報道各位

2020 年 2 月 26 日
株式会社インプレスR&D
https://nextpublishing.jp/

Windows/Linux 両対応のサンプルで、段階的に技術を習得！

『Rust ではじめる OpenGL』発行

技術の泉シリーズ、2月の新刊

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『Rust ではじめる OpenGL』(著者:山口 聖弘)を発行いたしました。

最新の知見を発信する『技術の泉シリーズ』は、「技術書典」や「技術書同人誌博覧会」をはじめとした各種即売会や、勉強会・LT 会などで頒布された技術同人誌を底本とした商業書籍を刊行し、技術同人誌の普及と発展に貢献することを目指します。

『RustではじめるOpenGL』
https://nextpublishing.jp/isbn/9784844378556

著者:山口 聖弘
小売希望価格:電子書籍版 1600 円(税別)／印刷書籍版 2000 円(税別)
電子書籍版フォーマット:EPUB3／Kindle Format8
印刷書籍版仕様:B5 判／カラー／本文 158 ページ
ISBN:978-4-8443-7855-6
発行:インプレス R&D

＜＜発行主旨・内容紹介＞＞

本書は OSS のプログラミング言語 Rust を使って OpenGL プログラミングを行う入門書です。最も基本的な三角形の描画から実装をはじめ、少しずつソースコードを発展させていきます。照明を考慮しながら立体的なオブジェクトにテクスチャを貼り、GLSL を使って画面にエフェクトをかけるところまでを解説します。また、GUI を簡単に導入できるライブラリ「Dear ImGui」を使うことで、様々なパラメータをより直感的に操作できる親切設計になっています。

（本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。）

クロスプラットフォームライブラリー「SDL」を使ったプログラムを作成

crateについての情報を調べるとき、多くの人たちはRustのパッケージが登録されているサイト「crate.io」へ行きます。これは、Pythonでいうところの「PyPI」、Rubyでいうところの「RubyGems.org」です。

crates.io

```
https://crates.io/
```

このサイトで使おうとしているcrateを検索します。
今回利用するSDLのcrateは、こちらのページで見ることができます。

sdl2のcrate情報

```
https://crates.io/crates/sdl2
```

図2.1: crates.ioの中のsdl2

このページの中央に書いてあるcrateのバージョン情報をコピーして、Cargo.tomlにペーストします。今回のスクリーンショットの例では、「sdl2 = 0.32.2」と書いてあるので、それをペーストします。

リスト2.1:sdl2crateの追加

```
1: [package]
2: name = "opengl_app"
3: version = "0.1.0"
4: authors = ["Owner Name <owner_name@email.com>"]
5: edition = "2018"
6:
7: [dependencies]
8: sdl2 = "0.32.2"
```

次に、各プラットフォームごとにSDLライブラリーのインストールを行います。インストール方法については、sdl2crateのGitHubに記述があるので、それにそって解説していきます。今回ここで解説するのは、LinuxとWindowsの場合を解説します。

2.1.2 Linuxの場合

ディストリビューションごとにパッケージマネージャーを使って、SDLライブラリーをインストールしましょう。基本的には次に記述したコマンドでインストールができますが、SDLの補助ライブラリー（SDL_image、SDL_ttf、SDL_mixerなど）を使いたい場合には、自分でパッケージ名を検索しインストールする必要があります。そのときには、次に書くパッケージの検索コマンドを使えば、パッケージ名を知ることができるでしょう。

Debian、Ubuntuの場合

Debian系Linuxの場合は、apt-get、apt-cacheなどを使ってパッケージ管理をします。

apt-getによるインストール

```
sudo apt-get install libsdl2-dev
```

SDL関連パッケージの検索方法

```
sudo apt-cache search libsdl2
```

Fedoraの場合

FedoraのようなRed Hat系Linuxの場合は、dnfを使ってパッケージ管理をします。古いバージョンのLinuxを使う場合は、yumを使います。

dnfによるインストール

```
sudo dnf install SDL2-devel
```

SDL関連パッケージの検索方法

```
sudo dnf search SDL2
```

Archの場合

Arch Linuxを使う場合には、pacmanを使ってパッケージ管理をします。

OpenGLのプログラミングを丁寧に解説

です。
まず、バーテックスバッファーオブジェクトは、GPU側のメモリ上に頂点データを置いておくための領域です。GPUで描画するためには、CPU側からGPU側へデータを送る必要があるため、バーテックスバッファーオブジェクトを作って、その中に頂点データを保存しておくのです。バーテックスバッファーオブジェクトは、省略してVBOと書かれることも多いです。
バーテックスアレイオブジェクトは、頂点データをどのようなまとまりで使うのかを設定するものです。頂点データは、ただのfloat型の配列でしかありません。たとえば、その配列を3個のfloat型ごとに3次元の座標として扱い、ひとまとめにすることができます。（図3.2）少し複雑な例では、3次元の座標の他に、法線ベクトルと色情報もあわせた、10個のfloat型でひとまとめにすることもあります。（図3.3）まとめられたデータは、シェーダー上で効率よく使うことができます。バーテックスアレイオブジェクトも、VAOと省略されることがあります。

図3.2: データをまとめるバーテックスアレイオブジェクト

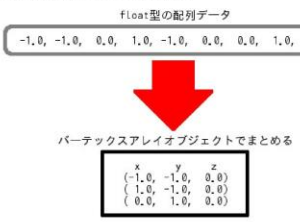
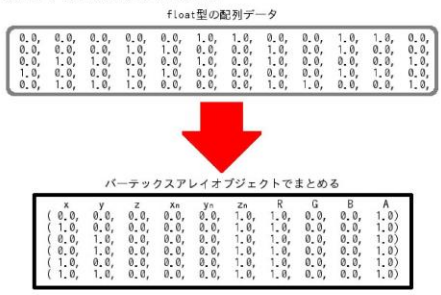


図3.3: 複雑なデータをまとめるバーテックスアレイオブジェクト



それではここに、VAOとVBOを扱うVertex構造体のためのソースコードを記します。

リスト3.9: VAOとVBOのソースコード: vertex.rs

```
1: use std::mem;
2: use std::os::raw::c_void;
3:
4: use gl::types::{GLenum, GLfloat, GLint, GLsizei, GLsizeiptr};
5:
6: pub struct Vertex {
7:     vao: u32,
8:     _vbo: u32,
9:     vertex_num: i32,
10: }
11:
12: impl Vertex {
13:     pub fn new(
14:         size: GLsizeiptr,
15:         data: *const c_void,
16:         usage: GLenum,
17:         attribute_type_vec: std::vec::Vec<GLenum>,
18:         attribute_size_vec: std::vec::Vec<GLint>,
```

OpenGL に様々な GUI を追加できるライブラリー「Der ImGui」を解説

第4章　Dear ImGui

この章では、OpenGLにさまざまなGUIを追加できる「Dear ImGui」(通称、ImGui)というライブラリーを導入します。

もともとDear ImGuiはC++で実装されていますが、面倒なOpenGLの処理の中に便利なGUIウィジェットを導入できるとあって、多くの言語にポーティングされています。ソースコードはGitHubで公開されているため、誰でも参照できます。

C++版 Dear ImGuiのGitHub

```
https://github.com/ocornut/imgui
```

Rustのimguicrateは、現在もより使いやすくするための改善がなされているため、近い将来にもウィジェットのインターフェイスが変更される懸念がありますが、それを差し引いてもOpenGLの効率的な開発を行う上で、大変便利なライブラリーとなるはずです。

参考のために、本書の始めにあるギャラリーのページに、Dear ImGuiのサイト上で紹介されているスクリーンショットを引用しました。この画像を見ると、テキスト、ボタン、リスト、プルダウンメニュー、図形やグラフなど、さまざまなウィジェットを作成できることがわかります。

4.1　準備

Cargo.tomlに、Dear ImGuiのためのcrateであるimgui、imgui-sdl2、imgui-opengl-rendererを追加します。

imgui、imgui-sdl2、imgui-opengl-rendererのcrate情報

```
https://crates.io/crates/imgui
https://crates.io/crates/imgui-sdl2
https://crates.io/crates/imgui-opengl-renderer
```

Cargo.tomlの[dependencies]のところに、それぞれのバージョンを追加します。

リスト4.1:imgui、imgui-sdl2、imgui-opengl-renderercrateの追加

```
[dependencies]
sdl2 = "0.32.2"
gl = "0.14.0"
cgmath = "0.17.0"
c_str_macro = "1.0.2"
imgui = "0.2.1"
imgui-sdl2 = "0.7.0"
imgui-opengl-renderer = "0.6.0"
```

imguicrateは、RustでImGuiを使えるようにするための、もっとも主要なcrateです。もし実装したいウィジェットを探す場合には、このcrateのドキュメントから探すとよいでしょう。

imgui-sdl2crateは、ImGuiをSDL2のフレームワーク上で使えるようにしたcrateです。ImGuiはさまざまなOpenGLフレームワーク上で動作することができるので、それらと合わせて使えるようにするためのグルーコードが多少なりとも必要なのです。

imgui-opengl-renderercrateは、上記imgui-sdl2crateの作者が合わせて作った、描画処理の部分を担当するcrateです。

4.2　プログラムの作成

Dear ImGuiを使って、SDL2の生成したウィンドウ内に、ImGuiが生成するOpenGL製のウィンドウを描画します。前の章までで作成したOpenGLのソースコードに、Dear ImGuiのソースコードを追加していきます。

4.2.1　初期化

Dear ImGuiを使う準備をしましょう。

次のソースコードを、前章のmain.rsソースコードの72行目に挿入しましょう。(Vertex構造体の準備が終わった直後、イベントループの直前)

リスト4.2: Dear ImGuiの初期化

```
    // init imgui
    let mut imgui_context = imgui::Context::create();
    imgui_context.set_ini_filename(None);

    // init imgui sdl2
    let mut imgui_sdl2_context = imgui_sdl2::ImguiSdl2::new(&mut imgui_context,
&window);
    let renderer = imgui_opengl_renderer::Renderer::new(&mut imgui_context, |s| {
        video_subsystem.gl_get_proc_address(s) as _
    });
```

imgui::Context::create()メソッドで、ImGuiの全体的な設定を管理しているコンテキストデータを取得します。

そのあと、imgui_context.set_ini_filename(None)をしています。もともとDear ImGuiには、ソフトウェアの終了時に自動でウィジェットの位置などを保存する機能があり、設定ファイルが作られます。そのような保存作業が不要な場合は、こうして引数にNoneを指定し、抑制することがで

＜＜目次＞＞

＜＜著者紹介＞＞

山口 聖弘

ソフトウェアエンジニア歴 12 年。好きなプログラミング言語は、C/C++、Python、Rust。大学・大学院ではデータマイニング・機械学習を研究。現在は、セキュリティー企業に勤務し、Developer も QA も、場面によって両方担当。業務によって、Android アプリ、Web システム、Linux カーネルなどを、薄く広く扱う。興味のある分野が多く、OpenGL による 3D 技術もその中の 1 つ。ネットワークの 3D 可視化技術の実装に情熱を注ぎ、その集大成として本書を完成させた。

＜＜販売ストア＞＞

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

【インプレス R&D】 https://nextpublishing.jp/

株式会社インプレス R&D（本社：東京都千代田区、代表取締役社長：井芹昌信）は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム（またはメソッド）の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド（POD）による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 https://www.impressholdings.com/

株式会社インプレスホールディングス（本社：東京都千代田区、代表取締役：唐島夏生、証券コード：東証 1 部 9479）を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「旅・鉄道」「学術・理工学」を主要テーマに専門性の高いメデイア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター
TEL 03-6837-4820
電子メール: np-info@impress.co.jp