ストップウォッチ 設計書

ビジュアライゼーション最終課題

情報システム学科3年 No.2172010有馬祥太

目次

1.概要

2.画面構成について

3.システム構造について

4.関数の定義

5.その他、参考文献

~設計書の書き方~

・図のリンクは、図番号と図の下段に図名を示しています。

・実際のソースコードからの抜粋はグレーの蛍光ペンで示しています。

1. 概要

　OpenGLのグラフィック描画を用いて、ストップウォッチを作成する。

　アナログ時計を再現することで、視覚的にも感覚的にも計測時間が分かるように描画しました。

グラフィック設定

　時計盤をXZ平面に描画し、視点をY軸の正から負の方向へ見るように設定しました。(図1:グラフィク設定)

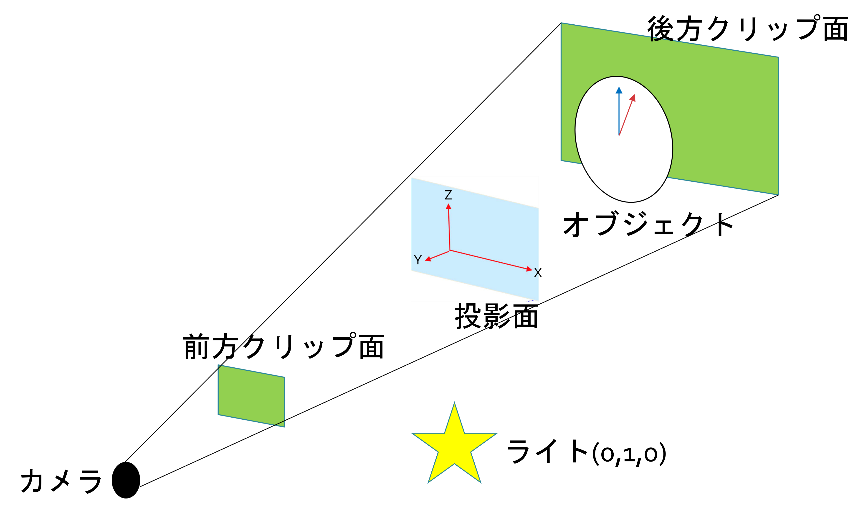


図1:グラフィック設定

1. 画面構成について

描画した要素は以下の通りです。

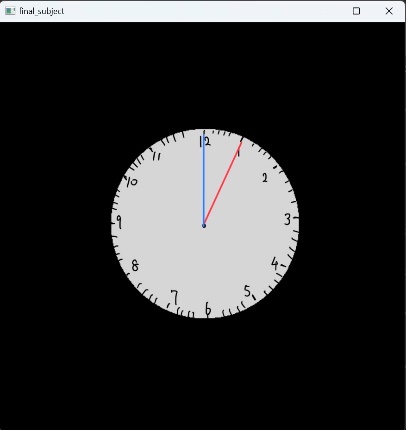


図2:時計盤のイメージ

・時計盤

・秒針、分針

・目盛り

画面中心を原点とし、時計盤とアナログ時計のような目盛りを描画します。

秒針を赤い針、分針を青い針で時間を表示する。

また、操作ごとにコマンドプロンプトでデジタル時刻を表示する。

完成イメージ図は以下の通りです(図2:イメージ図)

各要素の詳細な設定について記述します。

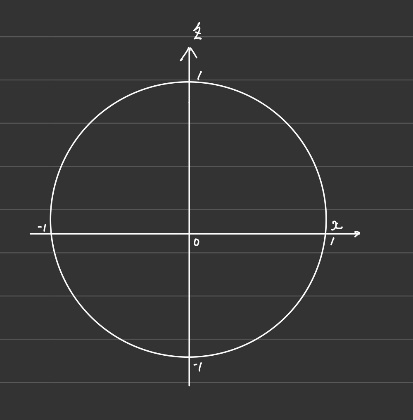


図3:時計盤のイメージ

・時計盤(図3:時計盤のイメージ)

点を50個打ち、GL\_POLYGONにより、円を近似しています。

半径は1に設定しています。

　 　要素の光の当たり方については、以下のパラメータを用意しました。

float diffuse1[] = { 1.0, 1.0, 1.0, 1.0 };

　float specular1[] = { 0.5, 0.5, 0.5, 1.0 };

　float ambient1[] = { 0.2, 0.2, 1.0, 1.0 };

　float shininess1 = 60.0;

　法線ベクトルはglNormal3f(0.0,1.0,0.0);と設定しています。

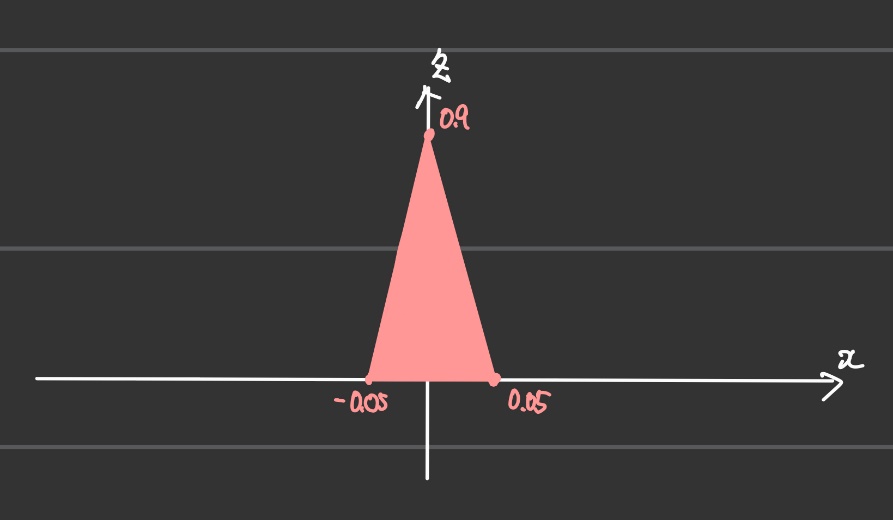


図4:分針、秒針のイメージ

・秒針、分針の描画(図4:分針、秒針のイメージ)

　GL\_POLYGONにより、三角形を生成します。

回転では、glRotatef(-theta ,0.0 ,1.0 ,0.0);を使用し、原点中心に、y軸を軸としてtheta角で回ります。また、-thetaにした理由としては、時計回りに回転させるためです。

秒針の光の当たり方は、以下のように設定しました。

float diffuse2[] = { 0.7, 0.1, 0.0, 1.0 };

float specular2[] = { 0.8, 0.0, 0.0, 1.0 };

float ambient2[] = { 0.1, 0.1, 0.1, 1.0 };

float shininess2 = 100.0;

法線ベクトルはglNormal3f(0.0,1.0,0.0);と設定しています。

分針の光の当たり方は、以下の通りです。

float diffuse3[] = { 0.0, 0.4, 7.0, 1.0 };

float specular3[] = { 0, 0, 0, 1.0 };

float ambient3[] = { 0.1, 0.1, 0.1, 1.0 };

float shininess3 = 128.0;

法線ベクトルはglNormal3f(0.0,1.0,0.0);と設定しています。

・小さな目盛り(図5:目盛り6°間隔\_イメージ)

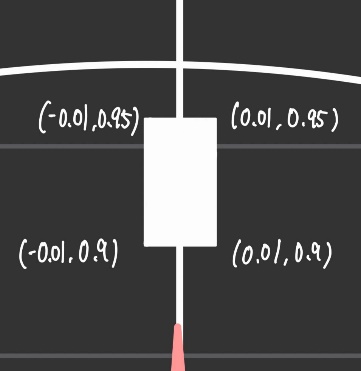


図5:目盛り6°間隔\_イメージ

GL＿POLYGONによって、小さな四角形を生成します。

60目盛り必要なため、for文で角度を調整し、描画する。

1目盛りは6°の間隔であるため、glRotatef(i\*6 ,0.0 ,1.0 ,0.0);で回転

目盛りの光の当たり方は、以下のように設定しました。

float diffuse4[] = { 0.0, 0.0, 0.0, 1.0 };

float specular4[] = { 0, 0, 0, 1.0 };

float ambient4[] = { 0.1, 0.1, 0.1, 1.0 };

float shininess4 = 128.0;

法線ベクトルはglNormal3f(0.0,1.0,0.0);と設定しています。

　・大きな目盛り(図5:目盛り30°間隔\_イメージ)

また、5分間隔で目盛りを大きくし、視認性を上げるため、生成する四角形を大きくします。

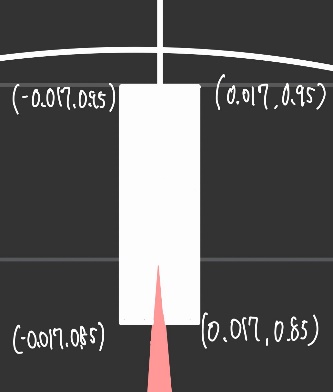


図5:目盛り30°間隔\_イメージ

for文で30°毎目盛りを12個描画します。

glRotatef(i\*30 ,0.0 ,1.0 ,0.0);で回転しています

目盛りの光の当たり方は、以下のように設定しました。

float diffuse4[] = { 0.0, 0.0, 0.0, 1.0 };

float specular4[] = { 0, 0, 0, 1.0 };

float ambient4[] = { 0.1, 0.1, 0.1, 1.0 };

float shininess4 = 128.0;

法線ベクトルはglNormal3f(0.0,1.0,0.0);と設定しています。

1. システム構造

ストップウォッチの状態として、「初期状態」「動作状態」「一時停止状態」の3状態を用意します。(図6:状態遷移図)

初期状態

　プログラム開始後の初期画面状態のことを指します。時間は00:00を示しています。「space」を押すことで、動作状態へ遷移します。

動作状態

　ストップウォッチが動作中の状態を指します。

一時停止状態

まず初めに、プログラムが起動されると「00:00:00」の状態でスタンバイします。これを初期状態とします。

スペースキーを押すことで、ストップウォッチが始まり、時計のグラフィックが始動します。これを動作状態とします。

再度スペースキーを押すことで、ストップウォッチが停止します。これを一時停止状態とします。

最後にｒキーを押すことで時間をリセットし、初期状態へ戻ります。