



Universidade Federal da Bahia - UFBA

Instituto de Matemática - IM

Departamento de Ciência da Computação - DCC

Curso de Bacharelado em Ciência da Computação

MATA65 - Computação Gráfica

Período: 2017.1

Data: 23/05/2016.

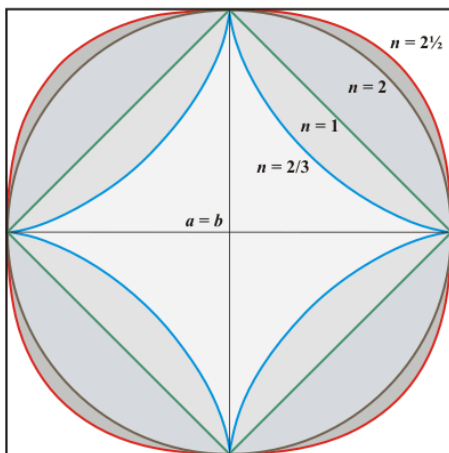
Prof. Antonio L. Apolinário Junior

Estagiário Docente: Rafaela Alcantara

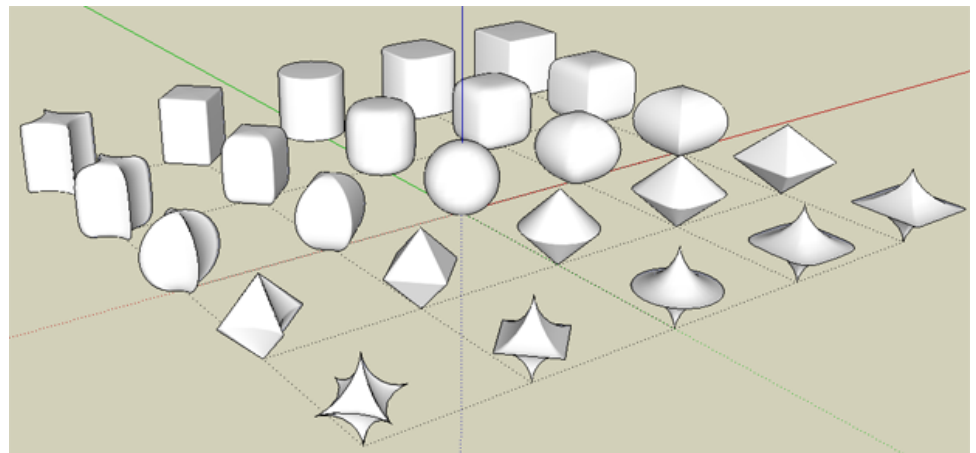
Atividade 2 - Objetos 3D

Motivação:

Uma das formas mais simples de se criar um objeto 3D para visualização em CG é construir uma expressão matemática que descreva sua forma [1]. Um exemplo desse tipo de abordagem pode ser visto na Figura 1. Nessa figura podemos ver uma forma bastante conhecida, na Figura 1(a) uma superelipse [2] e na Figura 1(b) um superelipsoide [3]. Ambos são generalizações da forma elíptica para 2D e 3D.



(a)



(b)

Figura 1 - Generalização da forma elíptica: (a) em 2D uma superelipse; (b) em 3D, um superelipsoide.

Um elipsóide pode ser gerado matematicamente a partir de sua forma implícita:

$$f(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \left[\left| \frac{x}{r_x} \right|^{2/n_2} + \left| \frac{y}{r_y} \right|^{2/n_2} \right]^{n_2/n_1} + \left| \frac{z}{r_z} \right|^{2/n_1}$$

onde \mathbf{r}_x , \mathbf{r}_y , e \mathbf{r}_z são, respectivamente os raios nas direções \mathbf{x} , \mathbf{y} e \mathbf{z} ; os valores de \mathbf{n}_1 e \mathbf{n}_2 definem os expoentes que regulam efetivamente a forma geométrica do superelipsoide.

Outra forma de representar a mesma forma geométrica é através da sua formulação paramétrica:

$$\begin{aligned}x &= r_x \cos^{n1}(\theta) \cos^{n2}(\beta) \\y &= r_y \cos^{n1}(\theta) \sin^{n2}(\beta) \\z &= r_z \sin^{n1}(\theta) \\ \text{where } \frac{-\pi}{2} \leq \theta \leq \frac{\pi}{2}, -\pi \leq \beta \leq \pi \\ \text{and } 0 < n1, n2 < \infty\end{aligned}$$

Os parametros que regulam a forma são os mesmos da formulação implícita, com **phi** e **beta** sendo os parametros, representando angulos.

Outra forma de construção semelhante é o supertoroide [4]. Sua forma pode ser vista na Figura 2.

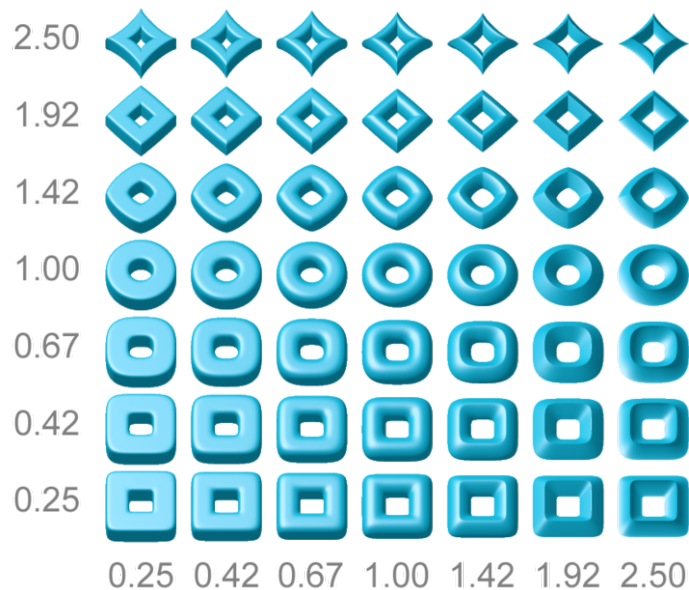


Figura 2 - Generalização da forma toridal: um supertorus.

Tal como a superelipse, sua forma geométrica pode ser descrita por uma formulação implícita:

$$z^2 + (\sqrt{x^2 + y^2} - a)^2 - b^2 = 0$$

Ou na sua forma paramétrica:

$$\begin{aligned}x &= \cos^{n1}(\theta) (r0 + r1 \cos^{n2}(\phi)) \\y &= \sin^{n1}(\theta) (r0 + r1 \cos^{n2}(\phi)) \\z &= r1 \sin^{n2}(\phi)\end{aligned}$$

onde os valores de **r0** e **r1** correspondem ao raio da seção circular e o raio do genus, respectivamente, e **n1** e **n2** regulam a forma do supertorus, como na Figura 2.

A atividade:

Desenvolver uma aplicação em **Three.JS** [5] e *Javascript* [6] que mostre as duas formas, um superelipsoide ou um supertorus.

Especificações:

1. Cada forma deve ser apresentada individualmente e ocupar o máximo possível do canvas;
2. A construção das formas deve utilizar o suporte do Three.JS para a estrutura de dados de uma malha poligonal;
3. A visualização deve ser feita no formato wireframe;
4. Deve ser possível visualizar a mudança dos parametros que definem a forma dos 2 objetos de forma automática;
5. As cores das duas formas devem variar ao longo da superfície e ao longo da animação.

DESAFIO: Defina duas tonalidades que deverão variar ao longo das formas, semelhante ao que acontece na Figura 3 (sem considerar a variação de intensidade luminosa nesse momento).

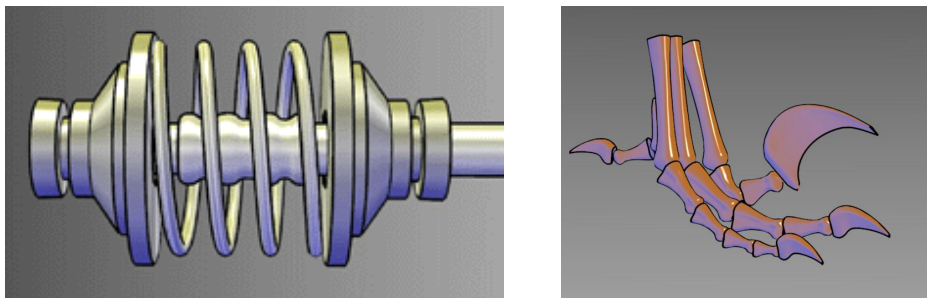


Figura 3: Uso de 2 tons para geração de uma paleta de cores em renderização não fotorealística. fonte: [6]

Entrega da atividade:

- O trabalho deverá ser submetido **somente** via **Moodle**, respeitando a data e hora limite para entrega. Em caso de qualquer problema de arquivos corrompidos ou similar, o trabalho será considerado como não entregue. Portanto, verifique bem o que for entregar!!
- A entrega no **Moodle** deve ser feita em **um único arquivo compactado (.tgz, .zip ou .rar) contendo um subdiretório com seu nome e dentro deste todos os arquivos necessários para a execução do seu código. Na falta de algum arquivo (libs, scripts, modelos, texturas, etc.), uso de caminhos absolutos, ou qualquer outra “falha” que necessite da edição do seu código fonte, a atividade será desconsiderada!!**
- A cooperação entre alunos é considerada salutar. No entanto, atividades com alto grau de similaridade serão tratadas como plágio, o que resultará em avaliação **zero** para **todos** os envolvidos.
- Qualquer dúvida, **não suponha** procure o professor¹ ou a estagiária² para esclarecimentos.

¹ antonio.apolinario@ufba.br

² rafa.alcantara23@gmail.com

Referências Bibliográficas:

- [1] Angel, Edward. **Interactive Computer Graphics - A top-down approach with WebGL**, 7th Editio. Addison-Wesley. 2014.
- [2] **Superellipse**, disponível em: <https://en.wikipedia.org/wiki/Superellipse>, acessado em 23/05/2017.
- [3] **Superellipsoid**, disponível em: <https://en.wikipedia.org/wiki/Superellipsoid>, acessado em 23/05/2017.
- [4] **Supertoroid**, disponível em: <https://en.wikipedia.org/wiki/Supertoroid>, acessado em 23/05/2017.
- [5] Dirksen, Jos. **Learning Three.js: the JavaScript 3D library for WebGL**. 2nd Edition. Packt Publishing Ltd, 2015.
- [6] Brown, Ethan. **Learning JavaScript: JavaScript Essentials for Modern Application Development**, 3rd Edition, O'Reilly, 2016.
- [7] Gooch, A., Gooch, B., Shirley, P., & Cohen, E. **A non-photorealistic lighting model for automatic technical illustration**. In Proceedings of the 25th annual conference on Computer graphics and interactive techniques (pp. 447-452). ACM. 1988.