

<https://eventos.utfpr.edu.br//sicite/sicite2019>

Implementação de módulo de serialização de objetos gráficos para impressão 3d

Graphic objects serialization module for 3d printing

RESUMO

Júlio Dutra Nicácio
julio_nicacio@hotmail.com
Universidade Tecnológica
Federal do Paraná, Santa
Helena, Paraná, Brasil

Davi Marcondes Rocha
davirocha@utfpr.edu.br
Universidade Tecnológica
Federal do Paraná, Santa
Helena, Paraná, Brasil

Dado aos avanços no campo da impressão 3D, por sua capacidade de auxiliar em várias áreas do conhecimento, é importante que esse processo seja estudado e aprimorado, com o objetivo de criar novas ferramentas e possibilidades para criação de objetos no plano virtual se tornarem palpáveis no plano material. Este trabalho teve por finalidade a implementação do módulo de serialização de objetos gráficos tridimensionais em um framework multiplataforma, para posterior impressão em equipamentos de prototipagem rápida (impressora 3D). Para isso foi utilizado a linguagem de programação C++, a API OpenGL e a biblioteca multimídia SDL2.

Embora não desenvolvido completamente, o algoritmo de serialização encontra-se em estado funcional, mostrando que o conceito de leitura/ gravação de objetos 3d é passível de ser implementado no framework sobre SDL2, possibilitando em trabalhos futuros a real impressão tridimensional de artefatos através de uma máquina de prototipação rápida.

PALAVRAS-CHAVE: Framework. OpenGL. Prototipação rápida.

Recebido: 19 ago. 2019.

Aprovado: 01 out. 2019.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição Internacional. 4.0



ABSTRACT

Bearing in mind the advances in the field of 3D printing, for the capacity to assist in any areas of human knowledge, it's very important wicth this process must be study and evolve with the mission to make real the build of new tools and possibilities to virtual objects become true in material world. This research aims the implementation of serialization module of three-dimensional graphical objects in multiplatform framework for a subsequent printing in 3D printer. Was used the C++ programming language,OpenGL API and multimedia library SDL2. Though not completely developed, the serialization algorithm finds in functional state, showing the concept of read/record in 3D object recording is susceptible of be implemented in a SDL2 based framework, making available in a future work a real three-dimensional print of artefacts through a fast prototype machine.

KEYWORDS: Framework. OpenGL. Fast prototype.

INTRODUÇÃO

A Computação Gráfica (CG) é um campo da Ciência da Computação a qual se dedica ao estudo e aperfeiçoamento de métodos e algoritmos para a síntese de imagens através de ferramentas computacionais e que, atualmente, estão presentes em diversas áreas do conhecimento (MANSSOUR E COHEN, 2006).

Segundo Santo et al. (2001), é constante a presença da CG na vida cotidiana, permeando desde sistemas CAD (Computer-Aided Design), usados para projetar utilidades de uso diário até o desenvolvimento de jogos e filmes de animação.

A modernização de recursos tecnológicos vem ocorrendo em ritmo acelerado, a exemplos, podem ser observadas a computação gráfica tridimensional de alta fidelidade, elaboração de objetos digitais em sistemas computacionais, realidades virtuais, criação jogos interativos e impressões tridimensionais realizadas com máquinas de prototipação rápida (WEN, 2009).

De acordo Wen (2016), apesar de ser resultado de pesquisas realizadas desde na década de 1980, recentemente, impressoras 3D se tornaram populares, devido à melhoria de qualidade e redução de preço de aquisição. Estes equipamentos permitem a produção física de estruturas virtuais, projetadas através de recursos digitais fornecidos pela CG.

Precisamente a prototipagem, modelagem e modelos em escalas, fazem parte de aplicações usadas no processo aditivo das impressoras 3D. Entretanto, estes campos são apenas uma pequena amostra da capacidade desta tecnologia, uma vez que a impressão 3D já beneficia grandes indústrias, como aeroespacial, automobilística, médica, ortodôntica, entre outras (DABAGUE, 2014).

Este trabalho teve por finalidade a implementação do módulo para serialização de objetos gráficos tridimensionais em um framework sobre a biblioteca SDL2 multiplataforma para posterior impressão em equipamentos de prototipagem rápida (impressora 3D).

METODOLOGIA

Para a elaboração do projeto utilizou-se um conjunto de tecnologias, como GNU C++ Compiler (versão 6.3), CMake (ferramenta de construção que verifica se as bibliotecas, compilador e arquivos de inclusão descritos no projeto estão disponíveis na plataforma onde o CMake está sendo executado) Make (que também descreve como o projeto deve ser compilado) e Git (sistema de versionamento de arquivos largamente utilizado para registrar mudanças no desenvolvimento de um projeto).

Devido a fácil leitura, uma vez que permite separar seu código em partes e arquivos para melhor estruturação, o trabalho foi escrito na linguagem de programação C++. Utilizou-se também a OpenGL (Open Graphics Library), biblioteca de rotinas gráficas e de modelagem, bidimensional e tridimensional portátil e rápida que, possibilita desenvolver programas interativos e gerar

imagens de cenas tridimensionais com alto realismo (MANSSOUR e COHEN, 2006).

A abstração do acesso aos periféricos do computador foi realizada utilizando a biblioteca multimídia SDL2 (*Simple DirectMedia Layer*), que teve como principal objetivo fornecer acesso a baixo nível ao áudio, teclado, mouse, joystick e gráficos via OpenGL e Direct3D.

RESULTADOS E DISCUSSÕES

Após a elaboração da primeira parte do algoritmo de serialização, pôde-se realizar a leitura de uma primitiva gráfica (cubo) e extração das informações dos descritores de um objeto tridimensional, onde foi lido o número de vértices do objeto 3D, as normais e suas faces. Estes dados compõem um arquivo em formato “.obj” (A Figura 01).

Figura 1 – Estrutura do arquivo “.obj”

```

5 v 1.000000 -1.000000 -1.000000
6 v 1.000000 -1.000000 1.000000
7 v -1.000000 -1.000000 1.000000
8 v -1.000000 -1.000000 -1.000000
9 v 1.000000 1.000000 -0.999999
10 v 0.999999 1.000000 1.000001
11 v -1.000000 1.000000 1.000000
12 v -1.000000 1.000000 -1.000000
13 vn 0.0000 -1.0000 0.0000
14 vn 0.0000 1.0000 0.0000
15 vn 1.0000 0.0000 0.0000
16 vn -0.0000 -0.0000 1.0000
17 vn -1.0000 -0.0000 -0.0000
18 vn 0.0000 0.0000 -1.0000
19 usemtl Material
20 s off
21 f 1//1 2//1 3//1 4//1
22 f 5//2 8//2 7//2 6//2
23 f 1//3 5//3 6//3 2//3
24 f 2//4 6//4 7//4 3//4
25 f 3//5 7//5 8//5 4//5
26 f 5//6 1//6 4//6 8//6

```

Fonte: Autoria própria (2019).

As linhas que começam com “#” são comentários. As linhas que começam com “v” e “vn” são vértices e as normais respectivamente, contendo três coordenadas cada. E as linhas que começam com “f” são as faces, contendo três

coordenadas que dizem qual face está ligada a qual separada por “//”. A Figura 2 apresenta o algoritmo desenvolvido.

Figura 2 – Leitura do objeto tridimensional

```

37     std::string type;
38     std::string name;
39
40     while(file >> type) {
41         if (type == "o") {
42             file >> name
43             file2 << name;
44         }
45         if (type == "v") {
46             file >> x >> y >> z;
47             file2 << x << y << z;
48
49             glm::vec3 v(x, y, z);
50             vertices.push_back(v);
51         } else if (type == "vn") {
52             file >> x >> y >> z;
53             file2 << x << y << z;
54
55             glm::vec3 v(x, y, z);
56             normals.push_back(v);
57         } else if (type == "f") {
58             file >> x >> y >> z;
59             file2 << x << y << z;
60
61             glm::vec3 v(x, y, z);
62             faces.push_back(v);
63         }
64     }
65
66     obj.name = name;
67     obj.normals = normals;
68     obj.vertices = vertices;
69     obj.faces = faces;
70
71     return obj;
72 }

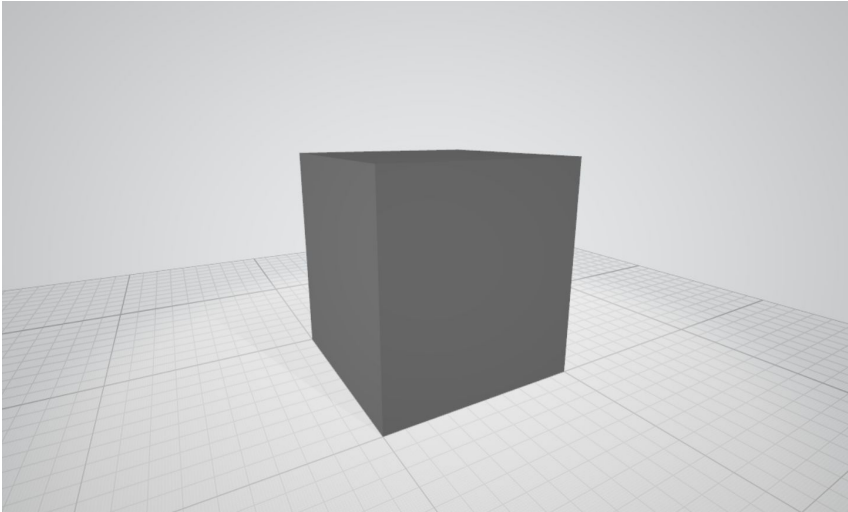
```

Fonte: Autoria própria (2019).

Após receber o endereço de um objeto virtual, cria-se três vetores, no qual cada um representará um elemento do arquivo, sendo um para os vértices, outro para as normais e outro para as faces. Em seguida o arquivo é lido e, na linha que começar com “o”, é obtido o nome do arquivo. Nas linhas que começam com “v”, “vn” e “f” são capturadas as coordenadas dos vértices, normais e faces, respectivamente. Ao final é realizado os apontamentos das variáveis que foram lidas para as variáveis da estrutura do objeto.

A Figura 03 apresenta o objeto renderizado (cubo) pelo framework após leitura do arquivo “.obj”.

Figura 3 –Arquivo “.obj” renderizado.



Fonte: Autoria própria (2019).

CONCLUSÕES

O estudo realizado mostrou que, embora não desenvolvido completamente o algoritmo, o mesmo encontra-se em estado funcional e o conceito de leitura e gravação de objetos tridimensionais é passível de ser implementado no framework em questão, possibilitando em trabalhos futuros a real impressão tridimensional de artefatos através de uma máquina de prototipação rápida.

REFERÊNCIAS

- COHEN, M., MANSSOUR, I. H. OpenGL: uma abordagem prática e objetiva. Novatec, 2006. 478p.
- DABAGUE, L. A. M. O Processo De Inovação No Segmento De Impressoras 3d. Trabalho e Conclusão de Curso. Setor de Ciências Sociais Aplicadas, Universidade Federal do Paraná – UFPR. 2014
- SANTOS, E. T.; ZUFFO, M. K.; NETTO, M. L.; DE DEUS LOPES, R. Computação Gráfica: Estado da arte e a pesquisa na USP. Simpósio Nacional de Geometria Descritiva e Desenho Técnico, v. 15, p. 1333-1363, 2001.
- WEN, C. L. “Homem Virtual”. In: MARTINS, Milton de Arruda et al. (orgs.). Clínica Médica. Barueri, SP, v.1, p.988-991, 2009.



IX SEMINÁRIO DE EXTENSÃO E INOVAÇÃO
XXIV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA E TECNOLÓGICA
11 a 13 de Novembro | Pato Branco - PR



WEN, C. Lung. Homem Virtual (Ser Humano Virtual 3D): A Integração da Computação Gráfica, Impressão 3D e Realidade Virtual para Aprendizado de Anatomia, Fisiologia e Fisiopatologia. Rev. Grad. USP, v.1, n.1, p.7-16, 2016.