

João Pedro Miranda Miguel

Desenvolvimento de um ambiente tridimensional utilizando opengl

São José dos Campos - Brasil

Dezembro de 2017

João Pedro Miranda Miguel

Desenvolvimento de um ambiente tridimensional utilizando opengl

Relatório apresentado à Universidade Federal
de São Paulo como parte de um projeto desen-
volvido na disciplina de computação gráfica.

Docente: Prof^a. Dra. Regina Célia Coelho

Universidade Federal de São Paulo - UNIFESP

Instituto de Ciência e Tecnologia - Campus São José dos Campos

São José dos Campos - Brasil

Dezembro de 2017

Resumo

Esse relatório pretende discorrer sobre o projeto de um ambiente tridimensional contendo um personagem animado que apresenta interação com o usuário. Esse projeto foi realizado como parte da disciplina de computação gráfica da universidade federal de São Paulo campus São José dos Campos. O objetivo do projeto é então projetar um ambiente tridimensional que contenha ao menos um personagem responsável por realizar alguma tarefa. Nesse projeto optou-se por fazer algo inspirado em alguns jogos de fazenda onde o fazendeiro, que é o personagem, se movimenta e planta algo conforme o local do ambiente em que houve o *click* do *mouse*. Nesse projeto deverá haver também uma opção para que o personagem ande pelo cenário e cumpra uma animação pré definida de forma autônoma, isto é, sem esperar uma entrada do usuário. Além disso deverá haver também opções para se movimentar cada parte do corpo do personagem separadamente. Todo esse projeto é desenvolvido utilizando visão perspectiva e implementado utilizando-se o pacote gráfico `opengl`.

Palavras-chaves: `opengl`, ambiente tridimensional, computação gráfica.

Sumário

1	INTRODUÇÃO	5
2	OBJETIVOS	7
3	METODOLOGIA	9
3.1	Cenário	9
3.2	Personagem	10
3.3	Solos e plantas	11
3.4	Texturas	11
3.5	Animações	11
3.6	Câmera	12
3.7	Iluminação	13
3.8	Reconhecimento dos <i>clicks</i> do mouse	13
3.9	Menus	13
3.10	Dependências	13
4	RESULTADOS E DISCUSSÕES	15
5	CONCLUSÃO	23
	REFERÊNCIAS	25

1 Introdução

Um dos fatores decisivos para a popularização dos computadores foi a invenção da interface gráfica. Graças a isso tornou-se possível, para alguém sem conhecimentos profundos sobre o funcionamento dos computadores ou de alguma linguagem de programação, utilizá-lo sem muito sofrimento. Com a popularização dos computadores e os avanços no poder de processamento tornou-se então possível a criação de ambientes gráficos virtuais tridimensionais para as mais variadas funções.

Pode-se citar como aplicações, das técnicas envolvidas na criação de ambientes gráficos virtuais, a criação de jogos eletrônicos, a aplicação de efeitos especiais em filmes entre várias outras como por exemplo a aplicação em simulações médicas a fim de melhor preparar os médicos para fazer uma cirurgia.

Visando então desenvolver habilidades práticas no desenvolvimento de aplicações gráficas, foi feito um projeto onde o objetivo era ter um humanóide em um cenário e que esse humanóide tivesse alguma função no ambiente em que ele estivesse inserido. Nesse projeto optou-se por fazer algo inspirado nos jogos de fazenda onde uma das funções do personagem é plantar. Então nesse projeto foi desenvolvido um ambiente tridimensional com um personagem em opengl onde, o personagem tem por objetivo plantar em todos os lugares do cenário em que houve um *click* do *mause*. Também foram incluídas opções para se movimentar cada parte do personagem e para executar uma animação automática que não depende de nenhuma entrada do usuário.

2 Objetivos

Foi proposto no projeto desenvolver um ambiente gráfico tridimensional com um personagem que realizará uma tarefa. Inspirado em jogos de fazenda e no popular título "minecraft" tem-se então como meta desenvolver um ambiente rural e um humanóide que irá plantar algo a depender do lugar em que houve algum *click* do *mouse*.

Também será implementada as opções de se ativar uma animação automática e de mover cada parte do humanóide. Na animação automática o humanóide irá dançar seguindo uma sequência de passos e andará pelo cenário, já na opção para se movimentar cada parte do humanóide será permitido escolher qual parte do corpo mover e esse movimento será realizado observando as limitações físicas de um ser humano real.

3 Metodologia

A seguir é apresentado como foi desenvolvida cada parte do projeto e na secção [3.10](#) é falado sobre o que é necessário para se compilar e rodar o programa.

3.1 Cenário

Tendo em mente que o personagem seria um fazendeiro e que teria como tarefa plantar, foi feito um ambiente, em céu aberto contendo árvores e uma casa simples, que remete à uma área rural. Para fazer a casa, as árvores, o chão e o céu utilizou-se de formas geométricas primitivas, no caso esferas e cubos.

No caso do céu optou-se por fazer uma esfera de cor azul englobando todo o cenário. O chão foi feito com um cubo bem grande transladado para baixo. As árvores foram feitas combinando-se esferas e cubos e a casa foi feita combinando-se cubos e triângulos.

No caso do chão e da casa foi necessário fazer funções para desenhar as formas geométricas de modo a poder se inserir textura. Foram feitas funções para se desenhar um cubo utilizando-se do conceito de *sweep* visto em aula para se determinar os vértices do cubo e utilizando o comando *glTexCoord2f* para se definir cada cordenada de textura do cubo. Foram feitas três funções diferentes para se desenhar cubos.

Duas das funções desenham um cubo com uma única textura repetindo ela em todas as faces do mesmo, a diferença é em como as cordenadas de textura são definidas, enquanto que em uma das funções o cubo tem cordenadas de textura de 0 a 1, na outra o cubo tem cordenadas de textura de 0 a 50. A segunda função foi criada para se mapear a textura no chão de forma a fazer com que a textura se repetisse várias vezes tendo em vista o tamanho pequeno da textura em relação ao tamanho do chão.

A outra função que desenha um cubo com textura única foi utilizada para se desenhar a casa que não precisava de repetição de textura devido ao tamanho da textura em relação à casa, essa função também foi utilizada para se desenhar o solo onde o personagem planta como é dito com maiores detalhes na secção [3.3](#) e para se desenhar algumas partes do personagem como é dito na secção [3.2](#).

A outra função para desenhar cubos desenha um cubo com seis texturas diferentes, uma em cada face, e foi utilizada para se desenhar algumas partes do humanóide, ela é explicada com maiores detalhes na secção [3.2](#).

Para se montar as árvores e a casa utilizou-se de transformações geométricas para se posicionar os cubos e esferas de modo a formar as árvores e a casa. Para se montar o

telhado da casa utilizou-se de cálculos trigonométricos a fim de definir os ângulos de rotação necessários para se posicionar as laterais do telhado, os cálculos foram feitos defindo-se um triângulo isósceles como sendo o formato do telhado. Já os triângulos frontais do telhado foram posicionados utilizando-se de translações.

Foram feitos dois tipos de árvores, um foi obtido através da combinação de cubos e esferas e o outro através da combinação de cones e cubos. Foi utilizado de transformações geométricas tais como escala e translação para se posicionar cada elemento das árvores a fim de se obter o resultado final.

Por fim, para se montar o cenário, foi feita uma função que define posições aleatórias onde serão desenhadas um determinado número de árvores de cada tipo. Essa função também é responsável por definir posições onde serão desenhadas algumas plantas e solos como é dito na secção 3.3. Essas posições são salvas em filas que são utilizadas todas as vezes que a tela tem que ser redesenhada para se definir onde desenhar as árvores e os solos e plantas.

3.2 Personagem

Inspirado no jogo popular "minecraft" o personagem foi desenhado através de cubos que foram transladados e sofreram transformações de escala afim de formar o personagem. Foram adicionadas esferas em cada articulação para que não ficassem buracos quando o personagem executasse alguma animação. Todo o personagem foi construído utilizando-se o conceito de hierarquia visto em sala com o intuito de se fazer as movimentações do personagem de modo coerente com a realidade.

Para se desenhar os cubos que formam a cabeça e o tronco do personagem foi feita um função que desenha um cubo com uma textura diferente em cada face a fim de se representar o rosto e a parte central das roupas do personagem. Os braços, pernas, pés, mãos e pescoço foram desenhados utilizando-se de uma função que desenha cubos com uma única textura repetindo ela em todas as faces do cubo. As texturas são passadas como argumento dessas funções através de vetores que contém os identificadores de cada textura. As funções criadas para desenhar os cubos com textura também definem as normais para cada face do cubo necessárias para os cálculos de iluminação relaizadas pelo opengl.

Utilizando-se do conceito de objetos filhos foi também criado uma enxada através de cubos e transformações geométricas que foi ligada como objeto filho das mãos do personagem. A enxada se encontra por hierarquia atada as mão do personagem e só é desenhada se uma determinada condição é satisfeita o que faz com que seja possível controlar quando e em que mão a enxada é desenhada ou não.

3.3 Solos e plantas

Quando há um *click* na tela, aparece um solo no chão no lugar onde houve o *click*. Depois de aparecer o solo o personagem se movimenta até o local e planta uma planta. Apartir dai o solo é então desenhado com uma planta em cima. Para se obter esse resultado o solo e a planta foram criados utilizando-se do conceito de objeto filho onde a planta é atada como objeto filho do solo e só é desenhada se uma determinada condição é satisfeita. O solo foi desenhado como um cubo com transformações de escala e de translação. Para se desenhar o cubo do solo utilizou-se da função criada para se desenhar cubos com uma única textura. As plantas foram desenhadas através de cubos e esferas, com transformações de escala e de translação. As esferas no topo das plantas foram desenhadas com cores aleatórias de modo a simular flores diferentes.

3.4 Texturas

As texturas foram carregadas utilizando-se uma biblioteca chamada SOIL que facilita o carregamento das mesmas. Os vetores com os identificares de cada textura carregada foram declarados como variáveis globais. Dentro das funções feitas para se criar cubos com textura são feitas chamadas para a função *glBindTexture* para que se defina que textura será aplicada sobre cada face do cubo.

No carregamento das texturas foi definido para que o opengl aplique os filtros de aumento e diminuição linear de textura. Na função feita para se desenhar um cubo com única textura utilizada para se fazer o chão foi definido também que o opengl faça a repetição das texturas nos eixos x e z.

3.5 Animações

Foram feitas três animações distintas, a do personagem andando, plantando e dançando. Cada uma dessas animações é guardada em uma fila através das posições do personagem e dos ângulos que cada membro do corpo do personagem deve girar a cada quadro. Todas as animações são calculadas em relação a posição atual do personagem sempre que ele necessita de executar alguma delas.

Para tanto foram feitas três funções, uma para cada animação, responsáveis por calcular a posição do personagem e os ângulos que cada parte do corpo do personagem deverá girar a cada quadro desenhado. Esses ângulos e posições são então armazenados em filas que são percorridas para executar a animação. As posições são utilizadas para se fazer translações e os ângulos as rotações necessárias a cada quadro a fim de se executar a animação.

A função que faz a animação do personagem andando também é responsável por calcular uma trajetória para que o personagem se desloque enquanto executa a animação de andar. Essa trajetória é calculada através do conceito de variação, visto em física, do ponto final para onde o personagem deverá se deslocar em relação ao ponto inicial de onde o personagem irá partir. A partir dessa variação é calculado então o quanto o personagem deverá andar no eixo x e no eixo z afim de chegar no destino.

Apartir disso é calculado a posição do personagem a cada quadro bem como a quantidade de passos que ele deverá dar nas direções dos eixos x e z. Para se calcular os ângulos das pernas do personagem e o tanto que ele deverá transladar, para baixo quando ele dobra o joelho ou para frente quando ele dobra a perna para trás, foram utilizadas relações trigonométricas.

Para a execução das animações foram feitas funções que percorrem as filas onde foram guardadas as animações e desenham o personagem na posição e com os ângulos descritos. O tempo das animações é dado pelo tempo que é definido para a função esperar entre o desenho de cada quadro.

3.6 Câmera

As opções de movimentação da câmera foram feitas com base em uma circunferência de raio 0.2, onde o foco da câmera se encontra em algum ponto sobre a circunferência e a origem da circunferência é a posição da câmera. Para se descrever os pontos da circunferência utilizou-se das equações paramétricas da circunferência.

Quando as teclas 'a' e 'd' são pressionadas é somado, ao local da câmera e ao ponto focal, o ponto que está a 90 graus, positivos se a tecla for 'a' e negativos se a tecla for 'd', do foco da câmera na circunferência. Isso faz com que a câmera sempre se desloque a uma distância do tamanho do raio da circunferência sempre que essas teclas são precionadas. Isso faz com que a câmera se desloque para os lados esquedo e direito.

Para a câmera se deslocar para frente e para trás, quando pressionadas as teclas 'w' e 's', foi feita a soma da origem da circunferência e do ponto focal com o ponto onde se encontra o foco no caso da tecla 'w' e com o ponto que se encontra a menos 180 graus do ponto de foco no caso de ser a tecla 's'. Isso faz com que a câmera se desloque para frente e para trás sempre percorrendo distâncias iguais ao tamanho do raio sempre que alguma dessas teclas são pressionadas.

Para se fazer o giro da câmera, quando pressionadas as teclas 'q' e 'e', o ponto focal é deslocado sobre a circunferência em mais 1,8 graus quando a tecla pressionada é 'q' e em menos 1,8 graus quando a tecla pressionada é 'e'. Isso faz com que a câmera gire no sentido anti-horário e no sentido horário.

3.7 Iluminação

Tendo em mente que se queria fazer o ambiente de modo que desse a entender que estivesse amanhecendo, foram utilizadas duas fontes de luz posicionadas próximas uma da outra de modo que se encontrassem perto de uma das extremidades do cenário e acima do chão. As características de cada fonte, isto é a intensidade de luz ambiente, especular e difusa, foram definidas de modo semelhante para as duas fontes. Optou-se pelo uso de duas fontes para se obter um efeito melhor de iluminação que não foi alcançado com uma única fonte.

Para que as fontes de luz ficassem imóveis conforme movimenta-se a câmera, toda as vezes que a câmera é movida, são feitas chamadas para a função *glLightfv* que define sempre a mesma posição para as fontes de luz. Isso é necessário porque no opengl as informações da posição das luzes são guardadas juntamente com as informações da câmera, o que faz com que quando a câmera sofra alguma mudança as luzes também o sofram.

3.8 Reconhecimento dos *clicks* do mouse

Para que fosse reconhecido o local do ambiente correspondente ao local em que houve o *click* do *mouse* na tela, foram aplicadas as transformações inversas das que o opengl aplica para calcular a projeção do modelo na tela. Para tanto os pontos da tela onde houve *clicks* do *mouse* tiveram seus valores de y invertidos para serem compatíveis com o sistema de coordenadas do opengl, depois os pontos foram normalizados e multiplicados pelas matrizes inversas das matrizes de projeção e de modelo do opengl. Como coordenada z da tela foi utilizado o valor do *depth buffer* no ponto x, y em que houve o *click* do *mouse*.

3.9 Menus

Os menus que aparecem com o click do botão direito do mouse foram feitos com o auxilio das funções *glutCreateMenu*, *glutAddSubMenu* e *glutAddMenuEntry*. Na implementação optou-se por fazer o controle do programa através desses menus de modo que cada opção altera os valores de algumas variáveis globais responsáveis por dizer ao programa como ele deve prosseguir.

3.10 Dependências

Para se compilar o programa desenvolvido é necessário ter instalado, além das bibliotecas padrões da linguagem c e da glu e glut do opengl, a biblioteca SOIL utilizada para o carregamento de texturas. Para se executar o programa é necessário haver uma pasta,

no mesmo diretório do executável, com o nome texturas onde deverão ser armazenadas as texturas que serão utilizadas pelo programa.

Para a instalação da biblioteca SOIL é necessário adicionar o compilado dessa biblioteca dentro da pasta que contém as bibliotecas do compilador. Também é necessário adicionar o arquivo 'h' da SOIL na pasta que contém os arquivos 'h' do compilador. Para a compilação desse projeto foi utilizado o compilador GCC no sistema operacional windows. Para se obter a biblioteca SOIL recomenda-se o site <<http://www.lonesock.net/soil.html>>.

4 Resultados e Discussões

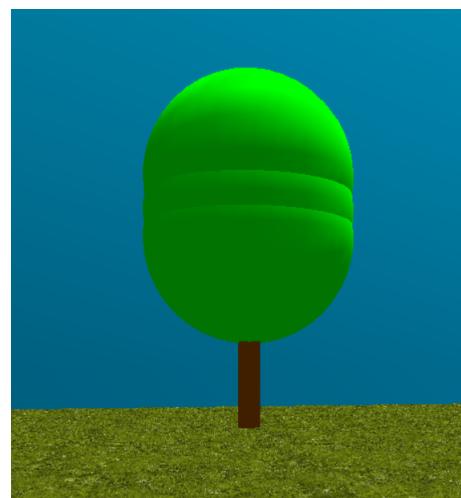
A [Figura 1](#) e a [Figura 2](#) mostram os dois tipo de árvores construídas para fazer parte do cenário, a árvore da [Figura 1](#) foi construída combinando-se cubos e cones com transformações de escala e de translação. Já a árvore da [Figura 2](#) foi construída combinando-se esferas e cubos.

Figura 1 – Árvore feita apartir da combinação de cones e cubos



Fonte: o autor

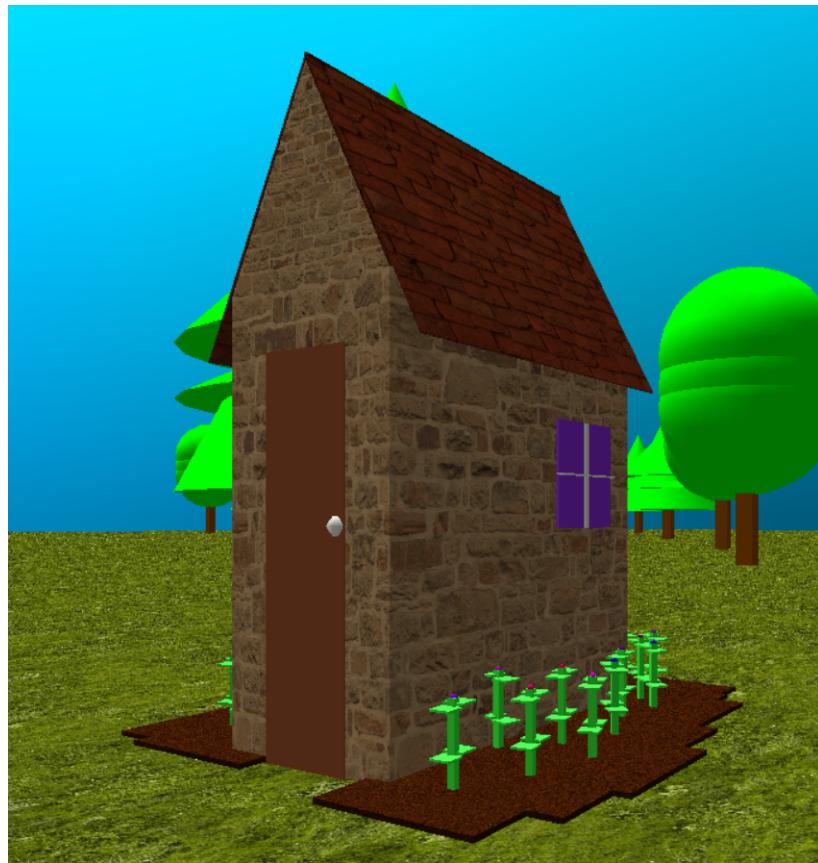
Figura 2 – Árvore feita apartir da combinação de esferas e cubos



Fonte: o autor

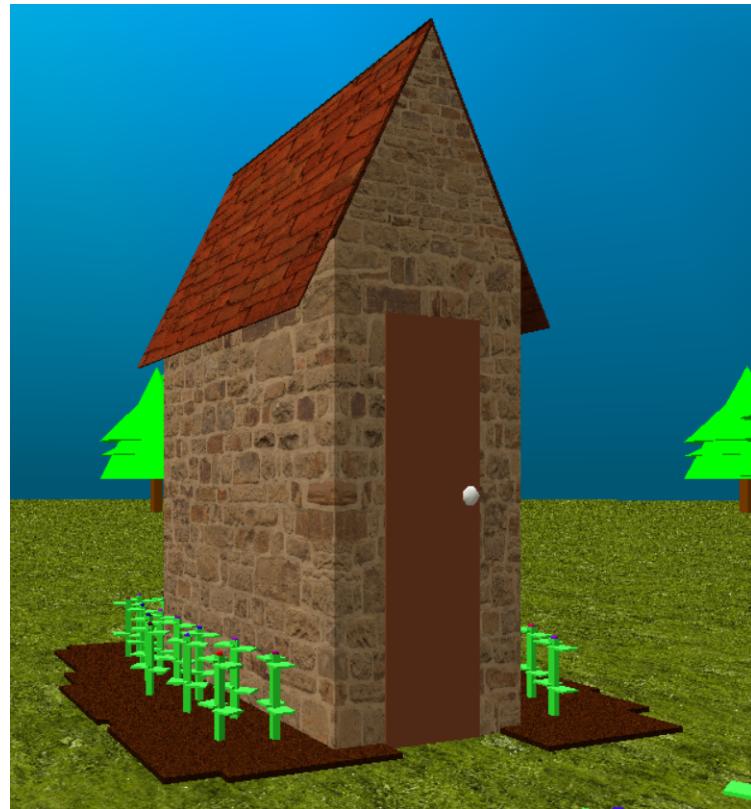
A [Figura 3](#) e a [Figura 4](#) mostram a casa que foi construída através da combinação de cubos, triângulos e algumas texturas.

Figura 3 – Casa feita apartir da combinação de cubos e triângulos



Fonte: o autor

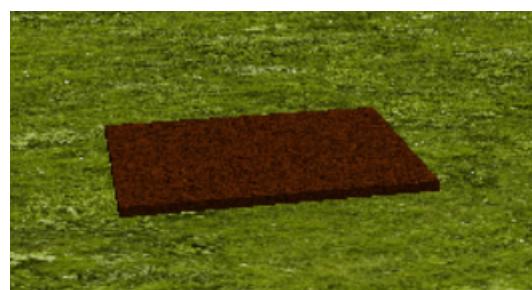
Figura 4 – Casa feita apartir da combinação de cubos e triângulos



Fonte: o autor

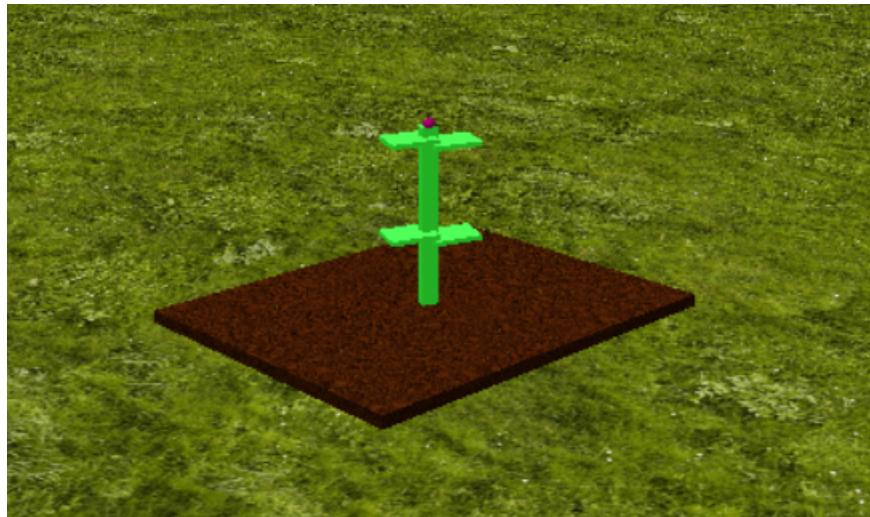
A [Figura 5](#) mostra o que aparece ao *click do mouse*. Definiu-se para esse objeto o nome de solo e ele foi criado através de um cubo com textura. A [Figura 6](#) mostra o solo logo após o personagem cultivá-lo. Nesse momento aparece no centro do solo uma planta que foi construída através de cubos e esferas. As esferas no topo das plantas tem cores definidas aleatoriamente.

Figura 5 – Solo que aparece ao *click do mouse*



Fonte: o autor

Figura 6 – Solo com a planta



Fonte: o autor

A [Figura 7](#) mostra o personagem, no caso um fazendeiro, que foi construído apartir de cubos com textura que foram escalonados e transladados. Em cada articulação do personagem foram inseridas esferas para que não aparecessem buracos quando o personagem movesse alguma parte do corpo. Isso pode ser visto na [Figura 8](#), nessa figura é também mostrado o que acontece quando é selecionada alguma opção no menu para se movimentar o personagem, quando isso acontece o personagem é desenhado em um cenário diferente e as setas do teclado podem então ser utilizadas para se movimentar a parte do personagem que foi selecionada.

Figura 7 – Personagem feito com cubos ao estilo minecraft



Fonte: o autor

Figura 8 – Movimentos do personagem sendo feito pelas setas do teclado



Fonte: o autor

A Figura 9 mostra o menu que aparece na tela ao *click* do botão direito do *mause*, nesse menu há as opções para se mover o personagem, que abre um submenu para se selecionar que parte do personagem movimentar, para executar a animação automática onde o personagem dança e anda pelo cenário, para o modo jogo onde o programa reconhece os *clicks* do botão esquerdo do *mause* como sendo os lugares em que o personagem tem que ir plantar e, por último a opção para se resetar o jogo onde as plantas que foram plantadas são excluídas e o programa começa a executar denovo como se tivesse sido acabado de ser inicializado, setando a posição do personagem e da câmera para a posição inicial.

A Figura 10 e Figura 11 mostram os submenus referentes a opção de mover o personagem presente no menu principal.

Figura 9 – Menu principal



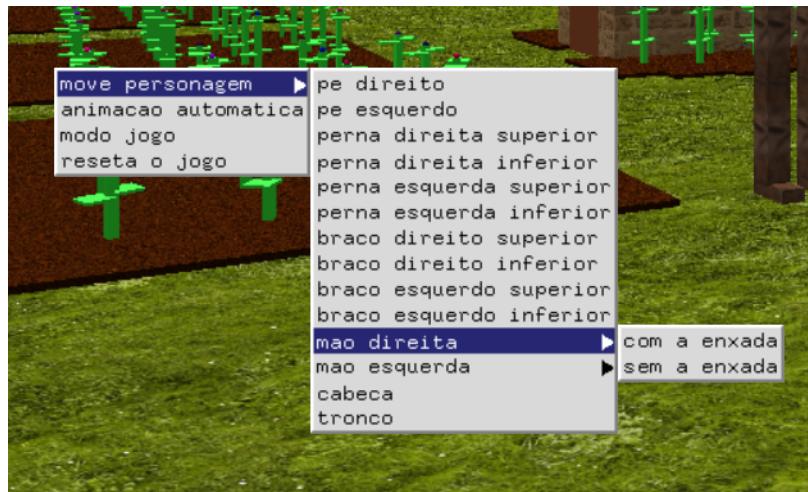
Fonte: o autor

Figura 10 – Menu principal com submenu de opções para se movimentar o personagem



Fonte: o autor

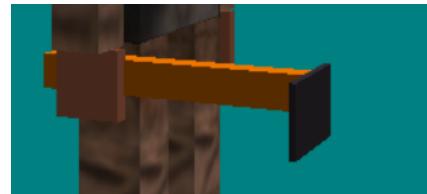
Figura 11 – Menu principal com submenu de opções para se movimentar o personagem mostrando as opções de se movimentar as mão do personagem com ou sem a exada



Fonte: o autor

A Figura 12 mostra a enxada que aparece na mão do personagem quando este está plantando ou quando é selecionado no menu de opções para se movimentar a mão do personagem com a enxada.

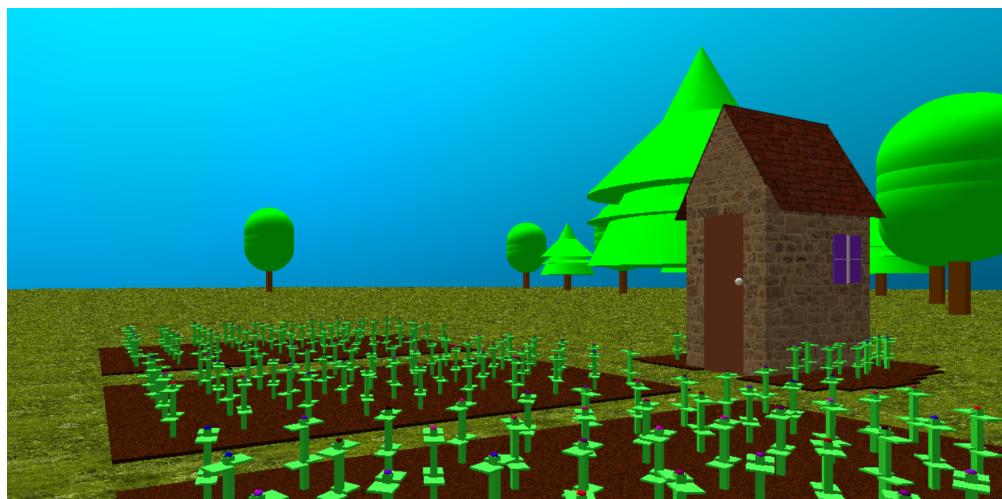
Figura 12 – Enxada que aparece na mão do personagem



Fonte: o autor

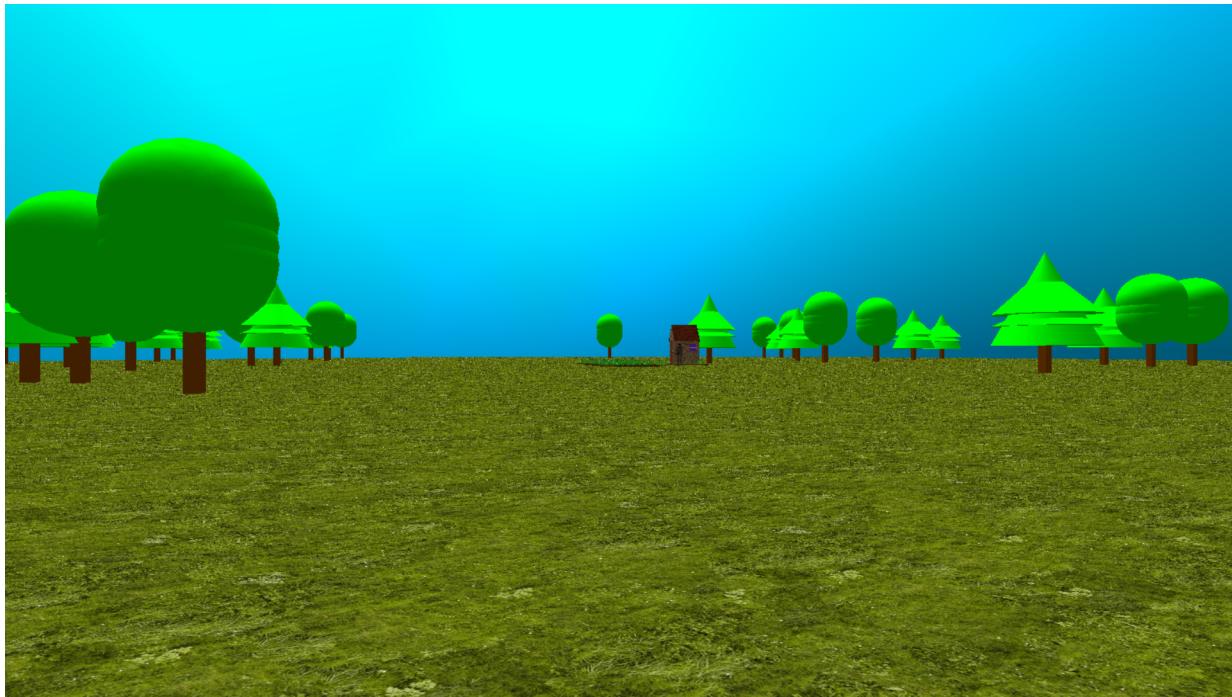
A [Figura 13](#) e [Figura 14](#) mostram como ficou o cenário depois de montado com os elementos já descritos aqui. Nessas imagens também da para ver melhor como ficou o chão com a textura e o céu com a iluminação de duas fontes de luz próximas uma da outra.

Figura 13 – Cenário construído



Fonte: o autor

Figura 14 – Cenário construído



Fonte: o autor

Também foram implementadas as opções para se mover a câmera de acordo com algumas teclas, são elas 'a', 's', 'd', 'q', 'w' e 'e'. As teclas 'a' e 'd' fazem a câmera se mover para o lado esquerdo e direito respectivamente, as teclas 'w' e 's' fazem a câmera ir para frente e para trás respectivamente e as teclas 'q' e 'e' fazem a câmera girar no sentido anti-horário e horário respectivamente.

5 Conclusão

Foi feito então um ambiente tridimensional baseado em um jogo de fazenda ao estilo minecraft, onde o personagem, no caso um fazendeiro, planta nos locais em que houveram *clicks* do *mause*. Para isso foi feita uma função que calcula o trajeto e os passos que o fazendeiro deverá executar.

Também foram feitos todos os objetos que constituem o cenário são eles as árvores, a casa, o solo e as plantas. Todos esses objetos foram criados apartir da combinação de figuras primitivas tais como cubos e esferas.

Foi implementado também o menu que aparece quando há um *click* do botão direito do mause, responsável por permitir ao usuário escolher se o personagem irá plantar, executar a animação automática ou qual parte do personagem mover.

Foram feitas as animações do personagem plantando, dançando e andando. Essas animações foram armazenadas em filas que são percorridas afim de se executar a animação.

Também foram implementadas as opções de movimento da câmera ao se precionar determinadas teclas do teclado. E foram inseridas texturas no personagem, no chão, no solo e na casa. Também foram definidas fontes de luz e suas especificações para iluminar os objetos.

Ao fim do desenvolvimento ficou mais claro como desenvolver uma aplicação gráfica e como funciona internamente um pacote gráfico como o opengl. Após algumas dificuldades para se fazer as animações compreendeu-se melhor o processo de se animar objetos tridimensionais e a importância dos pequenos detalhes no resultado final das animações.

Referências

- 1 ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 6028*: Resumo - apresentação. Rio de Janeiro, 2003. 2 p. Nenhuma citação no texto.
- 2 VIANNA, A. C. Computação gráfica. *TECGRAF-Pontifícia Universidade Católica do Rio de Janeiro*, 2001. Nenhuma citação no texto.
- 3 POZZER, C. T. Opengl-conceitos básicos. Nenhuma citação no texto.
- 4 WOO, M. et al. *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1999. Nenhuma citação no texto.
- 5 JR, R. S. W. et al. *OpenGL SuperBible: comprehensive tutorial and reference*. [S.l.]: Pearson Education, 2010. Nenhuma citação no texto.