



**Wellington Azevedo Barros**

## **PROJETO DE COMPUTAÇÃO GRÁFICA**

### **COLHEITA NO CÉU**

**Prof<sup>a</sup>: Dra. Regina Célia Coelho**

# **1. INTRODUÇÃO**

## **1.1 Motivação**

A colheita por mãos humanas aos poucos esta sendo substituída por máquinas capazes de realizar esta tarefa com qualidade e economia de tempo, no brasil onde há fazendas com um grande volume de pessoas que ainda executam esse tipo de trabalho manual, surgiu uma motivação para criação deste projeto, entretando este projeto é uma colheita no céu.

## **1.2 Resumo**

O objetivo do projeto é clicar em uma árvore e o personagem ir até ela e colher as frutas, depois que colheu todas as frutas clicadas dispara a animação final e o personagem levara as frutas para um lugar específico.

## **2. Objetivo**

O objetivo como uma função é simples colher todas as frutas clicadas com o botão do mouse, e retornar a um ponto específico com o balde de frutas em mãos. O personagem não é controlável, ou seja, suas tarefas apos um clique é automática e ele persegue o clique do mouse ao aparecer uma fruta na árvore, enquanto o humanoide esta buscando uma fruta pode-se clicar em outras arvores para aparecerem mais frutas essas frutas são adicionadas em um vetor de filas onde a busca será feita na ordem de FIFO. Após colher todas frutas, se e somente se, todas frutas estiverem colhidas e ainda pelo menos uma fruta tem que ser colhida para poder finalizar a tarefa, assim então no menu de opção terá a opção de finalizar tarefa.

## **2. Métodos, técnicas e conceitos**

### **2.1 Cenário**

O cenário foi criado pensando na facilidade do clique do mouse em tratar eixos negativos na lógica de programação, então o cenário é desenhado no eixo positivo X e Z onde  $X_{min} = 0$ ,  $X_{max} = 200$  e  $Z_{min} = 0$  e  $Z_{max} = 200$ . Uma cerca criada com cubos foi criada em volta dessas coordenadas delimitando o final do cenário, as árvores foram criadas em volta da posição inicial do personagem, o clique para aparecer uma fruta é avaliado por uma função `checaCliqueFruta` que poderá desenhar ou não a fruta na árvore então se clicar no solo, no

caule ou qualquer outro lugar que não seja nas folhas das árvores a fruta não aparece. O solo foi criado com a função `GL_QUADS` na cor verde. Para dar uma impressão de estar dentro do céu foi criada uma esfera gigante em torno do cenário e colocado uma textura com parâmetro de espera isso da uma impressão de fundo se movimentando quando o personagem anda.

## **2.2 Personagem**

O personagem foi criado em partes separadas como a seguir: braço direito, braço esquerdo, perna direita, perna esquerda, peitoral e rosto, usando apenas cubo no peitoral do personagem e esferas nas outras partes do corpo dando um realismo maior ao personagem, essas partes são escalados em x, y e z de forma que se pareçam mesmo com os membros de um humanoide. O uso de hierarquia em cada parte do corpo foi criada, então cada braço e perna rotacionam independentes e ainda os braços são divididos em parte superior, parte inferior e pulso (com mãos e dedos), logo as rotações nos braços são 6 rotações distintas, já as pernas são divididas em parte superior, parte inferior e pé, logo as rotações nas pernas são 5 rotações distintas. O nariz foi criado usando um cone, o chapéu foi dividido em duas partes, a primeira parte é a aba que foi criada com um `GL_POLYGON` de circunferência usando um for de  $i=0$  até 360, a outra parte foi criada com uma esfera dando a impressão do encaixe na cabeça do personagem. Os dedos foi dividido em 3 partes e o dedão em 2 com a intenção de rotação para abrir e fechar porém não consegui a tempo.

## **2.3 Câmera e Projeção**

A projeção utilizada no projeto foi a projeção perspectiva com  $Z_{near}$  e  $Z_{far}$  de 0.1 a 500.0, na declaração da função tem uma variável global que controla o zoom no personagem.

A câmera se move com rotações nas teclas e ainda acompanha o personagem quando o mesmo se desloca no cenário, nos parâmetros da câmera foi declarado como ponto focal as posição do personagem, e a posição da câmera de início é  $x=15, y=15$  e  $z=30$  conforme o personagem anda esses parâmetros são somados aos parâmetros anterior e isso faz com que a câmera irá sempre acompanhar o personagem onde é que ele esteja.

## **2.4 Clique do Mouse e Direção do Personagem**

O clique do mouse na projeção perspectiva, foi possível usando a função `gluUnProject` e habilitando o teste de profundidade, essa função analisa toda a área do clique e sua profundidade gerando uma posição precisa a cada coordenada passada nessa função em

x,y e z, quem chama essa função na main é a glutPassiveMotionFunc que pegar todas coordenadas do mouse em seu movimento.

A direção do personagem, como ele anda no plano XZ, usei a posição atual do personagem em X e Z e a posição final ou de algum clique, então se faz o delta X e delta Z desta distância e aplica o arco tangente e ele mostrará o ângulo correto que o personagem deveria percorrer.

## **2.5 Iluminação**

Na iluminação foi criada duas fontes de luz uma no cenário, na posição do centro do mesmo e a uma altura de y suficiente para iluminar todo o cenário e dar uma impressão de estar literalmente no céu. E a outra no personagem conforme ele anda uma iluminação direcional “spot” o acompanha com um ângulo de iluminação pequeno.

## **2.6 Textura**

Textura foi criada em todo o personagem utilizando tons de pele e roupa para dar um maior realismo, no caule da árvore uma textura de casca e no esfera que engloba todo o cenário dando a impressão de se estar no céu.

## **2.7 Como rodar o trabalho ?**

O projeto foi criado na plataforma Windows, logo o caminho das texturas estão conforme o windows, ao criar o projeto terá que importar no mesmo personagem.cpp e os cabeçalhos personagem.h e mapa.h (por causa das texturas no mapa) deixei as funções deles na main.cpp. Depois que construído o projeto clique em qualquer árvore irá aparecer uma ruta e o personagem irá colher esta fruta depois que repetir isso quantas vezes quiser abra o menu com o botão direito e clique em finalizar tarefa.

## **2.8 Funções que compõem o programa**

“main.cpp”

- getBitmapImageData → carrega a imagem para textura.
- CarregaTextura → utiliza getBitmapImageData e faz um contador de texturas.
- aplicaTextura → aplica a textura em um objeto
- iniciaTextura → utiliza CarregaTextura com as url das imagens.
- iniciaArvore → seta as posição de cada árvore criada.
- iniciaFruta → seta as posição de cada fruta criada.

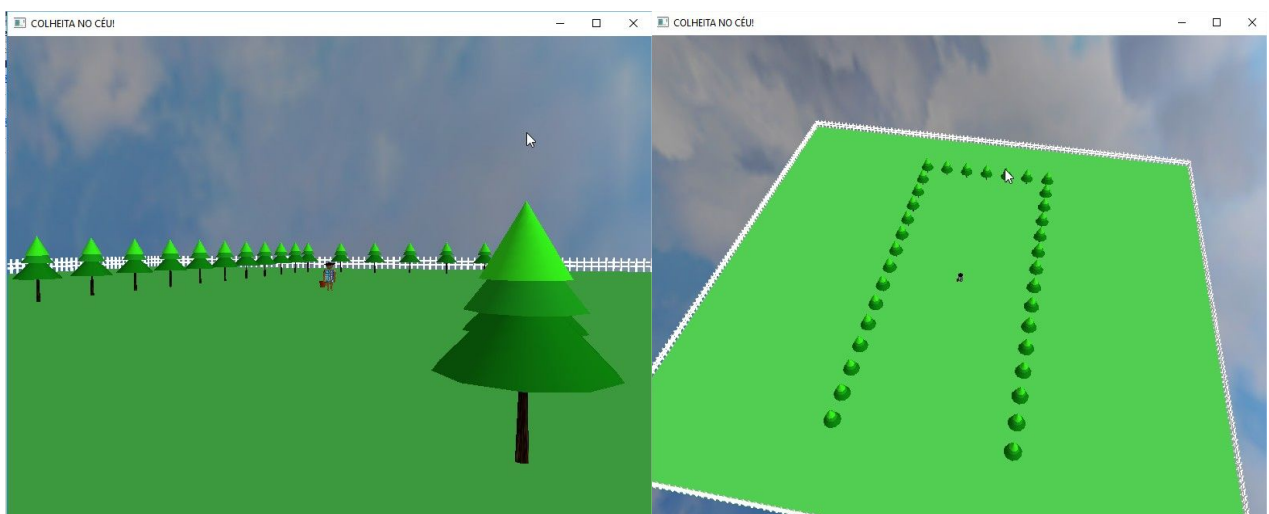
- `desenhaArvore` → desenha a árvore passando suas posições.
- `desenhaFruta` → desenha a fruta passando suas posições.
- `checaCliqueFruta` → checa se o clique do mouse foi em uma árvore.
- `resetaVariaveisMain` → reseta todas variaveis de rotações da main.
- `iniciaAgricultor` → cria a estrutura personagem e aloca memoria.
- `iniciaIluminacao` → inicia a iluminação do cenário e personagem.
- `desenhaCerca` → desenha toda a cerca no cenário.
- `desenhaChao` → denha o chão no plano XZ.
- `desenhaHumanoide` → desenha o humanoide por completo.
- `Display` → chamada na main é onde redesenha toda a cena.
- `cursor` → direção e profundidade do clique.
- `Reshape` → mantem a proporção da tela quando aumento ou diminui.
- `Keyboard` → funções do teclado.
- `GerenciaMouse` → funções do mouse.
- `criaMenu` → cria o menu de exibição.

“personagem.cpp”

- `cameraPerspectivaHumanoide` → estabelece a câmera atual no personagem.
- `andar` → anda na direção em que o angulo foi calculado.
- `animaHumanoide` → animação completa do personagem.
- `resetaTodosAngulosEstaticos` → reseta todos ângulos quando começa a andar.

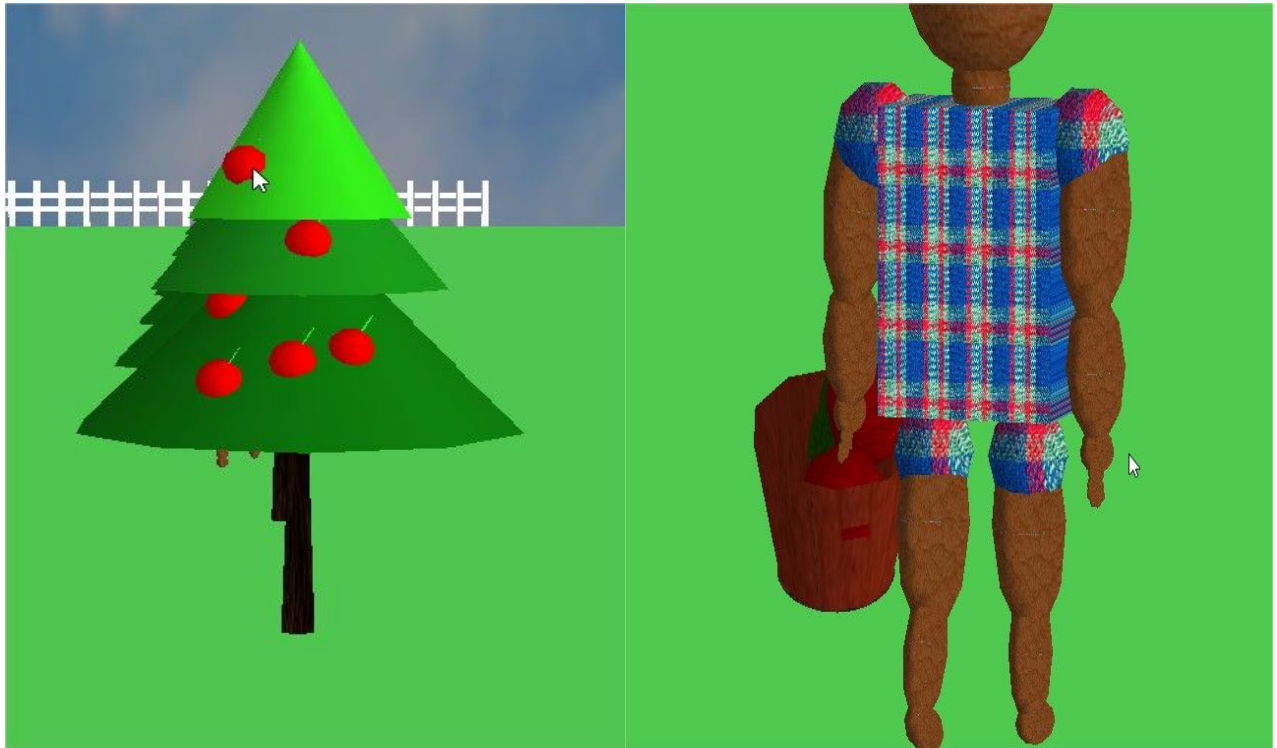
### 3. Resultados e discussões

O cenário em imagens frontal vista do chão e visto de cima cima.



No fundo do cenário se tem a sensação de estar no céu com a textura de nuvens.

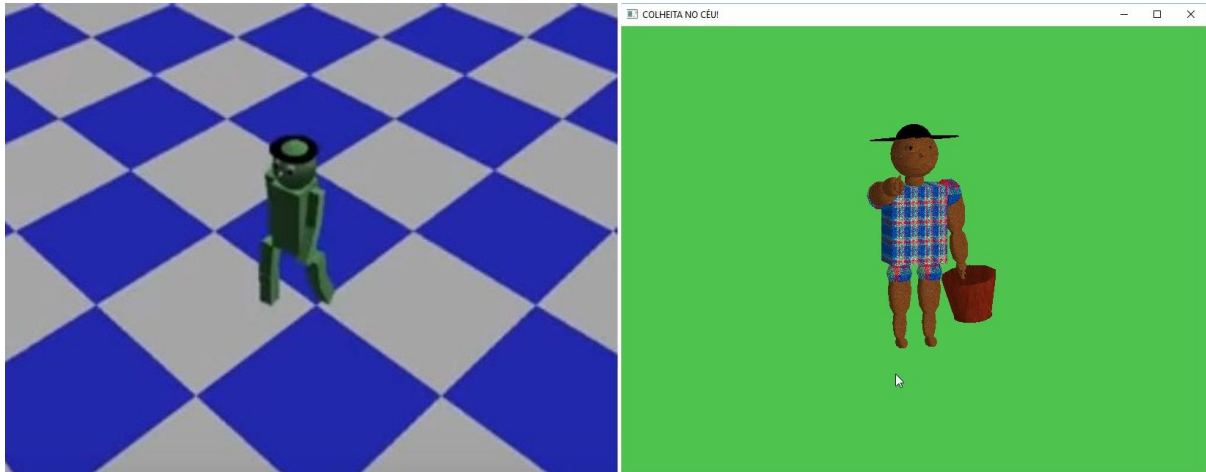
As árvores podem receber quantos cliques puderem e as frutas irão aparecer exatamente onde foi clicado e o personagem irá até ela recolher as frutas. No balde ficou uma simulação e não uma animação de frutas dentro do balde.



O clique do mouse pode ser dado em qualquer parte do cenário porém só irá aparecer frutas se for clicado em uma árvore e para isso foi criado valores de max e min de x,y e z na árvore onde a fruta aparecerá fora dela jamais aparecerá frutas, porém em algumas bordas ele reconhecerá que não está na árvore, existe um pequeno erro de clique a se considerar na árvore.

O personagem foi desenhado primeiramente com cubos depois aprimorado para esferas dando uma realidade bem melhor até de seus movimentos, porém quando desenhei tive problemas com a referência de rotação já que tudo era criado no eixo (0,0,0) depois transladado para o devido lugar, pois bem não percebi isso e já no desenho montado ao realizar as rotação com o personagem parado, estava tudo errado, demorei para descobrir o erro que eu estava cometendo, mas no final consegui resolver.

Abaixo está o antes e o depois, respectivamente do personagem.



Os maiores problemas enfrentados no projeto foi implementar o clique na projeção perspectiva isso de fato me custou muito tempo, mas no final acabei acertando o que faltava e funcionou perfeitamente. Outro problema foi direcionar o personagem para o clique, porque muitas vezes ele andava de costas ou de lado, então foi feito um estudo geométrico para obter o angulo certo.

## 3.1 Opções do teclado

- **Braço Esquerdo**
  - Tecla 'Q' ou 'q', rotaciona braço inteiro em X.
  - Tecla 'A' ou 'a', rotaciona braço inteiro em Z.
  - Tecla 'W' ou 'w', rotaciona antebraço em X.
  - Tecla 'S' ou 's', rotaciona pulso em X.
  - Tecla 'E' ou 'e', rotaciona pulso em Y.
  - Tecla 'D' ou 'd', rotaciona pulso em Z.
- **Braço Direito**
  - Tecla 'R' ou 'r', rotaciona braço inteiro em X.
  - Tecla 'F' ou 'f', rotaciona braço inteiro em Z.
  - Tecla 'T' ou 't', rotaciona antebraço em X.
  - Tecla 'G' ou 'g', rotaciona pulso em X.
  - Tecla 'H' ou 'h', rotaciona pulso em Y.
  - Tecla 'U' ou 'u', rotaciona pulso em Z.
- **Perna Esquerda**
  - Tecla 'I' ou 'i', rotaciona perna inteira em X.
  - Tecla 'J' ou 'j', rotaciona perna inteira em Z.
  - Tecla 'O' ou 'o', rotaciona canela em X.
  - Tecla 'K' ou 'k', rotaciona pé em X.

- Tecla 'p' ou 'p', rotaciona pé em Y.
- **Perna Direita**
  - Tecla 'L' ou 'l', rotaciona perna inteira em X.
  - Tecla 'M' ou 'm', rotaciona perna inteira em Z.
  - Tecla 'N' ou 'n', rotaciona canela em X.
  - Tecla 'B' ou 'b', rotaciona pé em X.
  - Tecla 'V' ou 'v', rotaciona pé em Y.
- **ZOOM**
  - Tecla '+', ZOOM-IN.
  - Tecla '-', ZOOM-OUT.
- **Rotação da câmera**
  - Tecla 'X' ou 'x', rotaciona câmera em X.
  - Tecla 'Y' ou 'y', rotaciona câmera em Y.
  - Tecla 'Z' ou 'z', rotaciona câmera em Z.

## 3.2 Opções do mouse

- **Clique esquerdo**
  - Se for um clique em uma árvore e aparecer uma fruta o personagem irá buscar a fruta.
- **Clique direito**
  - Abre o menu de opções.
- **Menu**
  - **1 → finaliza a tarefa**
  - **2 → sai do programa**
  - **3 → a partir desse ponto os comandos serão apenas feito por teclado. (opções do mouse)**

## 3.3 Conclusão

Nesse trabalho final de curso foi implementado um cenário e um personagem que percorre o cenário para colher frutas em uma ordem de FIFO e retorna a um ponto com as frutas no balde.

O trabalho feito na disciplina fez com que no processo de desenvolvimento fixássemos tudo que nos havia ensinado e ainda mais com os problemas enfrentados desenvolvemos técnicas de desviar de problemas futuros, a curva de aprendizagem do trabalho no começo é muito difícil de subir, OpenGL depende de uma base teórica muito forte, então conforme as coisas no projeto foram acontecendo a produtividade do mesmo só aumentou, a busca por conhecimento fora do escopo da aula teve que ser feita, isso demonstra a complexidade de se trabalhar com algo que não é simples, porém depende de muita teoria para saber o que acontece como um todo na tela.