

コンピュータグラフィックスS レポート課題

九州工業大学 情報工学部 システム創成情報工学科
講義担当：尾下真樹

レポートの提出方法

- レポートの締め切りは、8月上旬（期末試験終了後）を予定。（詳細は後日連絡）直前になって慌てなくても良いように、課題には、十分早く取りかかること。
- レポートは、Moodleの本講義のコースから提出すること。電子メールや紙での提出は、一切認めない。
- 課題を実現するプログラムを作成し、プログラムとレポートの両方のファイルを提出すること。
- レポートのファイル形式は、MS Word または PDF とすること。
- レポートのファイル名は学生番号にすること。例えば、学生番号 13236001 であれば、“13236001.docx”、“13236001.doc”、“13236001.pdf” などのファイル名で提出する。
- レポートには、自分の作成したプログラムのソースファイルを添付して提出すること。全ての課題を作成した1つのプログラムのソースファイルを提出する。（課題ごとに別々のファイルを提出する必要はない。）
こちら、ファイル名は学生番号にすること。（学生番号が 13233001 であれば、“13233001.c”、“13233001.cpp”、などのファイル名で提出）。ソースファイルの（コメントの日本語の）文字コードは、ユニコード、シフト JIS、EUC、いずれでも構わない。
- 上記の提出方法を守れなかった場合、レポートが評価されなかったり（0点）、減点の対象となったりすることがあるので、注意すること。
- その他、変更点や補足事項があれば Moodle の本講義のコースで連絡をするので、適宜確認すること。

レポートの作成方法

- レポートの先頭ページには、科目名、提出日、学生番号、氏名を書くこと。
- レポートの本文は、2 ページ目から始めて、課題ごとに章立てし、指示された内容を説明すること。長々と文章を書く必要はないので、指示された内容を適切に示すこと。
- プログラムは、授業の演習で作成したプログラムをもとに作成して構わないが、ソースファイルを見やすくするため、授業の演習で作成した内容のうち今回のレポートに関係のない不要な変数や処理はソースファイルから削除すること。不要な変数・処理が残っているプログラムは、減点の対象とする。

レポートの評価方法

- 本課題の成績は、提出されたレポート、及び、プログラムにより評価する。
- レポートについては、指示された内容（図・データ・数式・説明）が適切に示されているかどうかを評価する。仮にプログラムが正しく動いていても、プログラムの内容が正しくレポートで示されていないと、かなり低い評価となるので、注意すること。また、レポートの書き方（図、数式、説明がきちんと見やすく示されているか）も評価の対象とする。
- プログラムについては、単に正しく動くかどうかだけでなく、プログラムの中身（見やすく書かれているか、適切なコメントが書かれているか、余計な処理が入っていないか、インデントは適当か、など）も評価の対象とする。正しく動いていても、プログラムが見にくければ減点となるので、注意すること。
- カンニングは厳重な処罰や減点の対象となるので、絶対に行わないこと。例え一部分であっても、他人のレポートや資料を丸写ししたプログラムやレポートは、カンニングと判断される。どうしても分からないところを他人に相談することは構わないが、プログラムやレポートは全て自力で作成すること。（授業で資料として配布したプログラムについては、必要箇所をコピーして利用するのは構わない。）

サンプルプログラム

Moode に、各自がレポートでどのようなプログラムを作成すべきかを示す、サンプルプログラム (opengl_report2015) を用意している。

サンプルプログラムを起動すると、ログイン ID にもとづいて自動的に学生番号が取得され、学生番号に応じた、課題となる目標プログラムが示される。必ず、自分のログイン ID を使ってログインして、サンプルプログラムの内容を確認すること。なお、本授業の履修登録を行っていない場合は、エラーメッセージが表示され、学生番号や課題内容は表示されない。

目標プログラムと同じモデル描画 (課題 1)、視点操作 (課題 2)、アニメーション (課題 3)、テクスチャマッピング (課題 4) を実現するプログラムを作成すること。ただし、これら以外の指定されていない点については、サンプルプログラムと全く同じにする必要はない。例えば、オブジェクトの影の描画は不要である。

サンプルプログラムの操作方法は、以下の通り。

- A キーを押すと、アニメーションを再生する代わりに、途中のいくつかの位置・向きを同時に描画する。アニメーションの軌道を詳しく確認したいときなどに参考にとすると良い。
- スペースキーでアニメーションの一時停止と再開。
- N キーでコマ送り (アニメーションを一時停止した状態で)。
- R キーでアニメーションをリセット (開始時の位置・向きに戻る)。
- マウスのボタンをドラッグすることで視点操作 (課題 2 の説明の通り)。

課題 1. ポリゴンモデルの描画

サンプルプログラムで示されたものと同様のモデルを表示するようなプログラムを作成せよ。

モデルの形状は、できる限りサンプルプログラムで示された形状に近くなるようにすること。

(大幅に異なる形状や、勝手に簡略化された形状を表示するようなプログラムは、減点の対象とする)。

モデルの形状データは、必ずプログラム内に配列として持つようにすること。講義で最初に説明した方法 (glVertex3f() の中に直接頂点座標を記述する方法) は、実用的なやり方ではないため、認めない。

モデルの法線は、必ず、形状に合った正しい法線を用いること。

レポートには、以下の内容を記述すること。

どのような頂点座標・ポリゴンデータを使用したか (使用したデータの頂点・面・法線の数やそれぞれの値) を示すこと。配列データの数値だけではなく、使用したデータがどのようにモデルの形状を構成するのかが分かるように、必ず、三面図や透視図などの図を示すこと。また、各面の法線ベクトルの数値も示すこと。

課題 2. 視点操作インターフェースの追加

サンプルプログラムと同様の視点操作を実現するよう、下記のようなユーザインターフェースを追加せよ。

- マウスの右ボタンを押しながら、マウスを前後にドラッグすると、ワールド座標系の原点 (注視点) を中心にカメラが上下に回転 (原点から見たカメラの仰角が変化)。(演習で作成したものと同じ)
- マウスの左ボタンを押しながら、マウスを前後にドラッグすると、カメラと原点 (注視点) の間の距離が変化する。(演習で作成したものと同じ)
- マウスの右ボタンを押しながら、マウスを左右にドラッグすると、ワールド座標系の原点 (注視点) を中心にカメラが左右に回転 (原点から見たカメラの方位角が変化)。(演習では作成していないので新たに追加)

- マウスの左ボタンを押しながら、マウスを左右にドラッグすると、ワールド座標系の X 軸に沿って、カメラの注視点が移動（ワールド座標系の X 軸方向にカメラが移動）。(演習では作成していないので新たに追加)

上記の文章での説明でよく分らなければ、サンプルプログラムの動作を十分に確認して、同様の視点操作を実現すること。

レポートには、以下の内容を記述すること。

ワールド座標系からカメラ座標系への変換行列を平行移動行列・回転行列の積の形で記述せよ。

行列中で使用されている変数の定義を示すこと。それぞれが、どのような操作を実現するための変数であり、どのようなマウス操作によって変化するかを説明すること。

なお、プログラムで使用している変数との対応がきちんと示されていれば、説明の中の変換行列で使用する変数として、プログラムで使用している変数名を無理に使う必要はない。(例えば、プログラムで使用している `camara_pitch` という変数を θ とおいて、説明の中の変換行列は θ を使って記述しても構わない。ただし、プログラム中の変数と説明で使う変数の対応関係を、きちんと示すこと。)

課題 3. アニメーションの追加

サンプルプログラムと同様のアニメーションを実現する処理を追加する。

サンプルプログラムの通りに、4つのオブジェクトを動かし、アニメーションを表示すること。

また、オブジェクトの色も、サンプルプログラムと同様の色にすること。

レポートには、以下の内容を記述すること。

課題 2 と同様に、4つのオブジェクトのそれぞれについて、どのような変換行列を使用して描画すれば良いか、変換行列を平行移動行列・回転行列の積の形で記述せよ。

ただし、課題 2 で作成したワールド座標系からカメラ座標系への変換行列は、行列 A などと置いて省略して構わない。モデル座標系からワールド座標系への変換行列がきちんと分るように示すこと。

課題 4. テクスチャマッピング

サンプルプログラムと同様のテクスチャマッピングを行い、地面を描画する処理を追加する。

Moodle にアップロードされている画像ファイル `kyutech.bmp` を読み込み、適切なテクスチャ座標を設定して、地面の描画を行う処理を追加すること。(画像ファイルは提出時に添付する必要はない。)

地面は、授業中の演習で作成したプログラムと同様、ワールド座標系の原点を中心として X 軸方向・Z 軸方向にそれぞれ幅 10 の大きさの四角形として描画すること。シェーディングによる影響がきちんと生じるように、1枚の四角形として描画するのではなく、授業中の演習で作成したように、 10×10 の小さな四角形の集まりとして描画をすること。

また、光源は、(5, 3, 5) の位置にある点光源とすること。点光源・環境光源の色等の設定は、授業中の演習で作成したプログラムと同様にすること。

レポートには、テクスチャ画像のどの範囲を地面にマッピングするのか、テクスチャ座標 (u, v) を示した上で、 i 行 j 列目の四角形を描画する際の 4 点の頂点座標 (x, y, z)・テクスチャ座標 (u, v) の計算式を示すこと。(それぞれを i, j の関数の形で示す。)

付録 1. Linux-Windows 間でのファイルのコピー方法

情報科学センタの Linux 端末以外でプログラムを作成したい者は、下記の情報を参考にすること。ただし、情報科学センタ以外の環境については、一切サポートはしないので、やりたい者は自力でやること。また、提出されたプログラムは、情報科学センタの Linux 端末の環境で評価されるため、別の環境で正しく動いたとしても、情報科学センタの環境で正しく動作しなければ大幅な減点となるので、注意すること。

○ 情報科学センタの Linux 端末から、Windows マシンにファイルを移す方法

情報科学センタのオンラインガイドのページ (<http://edu.iizuka.isc.kyutech.ac.jp/Guide/>) に、USB メモリの使用法の説明が書かれているので、やり方が分からない人は参考にすること。この方法を使えば、USB メモリを使って、他のマシンにファイルを移すことができる。

USB メモリを持っていない人は、電子メールにファイルを添付して送り、別のマシンでメールを受信するなどの方法で、ファイルを移すこともできる。電子メールへのファイルの添付方法は、1年生のリテラシーの講義で習った通り。

○ プログラムの文字コードを変換する方法

Linux と Windows では、テキストファイルの 2 バイト文字コードや改行コードが異なるため、普通にファイルをコピーしても、コンパイルすることができない。日本語の文字コードとして、Windows ではシフト JIS やユニコード、Linux では EUC やユニコードが使用されている。また、改行コードとして、Windows では CR+LF、Linux では LF が使用されている。

授業で用いたサンプルプログラムはユニコードが使われているため、ユニコードが利用可能な環境で使用する場合は、文字コードの変換は必要ない。

ユニコードが使えない環境やソフトウェアを使って演習を行いたい場合は、下記のように、Linux にインストールされている `nkf` コマンドなどを使うことで、文字コードや改行コードを変換できる。

Windows にも、文字コード変換のためのフリーソフトなどが多くあるが、標準では付属していないため自分でインストールする必要がある。

演習環境 (Linux) 用の文字コードから、Windows 用の文字コード (S-JIS、CR+LF) へ変換する方法。

```
$ nkf -s -Lw opengl_euc.c > opengl_sjis.c
```

Windows 用の文字コードから、演習環境 (Linux) 用の文字コード (ユニコード、LF) へ変換する方法。

```
$ nkf -w -Lu opengl_sjis.c > opengl_euc.c
```

付録 2. よくある間違い

基本的には以上の説明に従ってレポートを作成すれば問題ないが、以下に、過去のレポートでよくある間違いを挙げておくので、特にこのような間違いをしないように十分注意すること。(ここで挙げられている以外の間違いにも、注意すること。)

- ポリゴンモデルの間違い。
最も多い間違いとして、法線ベクトルの向きが正しくないものや、法線ベクトルが正規化されていない、などがある。法線ベクトルは、面に対して垂直で、かつ長さが 1 となる必要がある。
頂点や面の情報と比べて、法線ベクトルの間違いにはやや気がつきにくいですが、法線ベクトルが間違っていると、実行結果を見たときに光による照らされ方が明らかにおかしくなることが多いので、プログラムを見直すだけではなく、実行結果も見て確認すると良い。
- 視点操作が正しく実現されてない。
特に、左ボタンを左右にドラッグしたときの注視点の移動を間違った座標系で行っていると、指定されたプログラムと異なる結果になってしまうので、十分に確認すること。
- 指定されたアニメーションが作成されていない。
それぞれの物体の動きの軌道を指定された通りにするのはもちろん、物体同士の運動周期も指定された通りになるよう、注意して作成すること。例えば、赤い物体が 1 回転する間に、青い物体が何回転するかなど、指定されたプログラムでの物体同士の運動周期やタイミングも再現するようにする。
- プログラムが見にくい。
プログラムの動作は正しくても、かなりプログラムが読みにくい人があるので、注意すること。プログラムが非常に読みにくい場合、減点の対象となる。
一般に、プログラムが見にくいと、バグを見つけるときなどに苦労するので、見やすいプログラムを書くよう心がけることが必要である。具体的には、インデントをきちんと揃える、適切なコメントや関数名・変数名をつける、無駄な空行は省く、などの工夫が必要。
- 提出されたプログラムにエラーがある。コンパイルできない。(論外)
まれに、コンパイルすらできないプログラムを提出する人がいるので、十分注意すること。
プログラム中に全角スペースが混じっている、コメントが適切にコメントアウトされていない、存在しないファイルをインクルードしている、など。
提出直前になってプログラムを読みやすくしようとして修正し、結果的に、修正を間違えてコンパイルできないプログラムを提出してしまう人がいるので、プログラムを修正した場合は、必ずコンパイルができて正しく動作することを確認した上で、提出すること。
また、たまたまコンパイルは通る場合でも、C 言語のルールに従っていないかたり、コンパイル時に警告が発生したりするような間違ったプログラムも、減点の対象となる。
- レポートの課題 1 で、ポリゴンモデルを構成する頂点や面の情報が図から分からない。
課題の説明で指示されている通り、どのようなデータを使用してポリゴンモデルを実現したのかを説明するためには、ポリゴンモデルの形状や頂点座標、各ポリゴンを構成する頂点番号、などが分かるような図を使って説明することが必要である。
レポートに図が全くない場合や、不十分な図しかない場合は、どのようなデータを使用したのかが分からず、減点の対象となるので、注意すること。
特に、頂点座標の情報だけでなく、物体をどのようなポリゴン(面)に分割して描画したかや、各ポリゴンが何番目のデータに対応するのか、といった、ポリゴンの情報もきちんと分かるようにすることが必要である。また、頂点の座標値が分かるように、座標軸・目盛を示すなどの工夫も必要である。
基本的に、定義した全てのデータ(頂点座標、法線、ポリゴンなど)について、定義したデータ(配列)が正しいかがきちんと検証できるような図や説明が求められる。
- レポートの課題 2~3 で、変換行列が正しく示されていない。あるいは、プログラムで用いられている変換行列とレポートで示されている変換行列が食い違っている。