

Paulo Belfi Dias da Silva

Sistema solar minimalista em LaTeX utilizando pacote TikZ

São Paulo – Brasil

2017

Paulo Belfi Dias da Silva

Sistema solar minimalista em LaTeX utilizando pacote TikZ

Relatório técnico, em conformidade com as normas ABNT, apresentado como requisito para aprovação na disciplina Projeto Integrador I.

Centro Universitário Senac
Bacharelado em Ciência da Computação

São Paulo – Brasil

2017

Resumo

Gerar um código que reproduza o movimento de translação dos planetas, por mais que desconsidere algumas circunstancias, podem acabar gerando alguns problemas por mais simples que pareça. Este relatório tem como objetivo demonstrar como foi reproduzir parte do movimento de translação dos planetas. E quais as dificuldades encontradas para gerar o código em \LaTeX , e transformá-lo em um PDF.

Palavras-chaves: latex. tikz. Sistema Solar. Minimalismo.

Sumário

1	INTRODUÇÃO	4
2	MATERIAIS E MÉTODOS	5
2.1	Motivações ao uso do material	5
3	DESENVOLVIMENTO	6
3.1	Tipo de documento e pacotes usados	6
3.2	Criação do modelo a ser usado	6
3.3	Criação do algoritmo que irá gerar o código LaTeX	7
3.4	Compilação do código	9
4	RESULTADO	11
4.1	Detalhes extras	11
5	CONCLUSÃO	12
	REFERÊNCIAS	13

1 Introdução

Este documento demonstra como foi criada a animação do sistema solar minimalista, utilizando o pacote Tikz dentro da linguagem LaTeX. Com seu código sendo gerado através de um programa em linguagem C. A motivação de realizar este trabalho foi devido a gostar do estudo de astronomia, e querer reproduzir esta ideia para um código que conseguisse demonstrar um pouco do universo.

Para gerar o código em Tikz LaTeX, foi necessário um programa em C, para automatizar o processo de gerar o cálculo da posição de cada planeta, utilizando conceitos de ângulos e círculo trigonométrico. E para gerar os slides da animação foram utilizados, códigos básicos, como *fill*, *definecolor*, *frame*.

2 Materiais e Métodos

Para gerar a animação foram utilizados diversas ferramentas. Iniciando pela definição de como seria o modelo a ser utilizado, usando o site overleaf.com, para definir as dimensões e cores que seriam utilizados pelos, planetas.

Após essa definição foi necessário o uso de um programa em C que iria gerar o código TikZ, a ser reproduzido.

E finalmente para compilar o código gerado, foi utilizado o compilador `pdflatex`, pois nenhum dos editores de código LaTeX testados suportavam o arquivo.

2.1 Motivações ao uso do material

As ferramentas utilizadas em sua maioria foram escolhidas devido a familiaridade do autor.

A escolha do site overleaf.com, foi devido a facilidade de uso do mesmo, por mais que não tenha o melhor desempenho em questão de tempo de compilação o site atende bem as expectativas de usuários que não tenham tanta familiaridade com o LaTeX.

Para escrever o código em C foi utilizado o software "Sublime Text 3", devido a sua facilidade de uso e conter diversos facilitadores para a escrita de código fonte em C.

Para compilar o código LaTeX foi utilizado o compilador `pdflatex` em uma plataforma Linux, pois foi o único testado que conseguia compilar o código gerado.

3 Desenvolvimento

O desenvolvimento foi inicialmente definir as cores, proporções e medidas que seriam utilizadas para gerar o código e posteriormente a definição da animação em latex.

3.1 Tipo de documento e pacotes usados

Como irá ser criada uma animação faz se necessário criar um documento que suporte animações, o escolhido foi o beamer, pois o seu uso é simples, e atendeu as necessidades do trabalho.

Para definir as cores que serão utilizadas nos planetas foi utilizado o pacote xcolor, pois assim consegue-se converter as cores em RGB, que serão extraídas da imagem de referencia para cores aceitadas em LaTeX.

Para a realização do desenho foi utilizado o pacote TikZ pois era uma das obrigatoriedades do trabalho, e é um pacote de fácil utilização.

3.2 Criação do modelo a ser usado

Para criar o modelo foi utilizado o site overleaf.com, aonde foi definido todo o primeiro slide da animação, ele será utilizado como base de todo o trabalho.

Utilizando dados reais dos tamanhos dos planetas, foi gerado uma imagem de teste aonde os planetas seguiam a mesma proporção do tamanho real. Porém quando a imagem era gerada, ficava imperceptível a existência de alguns planetas na imagem. Então a saída para este problema foi utilizar de licença poética e dar tamanhos não proporcionais aos planetas.



Figura 1 – Imagem gerada como base

Definido o desenho a ser realizado pelo TikZ, foi preciso apenas definir o uso do primeiro slide e tempo de transição de cada slide. Cada slide contém a posição do planeta em determinado dia, ou seja cada slide é equivalente a um dia na Terra. Como alguns planetas demoram mais do que outros para completarem o seu movimento de translação, sendo o mais demorado do sistema solar Netuno, que demora cerca de 160 anos da Terra, para completar o movimento, foi necessário realizar uma aceleração, na transição dos slides. Inicialmente foi testado, o tempo de transição de cada slide em 24 slides por segundo, porém o tempo de duração total acaba sendo muito grande, cerca de 41 minutos, para completar o movimento de Netuno. Então foi alterado o tempo de duração para 60 slides por segundo, que totalizava cerca de 13 minutos, ainda bastante tempo, porém com maior fluidez no movimento.

3.3 Criação do algoritmo que irá gerar o código LaTeX

Após definido o modelo base de uso do sistema solar, o tempo de transição de cada slide, o próximo passo foi gerar o código que irá gerar o código LaTeX a ser gerado. Para isso foi utilizado o software "Sublime Text 3", um software de edição de texto, e o mesmo possui diversas ferramentas que auxiliam no momento de escrever o código em linguagem C. E o compilador de linguagem C, GCC em sua versão 6.3.0, em uma plataforma Windows.

Para gerar a simulação do movimento de translação dos planetas, existem dois cenários possíveis, o cenário aonde será simulado o movimento da forma mais fidedigna possível, e o cenário que irá simular um movimento que remeta ao movimento de translação. O primeiro cenário implica em conseguir traduzir para o algoritmo, diversos fatores, como massa dos planetas, movimentos elípticos, aceleração devido a distância entre corpos celestes, e outros diversos fatores. Já o segundo cenário, contém somente a ideia do movimento de translação simplificado, do qual funciona de forma semelhante a um círculo trigonométrico.

A posição de cada planeta, será definido pela sua posição (X,Y) , sendo X o seno do ângulo relativo ao dia multiplicado pela sua distancia do centro do Sol. Isso significa o total de slides, desde o inicio da animação, dividido pelo tempo de duração do ano do planeta e multiplicado pela distância do planeta ao Sol. E Y a mesma formula que X , porem alterando seno por cosseno.

Definido o método que deverá ser reproduzido, podemos partir então para a codificação do movimento, o código TikZ LaTeX, foi separado em duas partes, a que não possui repetição, área do código onde será definido os padrões que serão utilizados na animação, como por exemplo classe do documento, cores dos planetas, definições dos pacotes e etc. E a parte que será repetida, que é composta pela definição dos frames e das figuras a serem geradas.


```

1 \documentclass{beamer}
2 \usetheme(default)
3 \usepackage{tikz}
4 \usepackage{xcolor}
5 \beamertemplatenavigationsymbolsempy
6 \definecolor{sol}{rgb}{0.992156863,0.792156863,0.003921569}
7 \definecolor{mercurio}{rgb}{0.850980392,0.733333333,0.478431373}
8 \definecolor{venus}{rgb}{0.992156863,0.792156863,0.003921569}
9 \definecolor{terra}{rgb}{0.003921569,0.2,0.592156863}
10 \definecolor{marte}{rgb}{1,0.207843137,0.003921569}
11 \definecolor{jupiter}{rgb}{0.505882353,0.4,0.278431373}
12 \definecolor{saturno}{rgb}{0.850980392,0.733333333,0.478431373}
13 \definecolor{urano}{rgb}{0.4,0.4,0.4}
14 \definecolor{netuno}{rgb}{0.003921569,0.411764706,0.788235294}
15 \begin{document}

```

Figura 2 – Parte fixa do código LaTeX

```

18 \begin{frame}
19 \transduration{0.0166666666666667}
20 \begin{tikzpicture}[scale=0.5]
21 \fill[black] (-4,-4) rectangle (19,15);
22 \fill[sol] (5,6) circle(10.00000000mm);
23 \fill[mercurio] (6.000000000,8.000000000) circle(1.035053161mm);
24 \fill[venus] (6.000000000,8.500000000) circle(0.586951149mm);
25 \fill[terra] (6.000000000,9.000000000) circle(0.591639368mm);
26 \fill[marte] (6.000000000,9.500000000) circle(0.548795977mm);
27 \fill[jupiter] (6.000000000,10.500000000) circle(3.027182908mm);
28 \draw[saturno,thick,densely dotted] (6.000000000,11.500000000) circle(3.797758621mm);
29 \fill[saturno] (6.000000000,11.500000000) circle(2.86591954mm);
30 \fill[urano] (6.000000000,13.500000000) circle(0.867227011mm);
31 \fill[netuno] (6.000000000,14.000000000) circle(0.855004598mm);
32 \end{tikzpicture}
33 \end{frame}
34

```

Figura 3 – Parte de repetição do código LaTeX

Já dentro do código em linguagem C, serão utilizados, os pacotes "stdio.h", "stdlib.h", "math.h" e também será utilizado um arquivo ".tex" para a saída do código gerado, pois irá facilitar o processo para a compilação do código LaTeX. Inicialmente iremos inserir dentro deste arquivo ".tex", a parte fixa do código. Como visto na figura 4 logo acima.

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main ( void )
{
    FILE * Output;
    Output = fopen("Sistema_solar.tex", "a");
    char * iniciadocumento = "\\documentclass{beamer}\\n"
        "\\usepackage{tikz}\\n"
        "\\usepackage{xcolor}\\n"
        "\\beamertemplatenavigationsymbolsempy\\n"
        "\\definecolor{sol}{rgb}{0.992156863,0.792156863,0.003921569}\\n"
        "\\definecolor{mercurio}{rgb}{0.850980392,0.733333333,0.478431373}\\n"
        "\\definecolor{venus}{rgb}{0.992156863,0.792156863,0.003921569}\\n"
        "\\definecolor{terra}{rgb}{0.003921569,0.2,0.592156863}\\n"
        "\\definecolor{marte}{rgb}{1,0.207843137,0.003921569}\\n"
        "\\definecolor{jupiter}{rgb}{0.505882353,0.4,0.278431373}\\n"
        "\\definecolor{saturno}{rgb}{0.850980392,0.733333333,0.478431373}\\n"
        "\\definecolor{urano}{rgb}{0.4,0.4,0.4}\\n"
        "\\definecolor{netuno}{rgb}{0.003921569,0.411764706,0.788235294}\\n"
        "\\begin{document}\\n";
    fprintf(Output, "%s",iniciadocumento);
}

```

Figura 4 – Inserção da parte fixa do código

Para a geração dos slides será necessário o algoritmo que gera a posição relativa dos planetas e um loop que irá controlar o angulo relativo de cada planeta. Sendo assim o código em C ficará da seguinte forma. Após isso falta somente compilar o código

```

for(int i = 0; i < 360; i++)
{
    fprintf(Output, "\\begin{frame}[t]\\n");
    fprintf(Output, "\\transduration{0.0166666666666667}\\n");
    fprintf(Output, "\\begin{tikzpicture}[scale=0.5]\\n");
    fprintf(Output, "\\fill[black] (-4,-4) rectangle (19,15);\\n");
    fprintf(Output, "\\fill[sol] (5,6) circle(10.00000000mm);\\n");
    fprintf(Output, "\\fill[mercurio] (6.000000000,8.000000000) circle(1.035053161mm);\\n");
    fprintf(Output, "\\fill[venus] (6.000000000,8.500000000) circle(0.586951149mm);\\n");
    fprintf(Output, "\\fill[terra] (6.000000000,9.000000000) circle(0.591639368mm);\\n");
    fprintf(Output, "\\fill[marte] (6.000000000,9.500000000) circle(0.548795977mm);\\n");
    fprintf(Output, "\\fill[jupiter] (6.000000000,10.500000000) circle(3.027182908mm);\\n");
    fprintf(Output, "\\draw[saturno,thick,densely dotted] (6.000000000,11.500000000) circle(3.797758621mm);\\n");
    fprintf(Output, "\\fill[saturno] (6.000000000,11.500000000) circle(2.86591954mm);\\n");
    fprintf(Output, "\\fill[urano] (6.000000000,13.500000000) circle(0.867227011mm);\\n");
    fprintf(Output, "\\fill[netuno] (6.000000000,14.000000000) circle(0.855004598mm);\\n");
    fprintf(Output, "\\end{tikzpicture}\\n");
    fprintf(Output, "\\end{frame}\\n");
    fprintf(Output, "\\n");
    return 0;
}

```

Figura 5 – Inserção da parte variável do código

e executá-lo, para isso foi utilizado o compilador GCC ver. 6.3.0 para Windows. Para

compilar o arquivo se utiliza o comando GCC no terminal. Já com o terminal dentro da pasta aonde se localiza o arquivo ".c", utiliza-se o comando GCC passando como parâmetro o nome do arquivo ".c-o nome do arquivo de saída. Como no exemplo abaixo.

```
C:\Users\paulo\OneDrive\Documentos\GitHub\Senac>pdflatex "Sistema solar.tex"
This is pdfTeX, Version 3.14159265-2.6-1.40.18 (MiKTeX 2.9.6300 64-bit)
entering extended mode
("Sistema solar.tex"
LaTeX2e <2017-04-15>
Babel <3.9r> and hyphenation patterns for 75 language(s) loaded.
("C:\Program Files\MiKTeX 2.9\tex\latex\beamer\beamer.cls"
Document Class: beamer 2017/01/23 v3.41 A class for typesetting presentations
("C:\Program Files\MiKTeX 2.9\tex\latex\etoolbox\etoolbox.sty")
("C:\Program Files\MiKTeX 2.9\tex\latex\beamer\beamerbasemodes.sty"
("C:\Program Files\MiKTeX 2.9\tex\latex\beamer\beamerbaseencode.sty"))
("C:\Program Files\MiKTeX 2.9\tex\generic\oberdiek\ifpdf.sty")
("C:\Program Files\MiKTeX 2.9\tex\latex\beamer\beamerbaseoptions.sty"
("C:\Program Files\MiKTeX 2.9\tex\latex\graphics\keyval.sty"))
("C:\Program Files\MiKTeX 2.9\tex\latex\geometry\geometry.sty"
("C:\Program Files\MiKTeX 2.9\tex\generic\oberdiek\ifvtex.sty")
("C:\Program Files\MiKTeX 2.9\tex\generic\ifxetex\ifxetex.sty")
("C:\Program Files\MiKTeX 2.9\tex\latex\geometry\geometry.cfg"))
("C:\Program Files\MiKTeX 2.9\tex\latex\base\size11.clo")
("C:\Program Files\MiKTeX 2.9\tex\latex\pgf\basiclayer\pgfcore.sty"
("C:\Program Files\MiKTeX 2.9\tex\latex\graphics\graphicx.sty"
("C:\Program Files\MiKTeX 2.9\tex\latex\graphics\graphics.sty"
("C:\Program Files\MiKTeX 2.9\tex\latex\graphics\trig.sty")
("C:\Program Files\MiKTeX 2.9\tex\latex\graphics-cfg\graphics.cfg")
("C:\Program Files\MiKTeX 2.9\tex\latex\graphics-def\pdfTeX.def"
("C:\Program Files\MiKTeX 2.9\tex\generic\oberdiek\ifxetex.sty"))
```

Figura 6 – Compilação do código em C

3.4 Compilação do código

Como dito anteriormente nenhum dos softwares de edição de código LaTeX testados, foram capazes de abrir o arquivo ".tex"gerado. Então desta forma foi necessário compilar o código diretamente pelo terminal. Inicialmente foi testado no compilador pdflatex, para Windows, porém em determinado momento o compilador não conseguia prosseguir com a compilação do arquivo.

Então fez se necessário compilar o código LaTeX, no compilador pdflatex para Linux, já neste caso, do tempo de quase 1 hora compilando o arquivo, a compilação foi finalizada com sucesso.

Para compilar o arquivo se utiliza o comando pdflatex no terminal. Já com o terminal dentro da pasta aonde se localiza o arquivo ".tex", utiliza-se o comando pdflatex passando como parâmetro o nome do arquivo .tex.

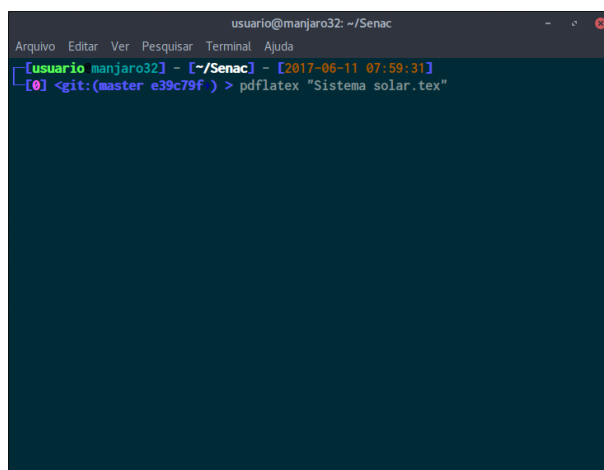


Figura 7 – Compilação do código LaTeX

4 Resultado

Como o trabalho desenvolvido é uma animação e não pode ser reproduzida neste relatório irei deixar um link para o download da mesma, e uma imagem demonstrativa do resultado final.

Link de download da animação: bit.ly/2rQkGzQ

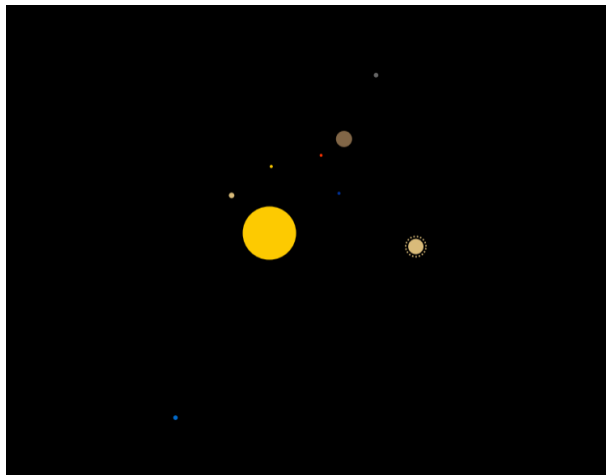


Figura 8 – Demonstração do resultado

4.1 Detalhes extras

O código fonte da animação contém cerca de 962.256 linhas de código, porém o código em linguem C, possui 49 linhas. Para compilar a animação foi necessário cerca de 1 hora. O tempo de duração da animação é de cerca de 14 minutos, possuindo 60140 slides, a 60 slides por segundo.

foram investidos cerca de 30 horas, para desenvolver o trabalho ao todo, sendo a maior parte buscando como iria ser calculada a posição dos planetas.

5 Conclusão

O TikZ é uma ferramenta ótima, para manipulações de formas geométricas que possamos exemplificar utilizando funções matemáticas, porém em seu uso para reproduzir desenhos, se faz um tanto quanto complexa. Tendo outras ferramentas que são de mais fácil uso.

Para reprodução de desenhos mais complexos, é necessário um grande estudo da ferramenta e um grande esforço por parte do usuário para se aprimorar na ferramenta, e conseguir executar o desenho.

O uso da ferramenta é interessante para desenhos que sejam mais simples, que não tenham tantas irregularidades e preferencialmente que possam ser expressadas de formas matemáticas, pois a ferramenta dispõe de uma série de funções para este tipo de imagem.

Referências

Gian Ferdinand, *Sistema solar minimalista*,
http://img06.deviantart.net/e750/i/2012/221/9/4/sistema_solar_minimalista_png_by_gianferdinand-d5aidex.png, 2012. Nenhuma citação no texto.

Till Tantau, *TikZ and pgf Manual for version 1.18*,
<https://www.bu.edu/math/files/2013/08/tikzpgfmanual.pdf>, 2007. Nenhuma citação no texto.