

コンピュータグラフィックスS 演習資料

サンプルプログラムのコンパイル方法

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

1. 開発環境

本授業の OpenGL プログラミング演習は、マルチメディア講義室で、Windows + Visual Studio2017 を使用して行う。本資料では、本環境でのサンプルプログラムのコンパイル方法について説明する。また、情報科学センターの端末室で Linux + gcc を使用してサンプルプログラムのコンパイルを行う方法についても簡単に説明する。

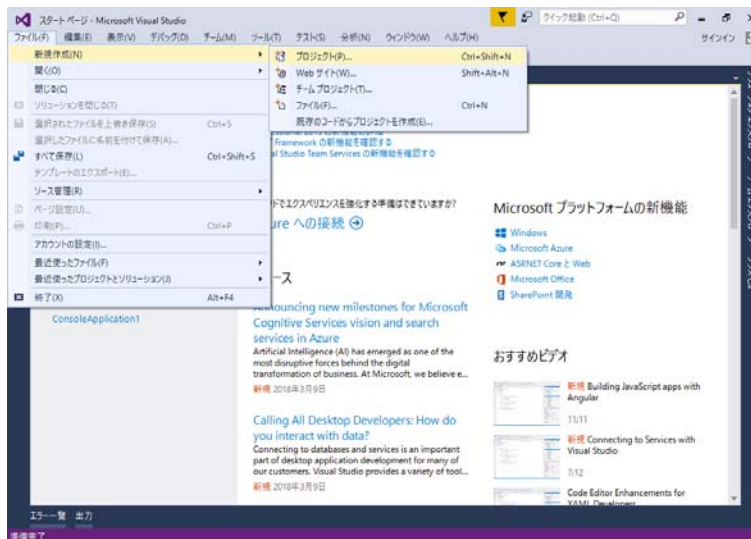
2. Windows + Visual Studio 2017 環境（マルチメディア講義室）でのコンパイル

Visual Studio の統合環境を使ってプログラミングを行う場合、最初に「プロジェクト」や「ソリューション」を作成する必要がある。「プロジェクト」はプログラムを構成するソースファイルやコンパイル方法の設定を定義したものである。また、「ソリューション」は複数の「プロジェクト」をまとめたものである。一つのソースファイルのみで構成される単純なプログラムをコンパイルする場合であっても、「プロジェクト」や「ソリューション」を作成する必要がある。

以下、プロジェクト+ソリューションの作成、ソースファイルの追加、コンパイル、実行の手順を示す。ここでは、Z:\cg\opengl_sample ディレクトリの下に、opengl_sample という名前のプロジェクトを作成して、あらかじめ用意されているソースファイル opengl_sample.cpp を追加し、コンパイル・実行する場合の例を示す。別のディレクトリにプロジェクトを作成したい場合や、別のソースファイルをコンパイルする場合は、適宜、読み替えること。

2.1. プロジェクト+ソリューションの作成

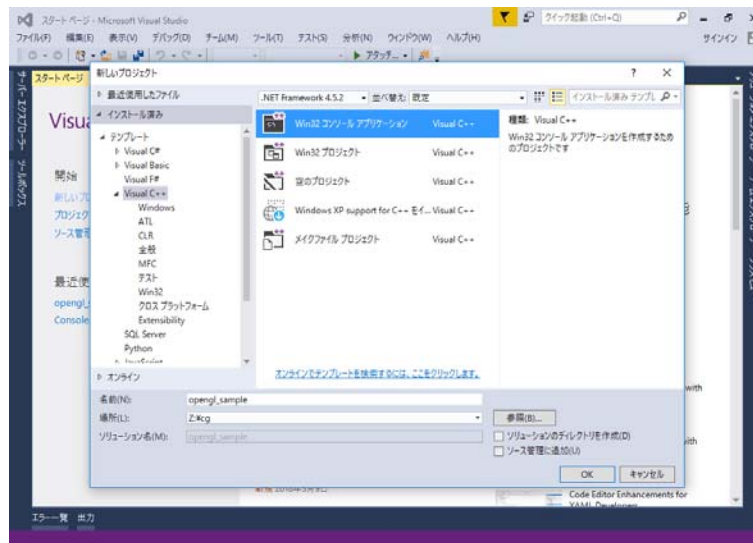
Visual Studio 2017 を起動する。起動時に環境設定画面が表示されたら、開発設定に「Visual C++」を選択する。（他の項目は、適当な設定で構わない。Visual Studio へのサインインも不要。）メニューから、ファイル → 新規作成 → プロジェクトを選択する。



新しいプロジェクトの画面から、「Visual C++」 → 「Win32 コンソールアプリケーション」を選択する。プロジェクトの「名前」と「場所」は、適当なものを指定する。

この例では、名前を「opengl_sample」、場所を「Z:\vcg」としている。実際には、指定した場所の下に指定した名前のディレクトリが作成されるため、この場合、プロジェクトのファイルは Z:\vcg\opengl_sample ディレクトリに生成されることになる。

また、「ソリューションのディレクトリを作成」のオプションは不要なので、無効にする。このオプションを有効にすると、ソリューションのディレクトリの下に、さらにプロジェクトのディレクトリが作成され、無駄にディレクトリ階層が深くなるため、一つのプロジェクトからなるソリューションを作成するときには無効にするのが良い。

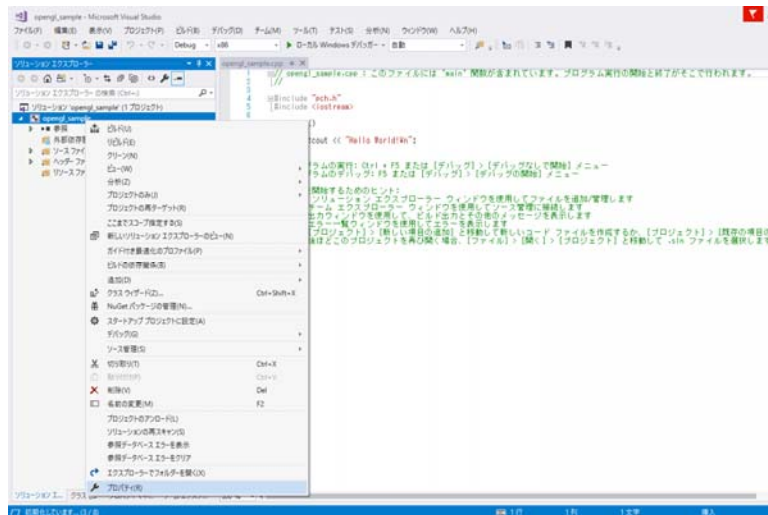


以上の操作により、ソリューション+プロジェクトのファイルが生成される。ソリューションは拡張子が `sln` のファイルであり、プロジェクトは拡張子が `vcxproj` のファイルである。それ以外にも、Visual Studio が使用するファイルやディレクトリがいくつか作成される。

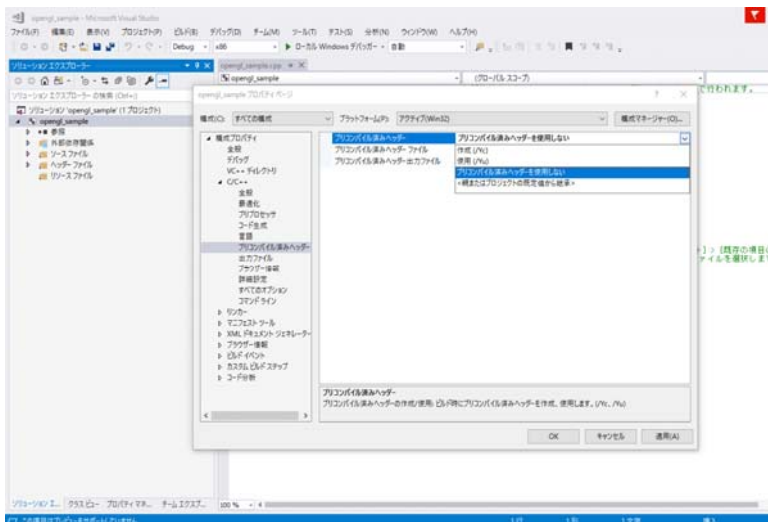
2.2. プロジェクトの設定

プロジェクトのコンパイル・リンク方法の設定を変更するためには、「ソリューションエクスプローラ」内のプロジェクトを右クリックして、メニューから「プロパティ」を選択する。

(※ Visual Studio 2015 までは、プロジェクトを生成するときに、プリコンパイル済みヘッダーの利用の有無などをウィザード形式で選択できるようになっていたが、Visual Studio 2017 では、デフォルトの設定で自動的にプロジェクトが生成されるため、必要に応じて手動で設定を変更する必要がある。)



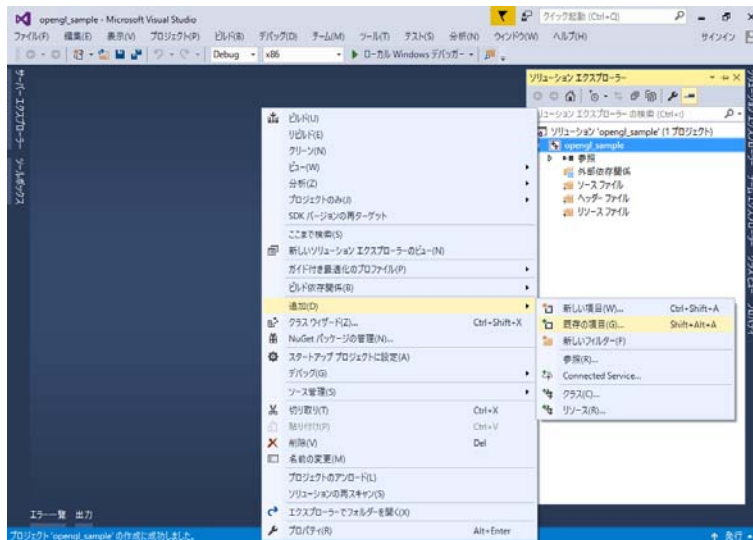
プロジェクトの設定の「すべての構成」の、「C/C++」→「プリコンパイル済みヘッダー」を、「プリコンパイル済みヘッダーを使用しない」にする。
 プリコンパイル済みヘッダーは、Visual Studio 独自の機能で、ヘッダーファイルのコンパイル結果を保存してコンパイルを高速化する機能だが、使用方法がやや複雑になるため、よく分からない場合は、使用しない方がよい。



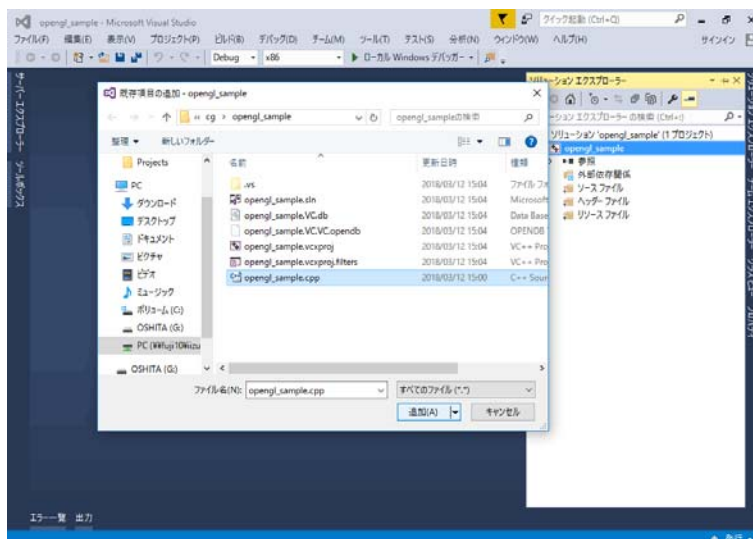
2.3. プロジェクトにソースファイルを追加

ソリューション+プロジェクトが生成されたら、「ソリューションエクスプローラ」から、ソースファイルの追加や削除ができる。
 プリコンパイル済みヘッダーを使用しない場合は、デフォルトにプロジェクトで追加されているいくつかのソースファイル (pch.cpp, pch.h などのファイル、または、stdafx.h, stdafx.cpp, version.h などのファイル) は、全てプロジェクトから「削除」して構わない。

既存のソースファイル opengl_sample.cpp をプロジェクトに追加する手順は、以下の通りである。
 「ソリューションエクスプローラ」内のプロジェクトを右クリックして、メニューから 追加 → 既存の項目 を選択する。



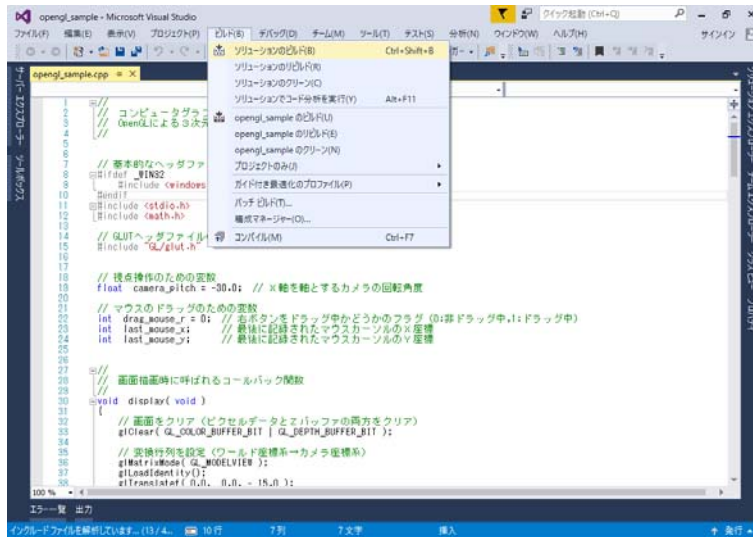
ファイルダイアログが表示されるので、プロジェクトに追加するソースファイル `opengl_sample.cpp` を選択する。あらかじめ、プロジェクトのディレクトリにソースファイル `opengl_sample.cpp` をコピーしておき、そのファイルを選択する。



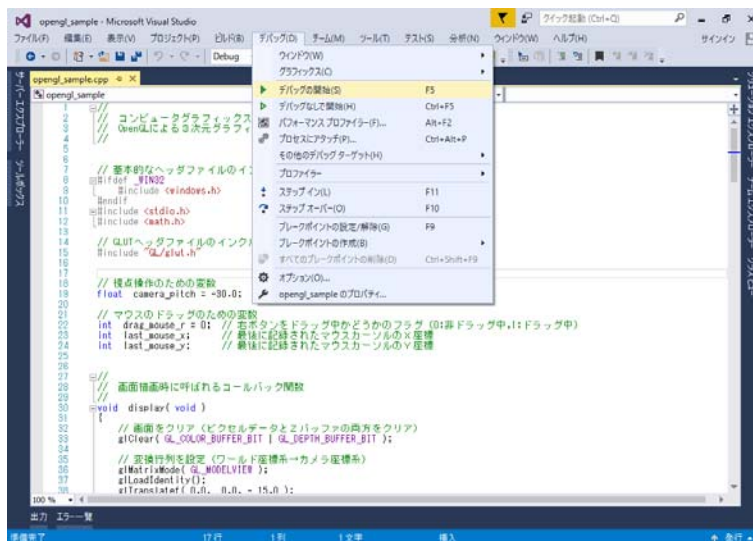
「ソリューションエクスプローラ」に追加されたソースファイルをダブルクリックすると、ソースファイルの中身が表示され、編集できる状態になる。

2.4. コンパイルと実行

ソースファイルをコンパイルするときには、メニューから **ビルド** → **ソリューションのビルド** を選択する。

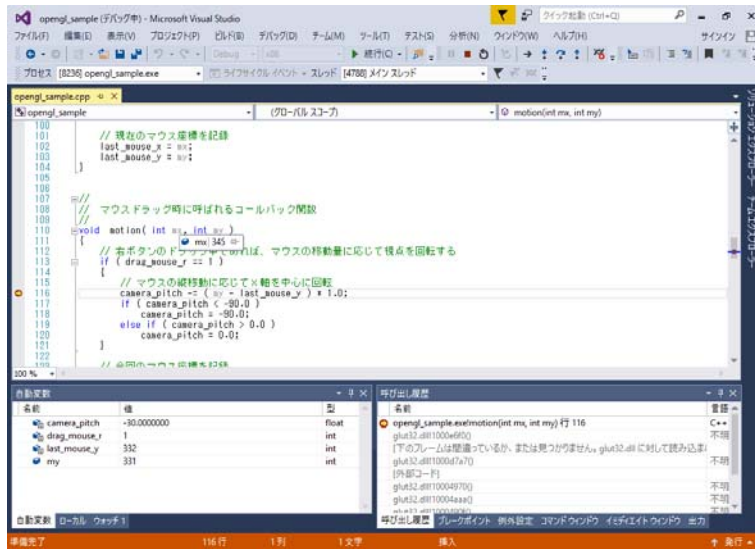


コンパイルに失敗した場合は、エラーメッセージが表示される。コンパイルができない原因や、ソースファイルの何行目でそのエラーが発生したか、といった情報が表示されるので、それらを参考に、ソースファイルを修正する。
 コンパイルに成功した場合は、メニューから **デバッグ** → **デバッグの開始** を選択することで、プログラムを実行できる。



2.5. デバッグ

Visual Studio はデバッガの機能を備えているため、プログラムが正しく動作しないときには、デバッガの機能を利用してプログラムのデバッグを行える。
 ソースファイルの左端をクリックすることで、その場所にブレークポイントを設定し、プログラムの実行中に一時停止することができる。また、プログラムを一行ずつステップ実行したり、プログラム内の変数の値を表示して確認したりすることができる。これらの機能を利用することで、自分が作成したプログラムの間違いや問題を効率的に探すことができる。



3. Windows + Visual Studio 2015 環境（旧マルチメディア講義室）でのコンパイル

ここでは、Visual Studio 2015 を使う場合のコンパイル方法を説明する。マルチメディア講義室の端末環境は、2019年度より、Visual Studio 2017 がインストールされているが、別の環境などで、Visual Studio 2015 を利用する場合には、本節の説明を参照する。プロジェクトの設定の方法に、やや違いがある。

Visual Studio の統合環境を使ってプログラミングを行う場合、最初に「ソリューション+プロジェクト」を作成する必要がある。プロジェクトはプログラムを構成するソースファイルや設定を定義したものである。また、ソリューションは複数のプロジェクトをまとめたものである。一つのソースファイルのみで構成される単純なプログラムをコンパイルする場合でも、ソリューションやプロジェクトを作成する必要がある。

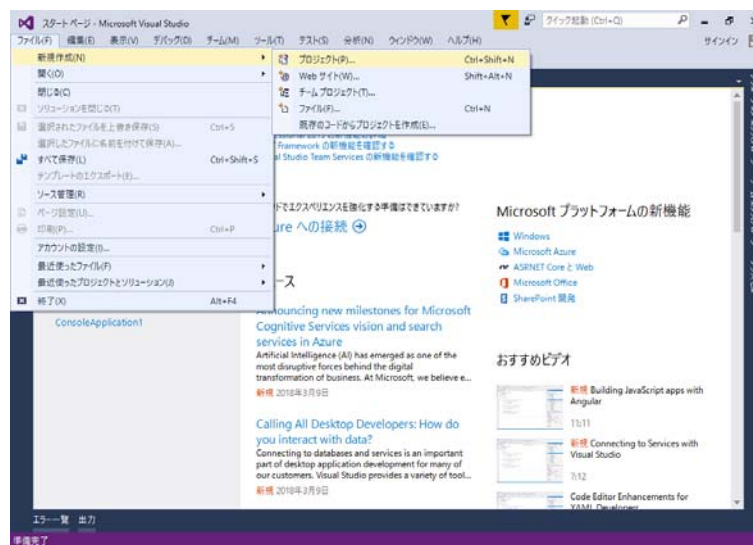
以下、プロジェクトの作成、ソースファイルの追加、コンパイル、実行のための手順を示す。

ここでは、Z:\cg\opengl_sample ディレクトリの下に、opengl_sample という名前のプロジェクトを作成して、あらかじめ用意されているソースファイル opengl_sample.cpp を追加し、コンパイル・実行する場合の例を示す。別のディレクトリにプロジェクトを作成したい場合や、別のソースファイルをコンパイルする場合は、適宜、読み替えること。

3.1. プロジェクト+ソリューションの作成

スタートメニューから、Visual Studio 2015 を起動する。

Visual Studio のメニューから、ファイル → 新規作成 → プロジェクトを選択する。

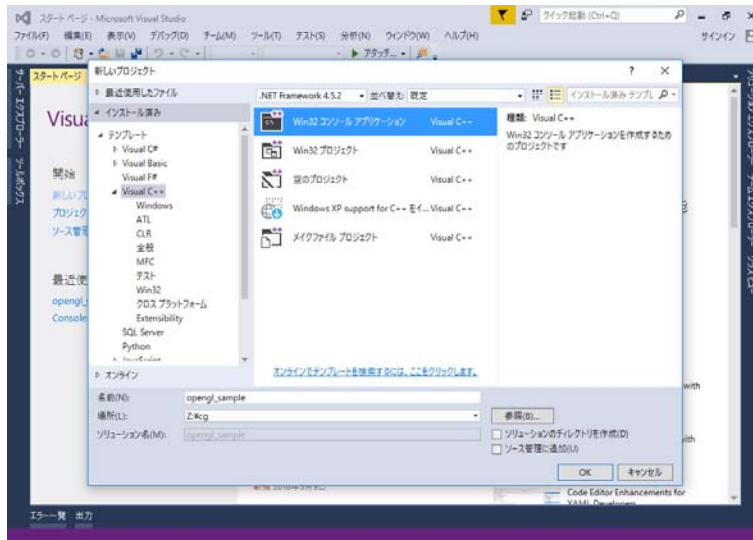


新しいプロジェクトの画面から、「Visual C++」→「Win32 コンソールアプリケーション」を選択する。

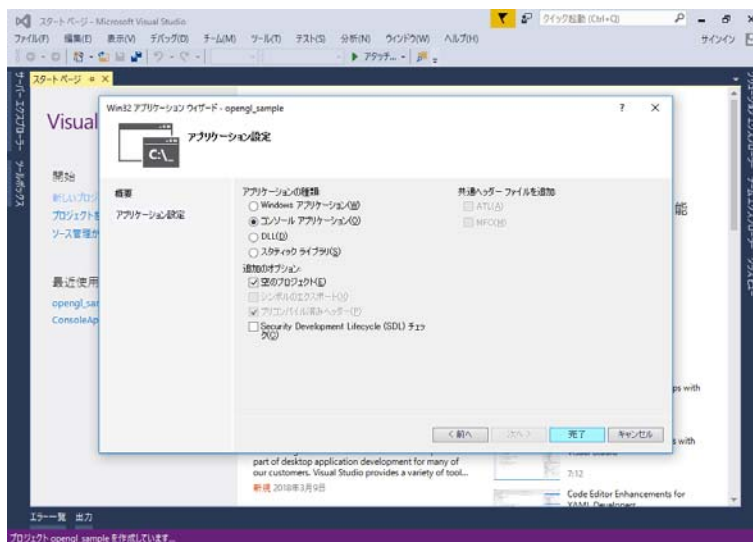
プロジェクトの「名前」と「場所」は、適当なものを指定する。

この例では、名前を「opengl_sample」、場所を「Z:\cg」としている。実際には、指定した場所の下に指定した名前のディレクトリが作成されるため、この場合、プロジェクトのファイルは Z:\cg\opengl_sample ディレクトリに生成されることになる。

また、「ソリューションのディレクトリを作成」のオプションは不要なので、無効にする。このオプションを有効にすると、ソリューションのディレクトリの下に、さらにプロジェクトのディレクトリが作成され、無駄にディレクトリ階層が深くなるため、一つのプロジェクトからなるソリューションを作成するときには無効にするのが良い。



アプリケーションウィザードが表示されたら、「アプリケーション設定」で、追加のオプションの「空のプロジェクト」を有効にする。このオプションを有効にすると、プリコンパイルヘッダ用の設定やファイルが生成されて複雑になるので、これらの機能を使用しない場合は、無効にしておいた方がよい。



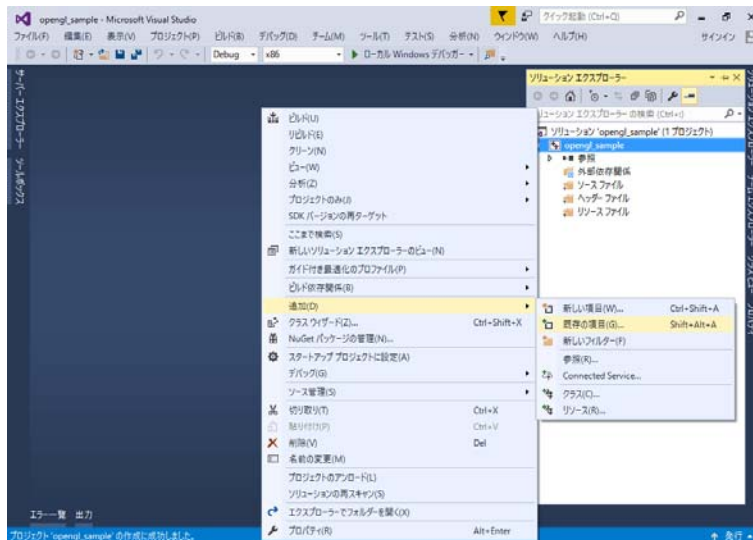
以上の操作により、ソリューション+プロジェクトのファイルが生成される。ソリューションは拡張子が `sln` のファイルであり、プロジェクトは拡張子が `vcxproj` のファイルである。それ以外にも、Visual Studio が使用するファイルやディレクトリがいくつか作成される。

3.2. プロジェクトにソースファイルを追加

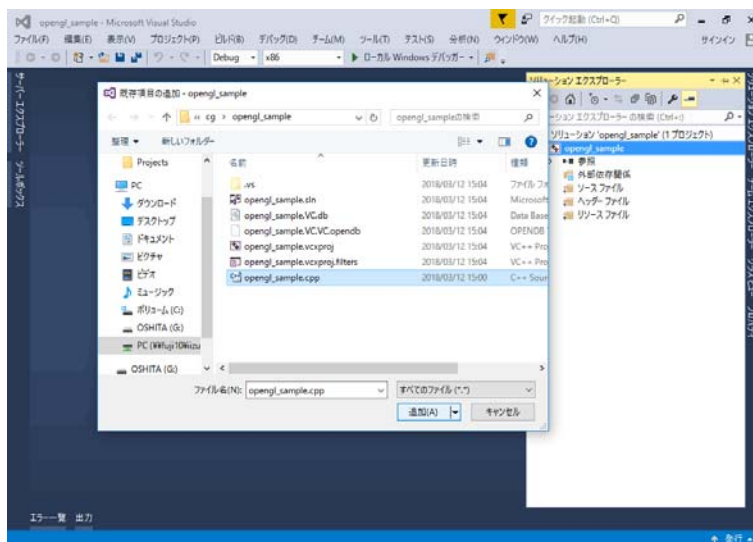
ソリューション+プロジェクトが生成されたら、「ソリューションエクスプローラ」から、ソースファイルの追加や削除ができる。

既存のソースファイル `opengl_sample.cpp` をプロジェクトに追加する手順は、以下の通りである。

「ソリューションエクスプローラ」内のプロジェクトを右クリックして、メニューから 追加 → 既存の項目 を選択する。



ファイルダイアログが表示されるので、プロジェクトに追加するソースファイル `opengl_sample.cpp` を選択する。あらかじめ、プロジェクトのディレクトリにソースファイル `opengl_sample.cpp` をコピーしておき、そのファイルを選択する。



「ソリューションエクスプローラ」に追加されたソースファイルをダブルクリックすると、ソースファイルの中身が表示され、編集できる状態になる。

3.3. コンパイルと実行、デバッグ

上記の Visual Studio 2017 環境の場合と同じなので、上記の説明を参照する。

4. Linux + gcc 環境（情報科学センタ端末室）でのコンパイル

ターミナル上で、ソースファイルが置かれているディレクトリに移動し、以下のコマンドを入力することで、GNU C コンパイラ (gcc) を使用して C プログラムをコンパイルし、実行形式のファイルを生成することができる。

```
gcc opengl_sample.cpp -L/usr/X11R6/lib -lglut -lGLU -lGL -lXmu -lm -o opengl_sample
```

コマンドの引数にある、`opengl_sample.cpp` と `opengl_sample` は、入力のソースファイル名と、出力の実行形式のファイル名を表している。必要に応じて適切なものに変更する。（出力ファイル名を省略すると、`a.out` というファイル名で出力される。）

また、`-L/usr/X11R6/lib` は、ヘッダーファイルを検索する追加ディレクトリを指定している。`-lglut -lGLU -lGL -lXmu -lm` は、それぞれ、`glut`, `glu`, `OpenGL`, `X` ウィンドウシステムのユーティリティライブラリ、算術計算ライブラリをリンクすることを指定している。これらのオプションについては、本演習で扱うプログラムでは全て共通なので、変更する必要はない。

コンパイルに成功すると、出力ファイル名で指定したファイルが生成されるので、以下のようにターミナルからコマンドを入力して、コンパイル結果のプログラムを実行する。

```
opengl_sample
```

コンパイルに失敗した場合は、エラーメッセージが表示される。コンパイルができない原因や、ソースファイルの何行目でそのエラーが発生したか、といった情報が表示されるので、それらを参考に、ソースファイルを修正する。