Universidade de Aveiro

departamento de electrónica, telecomunicações e informática

47053- Computação Visual

Puzzle

Implementação em WebGL de um Puzzle

8240 - mestrado integrado em engenharia de computadores e telemática

António Rafael da Costa Ferreira
NMec: 67405

Rodrigo Lopes da Cunha
NMec: 67800

Docente: Joaquim João Estrela Ribeiro Silvestre Madeira

Dezembro de 2015
2015-2016

Conteúdos

1 Introdução

O trabalho proposto para o projeto da unidade curricular de Segurança é um IEDCS: Identity Enabled Distribution Control System. Para o efeito foi necessário implementar uma Ebook Webstore, um WebService e um Player de reprodução dos Ebooks em formato de texto.

O objetivo deste sistema é garantir a máxima e possível segurança do serviço, utilizando os conhecimentos adquiridos na unidade curricular de Segurança. Para isso são necessários vários processos como por exemplo, a utilização de certificados HTTPS, a cifragem de todo o material existente, derivação de chaves e registo de utilizadores.

O relatório reflete todos os passos e decisões tomadas na criação do sistema, assim como uma análise ao que foi mostrado na primeira apresentação e decisões que se tomaram depois desta, tecnologias utilizadas, descrição dos vários processos existentes e conclusão.

2 JavaScript e WebGL

2.1 Arquitectura da implementação

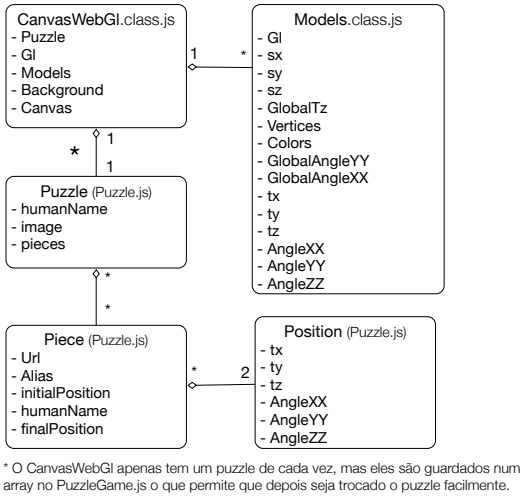

Figura 1: Diagrama da implementação desenvolvida

Na implementação desenvolvida procurou-se uma solução que permitisse reutilizar o código e instanciar quantas peças e puzzles fosse preciso. Para isso criou-se uma class Models, que tem como atributos os que estão descritos em cima, que instância um modelo, independente dos outros que irá fazer uso das translações, rotações e outros métodos usados durante as aulas práticas. Para isso, esta class irá, no seu construtor instanciar uma única vez dois buffers, um chamado *triangleVertexPositionBuffer* e outro *triangleVertexColorBuffer*, de resto, o *initBuffers* é sempre chamado sempre que for feito um *drawScene* para os arrays de buffers serem atualizados.

Já na class *CanvasWebGl*, é onde o puzzle instância todos os modelos (peças), aplica as translações globais e independentes, é desenhada a cena, inicializado o modelo de fundo e inicializado o WebGl.

Para alimentar a class *CanvasWebGl*, foi criada uma class *Puzzle* que tem todos os atributos necessários para instanciar um puzzle. O puzzle irá ter o seu *humanName* que será apresentado ao utilizador, a *image* que é a imagem final da solução do puzzle, e as *pieces* que são as várias peças do puzzle.

A peça, terá o URL sendo este onde será obtido a lista de vértices e de cores para a peça. O alias é o usado para identificar a peça, tem de ser único para todas as peças existentes no Puzzle, o *initialPosition* que identifica a posição inicial da peça e o *finalPosition* que identifica a posição final da figura.

2.2 JSON

Para uma melhor definição de todos os puzzles usados no jogo foi criado um ficheiro JSON onde é possível criar todas as peças de cada puzzle e definir todos os atributos anteriormente detalhados. Este ficheiro é carregado inicialmente sempre que o jogo é iniciado, apresentando assim ao utilizador a lista de puzzles e inicializando o jogo com o puzzle inicial.

```
{
  "puzzles" : [
    {
      "humanName": "Level 1",
      "image" : "img/puzzles/puzzle1.png",
      "pieces" : [
        {
          "alias": "triangulo",
          "url": "modelos/trianguloBlue.txt",
          "humanName": "Triangulo Blue",
          "initialPosition" : {
            "tx": 0.2,
            "ty": 0.4,
            "tz": 0.5,
            "angleXX": 225,
            "angleYY": 45,
            "angleZZ": 45
          },
          "finalPosition" : {
            "tx": 0,
            "ty": 0,
            "tz": 0,
            "angleXX": 0,
            "angleYY": 0,
            "angleZZ": 0
          }
        }
      ]
    }
  ]
}
```

2.3 JSON