



DOCUMENTAÇÃO TRABALHO PRÁTICO
COMPUTAÇÃO GRÁFICA
ANA PAULA PIOVESAN MELCHIORI

EDUARDO CARVALHO LEITE
LUIZ CARLOS BESSA DE LIMA
TAIANE RODRIGUES DE SOUSA

LAVRAS - MG
2022

1. INTRODUÇÃO

Durante a disciplina de Computação Gráfica, de acordo com a ementa, devemos aprender tópicos como representação vetorial e matricial, transformações geométricas, algoritmos de projeção, determinação de superfícies visíveis, representação de cores, métodos de iluminação, algoritmos de conversão matricial de primitivas gráficas, técnicas anti-serrilhado e texturização. E para aplicar tais conhecimentos de forma prática, é sugerida a utilização da ferramenta OpenGL como técnica de ensino.

Segundo o Wikipedia, OpenGL (Open Graphics Library) é uma API livre utilizada na computação gráfica, para desenvolvimento de aplicativos gráficos, ambientes 3D, jogos, entre outros. Assim como Direct3D ou Glide, é uma API (Application Programming Interface), termo usado para classificar uma biblioteca de funções específicas disponibilizadas para a criação e desenvolvimento de aplicativos em determinadas linguagens de programação. À princípio a OpenGL foi produzida para ser utilizada com C e C++, mas pode ser utilizada para diversas outras linguagens com um alto nível de eficiência.

Dessa forma, o objetivo deste trabalho é aplicar conhecimentos teóricos e práticos adquiridos durante a disciplina GAC104 - Computação Gráfica, da matriz curricular de Ciência da Computação na Universidade Federal de Lavras. Neste documento serão abordados levantamento dos requisitos, estratégias de codificação, resultados esperados e conclusão da proposta do projeto sendo a representação de um relógio de pulso atendendo os pré-requisitos definidos na mesma.

2. LEVANTAMENTO DE REQUISITOS

- **Sobre os requisitos**

Os requisitos de sistema ou software são as características e comportamentos esperados de um sistema, e são classificados em dois principais tipos:

- Requisitos funcionais: descrevem as maneiras como o sistema deve se comportar.
- Requisitos não funcionais: conhecidos como atributos de qualidade e descrevem as características gerais do sistema.

- **Identificação dos Requisitos**

RF é utilizado para identificar Requisitos Funcionais e RNF é utilizado para identificar Requisitos Não Funcionais. Ambas siglas vem acompanhada de um número que é o identificador único dos requisitos. Por exemplo, o requisito [RF001] indica um requisito funcional de número 1.

2.1 Requisitos funcionais

RF 001	Fechamento da Janela
DESCRIÇÃO	Ao pressionar a tecla “ESC” do teclado, a janela deve fechar.

RF 002	Transladar relógio acima
DESCRIÇÃO	Ao pressionar a tecla “w” o relógio se move para cima.

RF 003	Transladar relógio abaixo
DESCRIÇÃO	Ao pressionar a tecla “s” o relógio se move para baixo.

RF 004	Transladar relógio à esquerda
DESCRIÇÃO	Ao pressionar a tecla “a” o relógio se move para esquerda.

RF 005	Transladar relógio à direita
DESCRIÇÃO	Ao pressionar a tecla “d” o relógio se move para direita.

RF 006	Rotacionar relógio à direita
DESCRIÇÃO	Ao pressionar o botão direito do mouse o relógio gira para direita.

RF 007	Rotacionar relógio à esquerda
DESCRIÇÃO	Ao pressionar o botão esquerdo do mouse o relógio gira para esquerda.

RF 008	Rotacionar ponteiro de hora do relógio no sentido horário
DESCRIÇÃO	Ao pressionar a tecla “k” o ponteiro rotaciona para direita.

RF 009	Rotacionar ponteiro de minuto do relógio no sentido horário
DESCRIÇÃO	Ao pressionar a tecla “l” o ponteiro rotaciona para direita.

RF 010	Rotacionar ponteiro de hora do relógio no sentido anti horário
DESCRIÇÃO	Ao pressionar a tecla “z” o ponteiro rotaciona para direita.

RF 011	Rotacionar ponteiro de minuto do relógio no sentido anti horário
DESCRIÇÃO	Ao pressionar a tecla “x” o ponteiro rotaciona para direita.

RF 012	Escalar aumentando tamanho do relógio
DESCRIÇÃO	Ao pressionar a tecla “+” o relógio deve aumentar de tamanho.

RF 013	Escalar diminuindo tamanho do relógio
DESCRIÇÃO	Ao pressionar a tecla “-” o relógio deve diminuir de tamanho.

RF 014	Ponteiro de segundo do relógio
DESCRIÇÃO	O ponteiro de segundos deve girar sozinho ao inicializar o programa.

RF 015	Pausar ponteiro de segundo do relógio
DESCRIÇÃO	Ao pressionar a tecla “p” o ponteiro para a movimentação.

RF 016	Ativar/Desativar iluminação ambiente no vidro do relógio
DESCRIÇÃO	Ao pressionar a tecla “i” a iluminação poderá ser ativada ou desativada.

RF 017	Ativar/Desativar textura nos braceletes do relógio
DESCRIÇÃO	Ao pressionar a tecla “t” a textura poderá ser ativada ou desativada.
Adição de alteração	Não havia sido definido anteriormente como seria aplicada a textura, dessa forma foi adicionado mais um requisito funcional para contemplar esta funcionalidade.

2.2 Requisitos não funcionais

RNF 001	Ambiente
DESCRIÇÃO	O desenvolvimento de ambiente 2D ou 3D em OpenGL.

RNF 002	Cores
---------	-------

DESCRIÇÃO	Deve conter colorização diferente para objetos distintos.
------------------	---

RNF 003	Interação
DESCRIÇÃO	Deve conter interatividade com teclado e mouse.

RNF 004	Operações de transformação
DESCRIÇÃO	Deve conter rotação, translação e escala.

RNF 005	Iluminação
DESCRIÇÃO	Deve conter iluminação permitindo ligar/desligar luz.

RNF 006	Animação
DESCRIÇÃO	Deve conter animação de pelo menos 1 objeto.

RNF 007	Textura
DESCRIÇÃO	Deve conter textura em pelo menos 1 objeto.

RNF 008	Estrutura de manipulação
DESCRIÇÃO	Deve conter manipulação de objetos hierárquicos.

RNF 009	Software
DESCRIÇÃO	O programa deve funcionar em qualquer ambiente com sistema operacional Linux com a biblioteca GLUT instalada.

RNF 010	Software
DESCRIÇÃO	Em casos de sistema operacional Windows, deve conter o WSL instalado com uma distro Linux e com a biblioteca GLUT instalada.

RNF 011	Hardware
DESCRIÇÃO	O computador deve conter um monitor com resolução mínima de 1366x768.

RNF 012	Hardware
DESCRIÇÃO	O computador deve conter uma placa de vídeo compatível com o OpenGL.

3. ESTRATÉGIAS DE CODIFICAÇÃO

Neste tópico serão apresentados, além das estratégias de codificação, o ambiente e as tecnologias que serão utilizadas durante o desenvolvimento do projeto proposto.

3.1 Ambiente e tecnologias

Para o desenvolvimento e realização deste trabalho, será utilizado o sistema WSL que trata-se de um recurso disponível no Windows 11 que te permite executar binários e scripts em Linux diretamente no Windows, traduzindo as instruções enviadas para o sistema (as chamadas de sistema ou system calls) para uma instrução válida para o kernel do Windows. Com ele é possível ter um ambiente idêntico a de uma distribuição Linux sem precisar usar uma máquina virtual ou algo do tipo para isso.


A codificação será feita no editor de código Visual Studio Code, utilizando a linguagem de programação C++, vista durante a graduação desde o primeiro contato nas disciplinas básicas como Introdução aos Algoritmos e Estruturas de Dados.

E como já abordado anteriormente, utilizaremos a API OpenGL para aplicar os conceitos vistos durante a disciplina de Computação Gráfica, utilizando a biblioteca GLUT.

De acordo com o Wikipédia, GLUT (OpenGL Utility Toolkit) é uma biblioteca de funcionalidades para OpenGL cujo principal objetivo é a abstração do sistema operacional fazendo com que os aplicativos sejam multiplataforma. A biblioteca possui funcionalidades para criação e controle de janelas, e também tratamento de eventos de dispositivos de entrada (mouse e teclado). Também existem rotinas para o desenho de formas tridimensionais pré-definidas (como cubo, esfera, bule, etc).

3.2 Codificação

Toda biblioteca possui muitas funções e métodos prontos que são disponibilizados para facilitar que o desenvolvimento seja mais rápido e fácil. Na biblioteca GLUT, todos os nomes de funções começam com o prefixo glut. Ao longo da disciplina, foram utilizados uma boa parte dessas funções e também serão utilizados com grande importância para o funcionamento do nosso projeto. Abaixo tem a função main com as principais funções utilizadas da biblioteca com uma breve descrição das mesmas.



```

1  int main(int argc, char* argv[]) {
2      glutInit(&argc, argv);
3      glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
4      glutInitWindowSize(600, 600);
5      glutInitWindowPosition(0, 0);
6      glutCreateWindow("GAC 104 Computação Gráfica - Relógio de Pulso");
7      init();
8      glutDisplayFunc(renderScene);
9      glutReshapeFunc(reshape);
10     glutKeyboardFunc(keyboardEvent);
11     glutMouseFunc(mouse);
12     glutMainLoop();
13     return 0;
14 }

```

Legenda: Função main, ponto de partida para a execução do programa

glutInit(&argc, argv) : Utilizada para inicializar a biblioteca GLUT e recebe como parâmetros as opções de execução via linha de comando (command line options). Ela deve ser chamada ANTES de qualquer rotina GLUT que não seja de inicialização.

glutInitDisplayMode() : Define o modo de exibição inicial da tela.

glutInitWindowSize() : Especifica o tamanho inicial da janela.

glutInitWindowPosition() : Especifica a posição da janela na tela.

glutCreateWindow() : Cria uma janela principal com o nome recebido como parâmetro.

glutDisplayFunc() : Registra a rotina para exibição da janela.

glutReshapeFunc() : Registra a rotina para reexibição da janela.

glutKeyboardFunc() : Registra a rotina para tratamento de eventos gerados pelo teclado.

glutMouseFunc() : Registra a rotina para tratamento de eventos originados pelo clique do mouse.


glutMainLoop() : Esta rotina deve ser chamada após a inicialização. O controle do programa é entregue a GLUT que passa a monitorar os eventos e chamar as rotinas de callback apropriadas.

As transformações geométricas são essenciais para a construção de um programa interativo e animado em OpenGL. Por meio dos métodos **glRotatef()**, **glTranslatef()** e o **glScalef()** é possível aplicar rotações, translações e operações de escala. Apoiados por variáveis de controle que são atualizadas nas funções de interação por teclado e mouse, estes métodos permitiram as animações e movimentos de câmera implementados no projeto, além do suporte às operações para renderizar a cena.

Duas importantes funções do OpenGL que são responsáveis por empilhar e desempilhar matrizes de desenho são as **glPushMatrix()** e **glPopMatrix()**. Essas chamadas permitem isolar transformações geométricas. E para a implementação das cores dos objetos será utilizado o método de definição de cor **glColor3f()**, que recebe três parâmetros do tipo float que integra a cor no formato RGB.

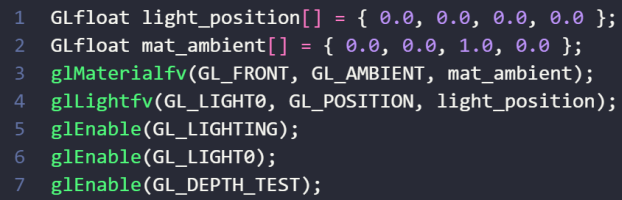
Além das funções principais de um programa em OpenGL utilizando a GLUT, também iremos aproveitar, como estratégia, algumas funções/implementações que realizam o procedimento que esperamos, disponibilizados nos exemplos do redbook.

Abaixo mostraremos algumas pretensões de utilização em nossa estratégia:



```
1 void makeCheckImage(void) {
2     int i, j, c;
3
4     for (i = 0; i < checkImageHeight; i++) {
5         for (j = 0; j < checkImageWidth; j++) {
6             c = (((i&0x8)==0)^((j&0x8)==0))*255;
7             checkImage[i][j][0] = (GLubyte) c;
8             checkImage[i][j][1] = (GLubyte) c;
9             checkImage[i][j][2] = (GLubyte) c;
10            checkImage[i][j][3] = (GLubyte) 255;
11        }
12    }
13 }
```

Legenda: Função que implementa a textura quadriculada



```
1 GLfloat light_position[] = { 0.0, 0.0, 0.0, 0.0 };
2 GLfloat mat_ambient[] = { 0.0, 0.0, 1.0, 0.0 };
3 glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient);
4 glLightfv(GL_LIGHT0, GL_POSITION, light_position);
5 glEnable(GL_LIGHTING);
6 glEnable(GL_LIGHT0);
7 glEnable(GL_DEPTH_TEST);
```

Legenda: Iluminação ambiente

4. RESULTADOS ESPERADOS

Primeiramente, devemos instalar as dependências necessárias para a preparação do ambiente de desenvolvimento de acordo com as ferramentas definidas.

Inicialmente, é feita a configuração no WSL visto que optamos por utilizá-lo pela questão de inconsistências durante o desenvolvimento separadamente em ambiente Windows e Linux, e para evitar esse tipo de problema o WSL simula dentro do sistema operacional Windows um ambiente Linux, onde todos da equipe e qualquer usuário Linux consegue executar o programa tendo os requisitos não funcionais, referentes a hardware e software, definidos neste documento.

4.1 Configuração do ambiente

O WSL dá suporte inicialmente apenas para um terminal, porém com algumas configurações adicionais é possível instalar o suporte para aplicativos de GUI do Linux

Pré-requisitos

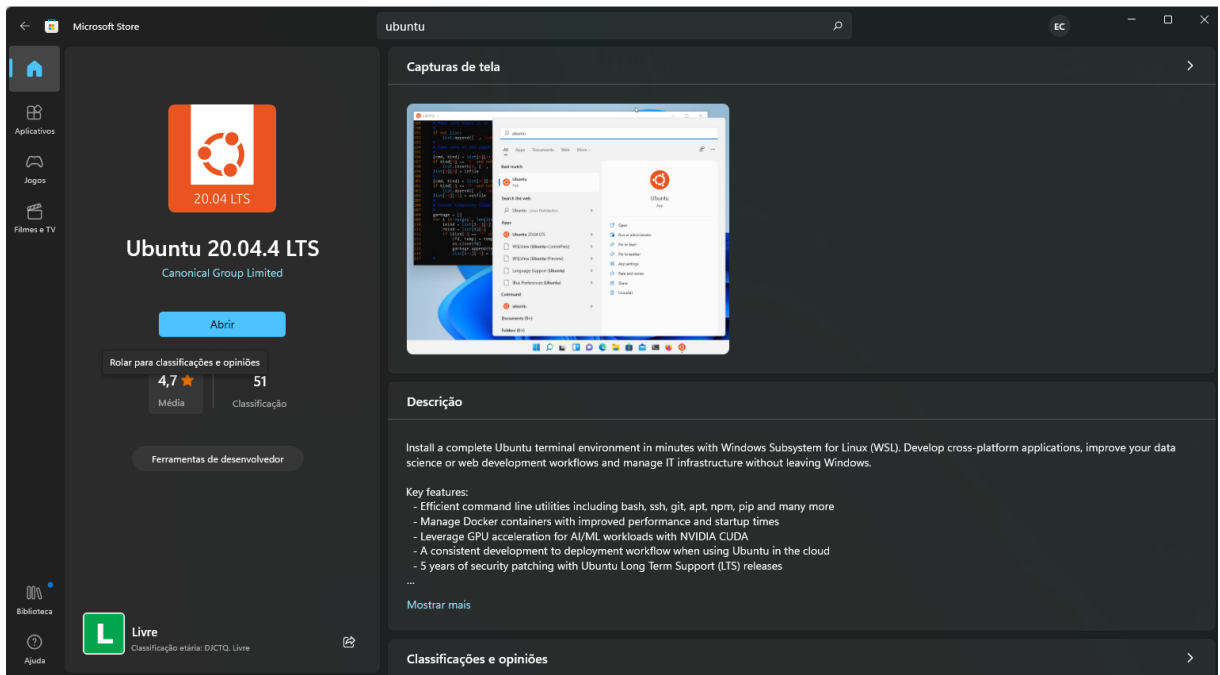
- É necessário estar no Windows 11 Build 22000 ou posterior para acessar esse recurso.
- Driver instalado para vGPU

Para executar aplicativos de GUI do Linux, primeiro você deve instalar o driver correspondente ao seu sistema abaixo. Isso permitirá que você use uma GPU virtual (vGPU) para que você possa se beneficiar da renderização do OpenGL acelerada por hardware.

- [Intel Driver de GPU para WSL](#)
- [AMD Driver de GPU para WSL](#)
- [NVIDIA Driver de GPU para WSL](#)

Passo 1: Instalar o WSL 2

- Pode ser instalado diretamente da Microsoft Store ou via terminal;
- Basta escolher a distribuição de preferência e fazer a instalação;



Passo 2: Instalar aplicativos X11

- X11 é o sistema de janelas do Linux e essa é uma coleção diversa de aplicativos e ferramentas que são fornecidos com ele, como xclock, calculadora xcalc, xclipboard para corte e colagem, xev para teste de eventos etc. Consulte os documentos de x.org para obter mais informações.
- Rodar comando abaixo já no terminal WSL instalado.

```
$ sudo apt install x11-apps -y
```

É possível testar se deu certo, executando algumas ferramentas nativas do Linux. Por exemplo:

- xcalc;
- xclock;
- xeyes.

Passo 3: Configurar variável DISPLAY

- No terminal do WSL abra o arquivo de configuração do seu shell. ex .bashrc, .zshrc...

```
$ nano .zshrc
```

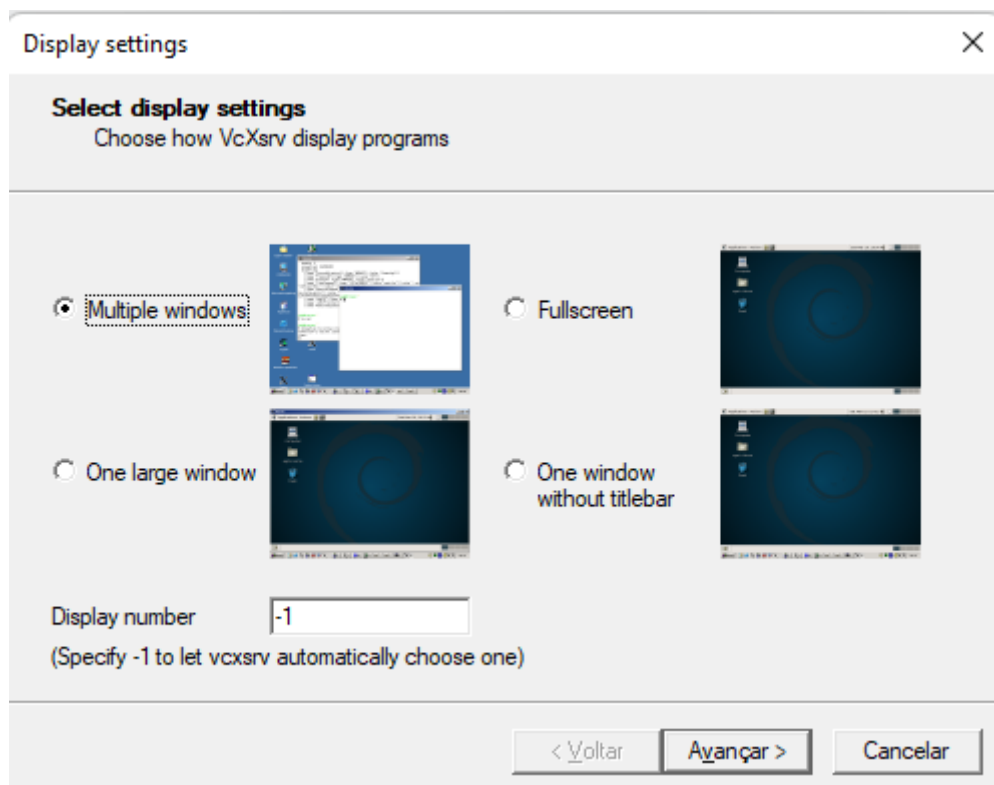
- Adicione o export abaixo:

```
export DISPLAY=$(grep -m 1 nameserver /etc/resolv.conf | awk '{print $2}'):0.0
```

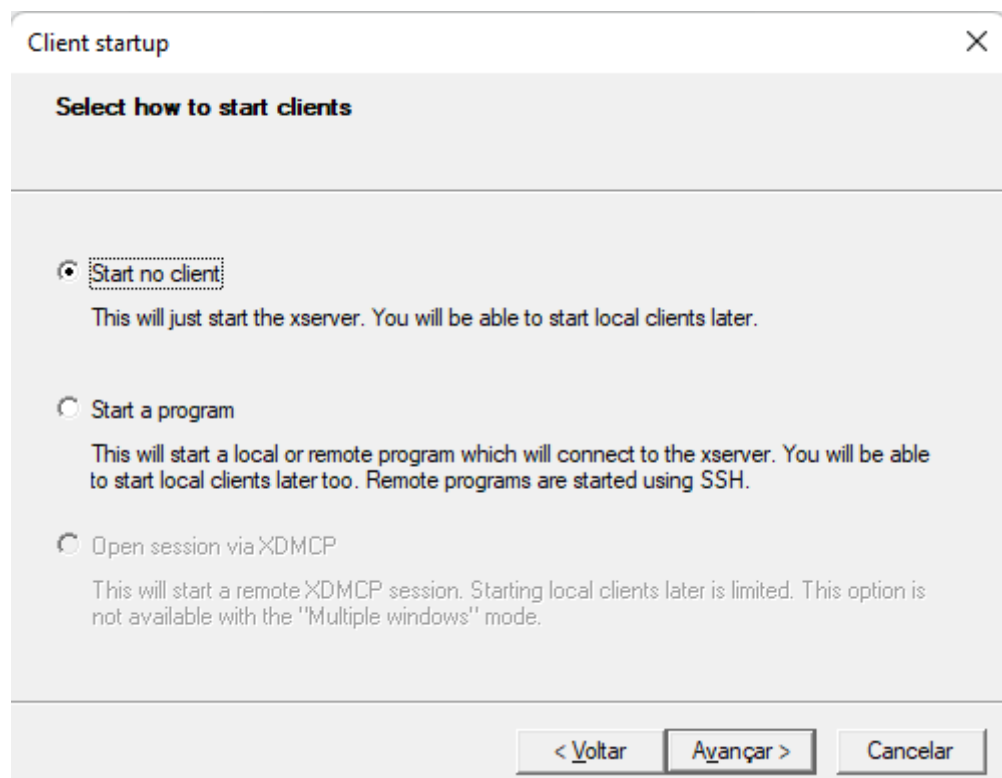
Informações adicionais

Se estiver utilizando uma versão anterior do Windows 11 ou o Windows 10, ou mesmo após seguir esses passos, receber erro de display, basta instalar o VcXsrv, para conseguir exibir GUIs do Linux no Windows.

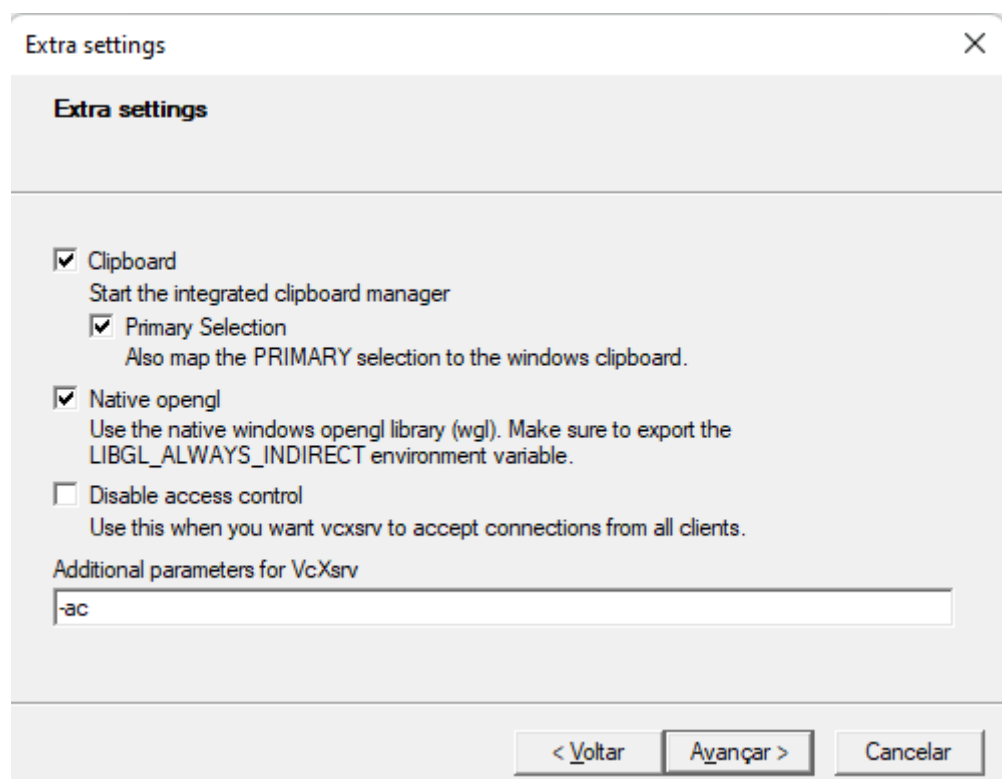
- Instalar VcXsrv Windows X Server;
- Executar o aplicativo xLaunch que foi instalado com o pack acima;
- Selecionar Multiple windows e preencher o display number com -1;



- Avançar e selecionar “start no client”;



- Avançar novamente, selecionar as três primeiras opções e preencher o campo de parâmetros adicionais com “-ac”;

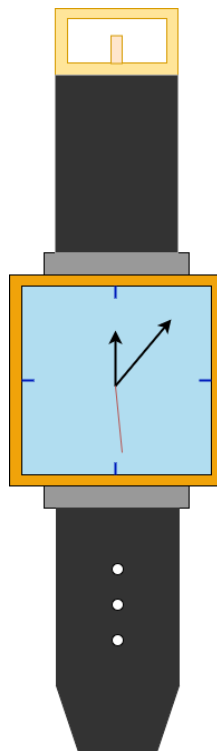


Após as devidas configurações do ambiente, será definida a estrutura inicial de um programa em OpenGL utilizando a biblioteca GLUT e a linguagem de programação C++.

Sabendo que o OpenGL é uma API gráfica e não uma plataforma própria, ele requer uma linguagem para operar e escolhemos a C++ para o desenvolvimento deste trabalho pela familiaridade dos integrantes da equipe com a mesma, visto que como dito anteriormente, é a linguagem de contato inicial durante a graduação. Desse modo, esperamos conseguir aplicar os conhecimentos adquiridos nas disciplinas base, complementando com os conceitos que foram obtidos em Computação Gráfica como parte das exigências apresentadas no plano de curso.

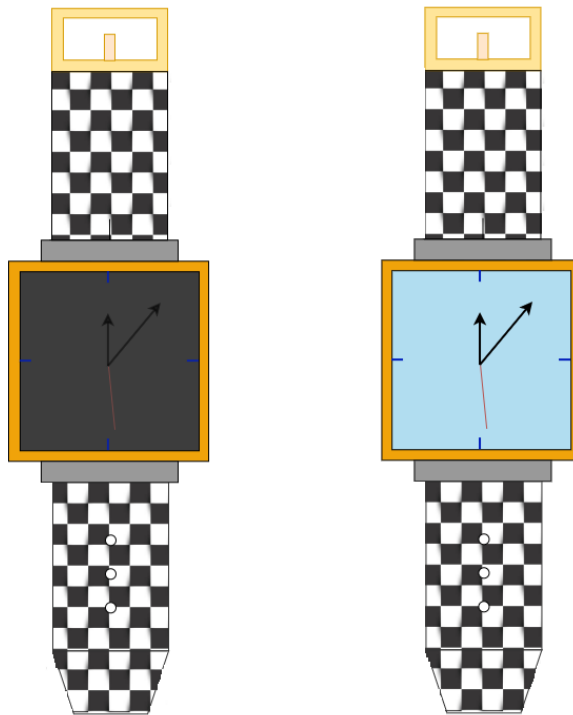
4.2 Resultados a serem alcançados

Como definido na proposta, é esperado como resultado desenhar um relógio de pulso em 2D aplicando os conceitos de colorização em cada parte do objeto.



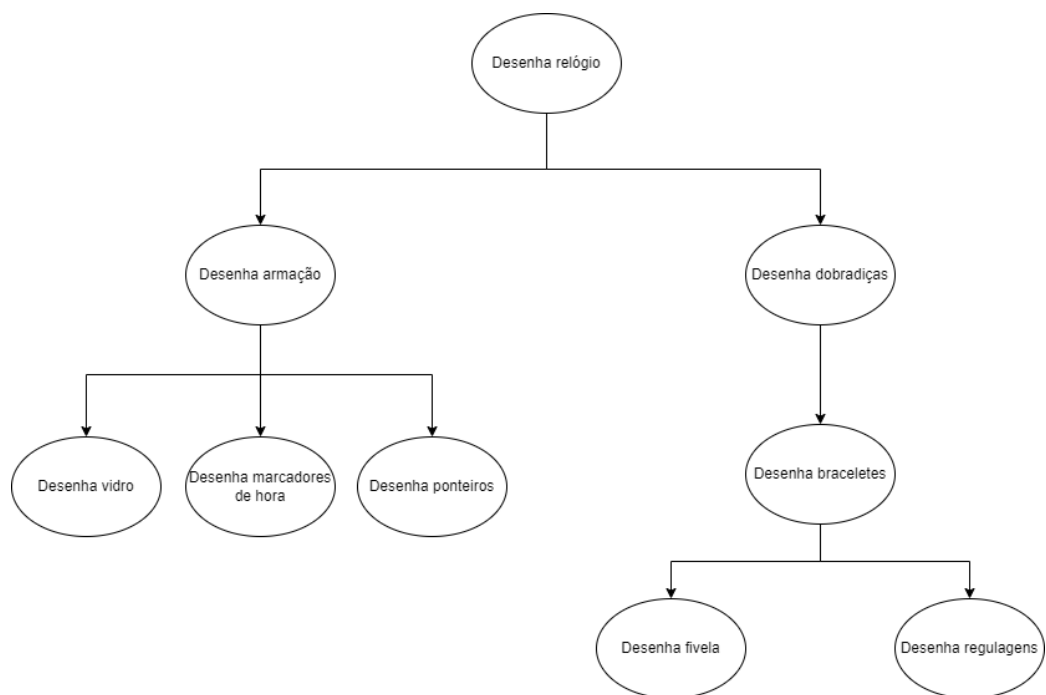
Legenda: Relógio idealizado

Abaixo temos um protótipo de exemplo representando o objetivo a ser alcançado atribuindo a textura e iluminação:



Legenda: Relógio esquerdo com iluminação desativada e relógio direito com iluminação ativada. ambos com textura nos braceletes

O relógio será dividido em 8 partes que serão construídas. Abaixo temos uma representação hierárquica:



Legenda: Estrutura hierárquica dos objetos

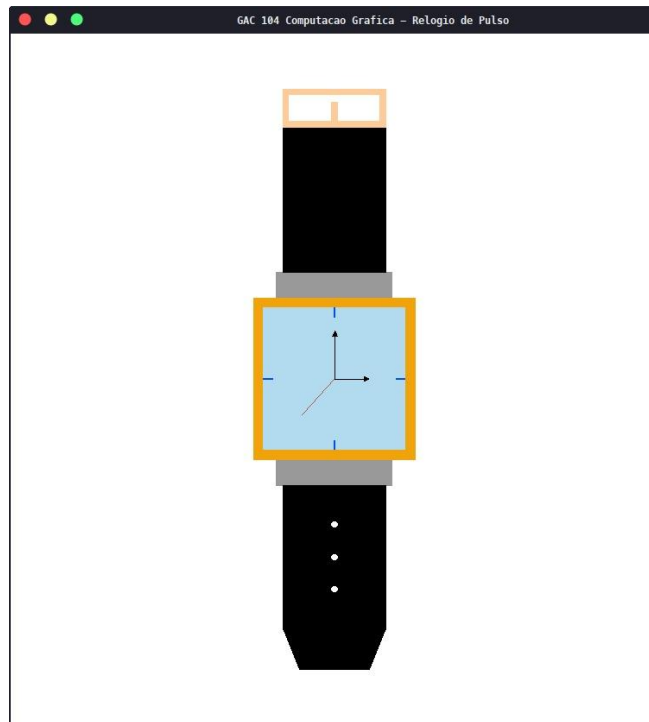
E descrevendo de forma mais detalhada, a armação do relógio será na cor dourado, o vidro será azul claro, os marcadores de hora serão na cor azul escuro, os ponteiros de hora e minuto serão na cor preto, as dobradiças serão na cor cinza claro, a fivela será na cor laranja claro, e a regulagens do bracelete serão brancos dando alusão de furos. A textura será aplicada no bracelete, nas cores preto e branco como em um tabuleiro de dama.

Para manipular transformações geométricas no relógio como um todo será utilizado a interação do teclado e mouse. Para operação de escala será utilizada a tecla “+” para aumentar o objeto e a tecla “-” para diminuir. Para a translação serão utilizadas as teclas “a”, “d”, “w”, “s”, para movimentar o relógio para esquerda, direita, cima, baixo, respectivamente. (a translação ocorre por padrão ao aplicar as demais transformações). A rotação será manipulada pelos botões do mouse. O botão direito irá rotacionar o relógio para direita e o botão esquerdo irá rotacionar para esquerda.

Para manipular as interações com os ponteiros do relógio como alusão ao ajuste do tempo, será utilizada a tecla “k” para rotacionar no sentido horário o ponteiro de hora e a tecla “l” para rotacionar no sentido horário o ponteiro de minuto. Ao executar o código, o ponteiro dos segundos já estará em funcionamento e por essa razão será utilizada a tecla “p” para pausar esse ponteiro. A parte interna do visor do relógio contemplará com a iluminação ambiente e será ativado/desativado com a tecla “i”. E com a tecla “t” será habilitada/desabilitada a textura nos braceletes do relógio.

Desse modo, será finalizado o relógio e suas funcionalidades dentro dos critérios dos requisitos, alcançando as competências para conclusão da disciplina de Computação Gráfica.

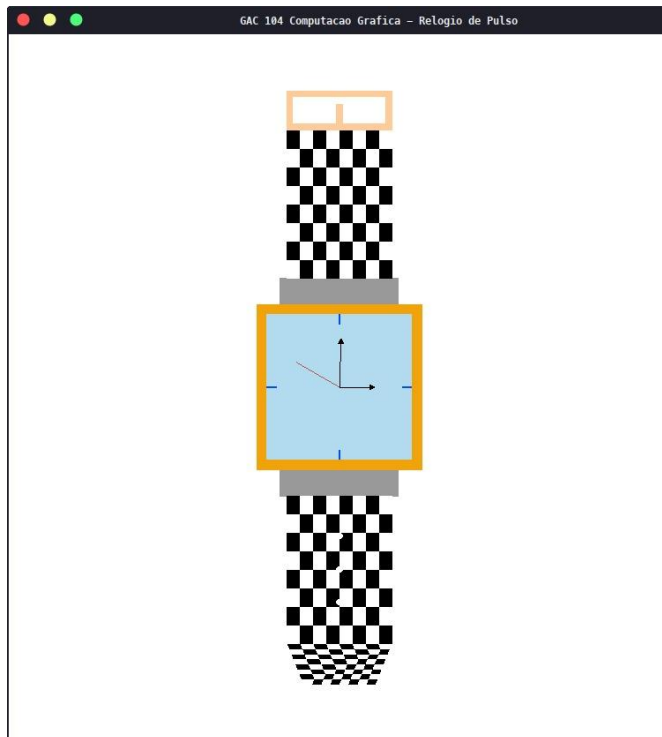
4.3 Resultados alcançados



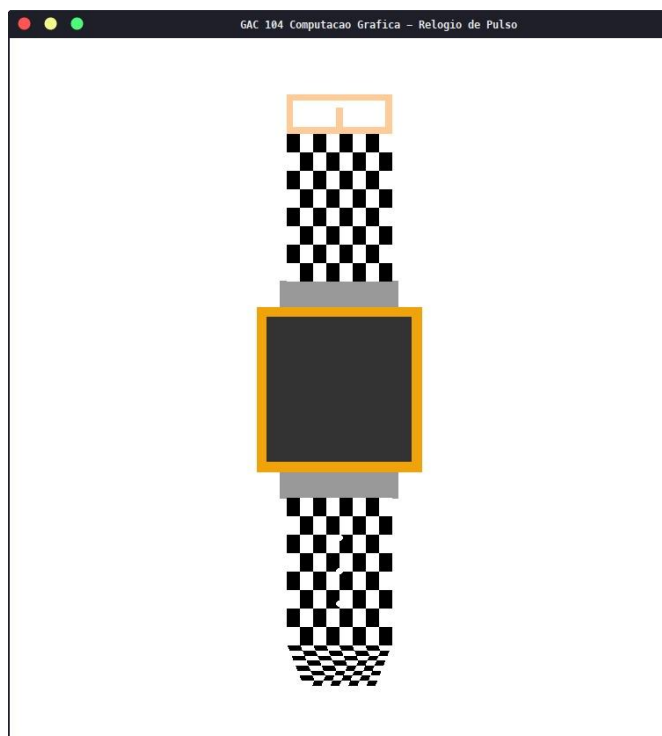
Legenda: Relógio ao iniciar programa



Legenda: Relógio com iluminação desativada



Legenda: Relógio com textura aplicada nos braceletes



Legenda: Relógio com iluminação desativada e textura aplicada nos braceletes

5. CONCLUSÃO

Ao longo do estudo da disciplina Computação Gráfica e execução deste trabalho, é evidenciada a extrema relevância da mesma para a nossa formação, visto que se tornou uma área fundamental para diversos setores que envolvem processamento de imagens, efeitos especiais, criações em 3D e desenvolvimento de interfaces gráficas.

O desenvolvimento deste projeto proporciona a aplicação prática de todos os conceitos e exercícios vistos em sala de aula, como forma de absorver e entender os conteúdos básicos da área. E nossa proposta teve como objetivo contemplar os pré-requisitos definidos para o projeto final da disciplina.

Através da utilização do OpenGL neste projeto é possível aplicar as principais manipulações desejadas, sendo elas: transformação geométrica, colorização, interação do teclado e mouse, iluminação e textura. E para a realização, é consultado os materiais e redbooks disponibilizados pela professora, Ana Paula Piovesan Melchiori, que está ministrando a disciplina atualmente, e conteúdos na internet referentes ao assunto para dar suporte a leitura da documentação do OpenGL que não é muito detalhada.

Com isso, concluímos até o momento que com a biblioteca GLUT e a API OpenGL conseguimos projetar objetos e interações tanto em baixo nível quanto em alto nível, até mesmo próximo de jogos/interfaces muito conhecidas e que por ser uma área muito ampla, produtos mais robustos exigem maior complexidade e conhecimento para manipulação.

Como dificuldades até o momento, tivemos a de entender o funcionamento da textura de acordo com os exemplos disponibilizados no site do redbook, tínhamos o objetivo de mudarmos para a cor que gostaríamos e dessa forma definimos que inicialmente não é aplicada a textura mantendo a cor definida e após habilitá-la, mudaria para uma textura quadriculada. E também tivemos dificuldade em diminuir a opacidade da iluminação para que após desativada não ficasse parecendo uma cor sólida.

REFERÊNCIAS

Wikipedia contributors. (n.d.). OpenGL. Wikipedia, The Free Encyclopedia. <https://pt.wikipedia.org/w/index.php?title=OpenGL&oldid=63672178>

Ementa Computação Gráfica UFLA. (n.d.). Ufla.Br. Retrieved August 27, 2022, from https://sig.ufla.br/modulos/publico/matrizes_curriculares/gerar_ementa.php?cod_disciplina=774

O que é Windows Subsystem for Linux (WSL)? (n.d.). Com.br. Retrieved August 27, 2022, from <https://www.treinaweb.com.br/blog/o-que-e-windows-subsystem-for-linux-wsl>

Wikipedia contributors. (n.d.). GLUT. Wikipedia, The Free Encyclopedia. <https://pt.wikipedia.org/w/index.php?title=GLUT&oldid=62416533>

Zanette, A. (2017, May 30). Framework x Biblioteca x API. Entenda as diferenças! Bencode. <https://bencode.com.br/framework-biblioteca-api-entenda-as-diferencas/>

craigloewen-msft. (n.d.). Executar aplicativos de GUI do Linux com o WSL. Microsoft.com. Retrieved August 29, 2022, from <https://docs.microsoft.com/pt-br/windows/wsl/tutorials/gui-apps>