



Universidade do Minho
Departamento de Informática

Computação Gráfica
Grupo 32
3ª Fase



Artur Leite- A97027



Afonso Magalhães-A95250



Diana Teixeira- A97

Índice

Índice.....	2
1. Introdução.....	3
2. Engine.....	3
2.1. Conversão para VBOs.....	3
2.2. Transformações com tempo.....	3
2.2.1. Rotações com Tempo.....	4
2.2.2. Translações com Tempo	4
2.3. Movimentação da Câmara.....	4
2.4. Keybinds.....	4
3. Generator.....	5
3.1. Leitura e Interpretação de <i>patches de Bezier</i>	5
4. Sistema Solar-Elementos Adicionados.....	5
5. Conclusão	6

1. Introdução

Nesta fase do trabalho, atualizamos o nosso programa com o intuito de implementar algumas melhorias. Uma destas melhorias inclui a utilização de VBOs, tornando mais eficiente o processo de desenhar objetos. Adicionalmente, através da implementação das curvas *Catmull-Rom*, os corpos celestes incluídos no nosso programa conseguem efetuar as suas respetivas trajetórias em torno de outros corpos celestes. Devido a estas melhorias, deparamo-nos com a necessidade de atualizar a *script* utilizada para gerar o cenário correspondente ao Sistema Solar.

Para além disso, devido à adição dos *patches* de *Bezier*, existiu a necessidade de adicionar uma nova primitiva ao nosso *generator*.

2. Engine

2.1. Conversão para VBOs

Em fases anteriores, os modelos eram desenhados de forma imediata. Com o intuito de tornar este processo mais rápido e otimizado, foram utilizados os VBOs (*Vertex Buffer Objects*). Desta forma, a informação em relação aos vértices que definem os objetos são guardados num buffer que é mais facilmente acedido e processado pela GPU do que se estes dados estivessem armazenados em memória local, resultando numa melhoria de desempenho exponencial.

Para Implementamos um sistema de *caching* para que caso o ficheiro *.3d* passado já tenha sido lido, ou seja já tenha sido criado um VBO para os vértices desse ficheiro, a *engine* reutiliza esse mesmo VBO, caso contrário irá ler o ficheiro e criar um novo VBO com os dados do mesmo.

2.2. Transformações com tempo

Com o intuito de adicionar movimentações cíclicas semelhantes aos movimentos dos corpos celestes do nosso Sistema Solar, optamos por incluir um parâmetro de tempo ao nosso método *apply*

implementado em fases anteriores, que está encarregue de aplicar as transformações aos objetos. Este parâmetro descreve o tempo que decorreu desde o início da aplicação.

2.2.1. Rotações com Tempo

Para fazer com que o tempo que as rotações devem demorar seja semelhante ao real, efetuamos um cálculo que, em função do tempo decorrido, tem em consideração o ângulo do instante em questão e consequentemente aplica a rotação necessária para efetuar o ciclo corretamente.

2.2.2. Translações com Tempo

Tendo em conta que as translações do nosso modelo vão de acordo com as curvas de *Catmull-Rom*, que são definidas por um conjunto de pontos de controlo que definem o formato da curva, temos de definir estes mesmos pontos de controlo para cada órbita dos corpos celestes. Desta forma, através da interpolação linear, podemos calcular a posição de cada corpo em cada instante.

2.3. Movimentação da Câmara

No nosso programa, disponibilizamos dois modos de visualização do cenário do Sistema Solar. Um destes é o Explorer, que permite ao utilizador movimentar a câmara de acordo com inputs do teclado, sendo que já tinha sido implementada na fase anterior. O outro é o modo FPS, onde o utilizador pode se movimentar pelo cenário, podendo visualizar os objetos em primeira pessoa. A implementação deste modo da câmara é atingida através de cálculos que envolvem dois ângulos: α (ângulo que a câmara possui em relação ao plano xOz) e β (ângulo que a câmara possui em relação ao plano xOy). Sempre que o utilizador movimenta o rato, é calculada a distância percorrida em relação ao ponto anterior, sendo posteriormente adicionada aos ângulos descritos anteriormente, obtendo então a nova imagem pretendida.

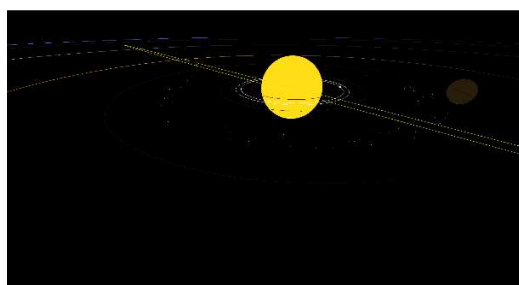


Figura 1- Imagem do cenário em modo FPS

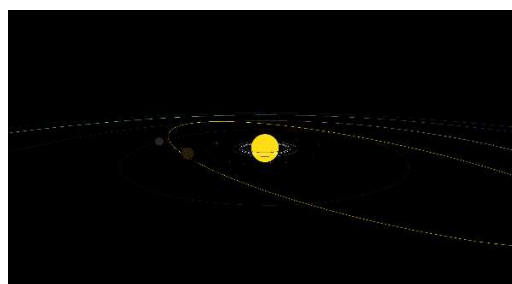


Figura 2- Imagem do cenário em modo Explorer

2.4. Keybinds

As seguintes *Keybinds* servem para facilitar o processo de visualização da simulação:

- WASD: Teclas utilizadas para movimentar a câmara no modo Explorer;
- F: Aumentar a velocidade da movimentação em ambos modos da câmara;
- L: Mostrar/Esconder as linhas que descrevem as trajetórias dos corpos celestiais;
- M: Mudar o modo da câmara.

3. Generator

3.1. Leitura e Interpretação de *patches* de Bezier

Nesta fase do projeto, com o intuito de utilizar as curvas de Bezier para representar certos elementos do nosso modelo, deparamo-nos com a necessidade de atualizar o nosso *generator* para ser capaz de interpretar os ficheiros .patch. Numa fase inicial, efetuamos a leitura dos *patches* presentes nesses ficheiros, guardando essa informação num vetor. Em seguida, tendo em conta que estes *patches* são compostos por índices que representam pontos, lemos estes respetivos elementos, sendo também guardados.

Agora que ambos estes conjuntos de informação estão guardados corretamente, passamos ao processo de cálculo das coordenadas para o ficheiro .3d, partindo de um nível de *tesselation* à escolha do utilizador. Para facilitar o processo de cálculo de matrizes para todos os pontos em questão, optamos por utilizar a seguinte expressão:

$$p(u, v) = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} M \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} M^T \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix}$$

4. Sistema Solar-Elementos Adicionados

Nesta fase do trabalho, foram adicionados alguns novos elementos ao nosso modelo do Sistema Solar, nomeadamente:

- Órbitas dos planetas, asteróides e do cometa Halley;
- Rotações dos Planetas;
- Adição do cometa Halley, representado por um *teapot*;

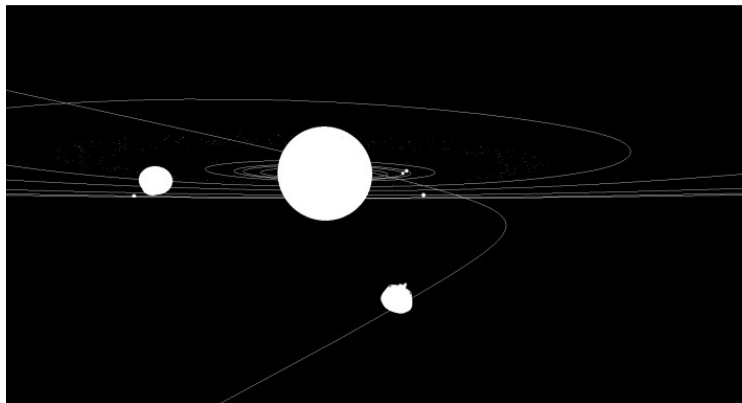


Figura 1- Cometa Halley, representado pelo teapot

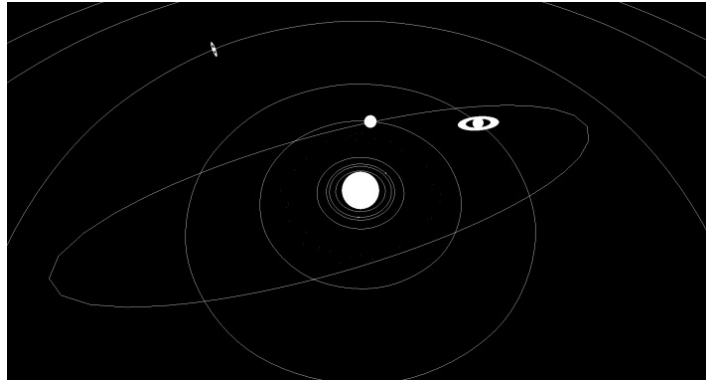


Figura 2- Órbitas dos corpos celestes

5. Conclusão

Após a realização desta terceira fase do trabalho, achamos que fomos capazes de integrar as melhoras ao nosso projeto de forma correta e concisa, sendo que nos deparamos com uma melhoria de desempenho considerável.

No futuro, gostaríamos de implementar uma interface gráfica com recurso à biblioteca ImGui para facilitar a utilização do programa.