

WebGLStudio

Matheus Filipe dos Santos Reinert

O que é?

Plataforma que permite criar cenas interativas em 3D diretamente no navegador

Permite editar visualmente cenas, programar seus comportamentos e editar shaders

Permite visualizar e alterar o grafo de comportamento sem precisar de código

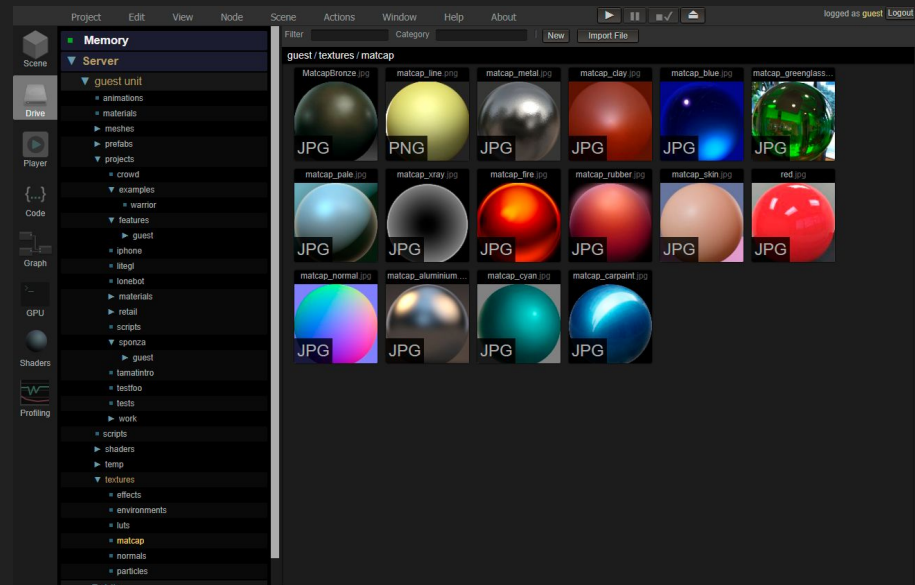
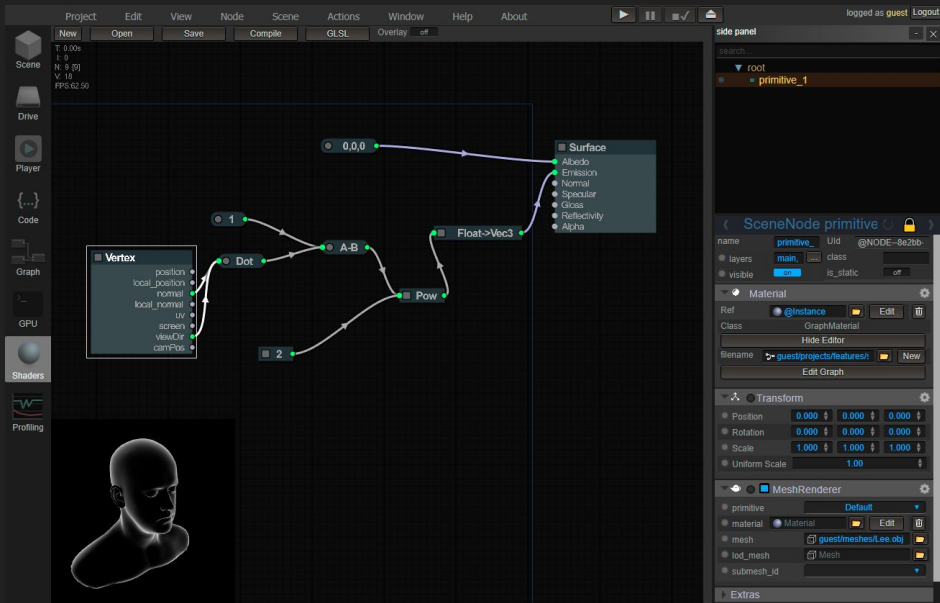
Afinal, o que é WebGL?

WebGL é uma API de javascript que permite utilizar aceleração de hardware para renderizar gráficos 2D e 3D no navegador nativamente.

WebGL conforma com OpenGL ES 2.0 e pode ser usado dentro da tag HTML `<canvas>`

Também existe o WebGL 2, que introduz suporte às funcionalidades do OpenGL ES 3.0

É importante notar que assim como o OpenGL, WebGL é apenas uma especificação, sendo a implementação concreta deixada para os navegadores



Cenas

Uma cena é uma coleção de objetos (texturas, câmeras, luzes, etc.) que juntos compõem um ambiente virtual

Podemos pensar que uma cena é um estúdio de cinema, onde possuímos a ambientação, objetos, atores e câmeras

SceneTree e SceneNode

Enquanto um SceneNode representa um objeto na cena, a SceneTree representa a cena em si

Todo nó pode conter um mais nós filhos, similar a como o DOM funciona

A SceneTree possui um nó raiz (`scene.root`) que todos os nós da cena são filhos

ProjectEditViewNodeSceneActionsWindowHelpAbout

SceneDrivePlayerCodeGraphGPUShadersProfiling

0: perspective

search...

▼ root

- ▶ sponza
 - light_1
 - FX
 - primitive_1
 - ▶ character
 - primitive_2
 - probe
 - camera_corridor
 - cache_day
 - ▼ cache_night
 - ceiling light

SceneNode character

visible on is_static off

Material

RefMaterialEdit

Create Material

Transform

Position657.398410.290-442.264

Rotation0.000-90.4000.000

Scale1.0001.0001.000

Uniform Scale1.00

PlayAnimation

animationguest/animations/waEdit

takedefault

root_nodet_e_@

playback_speed1.00

modeloop

playingon

current_time0.00s

blend_time0.00s

Extras

Add component

Add ScriptAdd Graph

Scene loaded

ResourcesTimeline+

Options#animations_walkTakedefaultDuration1.17sCurrent0.33s

- mixamorig_Hips/Transform/data trans10 [p]
- mixamorig_Spine/Transform/data trans10 [p]
- mixamorig_Spine1/Transform/data trans10 [p]
- mixamorig_Spine2/Transform/data trans10 [p]
- mixamorig_Neck/Transform/data trans10 [p]
- mixamorig_Head/Transform/data trans10 [p]
- mixamorig_LeftShoulder/Transform/data trans10 [p]
- mixamorig_LeftArm/Transform/data trans10 [p]
- mixamorig_LeftFoot/Transform/data trans10 [p]

Components

Component é um elemento que adiciona comportamento e representação visual a um nó

Cameras, Lights, MeshRenderers, etc. são todos components que são anexados a um nó

Esse sistema faz com que o motor gráfico seja dinâmico, podendo compor novos comportamentos somente anexando novos components a um nó

Components importantes

Transform: modifica posição, rotação e escalonamentos dos objetos na cena

Camera: dita de onde e como a cena vai ser renderizada

MeshRenderer: renderiza algo na cena

Light: ilumina a cena

Scripting

Um script é um código que adiciona comportamento a um nó ou cena no geral

Podemos fazer com que nós interajam entre si, ou que reajam a eventos como o pressionar de uma tecla ou clique do mouse

São expostas variáveis globais para o programador utilizar

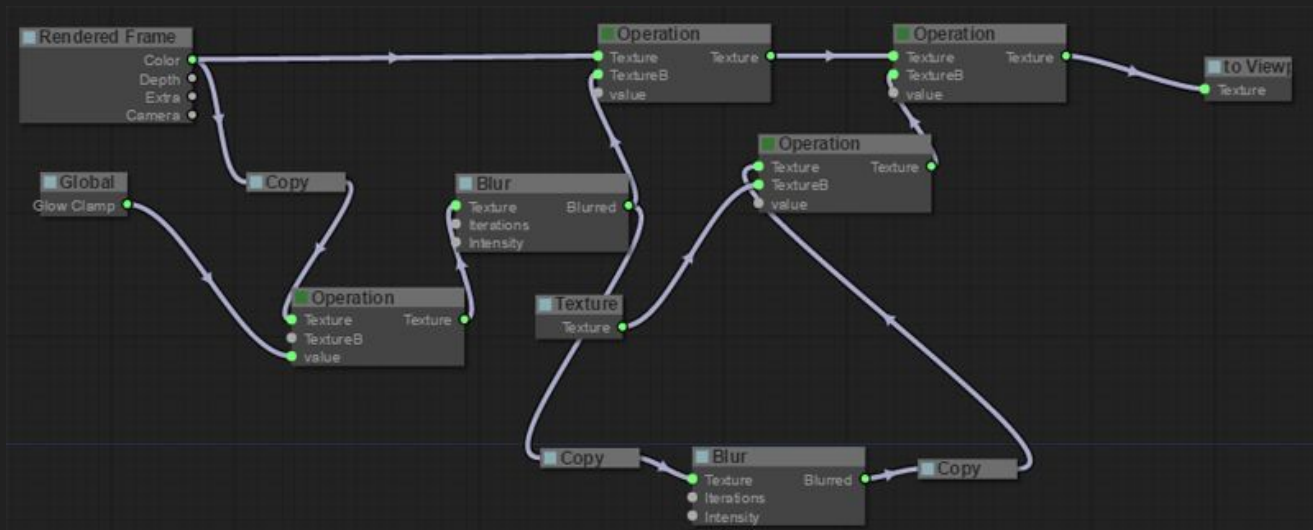
```
1  const rotationSpeed = 0.01;
2
3  function update() {
4      node.rotation.x += rotationSpeed;
5      node.rotation.y += rotationSpeed;
6      node.rotation.z += rotationSpeed;
7  }
8
9  function render() {
10     // Não é necessário fazer nada aqui
11 }
12
13 function animate() {
14     requestAnimationFrame(animate);
15     update();
16     render();
17 }
18
19 animate();
```

Graphs

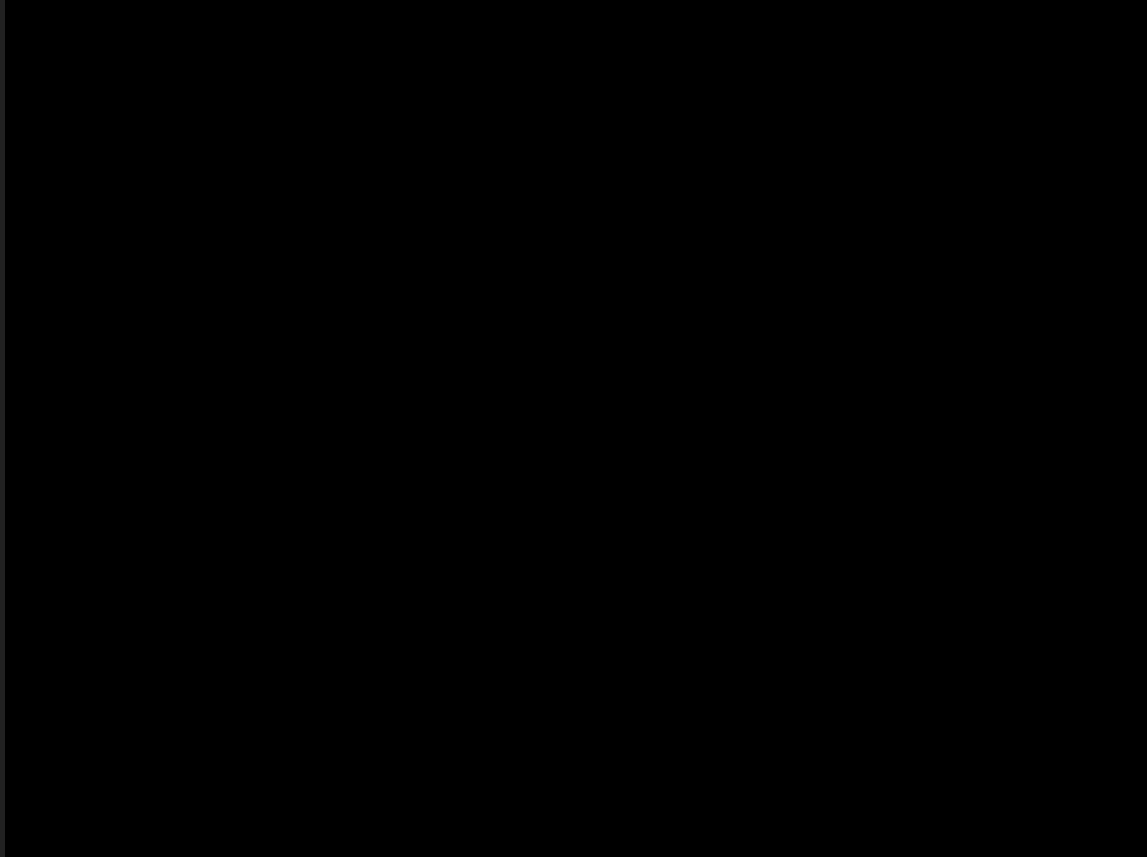
Funcionalidade que permite ao usuário controlar comportamento e a renderização

Um grafo possui nós representando os elementos da cena que pode se conectar a outros nós para transferir dados ou ações entre eles

Também é possível controlar os efeitos de pós-processamento da renderização utilizando o component `FXGraphComponent`



DEMO



Referências

<https://github.com/jagenjo/litescene.js/tree/master/guides>

https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API