1　課題 I

1.1　課題 I-1

OpenGL **関連技術**: 本テキストで扱う GLUT，GLU の他に OpenGL に関連するライブラリとして GLEW，GLFW，GLUI などがある．様々な実行環境で 2D/3DCG を実現するためのプログラミング技術として Vulkan，OpenGL ES，WebGL などがある．これらについて，機能や各ライブラリの関係性などを調査して整理してみよ．

1.1.1　GLEW

GLEW とは，グラフィックスハードウェアの拡張機能を使用可能にするライブラリである．特に Windows では，もともとサポートしている OpenGL のバージョンが 1.1 のため，新しい機能を使用することができない．そのため，GLEW を用いてグラフィックスハードウェアが持つ全ての機能を使えるようにする．

1.1.2　GLFW

GLFW とは，デスクトップでの OpenGL，OpenGL ES，Vulkan 開発用のオープンソースのマルチプラットフォームライブラリである．ウィンドウやコンテキストを作成し，入力を管理するためのシンプルな API を提供する．

1.1.3　GLUI

GLUI とは，ダイアログウィンドウで使用されるボタンやチェックボックスなどのコントロールを OpenGL で作成できるライブラリである．

1.1.4　Vulkan

Vukan とは，グラフィックス API のことで，直接 GPU にアクセスできる構造によって，これまでのグラフィックス API に比べてより速い描画をすることが可能となる．

1.1.5　OpenGL ES

OpenGL ES とは，電化製品や車両などの組み込みおよびモバイルシステムで，高度な 2D，3D グラフィックスをレンダリングするためのクロスプラットフォーム API である．

1.1.6 WebGL

WebGL とは，ウェブブラウザ上で OpenGL ES 相当の描画処理を行うことができる低レベルの API である．JavaScript の API として実装されているため，改めてプラグイン等をインストールすることなく実行できる．

1.2 課題 I-2

プログラミングのツール:GCC に含まれる make ユーティリティ，デバッガの機能や機能や利用法について学習せよ．また，コマンドラインからのプログラムの開発では，grep や touch などのツールがあると便利である．更に最近ではクロスプラットフォーム開発を支援する CMake，バージョン管理を支援する Subversion や Git などを利用することも一般化している．これらの機能や利用法について学習せよ．

1.2.1 make ユーティリティ

make とは大規模プログラムのコンパイルを簡略化するツールである．makefile にファイルの関係を記述することで各ファイルの更新を取得し，必要なものだけをコンパイルすることができる．以下のコマンドで makefile を実行することができる．

```
1  make
```

1.2.2 grep

grep とは，テキストファイルの中から正規表現と一致する行を検索し，出力するコマンドである．以下のコマンドで grep を使用することができる．

```
1 grep オプション [] 検索文字列正規表現 [()] ファイル名 []
```

1.2.3 touch

touch とはファイルの最終更新日を変更するコマンドである．以下のコマンドで touch を使用することができる．

```
1 touch オプション [] ファイル 1 ファイル 2 …
```

1.2.4 CMake

CMake とは，ソフトウェアをビルドやテスト，パッケージ化するために設計されたオープンソースのクロスプラットフォームツールである．プラットフォームやコンパイラに依存しないシンプルな設定ファイルを使用することでソフトウェアのコンパイルプロセスを制御し，makefile とワークスペースを生成するために使用される．

1.2.5 Subversion

Subversion とは，データの安全な避難所としての信頼性を特徴とするオープンソースの集中型バージョン管理システムである．個人から大規模なエンタープライズオペレーションまで，さまざまなユーザやプロジェクトのニーズをサポートすることができる．

1.2.6 Git

Git とは，小規模なプロジェクトから大規模なプロジェクトまで，効率的に処理できるように設計されたオープンソースの分散型バージョン管理システムである．非常に高速なパフォーマンスを備えた小さなフットプリントを備えている．

1.3 課題 I-3

C **言語**: 変数の「有効範囲 (スコープ)」,「記憶域期間 (記憶寿命)」について学習し，一般的な (ローカル) 変数とグローバル関数，スタティック変数の違いについて学習せよ．

1.3.1 有効範囲 (スコープ)

「有効範囲 (スコープ)」とは，変数などに与えられる識別子が通用する範囲のこと．

1.4 記憶域期間 (記憶寿命)

「記憶域期間 (記憶寿命)」には自動記憶域期間と静的記憶域期間がある．

自動記憶域期間とは auto を使用して宣言した変数が持つ記憶域期間のことで，オブジェクトは宣言したブロックに入ったときから終了するまで生存する．

静的記憶域期間とは，ファイル有効範囲のオブジェクトか static を使用して宣言したときにオブジェクトが持つ記憶域期間のことで，プログラムの開始から終了するまで生存する．

1.4.1 ローカル変数

main 関数など，宣言された関数内でのみ値を保持し，呼び出しできる変数．宣言された関数外で呼び出すことはできない．

1.4.2 グローバル変数

main 関数を代表とするような関数の外で宣言されたもの．プログラム内ならどこからでもアクセスできる．

1.4.3 スタティック変数

静的記憶域期間を持つ変数．プログラムが始まってから終わるまで値を保持し続ける．strtok 関数などに用いられていて，strtok 関数の扱いに注意が必要な原因である．

1.5 課題 I-4

星型正多角形を描き，回転させるプログラムを作成せよ．

1.5.1 プログラム

星形正多角形を描き，回転させるプログラムの主要部をソースコード 1 に示す．

Listing 1: 星形正多角形の描画と回転

```
1  void display() {
2      glClear(GL_COLOR_BUFFER_BIT);
3      glColor3d(1.0, 1.0, 1.0);
4
5      dt = 2.0 * M_PI / NUM;
6      theta = rotAng;
7
8      for (i = 0; i < NUM; i++) {
9          x[i] = cos(theta);
10         y[i] = sin(theta);
11         theta += dt;
```

```
12      }
13
14      for (i = 0; i < NUM; i++) {
15          glBegin(GL_LINES);
16          glVertex2d(x[i], y[i]);
17          glVertex2d(x[(i + 2) % NUM], y[(i + 2) % NUM]);
18          glEnd();
19      }
20
21      glFlush();
22      rotAng += 3.0 * M_PI / 180.0;
23  }
```

定数 NUM を変更することで，星型正多角形の角の数を変えることができる．

星型正多角形を描くには，全頂点において，2 つ隣の頂点に線を引くことで描画することができる．そのため，あらかじめ線を引く頂点の座標を配列に格納し，GL_LINES を用いることで線を引く．変数 theta をインクリメントすることで全頂点においてこの作業を行う．

1.6 課題 I-5

完全グラフを描き，回転させるプログラムを作成せよ．

1.6.1 プログラム

完全グラフを描き，回転させるプログラムの主要部をソースコード 2 に示す．

Listing 2: 完全グラフの描画と回転

```
1  void display() {
2      glClear(GL_COLOR_BUFFER_BIT);
3      glColor3d(1.0, 1.0, 1.0);
4
5      dt = 2.0 * M_PI / NUM;
6      theta = rotAng;
7
8      for (i = 0; i < NUM; i++) {
9          x[i] = cos(theta);
10         y[i] = sin(theta);
11         theta += dt;
12     }
13
14     for (i = 0; i < NUM; i++) {
15         for (j = i + 1; j < NUM; j++) {
16             glBegin(GL_LINES);
```

```
17              glVertex2d(x[i], y[i]);
18              glVertex2d(x[j], y[j]);
19              glEnd();
20          }
21      }
22
23      glFlush();
24      rotAng += 3.0 * M_PI / 180.0;
25  }
```

定数 NUM を変更することで，完全グラフの角の数を変えることができる．

完全グラフを描くには，全頂点において，他の頂点全てに線を引くことで描画することができる．そのため，あらかじめ線を引く頂点の座標を配列に格納し，GL_LINES を用いることで線を引く．その際，すでに線を引いてある 2 点間について，重ねて線を引かないよう，15 行目に書いてあるようにループ数を減らしていく．変数 theta をインクリメントすることで全頂点においてこの作業を行う．

1.7 課題 I-6

リスト 12 を解析し，数学関数を描くプログラムを作成せよ．

1.7.1 カージオイド

カージオイドを描画するプログラムの主要部をソースコード 3 に示す．

Listing 3: カージオイドの描画

```
1   void display(void){
2     glColor3d(0.0, 0.0, 1.0);
3     glBegin(GL_LINE_STRIP);
4
5     for (theta = 0; theta <= 2 * M_PI; theta += 2 * M_PI /
          100) {
6       x = cos(theta) * (1 + cos(theta));
7       y = sin(theta) * (1 + cos(theta));
8       glVertex2d(x, y);
9     }
10
11    glEnd();
12    glFlush();
13  }
```

1.7.2 サイクロイド

サイクロイドを描画するプログラムの主要部をソースコード 4 に示す.

Listing 4: サイクロイドの描画

```
1  void display(void){
2    glColor3d(0.0, 0.0, 1.0);
3    glBegin(GL_LINE_STRIP);
4
5    for (theta = 0; theta <= 2 * M_PI; theta += 2 * M_PI /
         100) {
6      x = theta - sin(theta);
7      y = 1 - cos(theta);
8      glVertex2d(x, y);
9    }
10
11   glEnd();
12   glFlush();
13 }
```

1.7.3 4 尖点の内サイクロイド

4 尖点の内サイクロイドを描画するプログラムの主要部を 5 に示す.

Listing 5: 4 尖点の内サイクロイドの描画

```
1  void display(void){
2    glColor3d(0.0, 0.0, 1.0);
3    glBegin(GL_LINE_STRIP);
4
5    for (theta = 0; theta <= 2 * M_PI; theta += 2 * M_PI /
         100) {
6      x = (cos(theta)) * (cos(theta)) * (cos(theta));
7      y = (sin(theta)) * (sin(theta)) * (sin(theta));
8      glVertex2d(x, y);
9    }
10
11   glEnd();
12   glFlush();
13 }
```

2　課題 II

2.1　課題 II-1

2.2　課題 II-2

2.3　課題 II-3

2.4　課題 II-4

3　課題 III

3.1　課題 III-1

3.2　課題 III-2

3.3　課題 III-3

3.4　課題 III-4

3.5　課題 III-5

4　感想

今まで学習していた内容もしっかり扱った上で，更に進んだ内容を学べれて良かったと感じる．色々なことが重なってしまったことでこの科目のみに集中できる時間をあまり取れなかったことが悔やまれる．総合制作はこのレポートとは違い，ギリギリにならない様に進めていきたい．

5　改善案

科目名から授業で扱う内容がなんとなくわかると良いと思った.「プログラミング演習 II」ではプログラミングをすることしか分からず，ややもったいないと感じる．

参考文献

1. 高橋章,R05-Ec5 プログラミング演習 II テキスト
2. https://tokoik.github.io/GLFWdraft.pdf
3. https://www.glfw.org/
4. https://forest.watch.impress.co.jp/article/1999/07/13/glui.html

5. https://www.dospara.co.jp/5info/cts_str_pc_vulkan
6. https://www.khronos.org/opengles/
7. https://wgld.org/
8. https://docs.oracle.com/cd/E19957-01/806-4833/Make.html
9. https://eng-entrance.com/linux-command-grep
10. https://atmarkit.itmedia.co.jp/ait/articles/1606/14/news013.html
11. https://cmake.org/
12. https://subversion.apache.org/
13. https://git-scm.com/
14. http://ki-www.cvl.iis.u-tokyo.ac.jp/thesis/senior/inaguma.pdf