Gustavo Rodrigues Sampaio

Marco Aurélio Ferreira Moreira

Tássis Fernando Soares Fernandes

**TRABALHO PRÁTICO - SISTEMA SOLAR**

Trabalho apresentado à disciplina de Computação com a criação de uma uma animação 2D representando o sistema solar.

Professora: Odilon

Timóteo

2022

**Sumário**

[**1. Introdução**](#_y817cagpm66y) **3**

[**2. Desenvolvimento**](#_3cvqsb4vph9g) **3**

[2.1. Cenário](#_jil0gv2xn77i) 3

[2.2. Animação](#_xsz5isxmnfyw) 7

[2.3. Controle](#_p4pv2ey4x61s) 8

[2.4. Cálculo](#_1q2lq5o3exz6) 12

[**3. Conclusão**](#_igpmjndm6y0k) **13**

# Introdução

O presente trabalho faz parte da disciplina de Computação Gráfica do curso de Engenharia de Computação do CEFET-MG, Campus Timóteo. O objetivo proposto é de elaborar uma animação em duas dimensões que represente o sistema solar, com os movimentos de rotação dos planetas, translação ao redor do sol e a representação de anéis e satélites naturais dos planetas que possuem.

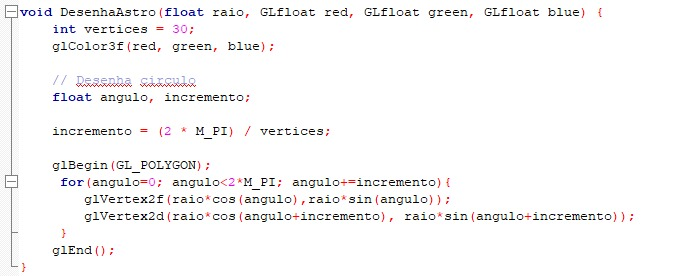
Além disso, por meio da interação do usuário com o programa por meio do teclado deve ser possível controlar a visualização dos planetas, das órbitas e pausar os movimentos de todos. Por fim, também se objetiva exibir o tempo passado no planeta terra medido pelo número de voltas do planeta ao redor do sol. O protótipo de inspiração encontra-se apresentado abaixo:

p

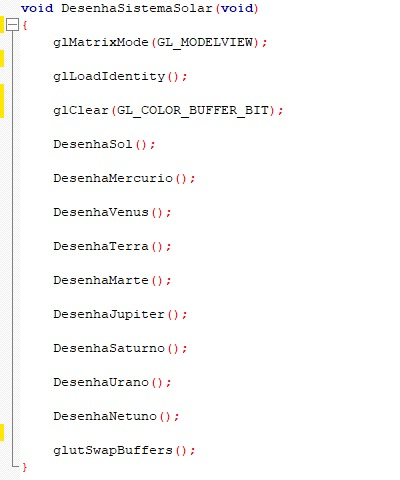
# Desenvolvimento

## Cenário

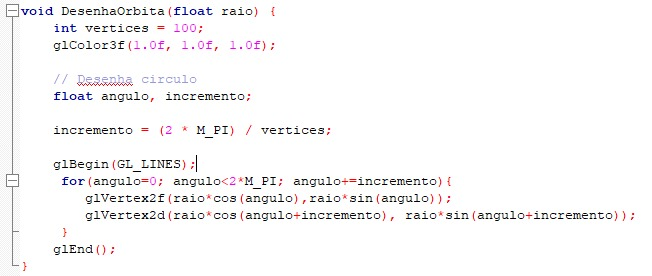
* Desenhar o sol, os planetas e suas órbitas

Para desenhar o sol e os planetas, foi criada a função DesenhaAstro(), que recebe parâmetros para definir sua cor e seu tamanho. O Sol foi desenhado sem translação nenhuma, após isso, foi necessário fazer a translação de cada planeta e logo depois chamar a função de desenhá-los, para que ficassem na posição correta do sistema solar. Para a melhor visualização os planetas foram desenhados alternando o eixo x entre positivo e negativo.  


Para cada astro foi criado uma função que são chamadas na função DesenhaSistemaSolar().

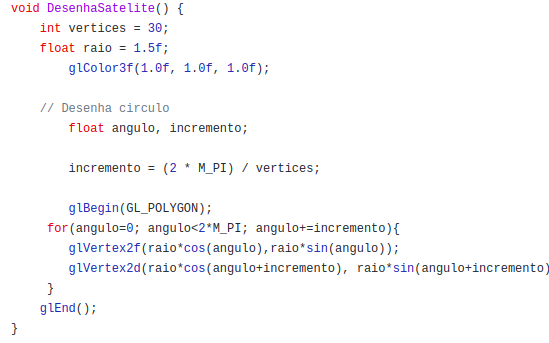


Para o desenho das órbitas foi criada uma função, que foi reutilizada para desenhar também os anéis, as órbitas foram desenhadas com a mesma translação dos planetas, para que tivessem o mesmo raio.



* Desenhar os satélites naturais

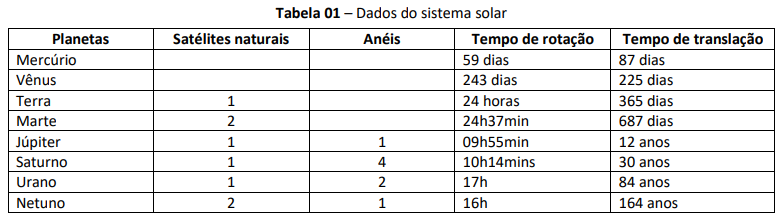
Para desenhar os satélites naturais, foi criada uma função específica para essa funcionalidade que tem o código apresentado abaixo.



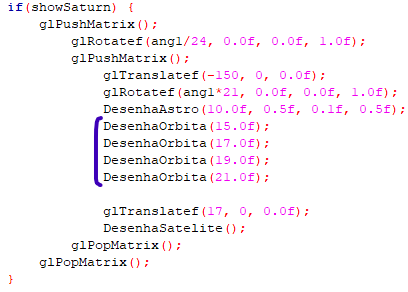
Inicialmente é definido a quantidade de vértices que compõem o círculo do satélite em questão (utilizamos o valor fixo igual a 30). Após isso declaramos o tamanho do raio de cada satélite e sua cor, ambos fixos. Depois é feito o cálculo para definir o incremento que o ângulo irá sofrer a cada iteração que será responsável por definir a posição dos satélites. O restante da função é referente a criação das órbitas.

* Desenhar os anéis

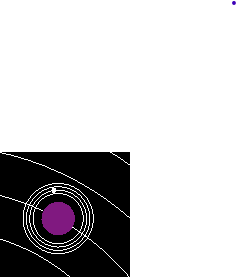
Para desenhar os anéis dos planetas, foram consideradas as quantidades indicadas na proposta do trabalho, representadas na tabela abaixo:



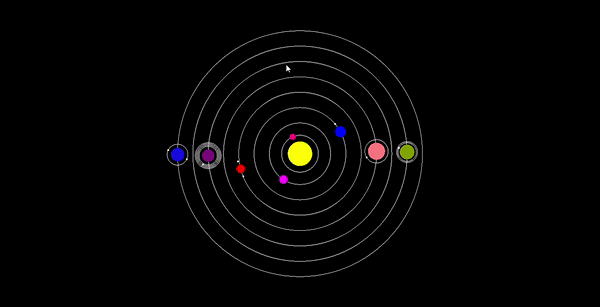
Para gerar os anéis em voltas dos planetas apresentados acima foi utilizado o mesmo método criado e já explicado de gerar as órbitas (DesenhaOrbita).



Como já está transladada a matriz para a posição que o planeta está desenhada, o que foi feito foi desenhar círculos de raios centrados no planeta. Portanto, no caso do Saturno, apresentado acima, ele já está transladado a uma distância de 150 do centro (Sol). Com isso, foi chamado o método de desenhar órbita 4 vezes com raios de 15, 17, 19 e 21, de forma a desenhar círculos crescentes em volta do astro.



Saturno



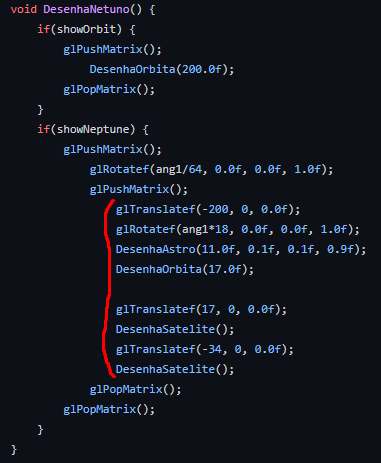
## Animação

* Criar o movimento de rotação

Para criar o movimento de rotação, antes de desenhar cada planeta foi chamado o método glRotatef, passando o ângulo, que é multiplicado ou dividido baseado na velocidade de rotação do planeta.

* Criar o movimento de translação

Para criar o movimento de translação foi feito um trecho de código dentro da respectiva função de cada planeta, segue o código abaixo:

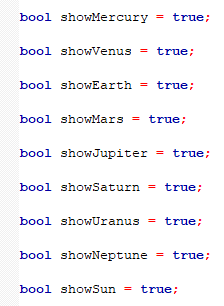


A execução começa ao realizar o glPushMatrix() para empilhar a atual matriz de transformação no topo da pilha. Após isso a matriz é transladada seguindo os parâmetros inseridos de forma fixa na função glTranslateF(-200,0,0.0f). Depois que ocorre a translação é feito a rotação do objeto em torno do ponto central e assim ocorre a cada iteração da execução e forma o movimento de translação dos astros.

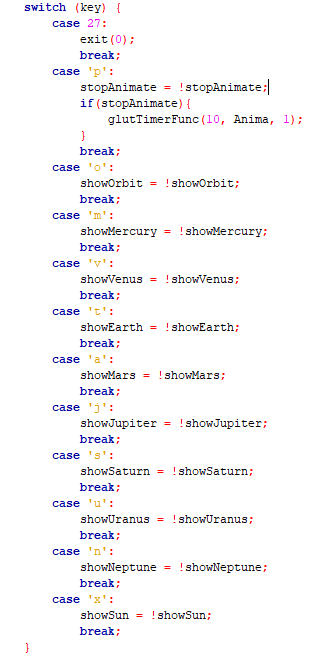
## Controle

* Ocultar e exibir um determinado planeta com seus satélites e anéis

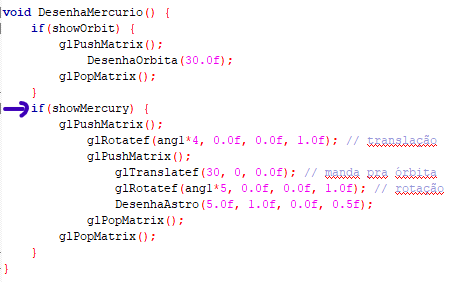
Para controlar a exibição de um planeta específico, junto de seus satélites e anéis, foram criadas variáveis globais de controle:



Elas são do tipo *bool* (*true/false)* e seu valor é alterado por meio da interação do usuário via teclado. Com isso, na função Teclado, que controla os eventos de teclas, existe uma estrutura de *switch-case* que trata para cada letra, que representa uma planeta, a mudança do estado da variável de controle:



Como já visto anteriormente, cada astro possui sua função específica que o desenha. Com isso, é possível adicionar estruturas condicionais verificando se a variável de controle desse astro está *true* ou *false*, sendo executado o código de desenho do astro, anéis e satélites apenas no caso de ser verdadeiro.



* Ocultar e exibir todas as órbitas

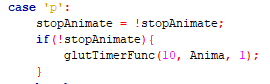
Para ocultar/exibir as órbitas, foi definida a letra ‘o’ do teclado para ser pressionada pelo usuário e ter esse controle. Na função Teclado, que gerencia os eventos de teclas, existe um tratamento para eventos com a letra ‘o’:



Um SE é colocado ao redor de cada função que desenha as órbitas dentro das funções que desenham os planetas, para que a cada vez que o usuário aperte o botão as órbitas de todos os planetas sejam mostradas ou escondidas.

* Pausar e continuar a simulação

Para pausar/continuar a animação, foi definida a letra ‘p’ do teclado para ser pressionada pelo usuário e ter esse controle. Na função Teclado, que gerencia os eventos de teclas, existe um tratamento para eventos com a letra ‘p’:

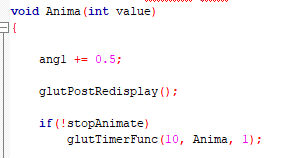


Para cada vez que a tecla é pressionada, o valor da variável de controle *stopAnimate* é alterado o valor. Como é uma variável do tipo bool, será sempre *true/false*. Ela inicia com *false*, indicando que a animação não está pausada:



Quando essa variável de controle está com o valor falso, a função glutTimerFunc é chamada, passando o intervalo de tempo de 10 ms que a função de callback Anima deve ser executada (a função Anima também é passada por parâmetro).

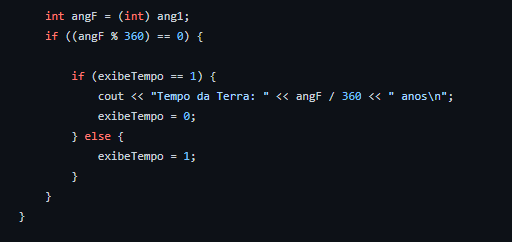
A função Anima é recursiva, portanto, internamente é feito o controle também com o valor da variável de controle *stopAnimate.* Dessa forma, só haverá a alteração dos ângulos do movimento de cada planeta, com a variação da variável global que controla o ângulo (ang1) se a variável de controle *stopAnimate* for *true,* o que gera o movimento dos planetas e satélites*.*

**

## Cálculo

* Calcular e exibir o tempo que a Terra está orbitando o Sol

Para realizar o cálculo do tempo em que a Terra está orbitando o Sol foi utilizado o trecho de código abaixo que está presente na função “Anima”.



Utilizamos uma variável com escopo local (angF) para receber o valor do ângulo atual da variável global “ang1”. Após a atribuição é realizado uma verificação para saber se o valor do ângulo atual dividido por 360º tem como resto 0, se tiver significa que acabou de concluir uma volta e executa o próximo “if”, caso contrário continua a execução sem esse trecho. Caso entre no “if (exibeTempo == 1)” (“exibeTempo” é uma variável global utilizada para fazer o controle de quando ou não imprimir com intuito de evitar impressões duplicadas) é realizada a impressão da quantidade de voltas (ou anos terrestres) que a Terra já concluiu em torno do Sol, caso contrário a impressão só ocorre na próxima iteração.

# Conclusão

Após a finalização do trabalho foi-se capaz de concluir que apesar de a princípio o trabalho parecer muito extenso, com a reutilização dos métodos foi possível realizar o trabalho de maneira mais simples e rápida.  
Foi possível aplicar alguns conceitos que vimos em sala de aula, como a translação e o sistema gráfico.  
O trabalho foi realizado com sucesso e foi possível realizar tudo o que foi requisitado.