

Trabalho Prático 1
Disciplina de Computação Gráfica – 2024
Instituto Federal de Educação, Ciência e Tecnologia do Norte de Minas Gerais – Campus Montes Claros

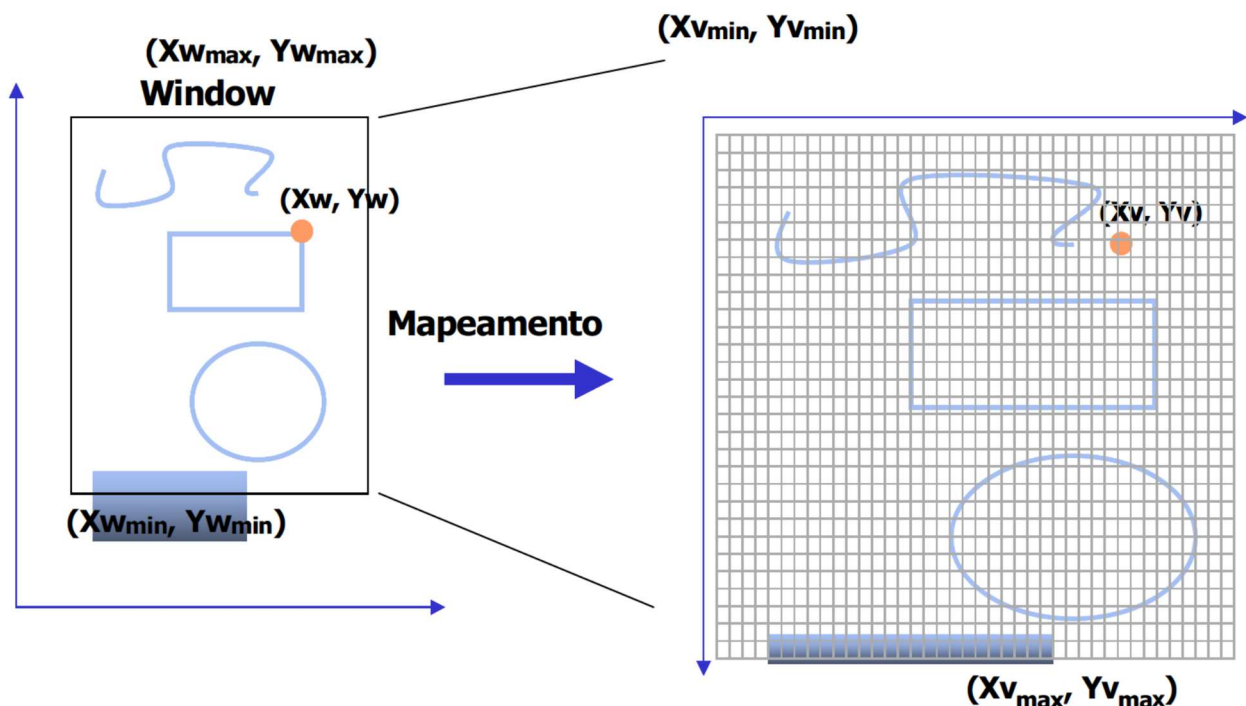
Tema:

Desenvolva um programa que seja capaz de ler um arquivo contendo informações a respeito dos objetos a serem desenhados, da *window* e da *viewport*, e gerar um arquivo de saída contendo os objetos no sistema de coordenadas de *viewport*.

Descrição:

Para desenvolver este trabalho prático, vocês terão que ler um arquivo texto contendo todas as informações de entrada necessárias (será dado um modelo de arquivo texto em xml para que vocês possam ter ideia do formato básico exigido) e, como saída, deverão exibir graficamente os objetos descritos no arquivo de entrada e gerar um arquivo com os dados destes objetos no sistema de coordenadas da *viewport* (que será obtido pela transformada de *viewport* levando-se em consideração os limites da *window* e as coordenadas de mundo dos objetos).

Considerem que sua *viewport* terá sempre uma margem extra de 10pixels em cada lado (à esquerda, à direita, acima, embaixo) isto será utilizado mais adiante para implementação auxiliar na visualização dos algoritmos de *clipping*.



Nesta primeira versão, vocês não precisarão efetuar o recorte dos objetos (*clipping*). Para a implementação da interface gráfica, é sugerido o uso da biblioteca QT.

Sobre os requisitos básicos do trabalho:

- Recomenda-se que o projeto seja desenvolvido em Python usando a biblioteca gráfica QT. A qualidade de código também será avaliada, (estruturem bem o seu código, criem classes para representação dos diferentes objetos do problema e enviem também instruções de compilação e informações a respeito de bibliotecas extras utilizadas. Por exemplo, para ler o arquivo, vocês podem usar um *parser* para leitura de arquivos XML).

- O arquivo de saída também deverá estar no formato XML seguindo o mesmo padrão do arquivo de entrada.
- No caso da utilização da linguagem Python, utilizem “*notebooks*” para documentar seu código. Dessa forma, ficará mais fácil de explicar o que cada parte do seu código faz.
- Criem classes para representar um conjunto básico de objetos: a *window*, o *ponto*, a *reta* e o *polígono*.
- Window: (é como uma janela de recorte do mundo), por enquanto, ela deve armazenar apenas os dados de seus limites (xwmin, ywmin, xwmax, ywmax).
- Ponto2D: 2 coordenadas reais.
- Reta2D: 2 Pontos não coincidentes.
- Polígono2D: lista de, ao menos, 3 pontos onde, o último ponto será ligado ao primeiro. Por enquanto, o polígono poderá ser vazado, ou seja, sem informações do seu interior, também chamado de polígono aramado (ou *wireframe*).

O cálculo da transformada de *Viewport* é definido como uma transformação linear simples entre a *window* e a *viewport*. Para encontrar o valor de x na *viewport*, x_{vp} , realizamos a transformação linear:

$$x_{vp} = \frac{x_w - x_{wmin}}{x_{wmax} - x_{wmin}} \cdot (x_{vpmax} - x_{vpmin})$$

Para encontrar o valor de y na *viewport*, y_{vp} , realizamos uma transformação um pouco diferente, levando em conta a inversão do eixo y , uma vez que você deverá considerar que a origem do seu sistema de visualização se encontra no canto superior esquerdo:

$$y_{vp} = \left(1 - \frac{y_w - y_{wmin}}{y_{wmax} - y_{wmin}}\right) \cdot (y_{vpmax} - y_{vpmin})$$

Observações finais:

Os trabalhos poderão ser feitos em DUPLAS (ou individualmente), porém, códigos iguais entre equipes diferentes não serão aceitos.

É necessário entregar um documento para informar como utilizar seu programa corretamente, isto será utilizado como guia para compilação e teste pelo professor.

A entrega do trabalho deverá ser feita na sua atividade correspondente no *google classroom*, em um **único arquivo** compactado contendo todo o projeto, as instruções de execução/compilação e a identificação e contato dos realizadores da atividade.