VR 3D-CG FEM CAD Cloud

UC-1 series UC-win series Suite series

スイート千鳥エンジン Ver.2

Operation Guidance 操作ガイダンス

FORUM8 フォーラムエイト®

フルカラー POD対応

本書のご使用にあたって

本操作ガイダンスは、おもに初めて本製品を利用する方を対象に操作の流れに沿って、操作、入力、処理方法を説明したものです。

ご利用にあたって

ご使用製品のバージョンは、製品「ヘルプ」のバージョン情報よりご確認下さい。
本書は、表紙に掲載のバージョンにより、ご説明しています。
最新バージョンでない場合もございます。ご了承下さい。

本製品及び本書のご使用による貴社の金銭上の損害及び逸失利益または、第三者からのいかなる請求についても、弊社は、その責任を一切負いませんので、あらかじめご承知置き下さい。
製品のご使用については、「使用権許諾契約書」が設けられています。

※掲載されている各社名、各社製品名は一般に各社の登録商標または商標です。

目次

第1章　製品概要

1　スイート千鳥エンジンの概要

スイート千鳥エンジンは、Visual Studioでの開発で使用できるライブラリ集となっております。3Dモデルの表示から各種イベントの制御までを、タスクシステムにより容易に管理できます。また、最新のVisual Studioへの対応や、FBXモデルの取り込みに対応しています。これにより、学生などは無償で利用できるVisual Studio Communityでの開発ができたり、統合型3DCGソフトShade3Dで作成した3Dモデルを取り込んだりすることができます。

必要システム

本製品は、Windows 8.1/10環境を有するOS上で動作します。
また、本製品を使用して開発するためには、Visual Studio 2019が必要となります。
papetプラグインを使用してパーティクルエフェクトデータを作成するためには、Autodesk Maya 2020が必要となります。
なお、Autodesk Maya LT 2020ではpapetプラグインはご利用いただけませんのでご注意ください。

ライセンスについて

スイート千鳥エンジンでは、ライセンス認証を行った場合のみ、作成されたゲームが正しく表示されます。そのため、ライセンス認証を行うためのコードを記述いただく必要があります。

2　フローチャート

第2章　操作ガイダンス

1　事前準備(Visual Stusdio 2019インストール)

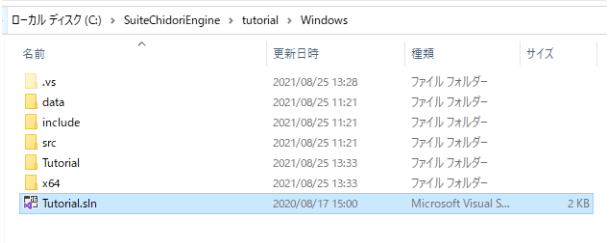
Visual Studio Community 2019をインストールします。
以下URL内の「Visual Studioのダウンロード」ボタンよりダウンロードできます。
https://docs.microsoft.com/ja-jp/visualstudio/releases/2019/release-notes

※Visual Studioは、製品版(Professional, Enterprise)の2019をお持ちの場合は、そちらをご使用いただいても構いません。
なお、2019以外のバージョンを使用すると、正しく動作しない場合があるため、必ず2019を使用してください。
※Visual Studio 2019のインストール時には、「C++ によるデスクトップ開発」にチェックを入れてください。

2　チュートリアル

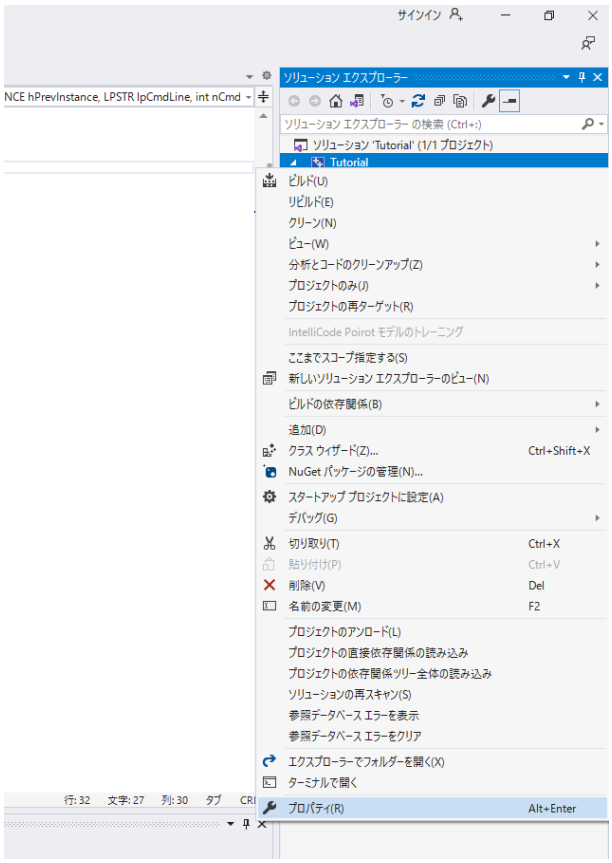
チュートリアル「Tutorial.sln」を例題として作成します。
本ドキュメントでは、モデルの表示とモーションの再生を行うチュートリアルプロジェクトの使用方法と、コード内容を説明します。
■各入力項目の詳細については製品の【ヘルプ】をご覧ください。

2−1　プロジェクトの読み込み

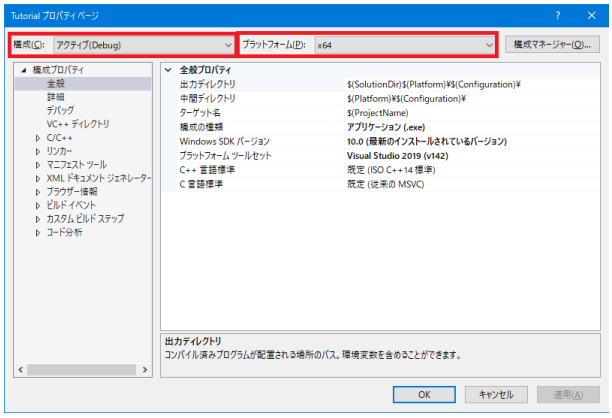


SuiteChidoriEngineのtutorialフォルダ内のWindowsフォルダを開き、「Tutorial.sln」をダブルクリックしてTutorialプロジェクトを開きます。

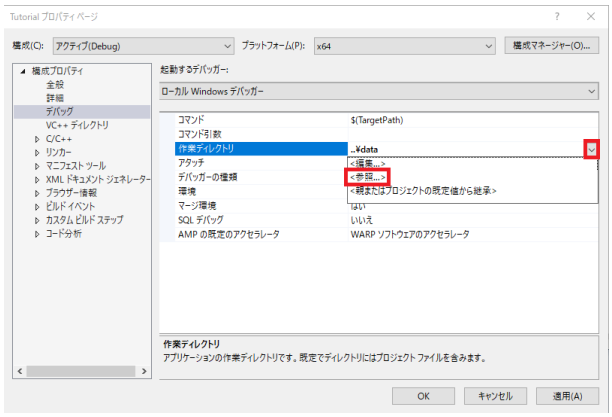
2−2　作業ディレクトリの設定



Visual Studio2019が開かれるので、右側「ソリューションエクスプローラー」-「Tutorial」を右クリックし、「プロパティ」をクリックしてプロパティページを開きます。



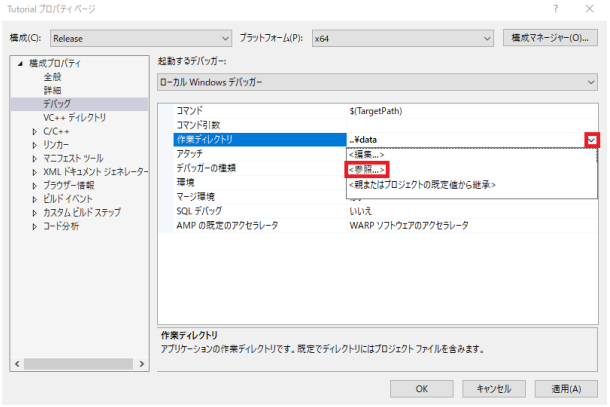
構成(C):で、「Debug」「Release」を切り替える事が出来ます。
「プラットフォーム」を変える必要はありません。



構成(C):
アクティブ(Debug)

プラットフォーム
x64

「構成プロパティ」-「デバッグ」-「作業ディレクトリ」
下矢印選択→<参照...>をクリック→下記フォルダを選択
C:¥SuiteChidoriEngine¥tutorial¥Windows¥data

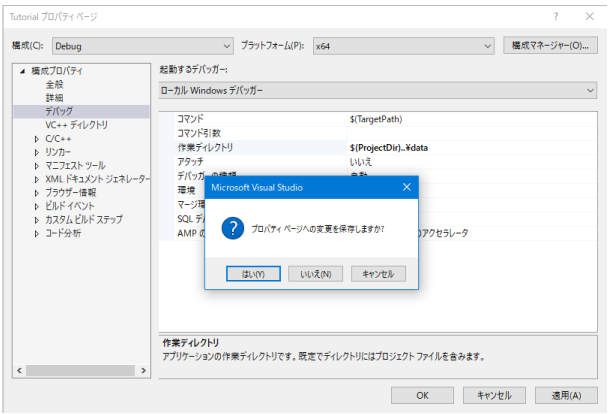


構成(C):
Release

プラットフォーム
x64

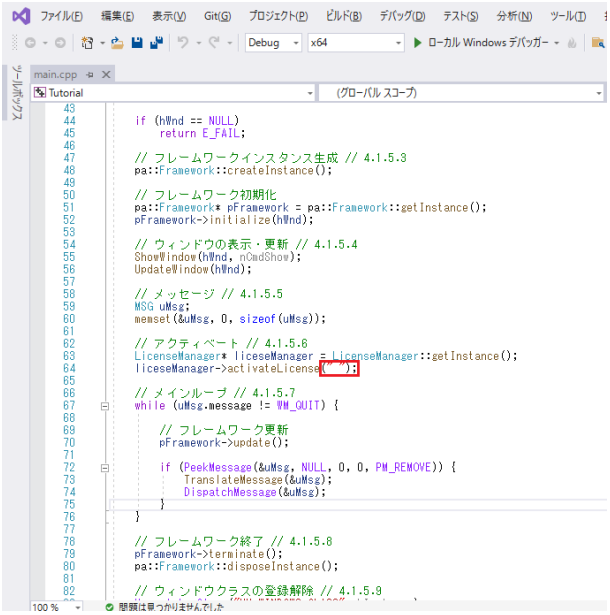
「構成プロパティ」-「デバッグ」-「作業ディレクトリ」
下矢印選択→<参照...>をクリック→下記フォルダを選択
C:¥SuiteChidoriEngine¥tutorial¥Windows¥data

設定が完了したら「OK」をクリックしてプロパティ画面を閉じます。



※「プロパティページへの変更を保存しますか?」とメッセージが出たら「はい」をクリックして下さい。

2−3　ライセンス認証

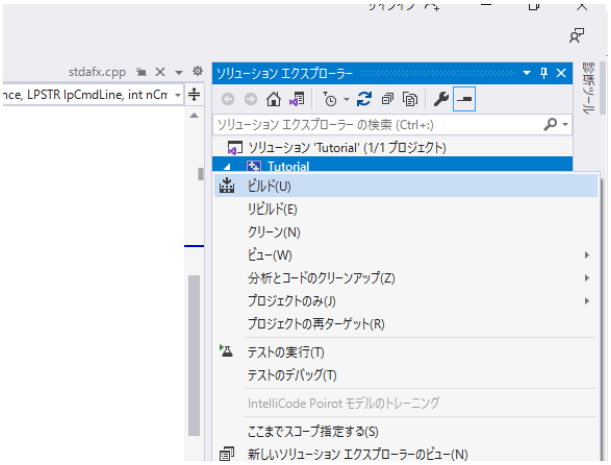


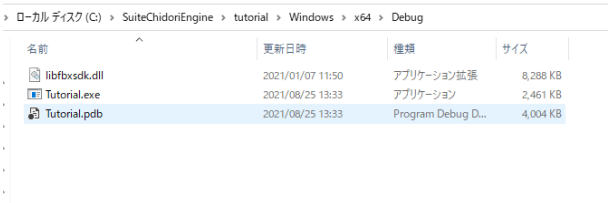
アクティベーションファイル(Chidori.ls)を作業ディレクトリに設置し、右側「ソリューションエクスプローラー」-「main」-「main.cpp」をクリックすると、作業ディレクトリに表示されます。
main.cppのwinMain関数内のactivateLicense(" ")の引数にシリアルコードを記述します。
シリアルコードはハイフンも含めて記述してください。

※アクティベーションファイル(Chidori.ls)とシリアルコードの素用方法について
(Q3-1.参照)
https://www.forum8.co.jp/faq/win/chidori-qa.htm#q3-1

2−4　ビルド

プラットフォームを「x64」に変更して、右側「ソリューションエクスプローラー」-「Tutorial」を右クリックし、「ビルド(U)」をクリックします。





libfbxsdk.dllの設置

生成された.exeファイルと同じディレクトリに「libfbxsdk.dll」を設置します。
※「libfbxsdk.dll」は「(インストール先)¥lib¥dll¥x64」にあるものを使用します。

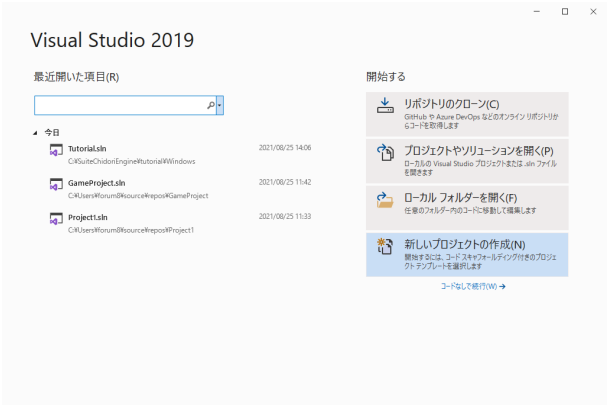
2−5　実行

「Visual Studio 2019」の画面に戻って「F5」を押して実行してください。
成功するとアプリが起動し、動くモデルが表示されます。

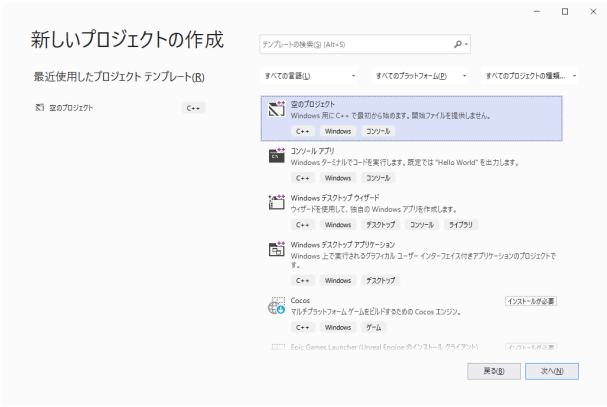
3　環境設定

スイート千鳥エンジンでアプリを作成するための環境設定を行います。

3−1　プロジェクトの作成



「Visual Studio 2019」を起動し、「新しいプロジェクトの作成(N)」をクリックします。



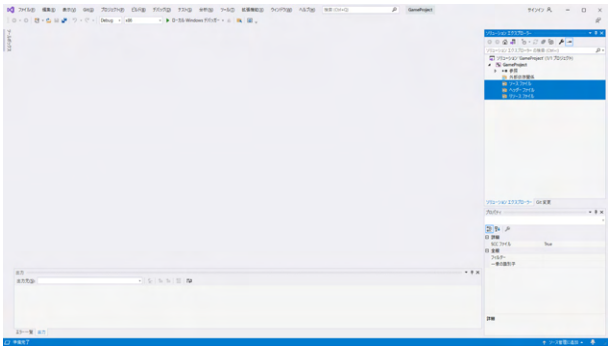
「空のプロジェクト」を選択して、「次へ(N)」をクリックします。

「プロジェクト名(N)」「場所(L)」（任意）を設定し、「ソリューションとプロジェクトを同じディレクトリに配置する(D)」をチェックして「作成(C)」をクリックします。

プロジェクト名
GameProject

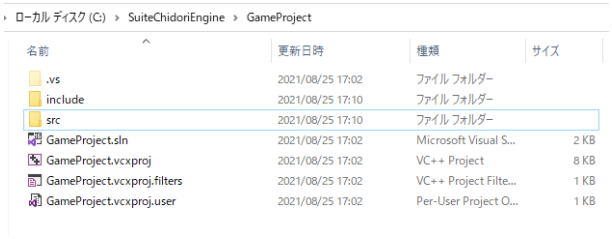
場所
C:¥SuiteChidoriEngine¥

ソリューションとプロジェクトを同じディレクトリに配置する(D)
チェックON

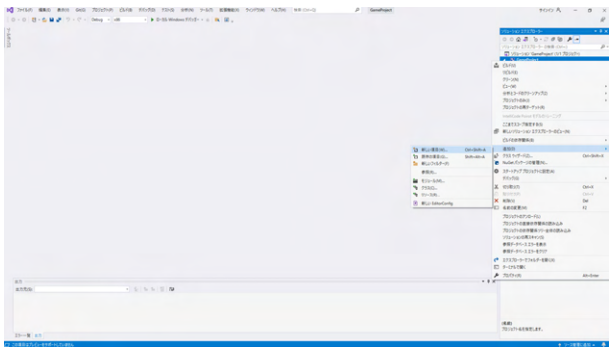


プロジェクトが開いたら、右側「ソリューションエクスプローラー」に表示されている下記フィルターを削除します。

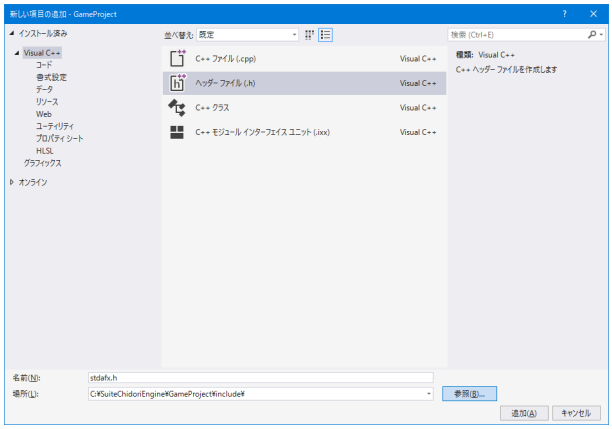
「ソース ファイル」
「ヘッダー ファイル」
「リソース ファイル」



エクスプローラーを開き、プロジェクトフォルダに
「include」
「src」
という名前のフォルダを新規作成します。



「Visual Stusdio 2019」を画面に戻り、右側「ソリューションエクスプローラー」-プロジェクト名(GameProject)を選択し、右クリックで「追加(D)」-「新しい項目(W)」をクリックします。



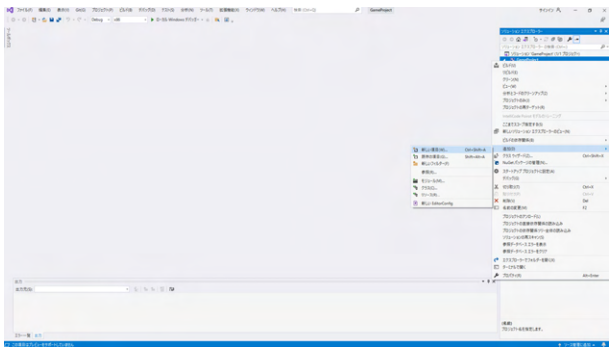
「新しい項目の追加」画面が開くので、設定を行います。

ファイル
ヘッダー ファイル (.h)

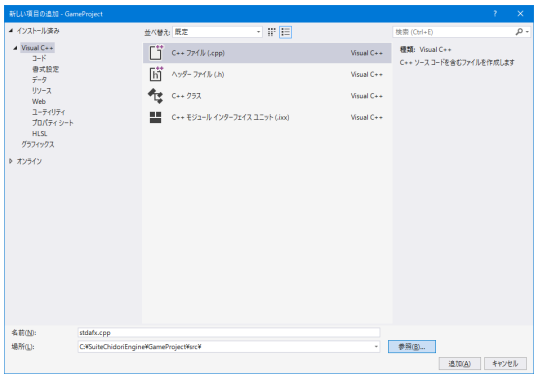
名前
stdafx.h

場所
C:¥SuiteChidoriEngine¥GameProject¥include¥
(先ほどプロジェクト下に作った「include」フォルダを選択します)

「追加(A)」をクリックします。



右側「ソリューションエクスプローラー」-プロジェクト名(GameProject)を選択し、右クリックで「追加(D)」-「新しい項目(W)」をクリックします。

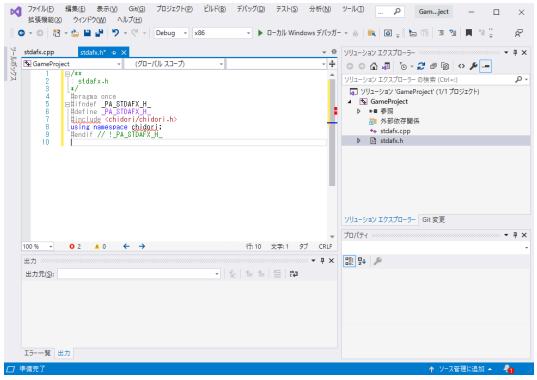


「新しい項目の追加」画面が開くので、設定を行います。

ファイル
C++ ファイル (.cpp)
名前
stdafx.cpp
場所
C:¥SuiteChidoriEngine¥GameProject¥src¥
(先ほどプロジェクト下に作った「src」フォルダを選択します)

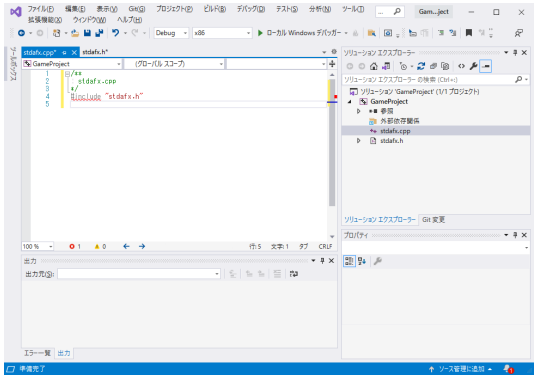
「追加(A)」をクリックします。

※今後は「ヘッダー ファイル」「C++ ファイル」はそれぞれ「include」「src」に追加していきます。



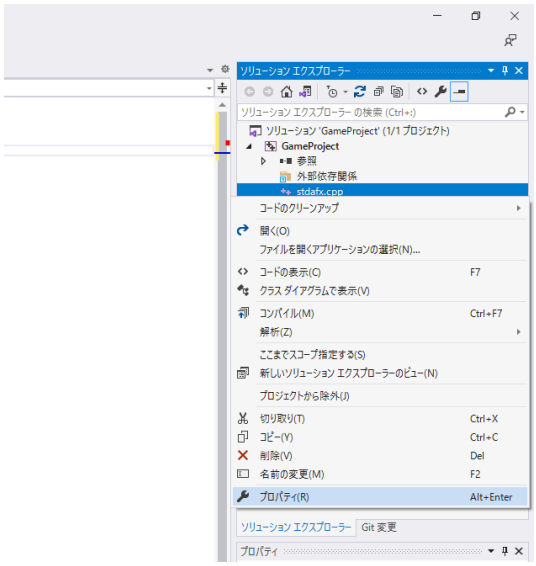
右側「ソリューションエクスプローラー」-「stdafx.h」をクリックし、作業ディレクトリに下記コードを入力します。

```
/**
 stdafx.h
*/
#pragma once
#ifndef _PA_STDAFX_H_
#define _PA_STDAFX_H_
#include <chidori/chidori.h>
using namespace chidori;
#endif // !_PA_STDAFX_H_
```

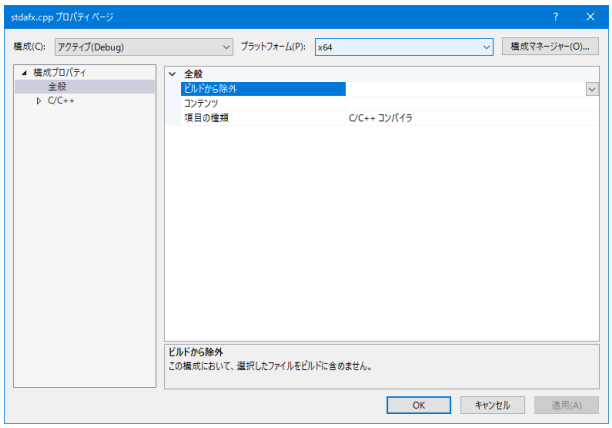


右側「ソリューションエクスプローラー」-「stdafx.cpp」をクリックし、作業ディレクトリに下記コードを入力します。

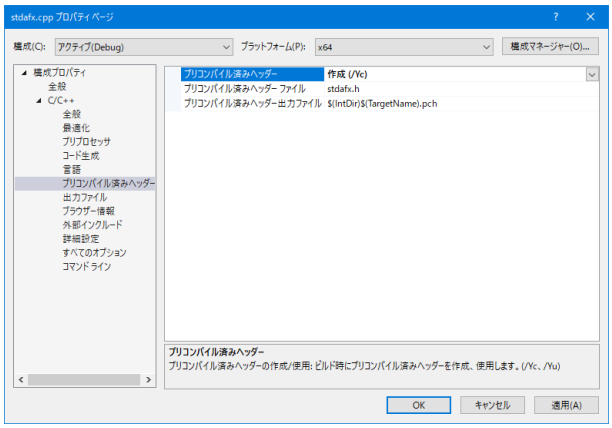
```
/**
 stdafx.cpp
*/
#include "stdafx.h"
```



右側「ソリューションエクスプローラー」-「stdafx.cpp」をクリックし、右クリックで「プロパティ(R)」を選択してプロパティページを開きます。

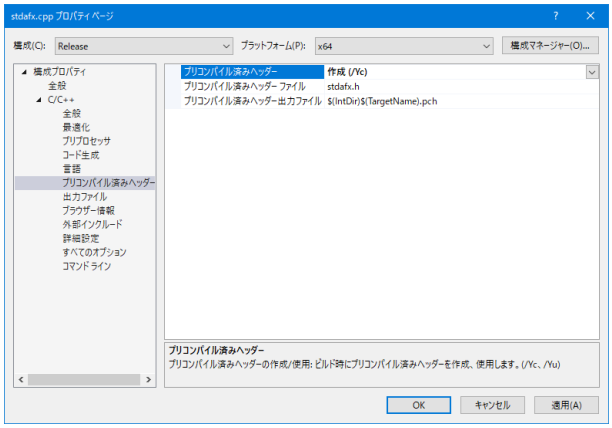


プラットフォーム
x64



「C/C++」-「プリコンパイル済みヘッダー」

プリコンパイル済みヘッダー
作成 (/Yc)



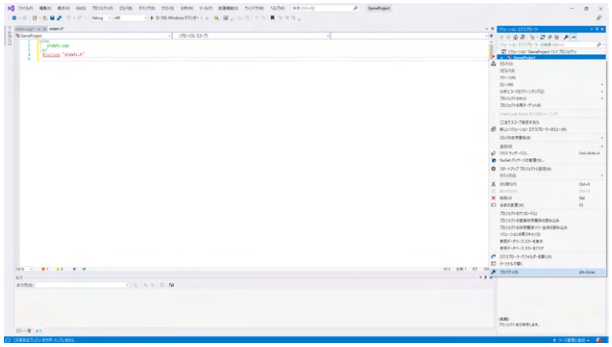
「構成(C):」を「Release」に変えて、同じく下記設定を行います。

「C/C++」-「プリコンパイル済みヘッダー」

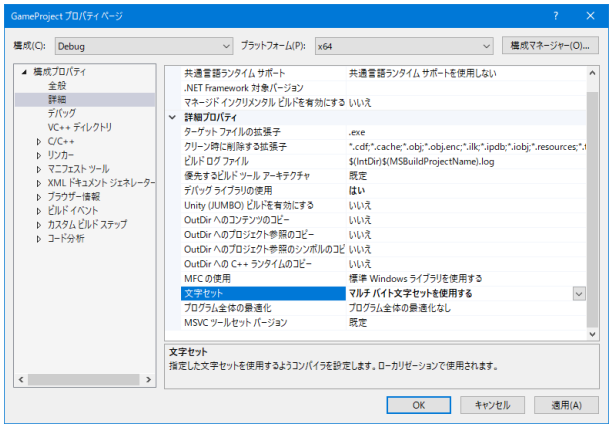
プリコンパイル済みヘッダー
作成 (/Yc)

設定が完了したら「構成(C):」を「Debug」に戻し、「OK」ボタンをクリックして画面を閉じます。

3−2　プロジェクトの設定（構成：Debug、プラットフォーム：x64）

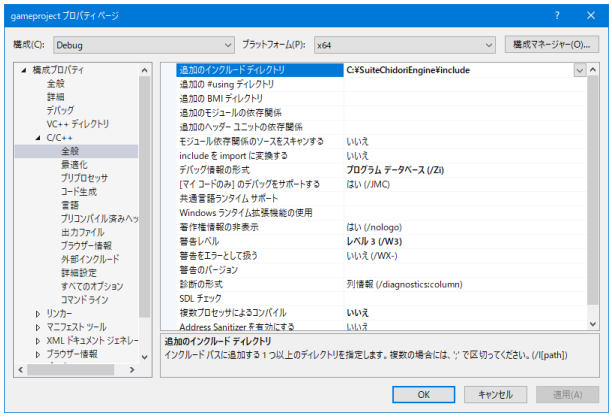


「構成:Debug」「プラットフォーム:x64」になっている事を確認し、右側「ソリューションエクスプローラー」-プロジェクト名(GameProject)を選択し、右クリックで「プロパティ(R)」をクリックします。



「構成プロパティ」-「詳細」

文字セット
マルチ バイト文字セットを使用する



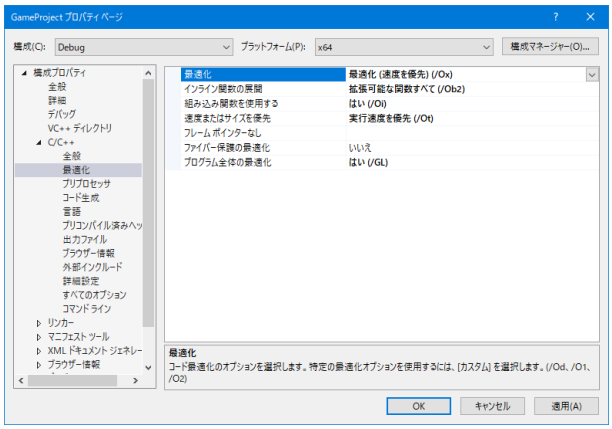
「C/C++」-「全般」

追加のインクルード ディレクトリ
C:¥SuiteChidoriEngine¥include

デバッグ情報の形式
プログラム データベース (/Zi)

SDL チェック
空白

複数プロセッサによるコンパイル
いいえ



「C/C++」-「最適化」

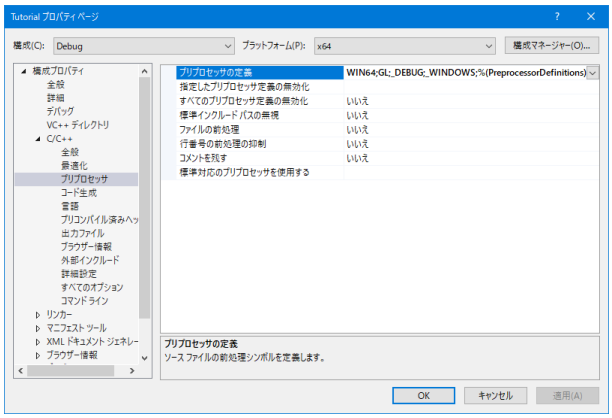
最適化
最適化 (速度を優先) (/Ox)

インライン関数の展開
拡張可能な関数すべて (/Ob2)

組み込み関数を使用する
はい (/Oi)

速度またはサイズを優先
実行速度を優先 (/Ot)

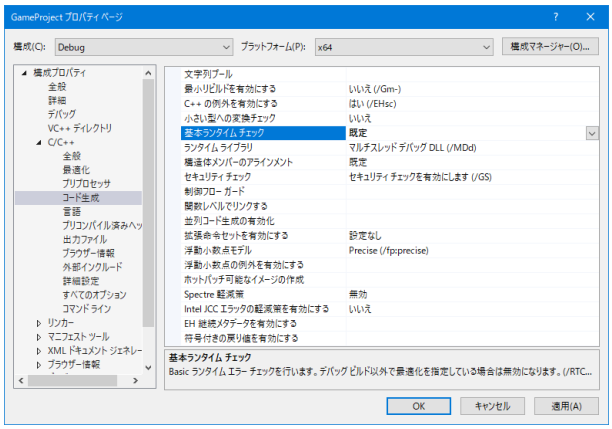
プログラム全体の最適化
はい (/GL)



「C/C++」-「プリプロセッサ」

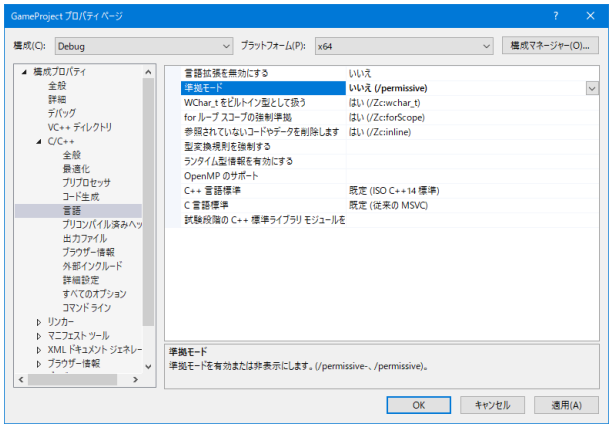
プリプロセッサの定義
※各シンボルはセミコロンで区切って入力してください。

WIN64;GL;_DEBUG;_WINDOWS;%(PreprocessorDefinitions)



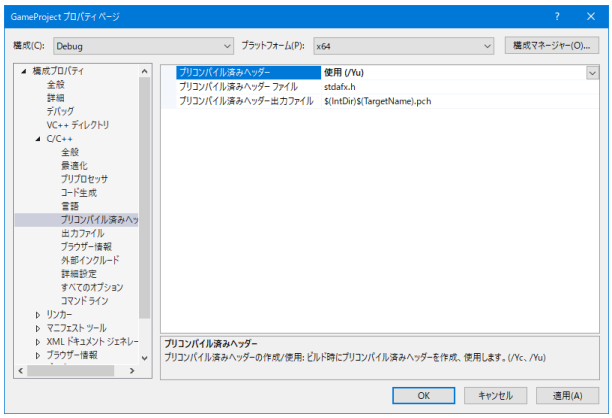
「C/C++」-「コード生成」

基本ランタイム チェック
既定



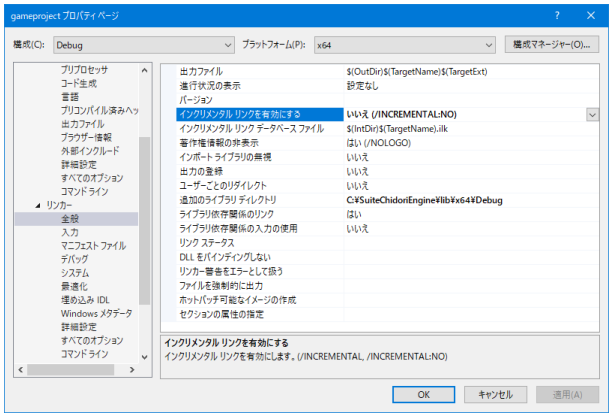
「C/C++」-「言語」

準拠モード
いいえ (/permissive)



「C/C++」-「プリコンパイル済みヘッダー」

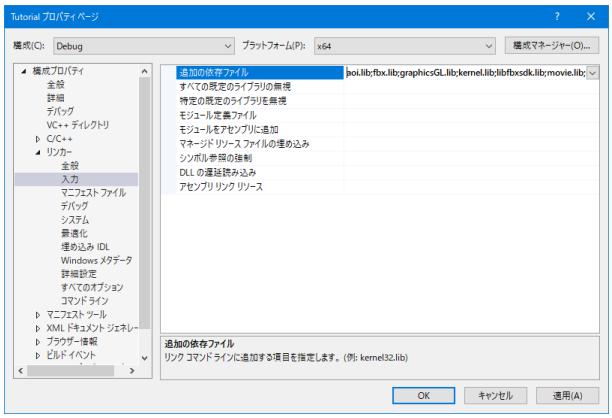
プリコンパイル済みヘッダー
使用 (/Yu)



「リンカー」-「全般」

インクリメンタル リンクを有効にする
いいえ (/INCREMENTAL:NO)

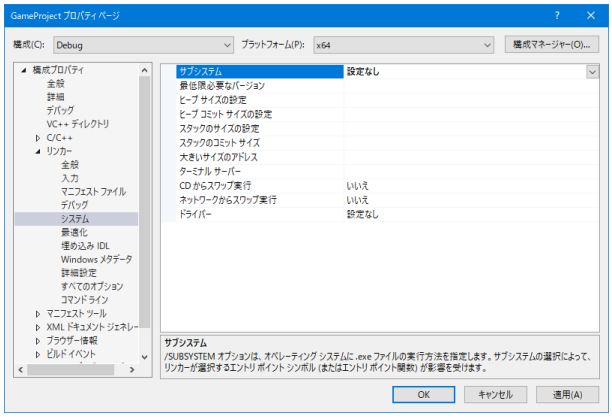
追加のライブラリ ディレクトリ
C:¥SuiteChidoriEngine¥lib¥x64¥Debug



「リンカー」-「入力」

追加の依存ファイル
※各シンボルはセミコロンで区切って入力してください。

aoi.lib;fbx.lib;graphicsGL.lib;kernel.lib;libfbxsdk.lib;movie.lib;net.lib;peripheral.lib;pet.lib;sound.lib;zlib.lib;%(AdditionalDependencies)

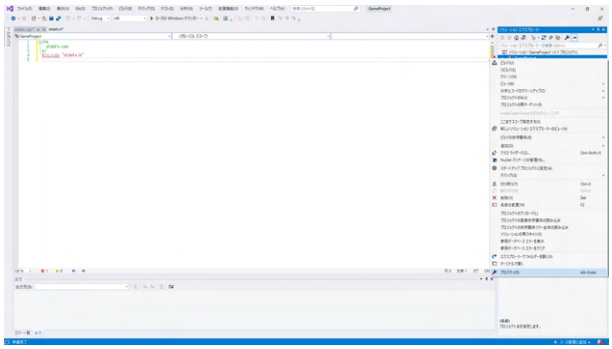


「リンカー」-「システム」

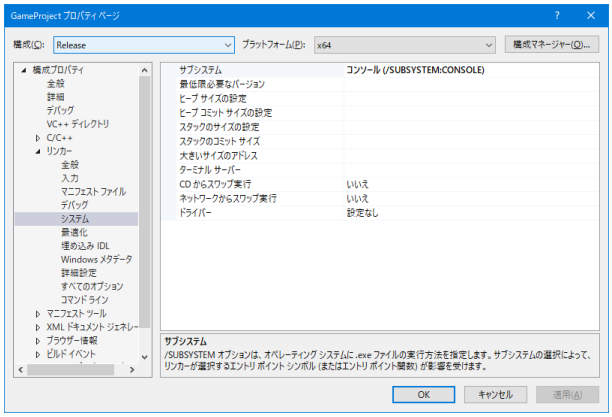
サブシステム
設定なし

設定が完了したら「OK」ボタンをクリックして、画面を閉じます。

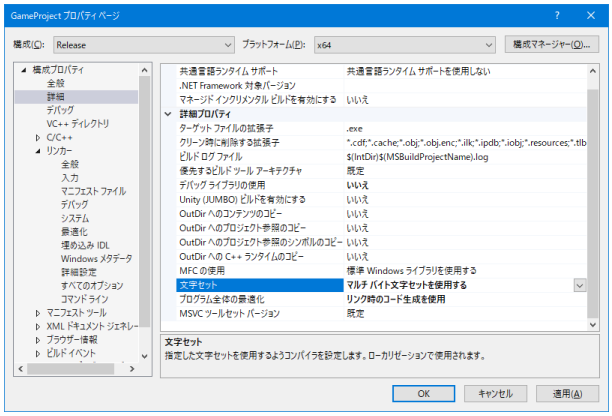
3−3　プロジェクトの設定（構成：Release、プラットフォーム：x64）



「構成:Release」「プラットフォーム:x64」になっている事を確認し、右側「ソリューションエクスプローラー」-プロジェクト名(GameProject)を選択し、右クリックで「プロパティ(R)」をクリックします。

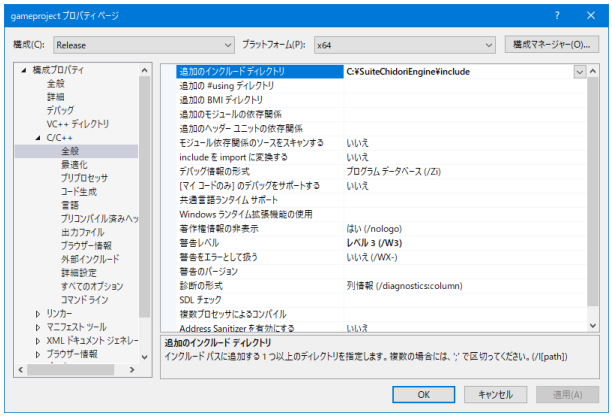


構成(C)
Release



「構成プロパティ」-「詳細」

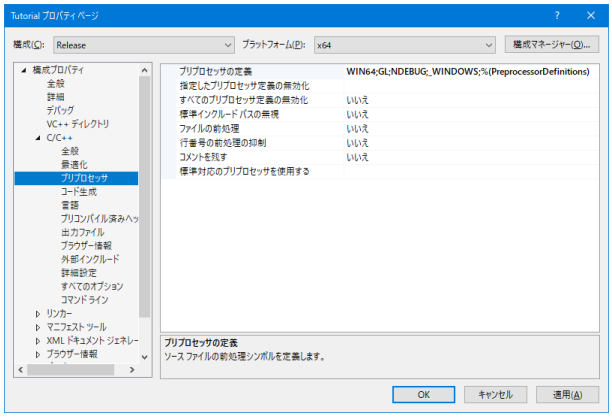
文字セット
マルチ バイト文字セットを使用する



「C/C++」-「全般」

追加のインクルード ディレクトリ
C:¥SuiteChidoriEngine¥include

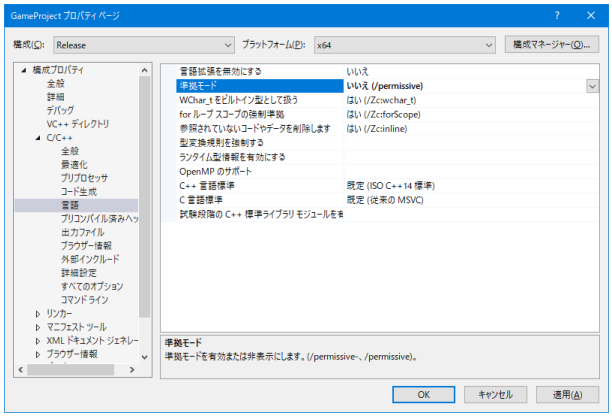
SDL チェック
空白



「C/C++」-「プリプロセッサ」

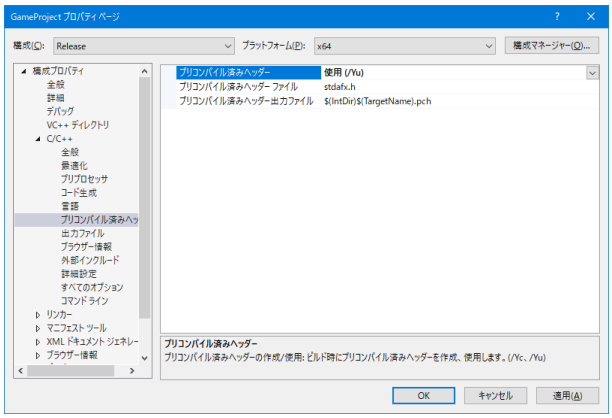
プリプロセッサの定義
※各シンボルはセミコロンで区切って入力してください。

WIN64;GL;NDEBUG;_WINDOWS;%(PreprocessorDefinitions)



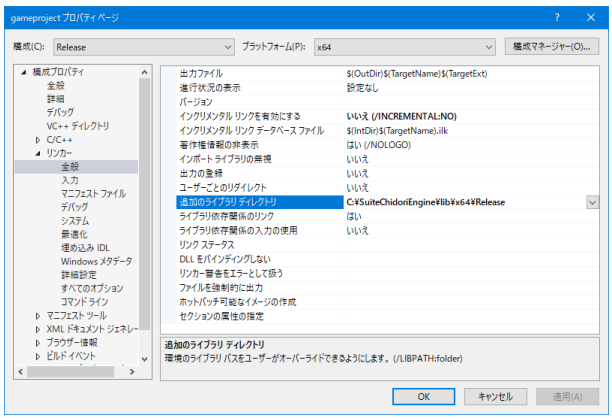
「C/C++」-「言語」

準拠モード
いいえ (/permissive)



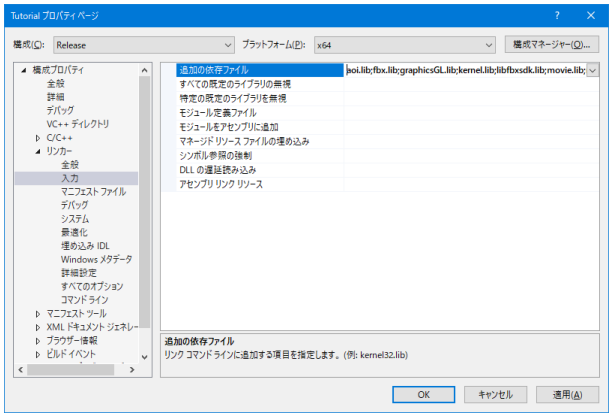
「C/C++」-「プリコンパイル済みヘッダー」

プリコンパイル済みヘッダー
使用 (/Yu)



「リンカー」-「全般」

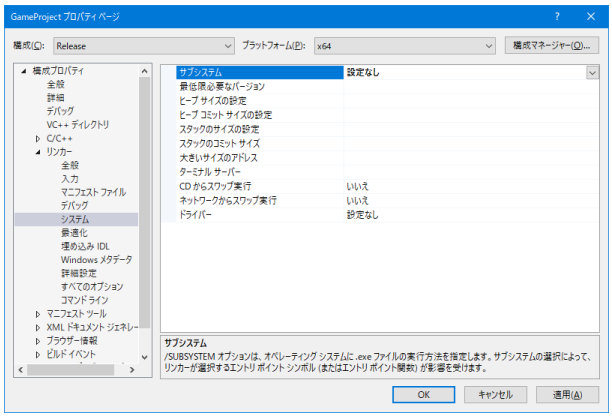
追加のライブラリ ディレクトリ
C:¥SuiteChidoriEngine¥lib¥x64¥Release



「リンカー」-「入力」

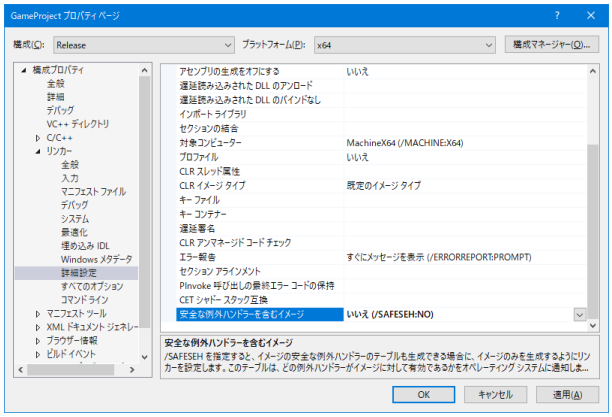
追加の依存ファイル
※各シンボルはセミコロンで区切って入力してください。

aoi.lib;fbx.lib;graphicsGL.lib;kernel.lib;libfbxsdk.lib;movie.lib;net.lib;peripheral.lib;pet.lib;sound.lib;zlib.lib;%(AdditionalDependencies)



「リンカー」-「システム」

サブシステム
設定なし

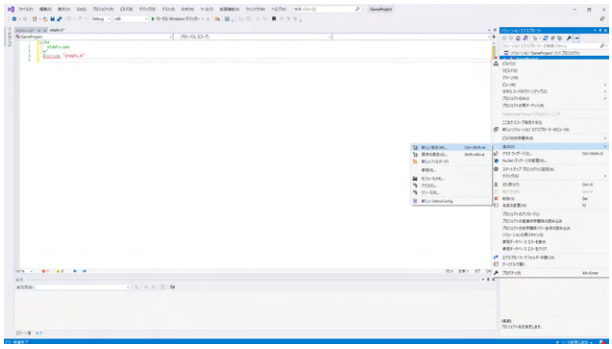


「リンカー」-「詳細設定」

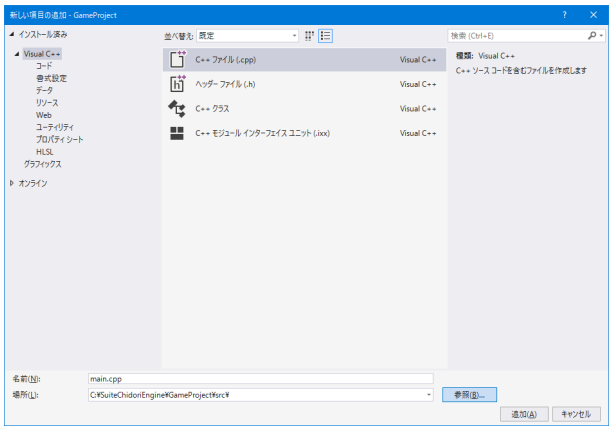
安全な例外ハンドラーを含むイメージ
いいえ (/SAFESEH:NO)

設定が完了したら「OK」ボタンをクリックして、画面を閉じます。

3−4　プロジェクトのビルド



右側「ソリューションエクスプローラー」-プロジェクト名(GameProject)を選択し、右クリックで「追加(D)」-「新しい項目(W)」をクリックします。



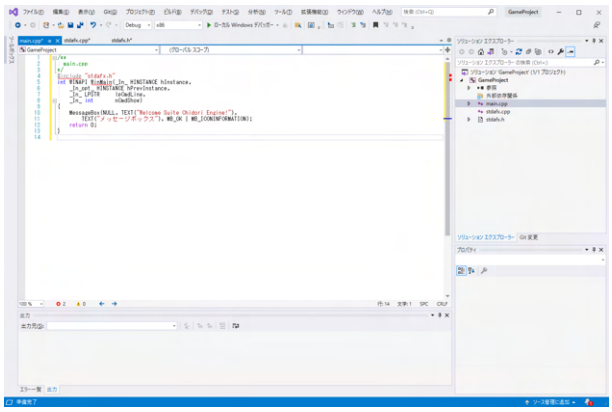
「新しい項目の追加」画面が開くので、設定を行います。

ファイル
C++ ファイル (.cpp)

名前
main.cpp

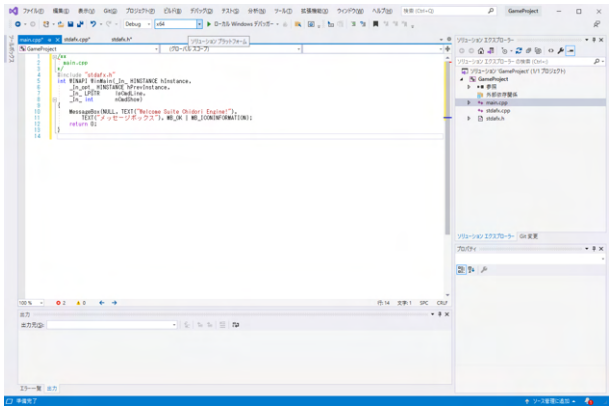
場所
C:¥SuiteChidoriEngine¥GameProject¥src¥
(先ほどプロジェクト下に作った「src」フォルダを選択します)

「追加(A)」をクリックします。

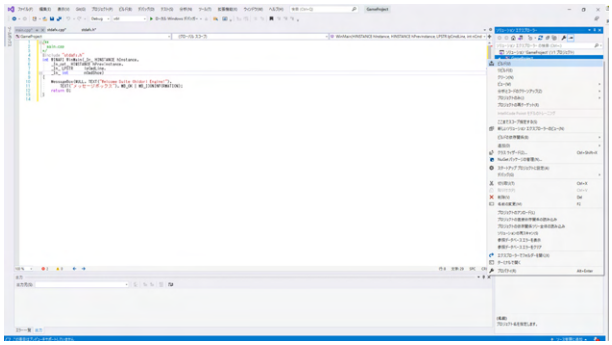


右側「ソリューションエクスプローラー」-「main.cpp」をクリックし、作業ディレクトリに下記コードを入力します。

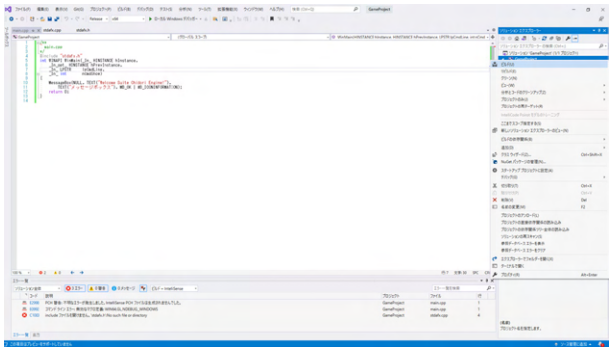
```
/**
  main.cpp
*/
#include "stdafx.h"
int WINAPI WinMain(_In_ HINSTANCE hInstance,
                   _In_opt_ HINSTANCE hPrevInstance,
                   _In_ LPSTR     lpCmdLine,
                   _In_ int       nCmdShow)
{
    MessageBox(NULL, TEXT("Welcome Suite Chidori Engine!"),
               TEXT("メッセージボックス"), MB_OK | MB_ICONINFORMATION);
    return 0;
}
```



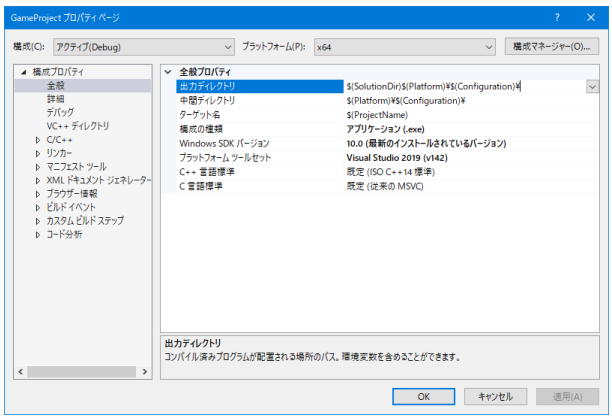
プラットフォーム
x64



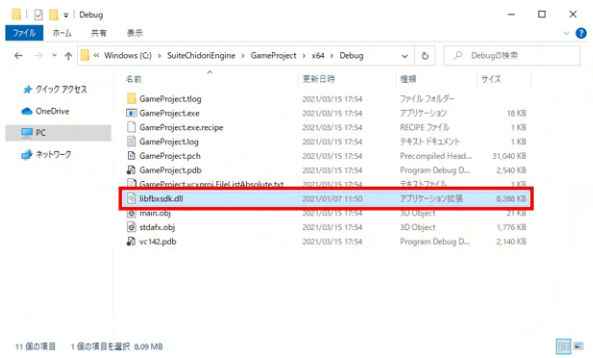
右側「ソリューションエクスプローラー」-プロジェクト名(GameProject)を選択し、右クリックして「ビルド(U)」をクリックします。



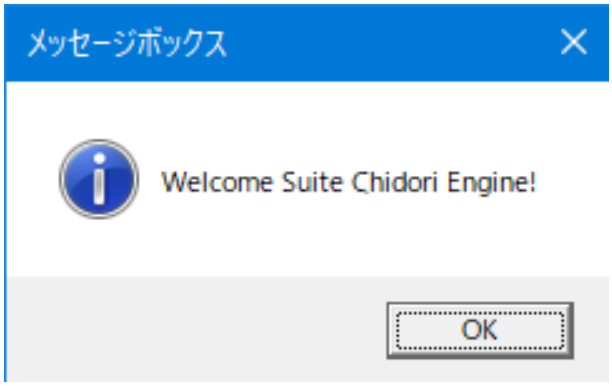
ソリューション構成を「Release」に切り替え、同じくプロジェクト(GameProject)を選択し、右クリックして「ビルド(U)」をクリックします。



ビルドが成功すると、「プロパティ(R)」-「構成プロパティ」-「全般」-「出力ディレクトリ」が示すフォルダに.exeファイルが生成されます。



生成された.exeファイルと同じディレクトリに「libfbxsdk.dll」ファイルを追加します。
※Debug・Releaseどちらにも追加してください
※libfbxsdk.dllは、(インストール先のフォルダ)\lib\dll\x64の中にあります。

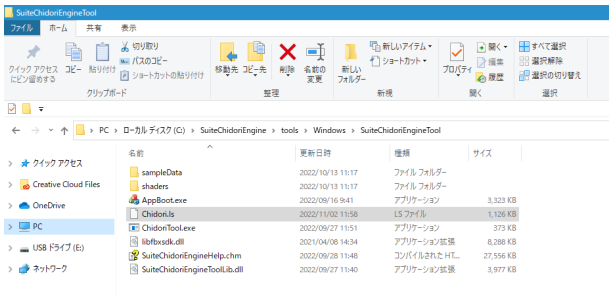


「Visual Stusdio 2019」画面で「F5」を押して実行し、「Welcome Suite Chidori Engine!」のダイアログが表示されれば成功です。

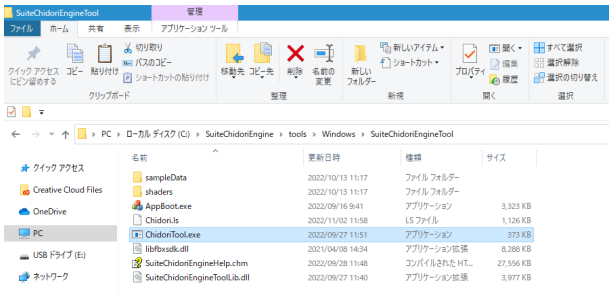
第3章　操作ガイダンス(スイート千鳥エンジンエディター)

1　事前準備

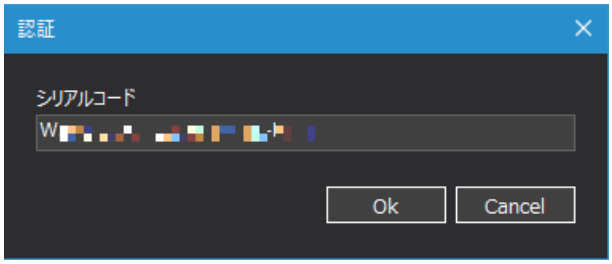
1−1　ライセンス認証



アクティベーションファイル(Chidori.ls)を「(インストール先)\tools\Windows\SuiteChidoriEngineTool」に設置します。



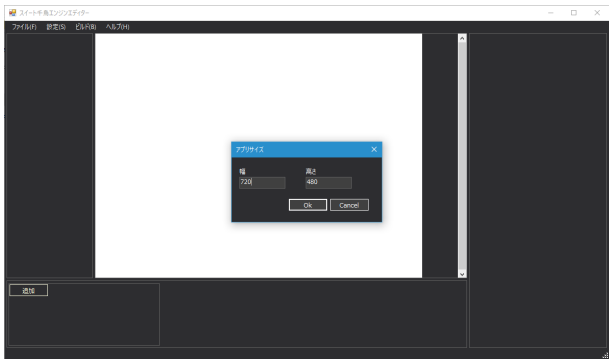
「ChidoriTool.exe」をダブルクリックします。



シリアルコードを入力するダイアログが表示されるので、入力して「OK」ボタンをクリックします。
シリアルコードはハイフンも含めて入力してください。
ライセンスが正しく認証されると、スイート千鳥エンジンエディターが表示されます。

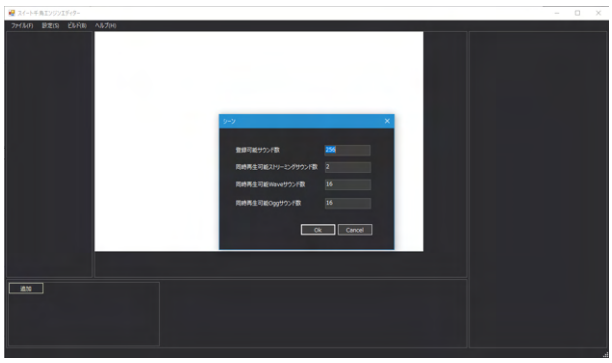
認証に失敗する場合は、アクティベーションファイル(Chidori.ls)が正しく設置されているかと、シリアルコードが正しく入力されているかを確認してください。

1−2　アプリサイズの設定

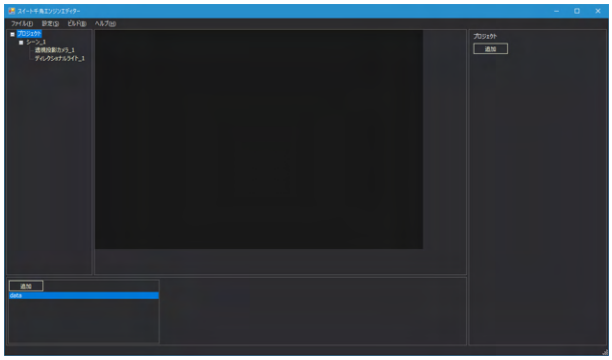


アプリサイズを変更する場合は、メニュー「設定」→「アプリサイズ」から、アプリサイズダイアログを表示して、サイズを指定します。
ここで入力したアプリサイズは、スイート千鳥エンジンエディターを再度起動するまでは反映されないので、ご注意ください。

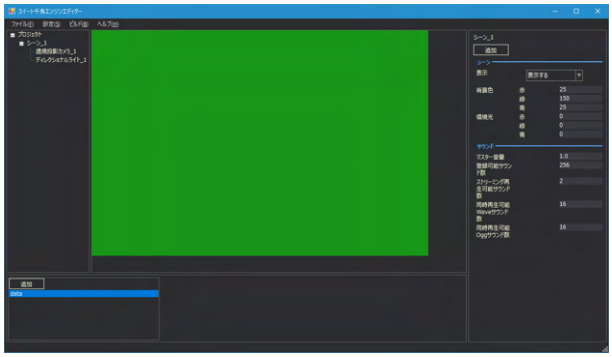
2　プロジェクト作成



アクティベーションファイル(Chidori.ls)を「(インストール先)\tools\Windows\SuiteChidoriEngineTool」に設置します。

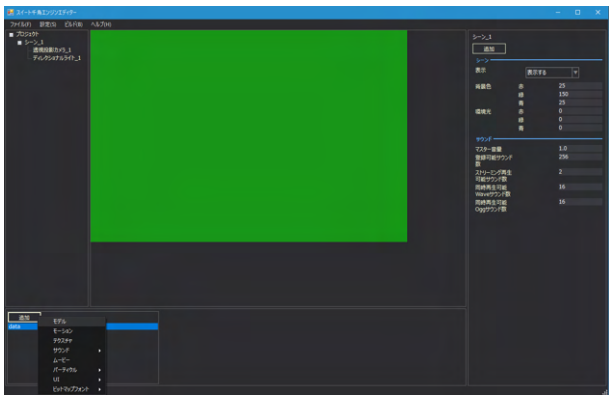


プロジェクトを作成すると、シーンと透視投影カメラ、ディレクショナルライト、アセットフォルダが1つずつ追加されます。
オブジェクトリストに表示されるプロジェクト、シーン、カメラ、ライトをクリックして選択することで、選択した要素のプロパティがプロパティリストに表示されます。



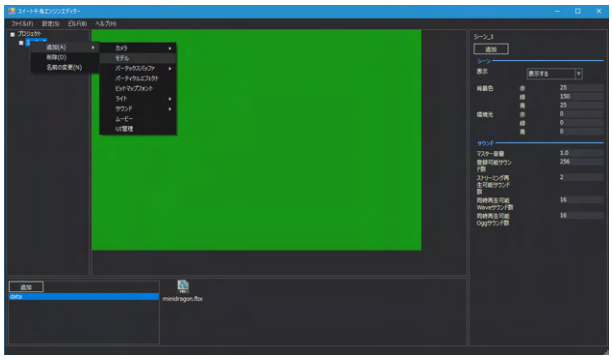
シーンの背景色を変更します。
オブジェクトリストから「シーン_1」をクリックして、プロパティリストにシーンのプロパティを表示します。
「背景色」の「緑」の値を「150」に変更して、アプリの背景色を緑色にします。

3　モデルの追加

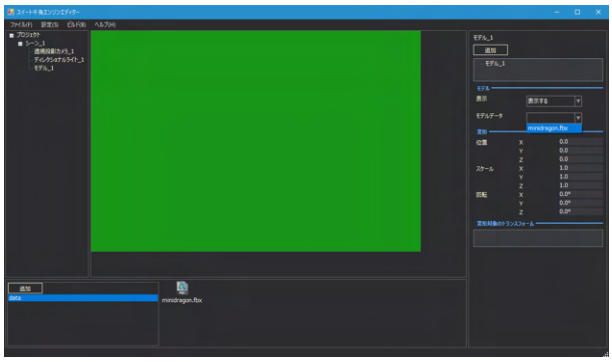


モデルを追加するために、まずモデルデータを追加します。
データリスト内の「data」をクリックして選択します。

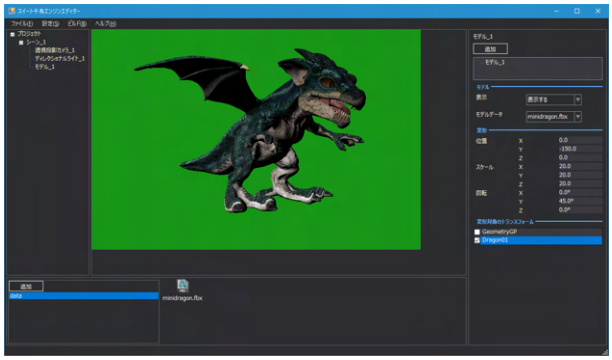
「追加」ボタンをクリックして表示されるリストから、
「モデル」を選択して、「(インストール先)\tools\Windows\SuiteChidoriEngineTool\sampleData\minidragon.fbx」を指定します。
指定したモデルデータがデータリストに表示されます。



モデルオブジェクトを追加します。
オブジェクトリストの「シーン_1」の右クリックメニューから「追加」→「モデル」を選択します。モデルオブジェクト「モデル_1」がシーンの子要素に追加されます。

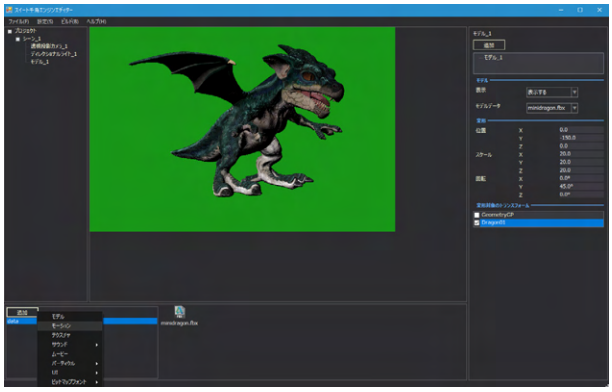


オブジェクトリストの「モデル_1」をクリックして、プロパティリストにモデルのプロパティを表示します。
「モデルデータ」のコンボボックスをクリックすると、追加したモデルデータ「minidragon.fbx」が表示されるのでクリックして選択します。これで選択したモデルがシーンビューに表示されます。

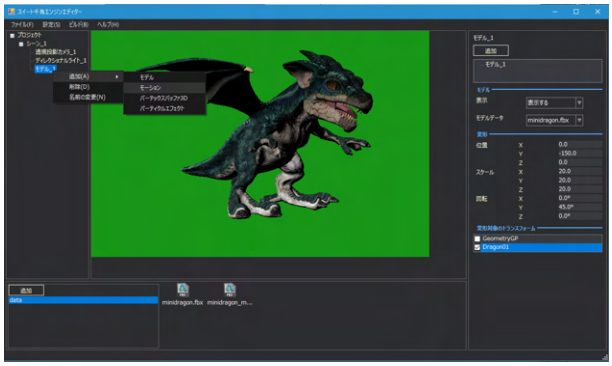


モデルの位置とサイズ、向きを調整します。
モデルの位置とサイズ、向きは、モデルが持つトランスフォームという部品に対して適用されますが、適用するトランスフォームが正しくないと、モデルが崩れたり、変更が適用されなかったりします。
今回読み込んだモデルでは、「Dragon01」というトランスフォームを使用するとよいので、「変形対象のトランスフォーム」内の「Dragon01」にチェックを入れます。 チェックを入れるとモデルの大きさが変化しますが、これはモデルに元々設定されていた値ではなく、モデルオブジェクトの初期値が反映されるためです。
「位置」の「Y」に「-150.0」を、「スケール」の「X」「Y」「Z」それぞれに「20.0」を、「回転」の「Y」に「45.0」を入力して、モデルの位置とサイズを調整し、右方向に向かせます。

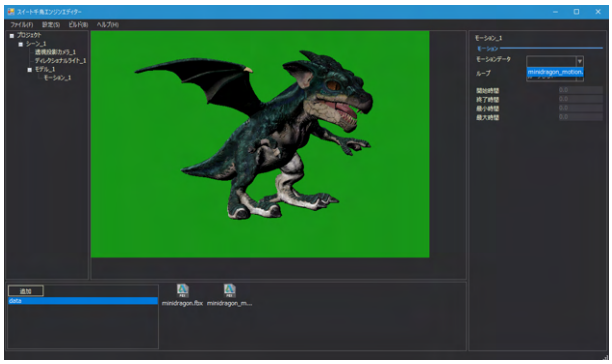
4　モーションの設定



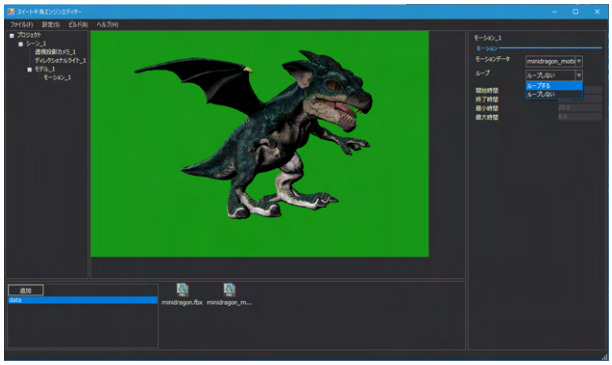
モーションを設定するために、まずモーションデータを追加します。
データリスト内の「data」をクリックして選択します。
「追加」ボタンをクリックして表示されるリストから「モーション」を選択して、「(インストール先)\tools\Windows\SuiteChidoriEngineTool\sampleData\minidragon_motion.fbx」を指定します。
指定したモーションデータがデータリストに表示されます。



モーションオブジェクトを追加します。
オブジェクトリストの「モデル_1」の右クリックメニューから「追加」→「モーション」を選択します。モーションオブジェクト「モーション_1」がモデルの子要素に追加されます。

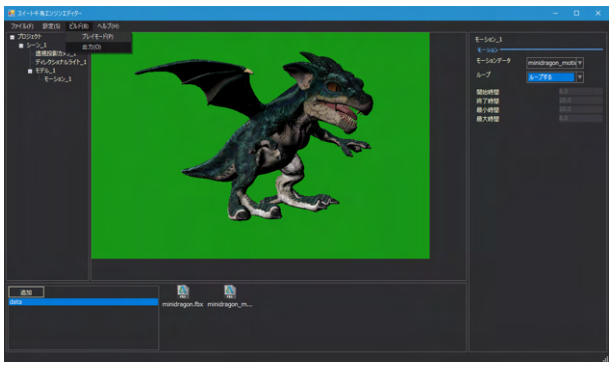


オブジェクトリストの「モーション_1」をクリックして、プロパティリストにモーションのプロパティを表示します。
「モーションデータ」のコンボボックスをクリックすると、追加したモーションデータ「minidragon_motion.fbx」が表示されるのでクリックして選択します。
なお、モーションはエディター上では再生されず、プレイモードという画面で再生されます。そのため、この時点ではまだモデルのモーションは再生されません。

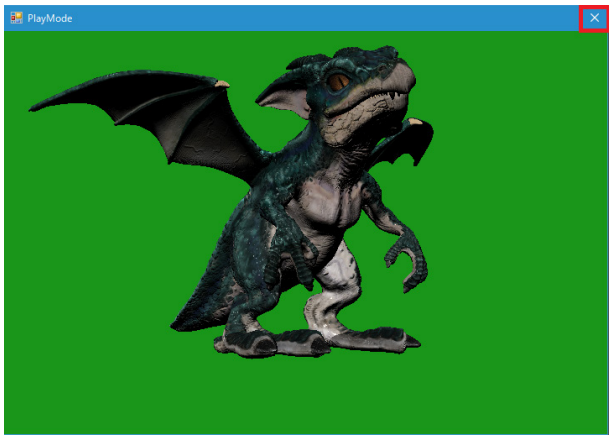


モーションをループ再生するようにします。
「ループ」のコンボボックスをクリックして「ループする」に変更します。

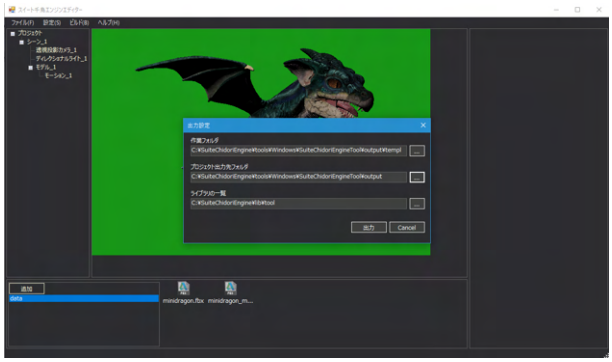
5　アプリ作成



アプリを作成する前に、今回作成したシーンの表示を確認します。
メニュー「ビルド」→「プレイモード」からプレイモード画面を表示します。
プレイモード画面では、エディター上では再生されないモデルのアニメーションやパーティクルエフェクト、サウンド、ムービーの再生を確認することができます。



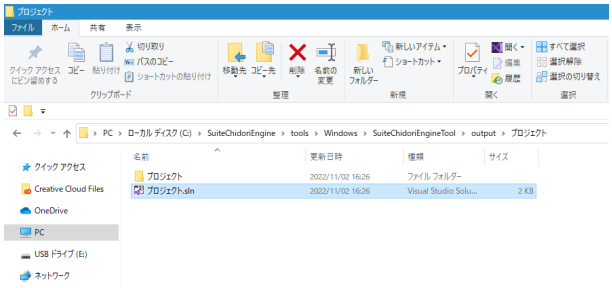
プレイモード画面では、マウス操作による視点変更が可能です。
プレイモードを終了するには、プレイモード画面をばつボタンで閉じます。



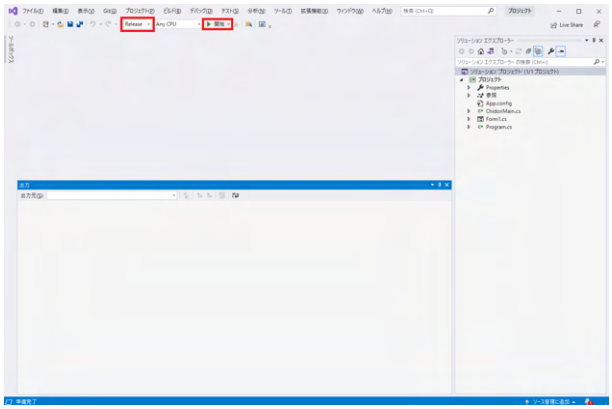
プレイモードで表示を確認したら、アプリを作成するためにVisual Studioのソリューションファイルを出力します。
メニュー「ビルド」→「出力」から出力設定ダイアログを表示します。
ダイアログで各設定を入力します。
「作業フォルダ」はソリューションファイルの作成時に一時的に使用するフォルダです。
ここでは例として「(インストール先)\tools\Windows\SuiteChidoriEngineTool\output\temp」を指定します(インストール先がデフォルトフォルダの場合は「C:\SuiteChidoriEngine\tools\Windows\SuiteChidoriEngineTool\output\temp」)。

「プロジェクト出力先フォルダ」にはソリューションファイル一式を出力するフォルダを指定します。
ここでは例として「(インストール先)\tools\Windows\SuiteChidoriEngineTool\output」を指定します(インストール先がデフォルトフォルダの場合は「C:\SuiteChidoriEngine\tools\Windows\SuiteChidoriEngineTool\output」)。

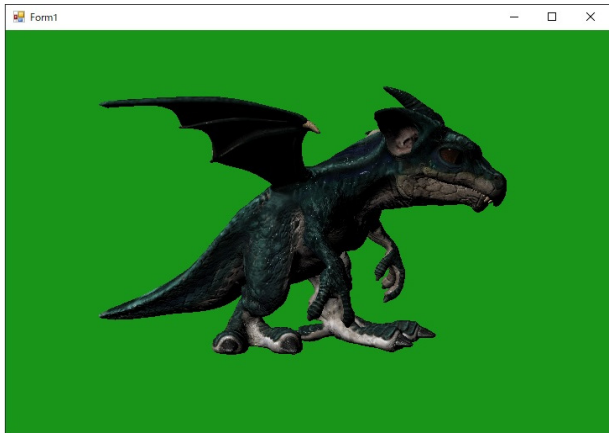
「ライブラリの一覧」にはアプリで使用するライブラリが設置されたフォルダを指定します。
「(インストール先)\lib\tool」を指定してください(インストール先がデフォルトフォルダの場合は「C:\SuiteChidoriEngine\lib\tool」)。

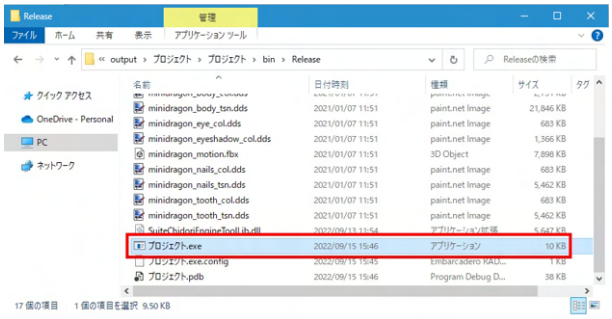


「出力」ボタンをクリックすると、「プロジェクト出力先フォルダ」にフォルダが生成され、ソリューションファイル一式が出力されます。
生成されたフォルダ内にプロジェクト名のフォルダ (ここでは「プロジェクト」フォルダ)があり、その中にプロジェクト名のソリューションファイル(ここでは「プロジェクト.sln」)があります。



ソリューションファイルをダブルクリックします。
Visual Studioが起動したら、「ソリューション構成」を「Release」にして「開始」をクリックします。
ウインドウが起動し、作成したアプリが表示されます。





アプリの実行ファイルは、「(プロジェクト出力先フォルダ)/(プロジェクト名フォルダ)/(プロジェクト名フォルダ)/bin/Release」フォルダに出力されています。
アプリを配布する場合は、フォルダ内のファイルを全て含めてください。

第4章　Q&A

1　スイート千鳥エンジンについて

Q1-1　スイート千鳥エンジンとは何ですか?

A1-1　スイート千鳥エンジンは、プログラミングソフトと組み合わせて使用するライブラリ集(ソースコード集)となります。

Q1-2　スイート千鳥エンジンでは、どのプラットフォーム向けのゲームが作成できますか?

A1-2　WindowsとiOSになります。なお、インストーラはWindows用ゲームを作成するためにWindowsで使用できるWindows版と、iOS用ゲームを作成するためにmacOSで使用できるiOS版とがあります。

Q1-3　スイート千鳥エンジンで使用するプログラミング言語は何ですか?

A1-3　スイート千鳥エンジンで使用するプログラミング言語はC++です。

2　導入について

Q2-1　スイート千鳥エンジンのインストーラについて教えてください。

A2-1　Windows版とiOS版の両方のバージョンのインストーラを提供しています。使用するプラットフォームごとに、下記の通りインストールしてください。

【Windows版】
Windowsで、「setup_スイート千鳥エンジン_Windows.exe」または「スイート千鳥エンジン_Windows.msi」を起動してインストールしてください。

【iOS版】
macOSで、「SuiteChidoriEngine_iOS.pkg」を起動してインストールしてください。

警告メッセージが表示され起動できない場合は、「SuiteChidoriEngine_iOS.pkg」を選択し、右クリックメニューの「開く」から起動すると、 表示される警告メッセージ内に「開く」ボタンが表示されますので、そのボタンから起動してください。

Q2-2　インストール後、まず何をすればよいですか?

A2-2　スイート千鳥エンジンをインストールした場所にSuiteChidoriEngineというフォルダが作成されます。
デフォルト設定では、
Windows版は「C:\SuiteChidoriEngine」
iOS版は「/Application/FORUM8/SuiteChidoriEngine」
スイート千鳥エンジンヘルプの［チュートリアル］を参考にして、スイート千鳥エンジンの使い方を確認してください。ライセンス認証の手順についてもここで学ぶことができます。

Q2-3　必要なプログラミングソフトは何ですか?

A2-3　使用するプラットフォームごとに、下記の通りとなります。

【Windows版】
Windows版はVisual Studio 2019を使用します。下記のマイクロソフトのダウンロードサイトより入手し、インストールしてください。
　https://docs.microsoft.com/ja-jp/visualstudio/releases/2019/release-notes
インストール時には、「C++ によるデスクトップ開発」にチェックを入れてください。
なお、Visual Studio 2019の全てのエディションにて動作しますが、使用にあたっては該当するエディションのライセンス条項をご確認下さい。また、個人利用の場合はコミュニティ版が無償で使用できます。コミュニティ版は上記URL内の「Download Commuity 2019」ボタンからダウンロードできます。

【iOS版】
iOS版はXcodeを使用します。App StoreよりXcodeをインストールしてください

Q2-4　ヘルプファイルはどこで確認できますか?

A2-4　スイート千鳥エンジンヘルプは、以下より確認できます。

【Windows版】
Windowsメニュー→［FORUM8］→［スイート千鳥エンジン ヘルプ］

【iOS版】
［アプリケーション］→［FORUM8］フォルダ→［SuiteChidoriEngine］フォルダ→［document］フォルダ→［iOS］フォルダ→［SuiteChidoriEngineHelp］フォルダ→［index.html］

Q2-5　サンプルデータはありますか?

A2-5　サンプルプロジェクト、および、スイート千鳥エンジンで作成したゲームを用意しております。

【Windows版】
・チュートリアル
モデルの表示とアニメーション、サウンド機能を確認できるサンプルです。スイート千鳥エンジンヘルプの［チュートリアル］を確認してください。
・ParticleSample
パーティクルエフェクト機能を確認できるサンプルです。スイート千鳥エンジンヘルプの［サンプル｜ParticleSampleの使用方法］を確認してください。
・PostEffectSample
ポストエフェクト機能を確認できるサンプルです。スイート千鳥エンジンヘルプの［サンプル｜PostEffectSampleの使用方法］を確認してください。
・ゲーム
スイート千鳥エンジンで作成したゲームです。スイート千鳥エンジンヘルプの［ゲーム］を確認してください。

【iOS版】
・チュートリアル
モデルの表示とアニメーション、サウンド機能を確認できるサンプルです。スイート千鳥エンジンヘルプの［チュートリアル］を確認してください。

3　ライセンス認証について

Q3-1　発行されたアクティベーションファイル(Chidori.ls)とシリアルコードは、どのように使用しますか?

A3-1　アクティベーションファイルとシリアルコードは、ライセンス認証に使用します。 使用方法は、スイート千鳥エンジンヘルプの［概要｜ライセンス認証の手順］および［チュートリアル｜Tutorialの使用方法］(Windows版)、［チュートリアル｜チュートリアルの使用方法］(iOS版)を確認してください。 ライセンスについては、［概要｜ライセンスについて］を確認してください。
なお、ライセンス認証を行わない場合は、正しくゲーム画面が表示されず、下記のようになります。

【Windows版】
ゲーム画面にスイート千鳥エンジンのロゴが常時表示されます。

【iOS版】
ゲーム画面が黒塗りとなり何も表示されません。

Q3-2　認証用の関数を呼んでいますが、認証が失敗してしまいます。

A3-2　グラフィックスの初期化より前に認証用の関数を呼んだ場合は、認証に失敗します。グラフィックスの初期化より後に認証用の関数を呼んでいるかを確認してください。なお、グラフィックスの初期化については、スイート千鳥エンジンヘルプの［チュートリアル｜5. フレームワーク処理の詳細説明］の5.2.3.3(Windows版)、［チュートリアル｜チュートリアルの構成］の2.2.5(iOS版)を確認してください。

4　スイート千鳥エンジンの機能について

Q4-1　**スイート千鳥エンジンにはどのような機能がありますか?**

A4-1　スイート千鳥エンジンにはグラフィック機能やファイルの入出力機能、タスク制御機能などがあります。主要な機能やそれぞれの使用方法については、スイート千鳥エンジンヘルプの［機能説明］を確認してください。

Q4-2　**スイート千鳥エンジンにはどのようなクラスや関数がありますか?**

A4-2　スイート千鳥エンジンで使用できるクラス、関数の一覧を記述したリファレンスマニュアルを用意しています。リファレンスマニュアルは以下より確認できます。

【Windows版】
Windowsメニュー→［FORUM8］→［スイート千鳥エンジン リファレンスマニュアル］

【iOS版】
［アプリケーション］→［FORUM8］フォルダ→［SuiteChidoriEngine］フォルダ→［document］フォルダ→［iOS］フォルダ→［SuiteChidoriEngineReference］フォルダ→［index.html］

5　スイート千鳥エンジンが対応しているファイル形式について

Q5-1　**スイート千鳥エンジンが対応しているモデルデータのファイル形式は何ですか?**

A5-1　スイート千鳥エンジンが対応しているモデルデータのファイル形式はFBX形式です。テクスチャは、Windows版ではDDS形式を、iOS版ではPVR形式をサポートしています。

Q5-2　**スイート千鳥エンジンが対応しているアニメーションデータのファイル形式は何ですか?**

A5-2　スイート千鳥エンジンが対応しているアニメーションデータのファイル形式はFBX形式です。

Q5-3　**スイート千鳥エンジンが対応しているサウンドデータのファイル形式は何ですか?**

A5-3　スイート千鳥エンジンが対応しているサウンドデータのファイル形式はWave形式とOgg形式です。

Q5-4　**スイート千鳥エンジンが対応しているムービーデータのファイル形式は何ですか?**

A5-4　スイート千鳥エンジンのWindows版が対応しているムービーデータのファイル形式はAVI形式で、映像はMotion JPEG、音声はPCMに対応しています。また、任意のムービーデータをこの形式に変換できる動画変換ツール「MJPEGコンバーター」を用意しています。MJPEGコンバーターについては、スイート千鳥エンジンヘルプの［ツールの使用方法｜動画変換ツールの使用方法］を参照してください。

6　その他

Q6-1　**今まで非商用で使っていましたが作ったゲームを販売したくなりました。どうすればよいでしょうか?**

A6-1　商用利用となる場合は、有償となります。大変お手数ですが、弊社営業までご連絡をお願いいたします。

Q6-2　**スイート千鳥エンジンで作ったゲームを販売した際、売り上げに応じてお金を支払う必要はあるのでしょうか?**

A6-2　いいえ、売り上げに応じた支払は必要ありません。

Q6-3　**スイート千鳥エンジンで作ったゲームを販売している間、常にスイート千鳥エンジンのサブスクリプションを継続する必要はあるのでしょうか?**

A6-3　いいえ、 サブスクリプションの継続はゲームの開発中のみでよいです。ただ、ゲームのアップデートや修正の予定がある場合は、スイート千鳥エンジンの最新版をご利用いただけるよう継続いただくことをお勧めいたします。

Q&Aはホームページ(https://www.forum8.co.jp/faq/win/chidori-qa.htm)にも記載しております。

スイート千鳥エンジン Ver.2 操作ガイダンス

2022年 11月　第1版

発行元　株式会社フォーラムエイト
〒108-6021　東京都港区港南2-15-1 品川インターシティA棟21F
TEL 03-6894-1888

お問い合わせについて

本製品及び本書について、ご不明な点がございましたら、弊社、「サポート窓口」へお問い合わせ下さい。
なお、ホームページでは、Q＆Aを掲載しております。こちらもご利用下さい。

ホームページ　www.forum8.co.jp
サポート窓口　ic@forum8.co.jp
FAX　0985-55-3027

スイート千鳥エンジン Ver.2

操作ガイダンス

www.forum8.co.jp

FORUM8
フォーラムエイト®