Teller Engine ドキュメント

たき

最終更新日 2022 年 6 月 3 日

はじめに

みなさんこんにちは、たきです。「たき」や「たきれん」など、複数名前がありますがたきです。このドキュメントは TellerEngine を改造したいだとか、ちょっと参考にしたいだとかを考えている人に向けた文書です。本書ではどのように設計したのか使用しているライブラリの話だとかをまとめたいと思います。説明不足の点もあると思いますが気付き次第改善する所存でありますので、どうかよろしくお願い致します。

目次

1　TellerEngine ことはじめ

1.1　外部ライブラリ

では記念すべき本文を書いていきましょう。まずはじめに使用している外部ライブラリのお話をします。以下のリストが使用している外部ライブラリです。

- Cinder 描画ライブラリ。他にもいろいろな機能があるが描画と画像のロードを担当している。imgui-node-editor を組み込んだカスタムビルド。
- imgui-node-editor imgui でノードエディタを作れる拡張。Cinder ライブラリのビルド時に一緒に入れた。
- nlohmann_json json 用パーサー。速くて有名な実績のあるパーサー。

他にもライブラリが見えていると思いますが使いません。

1.2　必要なもの

基本的に C++ で書いていて今のところ windows でしか動きません、おそらく。CMakeLists も書きかけのためまだ Linux、Mac の対応は先でしょう。というわけでまず windows マシンです。それ以外は知りません。筆者の環境をとりあえず載せます。

- Windows 10 21H2 build 19044.1706
- Visual Studio 2022 Community
- Git

以上があればビルドは通せるはずです。

1.3　はじめの一歩

まず Cinder からビルドを通しましょう。Github からクローンした場合既に imgui-node-editor もプロジェクトに入っているはずなので/Cinder/build/cinder.sln を開いて、Debug と Release のそれぞれでビルドしましょう。ビルドが終わったら/TellerEngin/TellerEngine.sln を開いてください。ビルドが通ったら準備完了です。

2 TellerEngin について

2.1 設計

TellerEngine は以下の基本的なクラスで構成されています。

- TellerCore
- Editor 及びその派生クラス
- ModuleCore 及びその派生クラス

2.2 TellerCore

TellerCore は上記のクラス、及びそれ以外の Animation クラスのなどを管理する中枢となるクラスです。このクラスから ModuleCore の派生クラスであったり Editor クラスを呼び出します。

2.3 ModuleCore

モジュールの基底クラス。詳細は後述。

2.4 Editor

エディターの基底クラス。

3 Module とは？

3.1 ざっくり

Module とは、ゲームを管理する基本となるクラスのことです。具体的には以下があります。

- GameModule
- SceneModule

GameModule はゲーム自体の礎となるクラス。これが進行を管理することになります。

これの下にシーンを管理する SceneModule があり GameModule は SceneModule を切り替えつつゲームを進行します。

3.2 SceneModule

シーンを構成するモジュールです。具体的には場面や CG シーンなどを移すときに使うことになります。このモジュールを GameModule にキューとして追加していきます。

4 簡単な使い方

4.1 プロジェクト作成

ではこの章から実際にビルドまでしてみましょう。せっかくなのでプロジェクトから作ってみます。VisualStudio で C++ から空のプロジェクトを作ってください。C++ の言語標準は C++17。ランタイムライブラリを debug、Release でそれぞれマルチスレッドデバッグ、マルチスレッドに設定。追加のインクルードディレクトリに Cinder/include と TellerEngine/src を追加。追加のライブラリディレクトリに cinder.lib のあるパスを追加します。その後追加の依存ファイルに cinder.lib を追加します。サブシステムを Windows に設定。以上で終了です。

4.2 TellerCore から Scene まで

Cinder はマルチプラットフォーム対応の描画ライブラリのためプログラマーが WinMain 関数を書く必要はありません。その代わり Cinder の App クラスを継承したクラスが WinMain 関数の代わりとなります。

ではまず Cinder::App を継承したクラスを作ります。main.cpp を作成してください。そして以下のコードを書きます。

Listing 1 hoge

```
1 #include "cinder/app/App.h"
2 #include "cinder/app/RendererGl.h"
3 #include "cinder/gl/gl.h"
4 class TellerEngineMain : public App {
5 public:
```

```
6         std::shared_ptr<TellerCore> mCore;
7
8         void setup() override;
9         void update() override;
10        void draw() override;
11 };
12
13 void BasicApp::draw()
14 {
15        gl::clear( Color::gray( 0.1f ) );
16
17        gl::end();
18 }
```