



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DO NORTE DE MINAS GERAIS – IFNMG
Criado pela Lei no.11.892, de 29/12/2008
Campus Montes Claros



SARAH EMANUELLE ALVES LINO
JOÃO KENNEDY SOUZA SOARES

RELATÓRIO : TRABALHO PRÁTICO 1
Visualização de Objetos 2D- Visualização de Viewport

Relatório apresentado ao Professor Wagner Ferreira de Barros como parte das exigências de avaliação da disciplina de Computação Gráfica do Curso de Bacharelado em Ciência da Computação do Instituto Federal do Norte de Minas Gerais, Campus Montes Claros.

MONTES CLAROS - MG
2025

Relatório de Desenvolvimento: Visualizador de Objetos 2D com Transformada de Viewport

A computação gráfica é uma área focada na geração de imagens a partir de modelos computacionais. Um dos seus conceitos mais fundamentais é a transformação de coordenadas, que consiste em mapear objetos de um sistema de referência para outro. No contexto da visualização 2D, a Transformada de Viewport é o processo matemático que converte as coordenadas de um objeto, definidas em um sistema abstrato conhecido como "mundo" (world), para as coordenadas de um dispositivo de exibição, como a tela de um monitor, conhecida como viewport.

Essencialmente, define-se uma "janela" (window) no mundo, que delimita a porção da cena que se deseja visualizar. A transformada, então, mapeia essa janela para a viewport, que é a área específica da tela onde a cena será desenhada. Este processo envolve operações de translação e escala para garantir que o conteúdo da window se ajuste corretamente à viewport, permitindo a criação de sistemas de visualização interativos, onde é possível navegar pela cena (movendo a window).

Este relatório descreve o desenvolvimento de um sistema de software, denominado Visualizador de Objetos 2D, criado para implementar e demonstrar na prática os conceitos da Transformada de Viewport.

2. Visão Geral e Objetivos do Projeto

O objetivo central deste projeto foi desenvolver uma aplicação gráfica interativa capaz de renderizar objetos geométricos 2D a partir de uma descrição de cena fornecida em um arquivo. O sistema foi projetado para aplicar a Transformada de Viewport em tempo real, permitindo ao usuário carregar, visualizar e navegar por um conjunto de objetos.

As funcionalidades principais implementadas incluem:

- Carregamento de objetos geométricos (pontos, retas e polígonos) a partir de arquivos no formato XML.
- Renderização dos objetos em uma área de visualização principal (viewport).
- Navegação interativa pela cena, permitindo que o usuário mova a "janela" de visualização utilizando as teclas direcionais do teclado.

- Exibição de um mini mapa que mostra a totalidade dos objetos e destaca a área atualmente em foco na viewport principal.
- Apresentação em tempo real da matriz de transformação 3x3 que mapeia as coordenadas do mundo para a viewport.

3. Desenvolvimento e Tecnologias

Para a construção do projeto, foram utilizadas ferramentas, com foco na simplicidade e na clareza do código.

- Linguagem de Programação: Python 3.12 foi escolhido como a linguagem principal devido à sua sintaxe clara e ao vasto ecossistema de bibliotecas.
- Interface Gráfica: A biblioteca Tkinter, que é o toolkit padrão de GUI para Python, foi utilizada para construir a interface do usuário, incluindo a janela principal, os menus e as áreas de desenho (canvas).
- Análise de Dados (Parsing): O módulo nativo do Python `xml.etree.ElementTree` foi empregado para analisar os arquivos de entrada em formato XML, extraindo as definições da cena e dos objetos geométricos.

A estrutura do código foi centralizada na classe `Visualizador`, que encapsula toda a lógica da aplicação. Suas principais responsabilidades, divididas em métodos, são:

- *init* : Inicializa a janela principal, os componentes da interface (como os canvas da viewport e do mini mapa) e as variáveis de estado.
- *carregar_arquivo*: Realizar o parsing do arquivo XML, extraindo as coordenadas da window e da viewport, bem como a lista de objetos a serem desenhados.
- *window2viewport (Matriz de Transformação)*: Aplica a matriz de transformação a um ponto, convertendo suas coordenadas do mundo para as coordenadas da viewport.
- *desenhar_viewport e desenhar_minimapa*: Métodos responsáveis por iterar sobre a lista de objetos e renderizar nas respectivas áreas de desenho.
- *_mover_e_recarregar*: Atualiza as coordenadas da window em resposta à entrada do usuário (teclas direcionais) e aciona a recalculação da matriz e o redesenho da cena.

4. Implementação da Transformada de Viewport

O projeto utiliza a implementação da transformação de coordenadas, ou seja, essa transformação é realizada através de uma matriz de transformação homogênea 3x3, que combina as operações de escala e translação necessárias para mapear um ponto (xw,yw) do mundo para um ponto (xv,yv) na viewport.

A matriz M utilizada no projeto é definida como:

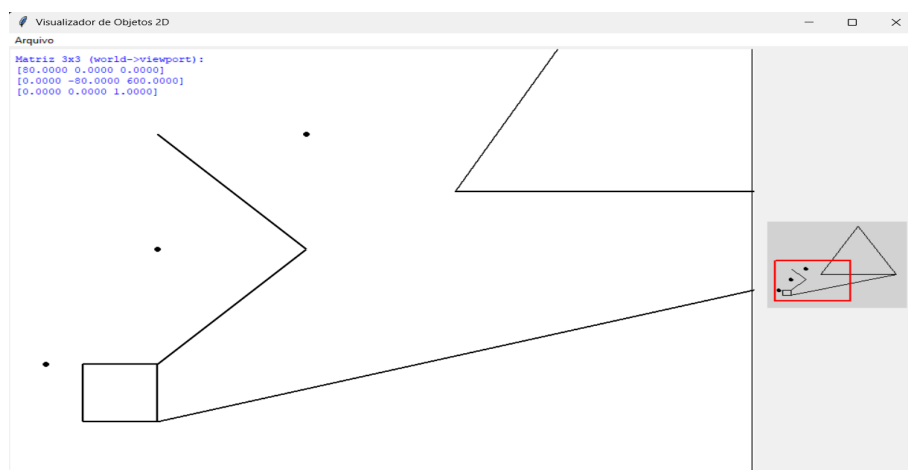
$$M = \begin{bmatrix} S_x & 0 & VPX_{min} - S_x \cdot W_{xmin} \\ 0 & -S_y & VPY_{min} + S_y \cdot W_{ymin} \\ 0 & 0 & 1 \end{bmatrix}$$

Onde os componentes são calculados da seguinte forma:

- $S_x = (vpx_{max} - vpx_{min}) / (w_{xmax} - w_{xmin})$: Fator de escala no eixo X.
- $S_y = (vpy_{max} - vpy_{min}) / (w_{ymax} - w_{ymin})$: Fator de escala no eixo Y.

Um detalhe crucial na implementação é o sinal negativo no fator sy. Isso é necessário porque, na maioria dos sistemas de coordenadas de tela, o eixo Y cresce de cima para baixo, enquanto no sistema de coordenadas do mundo, ele convencionalmente cresce de baixo para cima. A negação do fator de escala efetivamente inverte o eixo Y, corrigindo a orientação dos objetos na tela.

A cada vez que o usuário navega pela cena (movendo a window), os termos de translação da matriz são recalculados, e a cena é redesenhada com a nova transformação, criando o efeito de movimento.



5. Formato dos Dados de Entrada

Para desacoplar a lógica da aplicação dos dados da cena, foi definido um formato de arquivo de entrada baseado em XML. Este formato permite descrever de forma estruturada tanto os parâmetros da visualização quanto os objetos geométricos.

A estrutura do arquivo XML suporta as seguintes tags:

- **viewport:** Define a área da tela em pixels, com as tags filhas *vpmin* e *vpmax*.
- **window:** Define a região do mundo a ser visualizada, com as tags filhas *wmin* e *wmax*.
- **ponto:** Define um ponto único através de seus atributos *x* e *y*.
- **reta:** Define um segmento de linha composto por dois elementos *ponto*.
- **polígono:** Define um polígono fechado composto por três ou mais elementos *ponto*.

Essa abordagem torna o sistema flexível, permitindo que qualquer cena que siga essa estrutura possa ser carregada e visualizada sem alterações no código-fonte.

6. Geração de XML de Saída

Ao abrir o arquivo de entrada, foi adicionado um recurso que permite ao usuário gerar um arquivo XML contendo os dados originais da cena, juntamente com uma cópia dos objetos geométricos já transformados para as coordenadas da viewport. Essa funcionalidade, cumpre dois objetivos principais:

- Permitir que o usuário inspecione diretamente as coordenadas dos objetos após a aplicação da transformação de viewport, validando se o mapeamento de coordenadas foi realizado corretamente.
- O arquivo de saída pode ser utilizado como uma "cache" de dados renderizados, onde outro sistema gráfico pode carregar a cena com as coordenadas já otimizadas para a visualização, economizando o tempo de processamento da transformação.

O método é implementada dentro da classe Visualizador, onde o ocorre:

1. Transformação dos Objetos: O método *transformar_objetos_para_viewport()* é chamado para aplicar a matriz de transformação (*M*) a todos os pontos de cada objeto (ponto, reta, polígono) da cena. O resultado é uma nova lista de objetos com as coordenadas já em pixels, dentro do sistema de referência da viewport.

2. Criação da Estrutura XML: Uma nova seção é adicionada ao arquivo XML de origem. Essa seção é precedida por um comentário para facilitar a leitura e separação dos dados.
3. Inclusão dos Objetos Transformados: Cada objeto original é replicado na nova seção, mas com suas coordenadas x e y substituídas pelos valores calculados na transformação. Um novo atributo, *coords="viewport"*, é adicionado a cada objeto para diferenciá-lo dos objetos originais do "mundo".
4. Formatação e Salvamento: O arquivo é salvo com um nome e local escolhidos pelo usuário. A formatação com indentação é aplicada para garantir a legibilidade do arquivo de saída.

7. Conclusão

O desenvolvimento do Visualizador de Objetos 2D permitiu a aplicação prática e aprofundada dos conceitos teóricos da Transformada de Viewport, que é uma área utilizada dentro da computação gráfica. O projeto cumpriu com sucesso todos os seus objetivos, resultando em uma ferramenta funcional e interativa que não apenas renderiza objetos 2D, mas também expõe de forma clara e visual o funcionamento da matriz de transformação e seu impacto na cena. A arquitetura do projeto e o uso de tecnologias padrão garantem que o projeto sirva como um excelente material de estudo e uma base sólida para futuras expansões.

7. Git Hub

[Link GitHub - Transformada de Viewport](#)