

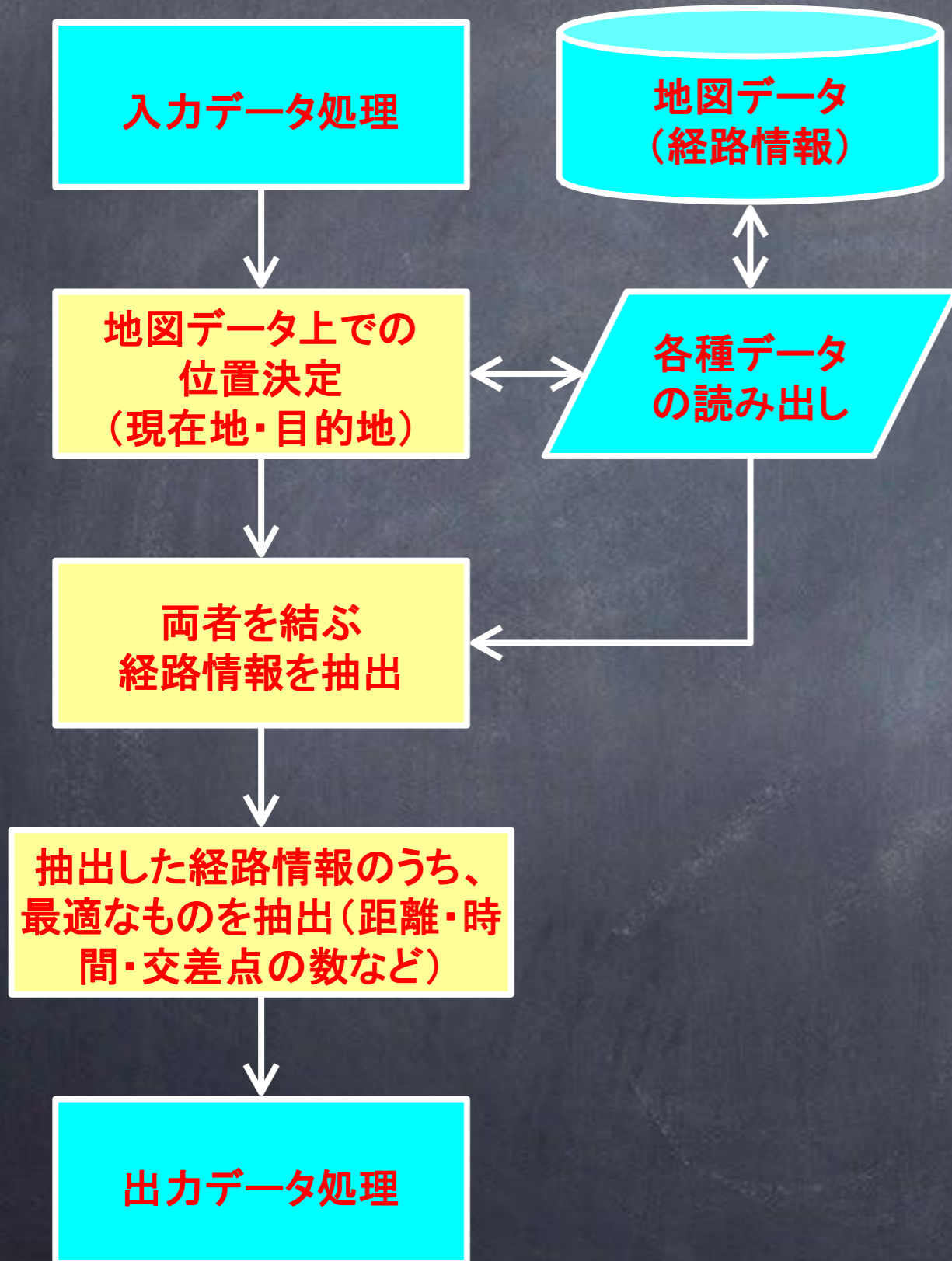
コンピュータ実習
第12回 2025年12月18日

カーナビゲーション5
座標変換

担当: 相澤 直人, 宍戸博紀

コンピュータ実習(第12回)

カーナビ作成に向けて



地図データの検索と抽出

→ 入出力・変数・構造体

抽出したデータを元に経路を割り出す

→ 条件検索(サーチ)

最適な経路を割り出す

→ ダイクストラ法

割り出した経路のうち、最適なものを選ぶ

→ 条件並び替え(ソート)

コンピュータ実習(第12回)

今後の授業の進行

1. 地図データを元に、目的地を決定
 - 地図データの読み込み (11/20)
 - 目的地(地図データ)の検索 (11/27)
2. 経路を探索
 - 条件に基づく経路の設定 (12/4: 簡易版)
 - 最適経路の計算 (12/11)
3. 経路をドライバーに分かりやすく提示
 - 画面上で車をアニメーション表示 (12/4)
 - グラフィック表示の見栄えを良くする
(座標変換)

 本日の講義

コンピュータ実習(第12回)

エイリアスの設定(第9回復習)

前ページにあるように、今回の地図を表示させるソースファイルのコンパイルを行うためには、

```
cc map.c -g -O2 -Wall -Wno-unused-result  
-I/usr/include/freetype2 -lftgl -lglfw -lGLU  
-lGL -lX11 -lXrandr -lm -o map
```

となり、一回一回すべて入力するのは面倒。。。

そこで、エイリアスを設定することをおススメします。

エイリアスの設定方法は、第6回の講義参考資料にありましたよね・・・？

コンピュータ実習(第12回)

エイリアスの設定の例

- ホームディレクトリに、`glcc.sh`というファイルを作成。
ファイル中に以下を書き込み、保存。

```
#!/bin/bash
```

```
cc $1.c -g -O2 -Wall -Wno-unused-result  
-I/usr/include/freetype2 -lftgl -lglfw -lGLU -  
lGL -lX11 -lXrandr -lm -o $1
```

cc ~ \$1は
1行で書く!

- `glcc.sh`に操作権限を与えるために以下を入力
`chmod +x glcc.sh`
- ホームディレクトリにある`.bashrc`に以下の一文を追加
`alias glcc='/home/gakusei2017/glcc.sh'`
- 上記の変更をh反映させるために以下を入力
`source ~/.bashrc`

コンピュータ実習(第12回)

エイリアスの設定の例

- 前回、`glcc.sh`の設定が上手くいかなかった方へ動作確認済のファイルをインターネットにアップロードしましたので、それを試してみてください

```
wget https://web.tohoku.ac.jp/kc_kyomu/  
computer_seminar1/c/qse-env/lecture/2024-12/  
glcc.sh
```

(ホームディレクトリで`wget`コマンドを使用すれば、その場所に`glcc.sh`がダウンロードできます)

- その後の`.bashrc`の設定は同じ

コンピュータ実習(第12回)

今回の内容: 地図表示(座標変換)



カーナビ作成に必要な要素

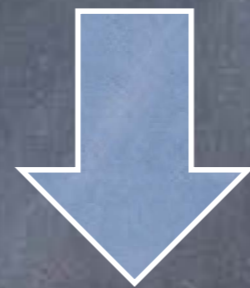
- データの用意
- データの読み込み
- データの検索(並べ替え)
現在地・目的地・最短経路
- グラフィックス表示
- 使い勝手・見栄えの改良

コンピュータ実習(第12回)

今回の内容: 地図表示(座標変換)

これまでに作ってきたプログラムでは,

- 地図の表示位置や大きさが固定されている.
- また, 地図の向きも常に北が上になっている.



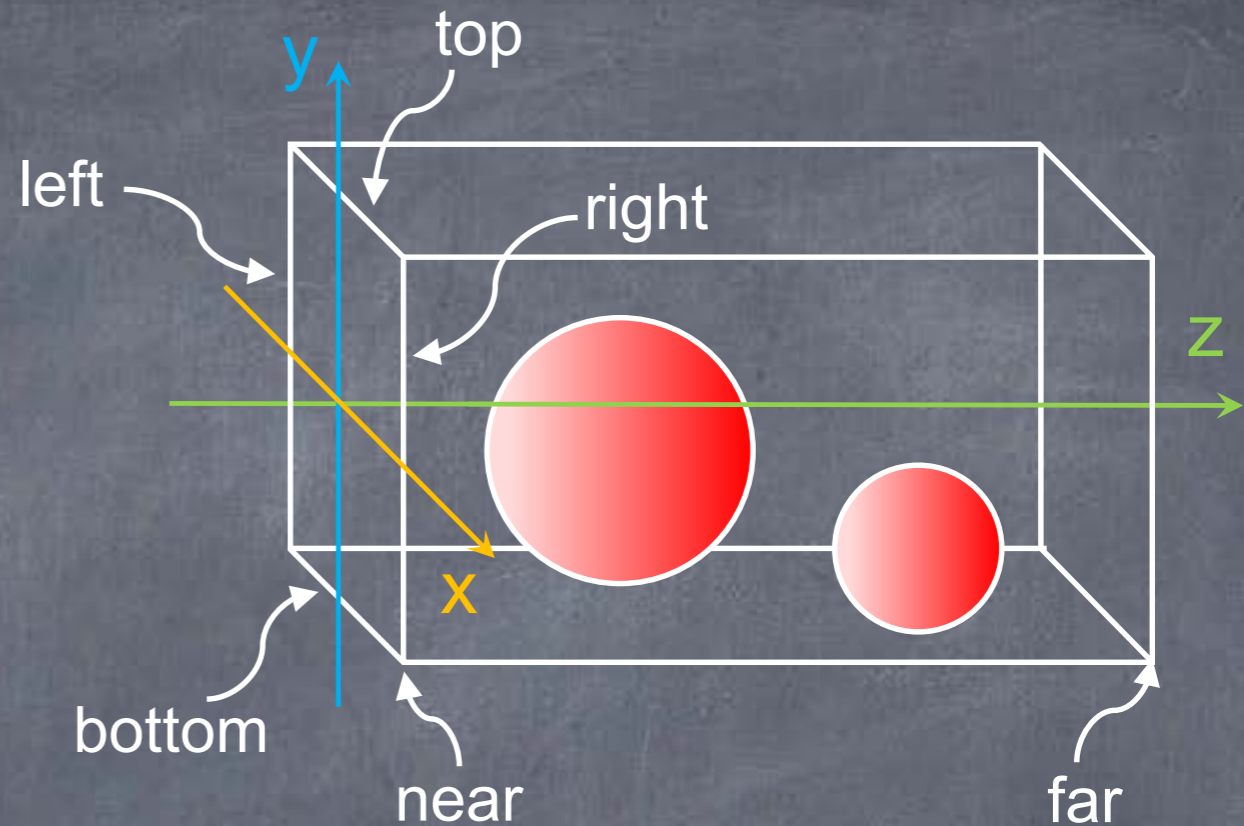
より見やすい表示を実現するために,

- 地図の任意の位置を拡大して表示する.
- 地図を回転して表示する.

コンピュータ実習(第12回)

マクロを利用した座標変換

```
/* 座標変換マクロの定義 */  
#define ORIGIN_X -2.0  
#define ORIGIN_Y 3.0  
#define REAL_SIZE_X 10.0  
#define REAL_SIZE_Y 10.0
```



glOrtho関数: 投影行列を正射影に設定する関数 (直交投影を定義する関数)

```
void glOrtho(left, right, bottom, top, zNear, zFar);
```

引数: left, right

左右の垂直座標の指定(左右の垂直のクリップ面の座標を指定する)

bottom, top

上下の水平座標の指定(上下の水平のクリップ面の座標を指定する)

zNear

視点から最も近いクリップ面の距離を指定

zFar

視点か最も遠いクリップ面の距離を指定

```
glMatrixMode(GL_PROJECTION); /*射影変換を使うと指示*/  
glLoadIdentity(); /*初期化(変換行列を単位行列にする)*/  
glOrtho(ORIGIN_X + REAL_SIZE_X * -0.5, ORIGIN_X + REAL_SIZE_X * 0.5,  
ORIGIN_Y + REAL_SIZE_Y * -0.5, ORIGIN_Y + REAL_SIZE_Y * 0.5, -1.0,  
1.0);  
/* (ORIGIN_X, ORIGIN_Y)を中心に、REAL_SIZE_X * REAL_SIZE_Yの範囲の空間をビューポートに投影する */
```

コンピュータ実習(第12回)

マクロを利用した座標変換

```
/* 座標変換マクロの定義 */  
#define ORIGIN_X -2.0  
#define ORIGIN_Y 3.0  
#define REAL_SIZE_X 10.0  
#define REAL_SIZE_Y 10.0
```

```
(left)      = -2.0 + 10.0 * (-0.5);  
(right)     = -2.0 + 10.0 * (0.5);  
(bottom)    = 3.0 + 10.0 * (-0.5);  
(top)       = 3.0 + 10.0 * (0.5);  
(zNear)     = -1.0;  
(zFar)      = 1.0;
```

(-2.0, 3.0)は地図の中心の(ウィンドウ上での)表示位置,
10.0 はx, y それぞれのスケールを表します.

これにより,

地図の平行移動&拡大縮小が実現できる.

※スケール変換は, ローカル座標系の軸そのものの伸縮に適用されるのであり,
ローカル座標系でのオブジェクトの各頂点座標は影響を受けない.

コンピュータ実習(第12回)

同次変換行列

平行移動, 回転移動, 拡大縮小を行列で表現するために同次座標を用いる.

これにより, 3つのモデリング変換を行列演算で表現できる為, 演算コストを低減できる.

$$\begin{pmatrix} x_{new} \\ y_{new} \\ z_{new} \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} + \begin{pmatrix} u \\ v \\ w \\ 1 \end{pmatrix}$$

回転移動

拡大・縮小

平行移動

コンピュータ実習(第12回)

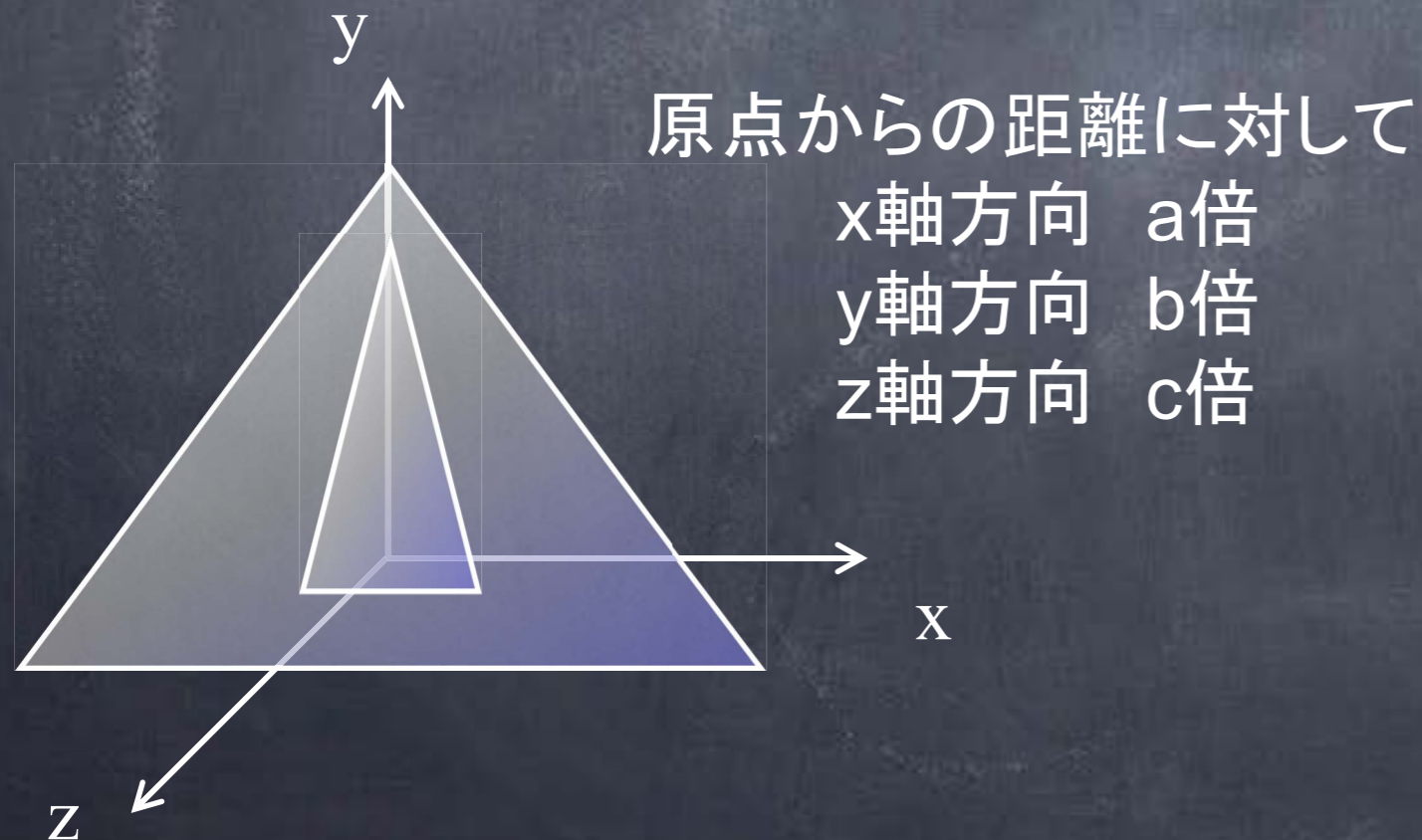
拡大 & 縮小 (scaling)

glScalef関数: 原点とオブジェクトの頂点座標の距離にスケール係数(a, b, c)を乗じて拡大・縮小するための関数

スケール係数が1より大きいならば拡大. 1未満ならば縮小となる.

```
void glScalef(a, b, c);
```

引数: a x軸方向のスケール係数
b y軸方向のスケール係数
c z軸方向のスケール係数



拡大・縮小

$$\begin{pmatrix} x_{new} \\ y_{new} \\ z_{new} \\ 1 \end{pmatrix} = \begin{pmatrix} p & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

コンピュータ実習(第12回)

回転移動(rotation)

glRotatef関数: 原点と(X, Y, Z)を結ぶ軸を中心として,
deg[deg]だけオブジェクトを回転させる関数

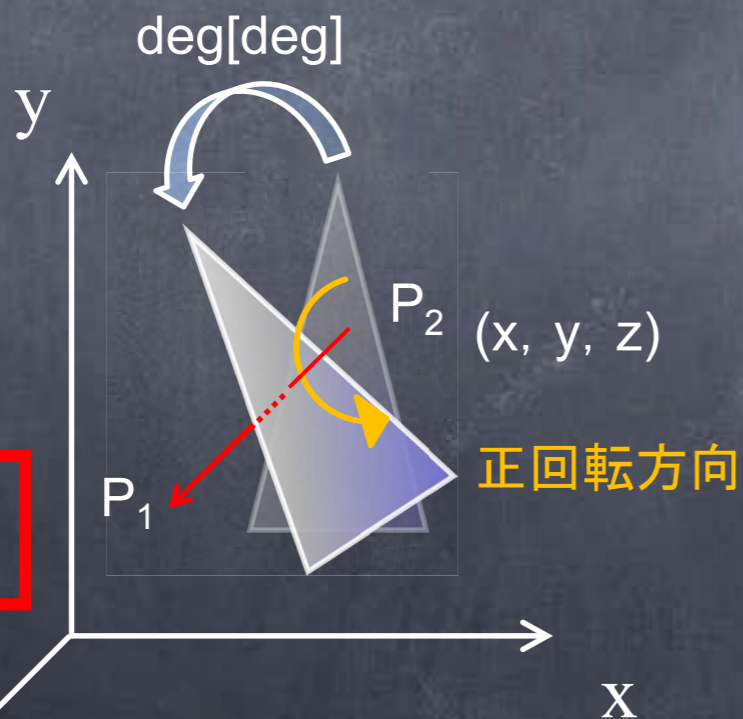
```
void glRotatef(deg, X, Y, Z);
```

引数: deg 回転角度

X x座標

Y y座標

Z z座標



OpenGLはP₁を
原点にとる.

P₁からP₂への方角で、
右ねじ回転が正回転

回転移動

$$\begin{pmatrix} x_{new} \\ y_{new} \\ z_{new} \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

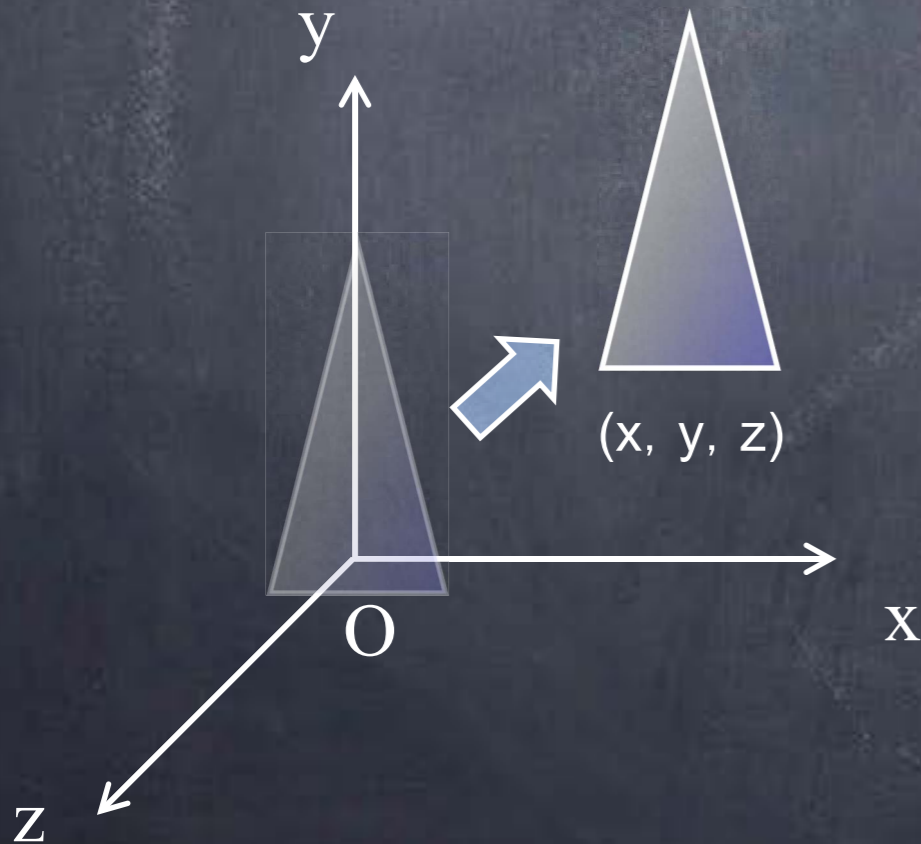
コンピュータ実習(第12回)

平行移動(translation)

glTranslatef関数: (x, y, z)の値だけオブジェクトを平行移動させる関数

```
void glTranslate(x, y, z);
```

引数: x x座標
y y座標
z z座標



平行移動

$$\begin{pmatrix} x_{new} \\ y_{new} \\ z_{new} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & u \\ 0 & 1 & 0 & v \\ 0 & 0 & 1 & w \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

コンピュータ実習(第12回)

今回の内容: 地図の移動アニメーション マーカではなく、地図を動かすには?

- これまでのカーナビプログラム
 - 固定表示の地図(常に北が上)の上をマーカが移動
 - マーカがウィンドウ外に飛び出たり、交差点が密集した場所を移動するときはマーカが見つらい!
- 地図自身を移動(または拡大縮小/回転)させることで、マーカの移動が終始追うことができる!
- 一般的なカーナビは、マーカが画面中央に固定されて、地図が移動する表示なので、皆さんもそのような表示ができるように書き換えてみましょう

コンピュータ実習(第12回)

今回の内容: 地図の移動アニメーション

ヒント1: 座標変換を動的に行う

- これまでのプログラムの座標の設定
 - #define文によるマクロ定義により実現
 - #define文を使用すると、プログラム動作中に設定したパラメータの数値が変更できないため、この方法では地図のアニメーション表示を行うことができない!
- プログラム動作中に逐次変更させるには、「変数(double型など)」として定義する必要がある
- 具体的には、座標変換パラメータ(ORIGIN_X, ORIGIN_Y, REAL_SIZE_X, REAL_SIZE_Y)をプログラム動作中に変える

コンピュータ実習(第12回)

今回の内容: 地図の移動アニメーション

ヒント2: アニメーション処理

- 地図をスクロールさせるなどのアニメーション表示を行うためには、座標変換パラメータ(`ORIGIN_X`, `ORIGIN_Y`, `REAL_SIZE_X`, `REAL_SIZE_Y`)をわずかに変化させて地図を再表示…を繰り返し行う
- または、座標変換関数(`glScalef`, `glRotatef`, `glTranslatef`)の引数を逐次変更することでもアニメーションを実装できる

コンピュータ実習(第12回)

最終面接について

カーナビゲーションプログラムの要件

これまでに演習してきた内容を組み合わせて、下記の4つの基準を満たすようなカーナビゲーションプログラムを完成させよう

また、これらの基準を満たした上で、より使いやすいカーナビゲーションになるようにプログラムに工夫・改良を加えること

最終面接ではそれら工夫点・改良点についても評価する

コンピュータ実習の単位取得条件

課題の最低限の合格ラインの目安

- 地図がグラフィック表示される
- 目的地と出発地を任意に設定できる(マウスによる選択でもキーボード入力による選択でも可)
- ダイクストラ法等の経路探索手法を利用し、最短経路を探索している
- 出発地から目的地までマーカが移動していくアニメーションが動く(マーカ自身は移動する方式でも地図表示が移動する方式でも可)

コンピュータ実習(第12回)

最終面接について

最終面接

1月15日(木)に行います。面接では作成したカーナビプログラムの動作を見せてもらいます。

その際、工夫点、改良点について質問します。

1月 8日17:00までにカーナビゲーションプログラムが完成した人は1月 8日に面接を受けても構わない。

その場合、課題の内容を判断し1月15日の実習の出席を免除する。

コンピュータ実習(第12回)

最終面接について

最終面接に関する提出物について

最終面接に先立ち、作成したカーナビゲーションプログラムのソースファイルと工夫点をまとめたレポートを Classroomの課題提出機能を使って提出すること。提出期限は1月15日14:40までとする。Classroomの課題提出ページは後日連絡する。

(1月8日に面接を受ける場合は、面接前までに提出すること。)

- レポートの題名は「2025コンピュータ実習最終課題(学籍番号)(名前)」とすること
- レポート中にはカーナビプログラムを完成させた際の工夫点を書くこと。
どんな些細なことでも良いので自分が考えたところをできる限り書くこと。
 - (例)〇〇の色を緑に変えた
- ソースファイルの名前は「report_学籍番号.c」とすること
- 優秀作品に選ばれた場合にお手本として公開することに同意しない場合は、その旨を書くこと。

コンピュータ実習(第12回)

最終面接について

高得点を取るためには・・・

オリジナリティを評価します

色やサイズが違うなどのパラメータを変えただけの新しさではなく、機能としての新しさを高く評価します。よって、他人のコピーでは高得点は望めません。

「こんな機能があったら便利」というものを盛り込めるように、是非チャレンジしてみてください！

他の授業・レポートと同じように、コピー&ペースト等の不正が発覚した場合は、コンピュータ実習 I が不合格になるだけでなく、当該セメスターの他の講義についても不合格となる可能性があります。不正行為は厳に慎んでください。