

Game Programming

白井 暁彦 (Akihiko SHIRAI)
shirai@ic.kanagawa-it.ac.jp / K1-501





入門ゲームプログラミング (Professional Game Developerシリーズ)

- 著者チャールズ・ケリー, Charles Kelly
- 価格 ¥ 3,132 (2014/04/20時点)
- 出版日2014/01/30
- 商品ランキング25,165位
- 大型本528ページ
- ISBN-104797374543
- ISBN-139784797374544
- 出版社SBクリエイティブ

- * 学修目標・学修内容：ゲームプログラミング技術は他のプログラミングと何が異なり、どのような設計手法、アルゴリズム、実装、データ構造、デバッグ、最適化手法、ユーザ行動分析がとられる必要があるのであろうか。実際の開発事例を分析しながら、初学者から専門技術者までの中間的技能を学ぶ。
- * 学修成果：ゲーム開発に長年関わるエンジニアとして必要となる技術、特にグラフィックス、ネットワーク、就職試験に出題されるアルゴリズムについて学び、専門基盤的な実力を身につけることができる。
- * 履修条件：応用プログラミングA(C++)
- * 履修推奨：ゲームグラフィックス、コンピュータアーキテクチャ
- * 授業形式、形態：講義および演習
- * 成績・評価：試験40% 小テスト15% 成果45%

講義の準備作業

- * Visual Studio
- * DirectX SDK (June 2010)
 - * www.microsoft.com/en-us/download/details.aspx?id=6812
- * Github
 - * <https://github.com/kaitas/ShiraiLabOpen/tree/master/GamePro>
- * Facebook Group
 - * <https://www.facebook.com/groups/KAIT.GamePro/>

新規プロジェクトの作成

スタートページ ×



Team Foundation Server への接続

新しいプロジェクト...

プロジェクトを開く...

最近使ったプロジェクト

- SkeletonBasics-WPF
- FaceTrackingBasics-WPF
- ColorBasics-WPF
- DepthBasics-WPF
- WiiMoteLib01
- DepthBasics-WPF
- DepthBasics-WPF
- DepthBasics-WPF
- WindowsFormsApplication1

新しいプロジェクト

最近使用したテンプレート

インストールされたテンプレート

Visual C#

Windows

Web

Reporting

Silverlight

WCF

Workflow

テスト

他の言語

Visual C++

ATL

CLR

全般

MFC

テスト

Win32

その他のプロジェクトの種類

データベース

テスト プロジェクト

オンライン テンプレート

.NET Framework 4

並べ替え基準: 既定

Win32 コンソール アプリケーション

Visual C++

MFC アプリケーション

Visual C++

Win32 プロジェクト

Visual C++

空のプロジェクト

Visual C++

ATL プロジェクト

Visual C++

MFC DLL

Visual C++

Windows フォーム アプリケーション

Visual C++

CLR コンソール アプリケーション

Visual C++

空の CLR プロジェクト

Visual C++

MFC ActiveX コントロール

Visual C++

Windows フォーム コントロール ライブラリ

Visual C++

名前(N): <名前を入力してください>

場所(L): c:\users\aki\documents\visual studio 2010\Projects

ソリューション名(M): <名前を入力してください>

参照

☒ ソリューション

☐ ソリューション

本日の課題

2章 Windowsプログラミングの基礎

- * HelloWorld (List 2.10)
- * Character Input (List 2.14)
- * 時間のある人は2章の終わりまで進めてください

2章 Windowsプログラミングの基礎(の続き)

- * デバイスコンテキスト
- * イベントハンドラ
- * メッセージ
- * Mutex

3章 DirectXの初歩

- * SDKライブラリの設定
- * ウィンドウ作成
- * フルスクリーン

教科書のサポートページも利用すること

- * <http://www.sbcr.jp/products/4797374544.html>

事後学習#2のポイント

インクルードファイルをコピーせよ

```
#include "constants.h"
```

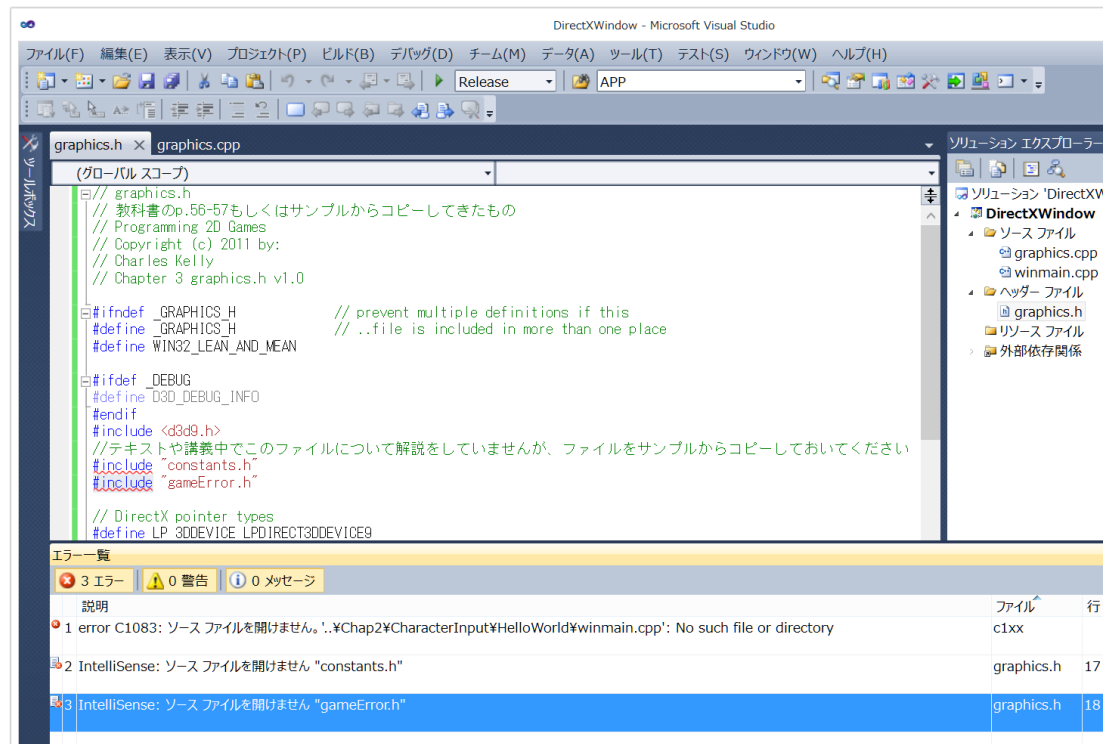
```
#include "gameError.h"
```

//認識されない場合はプロジェクトに追加する

Winmain.cpp (Line:49)

```
graphics->initialize(hwnd, GAME_WIDTH, GAME_HEIGHT, FULLSCREEN);
```

//FULLSCREENはどこで定義されているか確認せよ



3章 DirectXの初歩(の続き)

- * 前回の事後学習のポイント
- * ウィンドウ作成
- * フルスクリーン
- * DirectXDeviceCapabilities (p.80)

4章 ゲームエンジン

- * ゲームクラス
- * タイマー

事後学習 Chap4

サンプルのタイマー動作周波数を変更してみよう

事後学習#3のポイント

- * 英語コメントに日本語コメントを書き加えよう！辞書を引こう！
 - * Graphics, Inputなどクラス名を見分けられるか？
 - * Render: 表現する→レンダリング
 - * Entity: 実体・実在物
 - * Inherit: 継承
 - * Perform: 処理・実行
 - * Collision: 衝突
 - * Attempt: 試みる、企てる
- * 動作周波数？
 - * 高精度タイマー(p.94)、(追加)タイトルへのFPS書き込み
- * 消失デバイスのリセット、レンダリング、ゲームループ

ゲームループ速度の設定と計測

* P.99-101

経過した時間として計算します。プログラミングの一部の解説書やネット上にあるサンプルでは、フレームタイム値を、ゲームが1つのフレームをレンダーするために要した時間として計算している場合があります。この時間の値は、多くの場合、ゲームループの開始時と終了時にタイマーを読み取り、2つの時間の差を取得することによって計算されています。この方法は、ゲームループ外でWindowsや他のアプリケーションが使った時間が考慮されていないため、正確とは言えません。



どのような情報源のものであってもプログラムコードを鵜呑みにしないでください。コードに間違いは付き物です。常に対象を調査し、コードを使用する前にそれが正確かどうかを確認してください。

* constants.h

- * `const float FRAME_RATE = 200.0f; // 目標FpS (frames/sec)`
- * `const float MIN_FRAME_TIME = 1.0f/FRAME_RATE;`
- * `const float MAX_FRAME_TIME = 1.0f/MIN_FRAME_RATE;`

#4(5/19) ゲームエンジンと描画

第4章「ゲームエンジン」

- * 動作周波数
- * インプットデバイス
 - * (自習推奨) キーボード、マウス
 - * (可能なら) ゲームパッド、バイブレーター
- * ゲーム本体:: Spacewarクラスと純粹仮想関数
 - * spacewar.h, spacewar.cpp に日本語訳をつけよう

第5章「スプライトとアニメーション」

- * グラフィックスパイプライン
- * 透過性描画 (SpriteBorder, Planet)
- * (宿題) SpaceshipControl のコメント日本語化