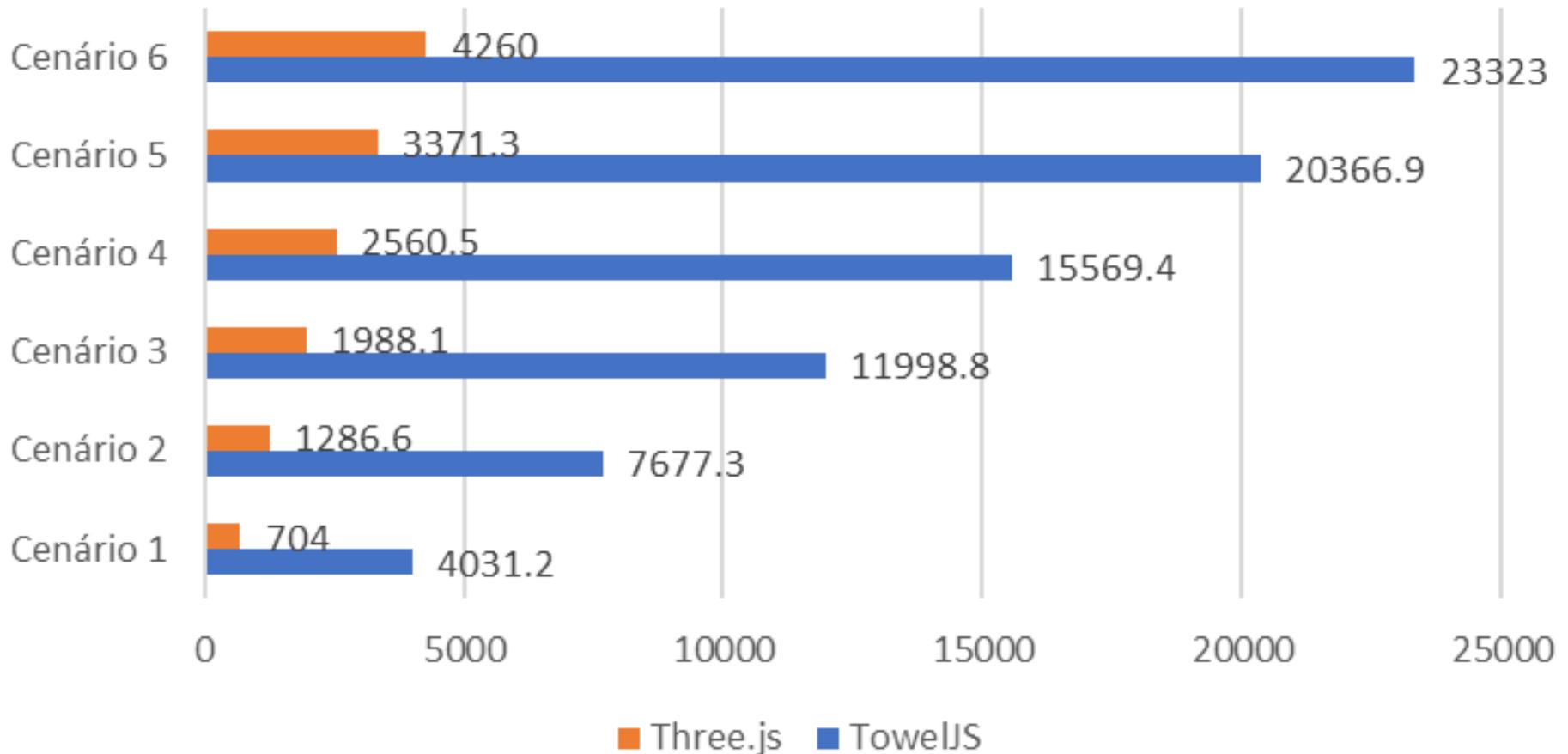
Resultados e Discussões

Consumo de memória (MB)



Resultados e Discussões

Tempo de criação das esferas (ms)



Resultados e Discussões

Características	Three.js	WebGLStudio.js	VisEdu-Engine	TowelJS
implementado em JavaScript	Sim	Sim	Sim	Sim
possuir grafo de cena	Sim	Sim	Não	Sim
sistema baseado em componentes	Não	Sim	Sim	Sim
gráfico em 3D	Sim	Sim	Não	Sim
possui um editor	Sim	Sim	Sim	Não
iluminação	Sim	Sim	Não	Sim
câmera sintética	Sim	Sim	Sim	Sim
seleção de objetos com o mouse	Sim	Sim	Sim	Não

Conclusões

- Utilização de componentes;
- Câmera sintética;
- Luzes;
- Criação de objetos;

Conclusões

- Grafo de cena;
- Poucos objetos;
- Melhoria no desempenho;
- Transformações geométricas confusas.

Sugestões

- permitir a troca de componentes de transformações geométricas e de desenho de objetos em tempo de execução;
- adicionar novos tipos de objetos;
- possuir uma forma de selecionar objetos com o mouse;
- importar objetos feitos em software de terceiros como Blender ou Maya;

Sugestões

- permitir aplicar texturas;
- construir mais testes para análise de performance;
- analisar melhor o uso de componentes;
- fazer teste em outros navegadores e em outros sistemas operacionais;
- construir um editor gráfico para facilitar o uso do motor de jogos desenvolvido.

Demonstração

TOWELJS: ENGINE 3D EM JAVASCRIPT USANDO ARQUITETURA BASEADA EM COMPONENTES

Aluno: Gabriel Zanluca

Orientador: Dalton S. dos Reis