



www.gdp.dcc.ufrj.br/urge

URGE: Motor para o Desenvolvimento de Jogos Eletrônicos



Alexandre Abdalla, Rodrigo de Toledo e Vitor Maia
DCC - UFRJ

Motivação

A proposta da URGE baseia-se em dois objetivos: completude e simplicidade. A ferramenta reúne várias técnicas de computação gráfica aplicadas em jogos eletrônicos, simulação física em tempo real e teleprocessamento em redes, e trabalha em computadores de baixo desempenho. A engine foi adaptada para programadores sem experiência em desenvolvimento de jogos. É uma ferramenta gratuita, desenvolvida por alunos de graduação em Ciência da Computação. Abaixo, um exemplo de código.

```
URGE_BEGIN
{
    gimme_window(800,600);

    FirstPersonCamera player;
    Scenario scenario;
    Generic enemy;
    Sky sky;
    Light light;
    Terrain terrain;
    Ocean ocean;
    FireBall fireball;

    ocean.loadNormalMap("media/water/wave_n.png");
    ocean.color(5,60,120);
    ocean.scale(500);

    player.position(0,20,0);

    enemy.load("media/models/mask/mask.md2");

    terrain.load("media/terrain/heightmap.jpg");
    terrain.loadTexture("media/terrain/heightmap_diffuse.jpg");

    light.directional();
    light.direction(-1,1,-1);

    sky.loadTexture("media/sky/cloudy");

    scenario.insert(player);
    scenario.insert(sky);
    scenario.insert(enemy);
    scenario.insert(terrain);
    scenario.insert(light);
    scenario.insert(ocean);

    scenario.prepare();

    do
    {
        if (Keyboard::hit(Keyboard::ESC)) break;

        // Fazendo enemy "seguir" player
        enemy.velocity() += (player.position() - enemy.position()).normalize()*5.0;
        enemy.direction() = enemy.velocity();

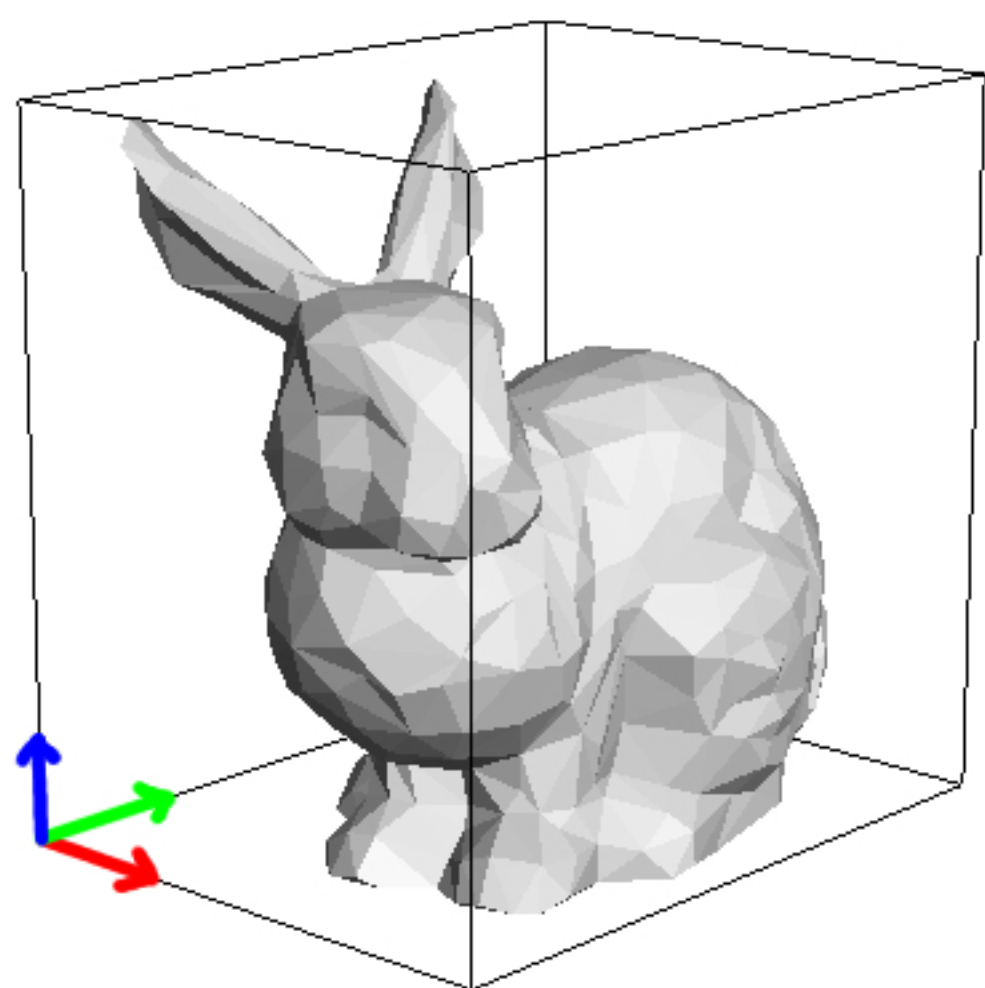
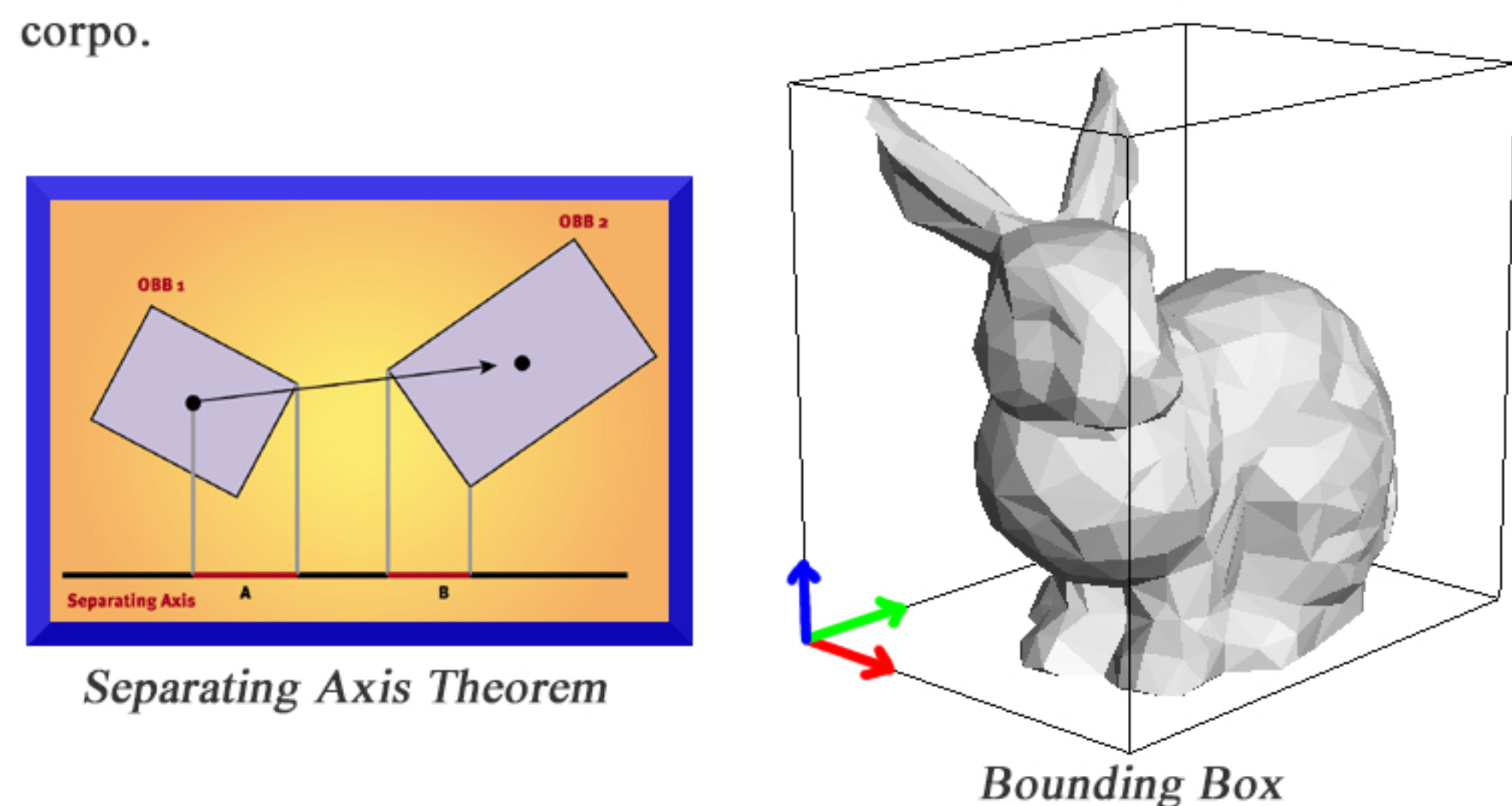
        scenario.update();

        next_frame();
    }while(true);

    thanks_byebye();
}
URGE_END
```

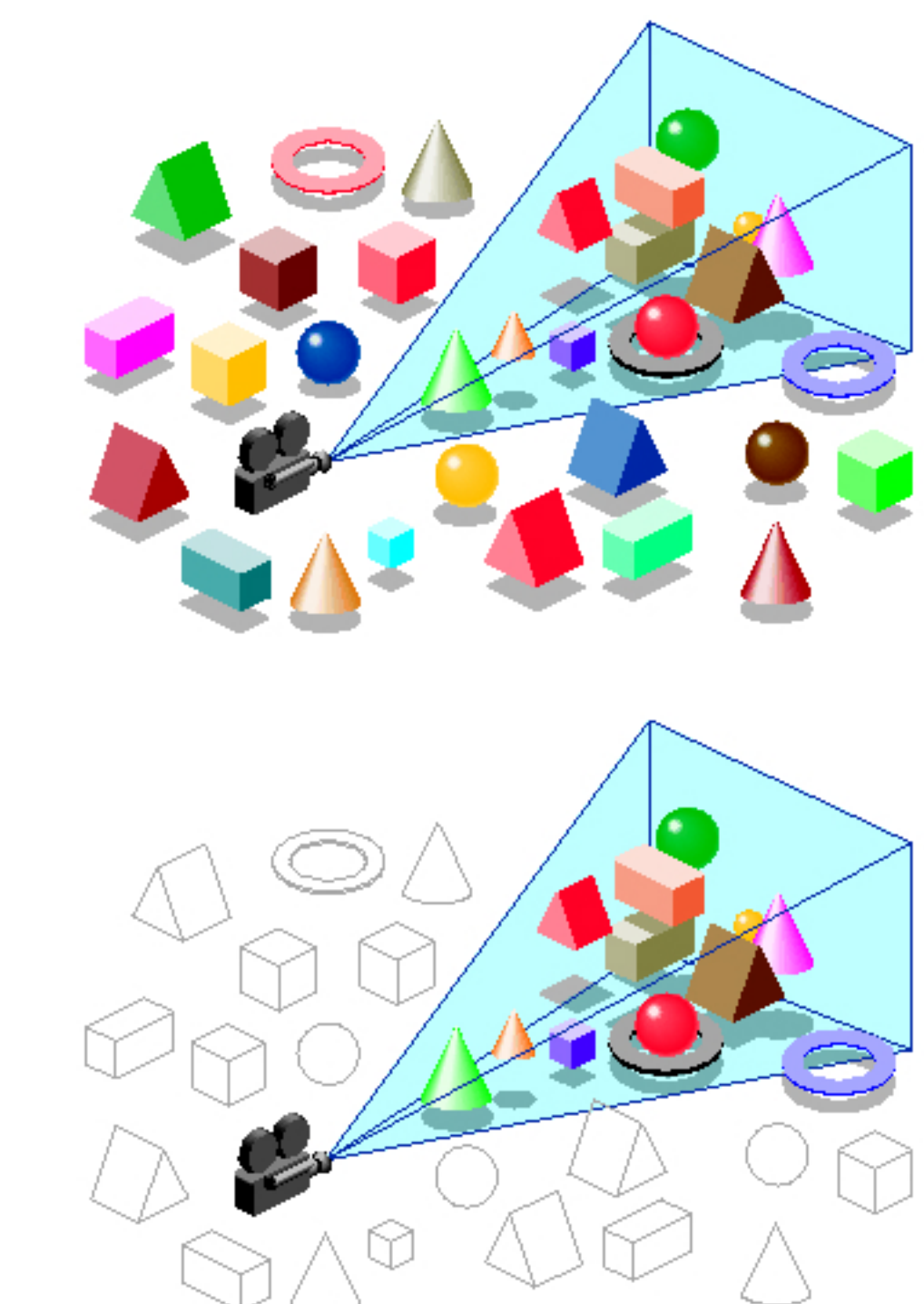
Simulação Física em tempo real

- Detecção de colisão por geometria aproximada - Esferas, Axis Aligned Bounding Boxes (AABB) ou Oriented Bounding Boxes (OBB) [ref.];
- Resposta de colisão interativa em função da distância de penetração de corpos [ref.];
- Dinâmica física baseada em impulso. Atualização da velocidade de acordo com atributos físicos (como massa e elasticidade) de cada corpo.

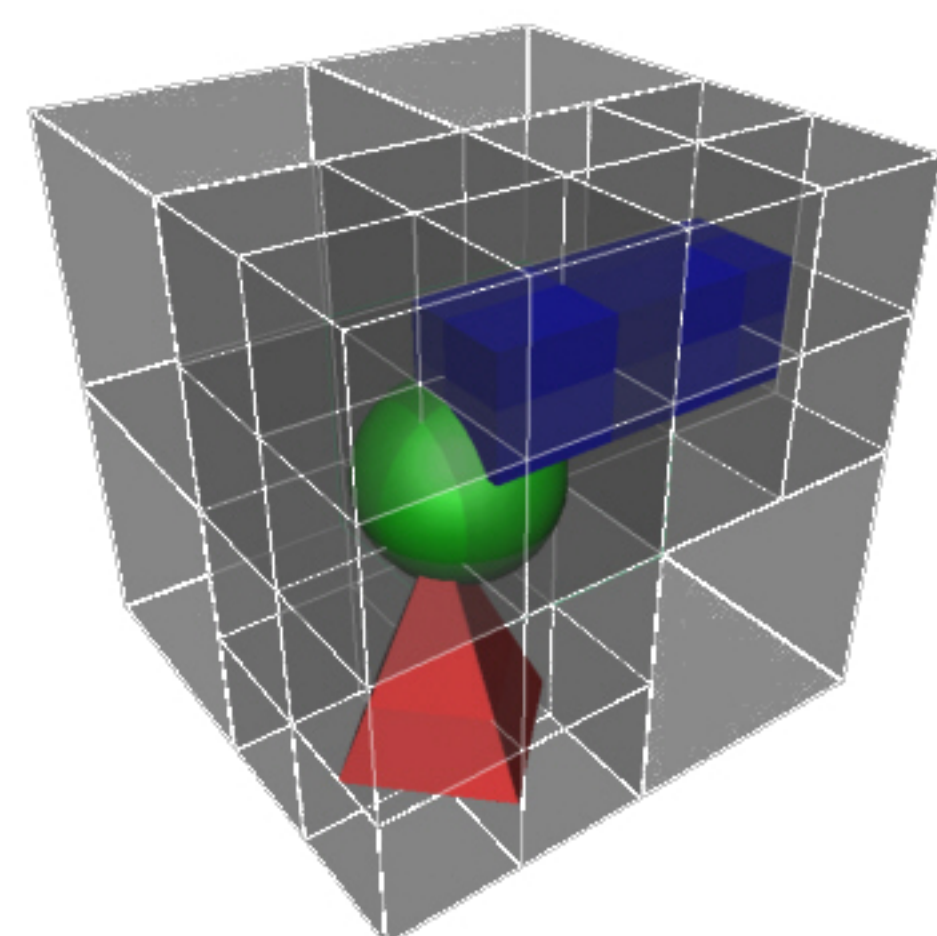


Gerenciamento de Cena

A URGE possui uma entidade de gerenciamento de cena. Seu objetivo é evitar overheads de visualização e de simulação, ou seja, evitar o desenho de objetos não vistos pelo jogador e evitar testes de colisão desnecessários. O gerenciamento executa algoritmos como *Viewing Frustum Culling* e *Octrees*.



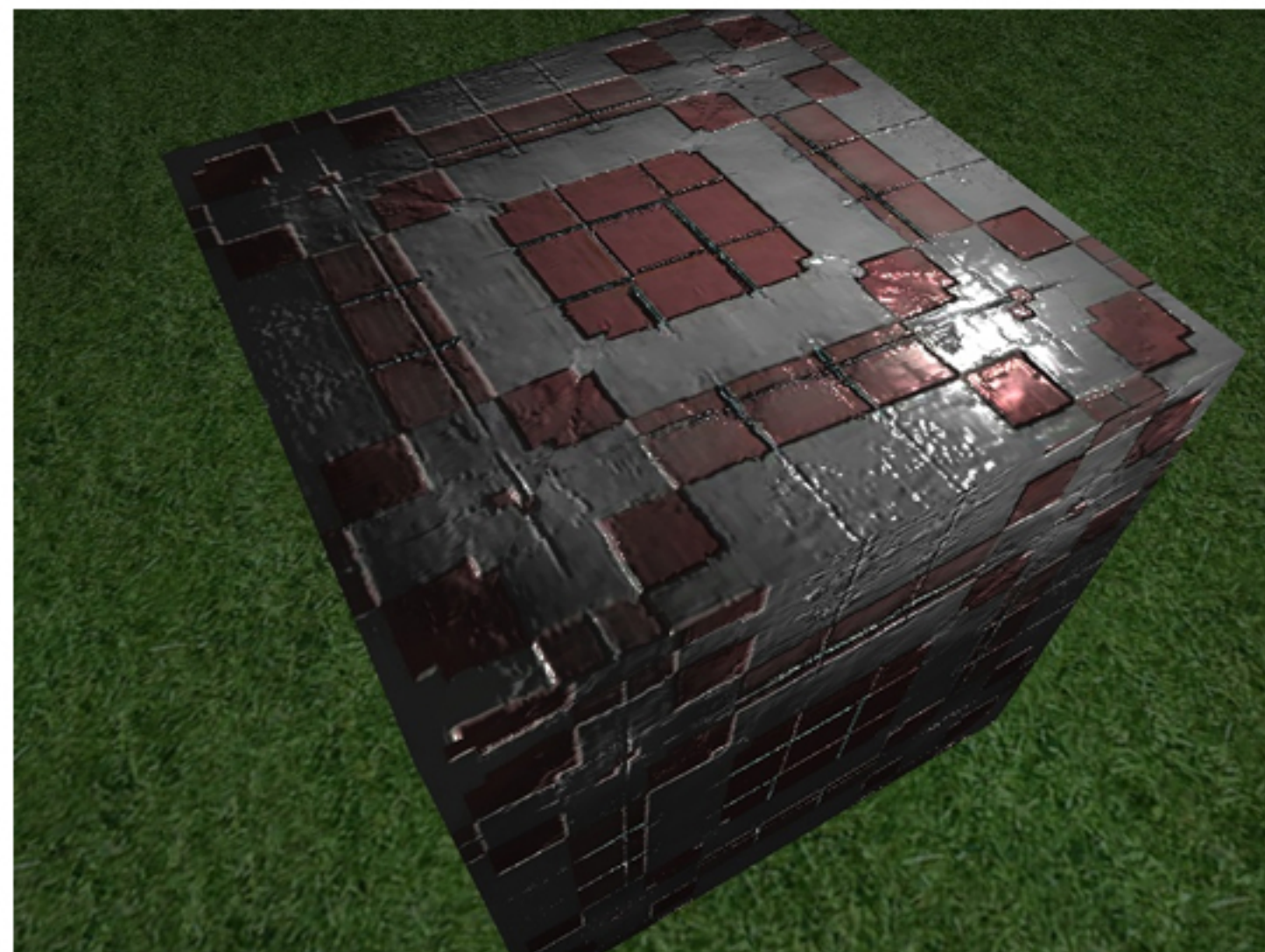
Frustum Culling



Octree

Visualização e Computação Gráfica

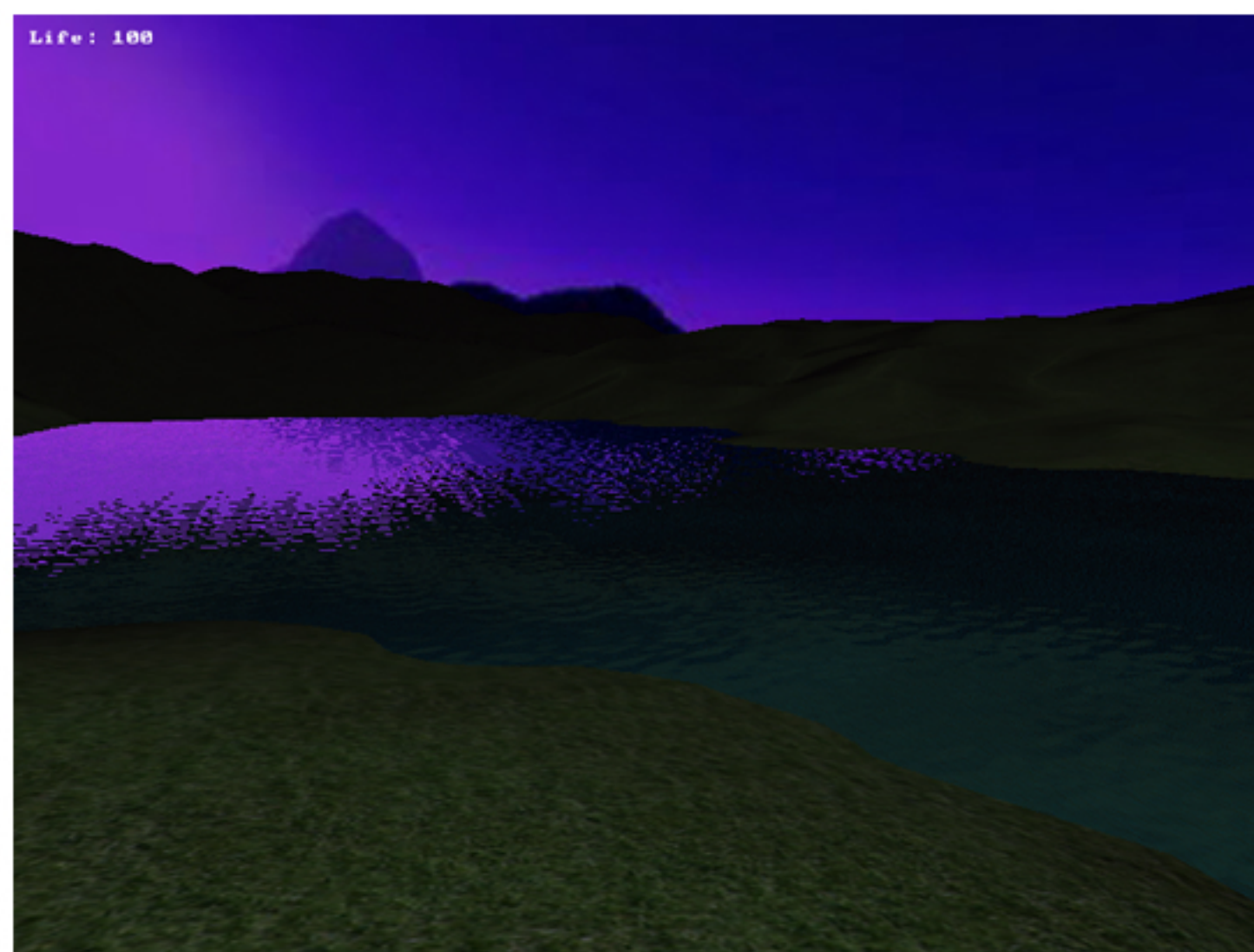
O foco do núcleo de visualização é realizar o desenho de materiais visíveis obtendo o maior proveito possível na relação entre qualidade gráfica e performance. Este sistema é responsável desde tarefas como desenho de primitivas até a renderização de terrenos, carregamento de modelos 3D, mapeamentos, iluminação via shaders e sistemas de partículas para simular fogo ou rastros luminosos.



Parallax Mapping



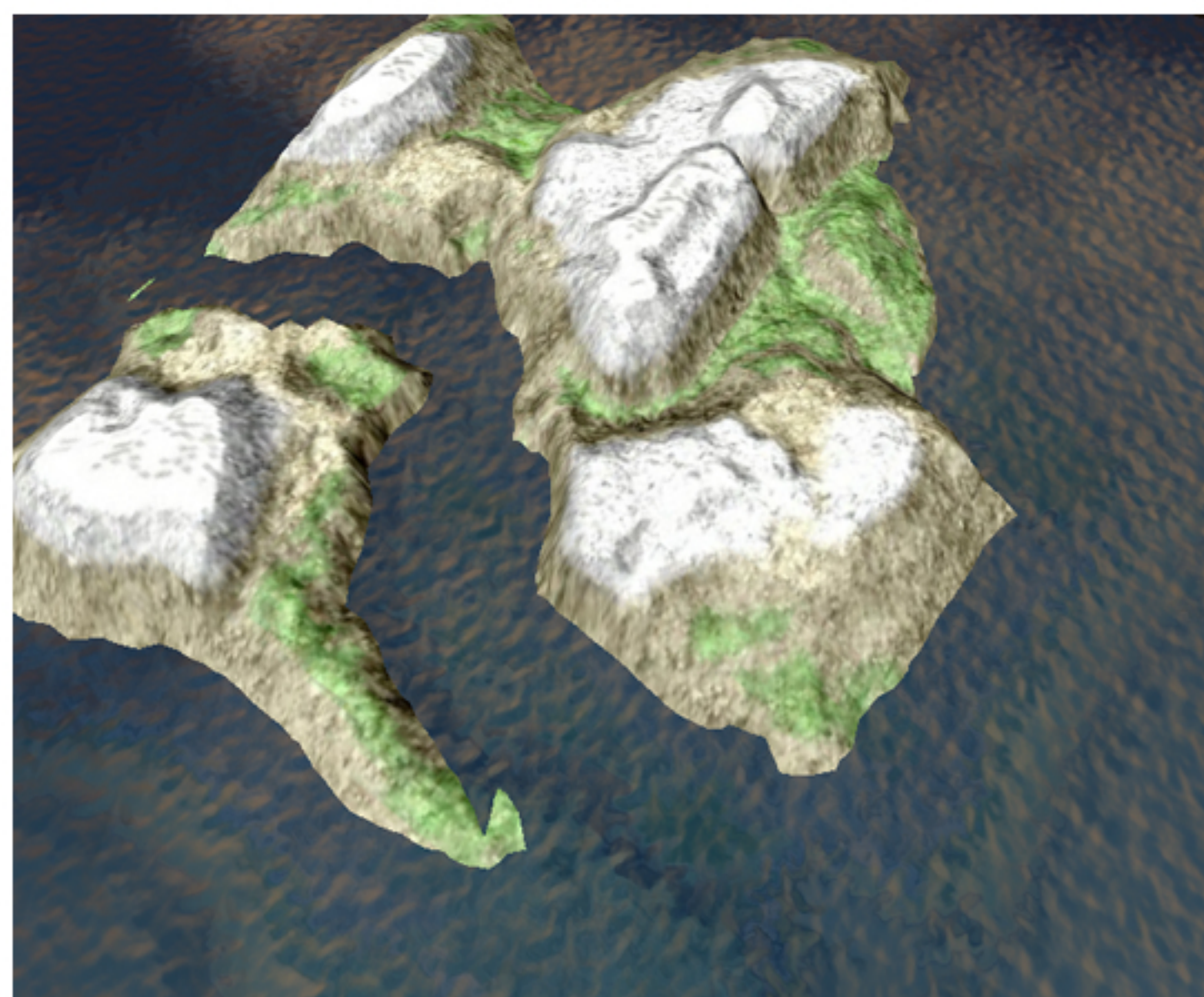
Partícula e modelo animado



Efeito de água



Sphere Mapping



Height Map



Carregamento de modelos



Iluminação



Primitivas

Referências

- [1] R. J. Rost, B. Licea-Kane, D. Ginsburg, J. M. Kessenich, B. Lichtenbelt, H. Malan, and M. Weiblen, OpenGL Shading Language, 3rd ed. Addison-Wesley Professional, 2009.
- [2] J. F. Blinn, Simulation of Wrinkled Surfaces, 3rd ed. ACM, 1978, vol. 12.
- [3] V. Krishnamurthy and M. Levoy, "Fitting smooth surfaces to dense polygon meshes," in Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996.
- [4] M. McGuire and M. McGuire, "Steep parallax mapping," I3D 2005 Poster, 2005.
- [5] B. Cabral, M. Olano, and P. Nemec, "Reflection space image based rendering," in Proceedings of the 26th annual conference on Computer graphics and interactive techniques, ser. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999.
- [6] D. Astle, More OpenGL Game Programming. Thomson/Course Technology, 2006.
- [7] Sibgrapi 2012, Proceedings of the XXV Brazilian Symposium on Computer Graphics and Image Processing. Macció, Brazil: IEEE, august 2012.
- [8] H. Hoppe, "View-dependent refinement of progressive meshes," in Proceedings of the 24th annual conference on Computer graphics and interactive techniques, ser. SIGGRAPH '97. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997.
- [9] OpenGL, "Opengl 2.1 reference," <http://www.opengl.org/sdk/docs/man/xhtmll>.
- [10] C. Ericson, Real-Time Collision Detection. Elsevier, 2005. [Online]. Available: <http://books.google.com.br/books?id=WGpL6Sk9qNAC>
- [11] S. Gottschalk, Collision Queries Using Oriented Bounding Boxes. University of North Carolina at Chapel Hill, 2000.
- [12] J. Van Verth and L. Bishop, Essential Mathematics for Games and Interactive Applications: A Programmer's Guide. Elsevier Science, 2008. [Online]. Available: <http://books.google.com.br/books?id=zKE9RIm4WkC>
- [13] B. Mirtich, Impulse-based Dynamic Simulation of Rigid Body Systems. University of California, Berkeley, 1996.
- [14] U. Assarsson and T. Möller, "Optimized view frustum culling algorithms for bounding boxes," J. Graph. Tools.