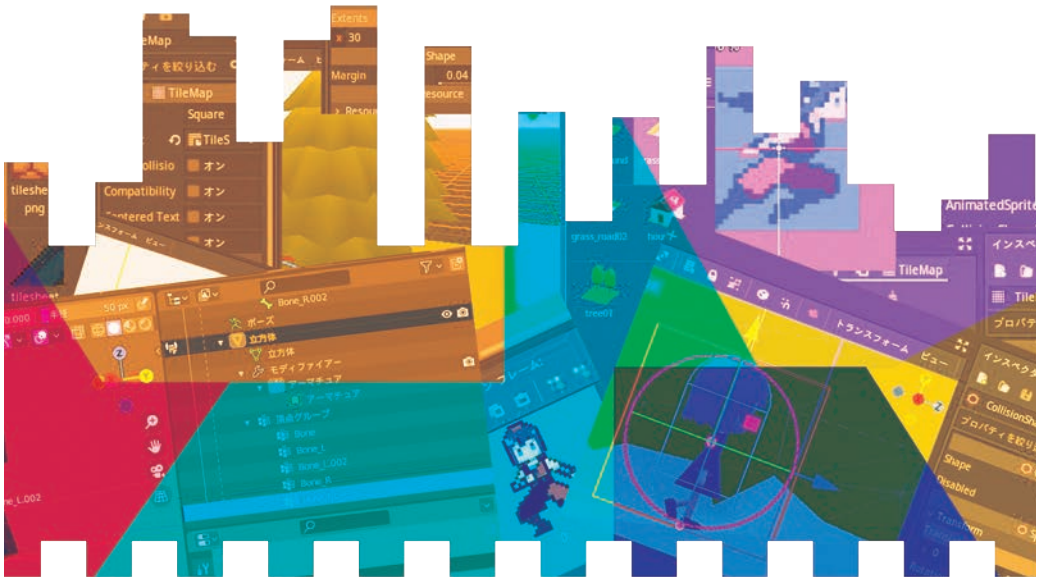
はじめてのゲームエンジン Godot

酒井雅裕◉著



CUTT
カットシステム

■**サンプルファイルのダウンロードについて**

本書掲載のサンプルファイルは、下記 URL からダウンロードできます。

https://----------

・本書の内容についてのご意見、ご質問は、お名前、ご連絡先を明記のうえ、小社出版部宛文書（郵送または E-mail）でお送りください。
・電話によるお問い合わせはお受けできません。
・本書の解説範囲を越える内容のご質問や、本書の内容と無関係なご質問にはお答えできません。
・匿名のフリーメールアドレスからのお問い合わせには返信しかねます。

はしがき

本書では Godot（ゴトー）※1 というゲームエンジンを用いて、GUI 操作・プログラミングを体験する。

Godot は以下のような特徴がある。

- シーンとノードによるプログラミング。シーンにノードを配置してゲーム全体を作り上げる。
- 3D データに関してはオープンソースの 3D モデル形式である glTF を扱える。これを Godot 側でインポートして 3 次元形状のノードを構成できる。アニメーションやボーンは Blender などの 3DCG ソフトで定義して出力する。glTF については章を改めて詳しく扱う。
- OpenGLES に準拠したシェーダによるリアルタイム 3DCG 表現が可能である。
- エンジン開発はオープンソースで進められている（Windows、MacOS、Linux で動作する）。
- SDK の別インストールなどが条件となるが、モバイルの環境に出力可能である。モバイルの開発には開発者の証明書のインストール・セッティングが不可欠であり、本書の範疇を超えてしまった。
- VR に関しても、AndroidSDK を経由して出力する※2。
- 本体と Windows 出力パッケージで 1GB 位と**軽い**。各環境によって容量は様々である。モバイルなどはそれぞれの SDK が必要である（概ね数 GB 程度）。

このように Godot はモバイルも含めた多くの環境への出力が可能なことからも大きな可能性を秘めたゲームエンジンである。また Godot の最も大きな特徴として、GDScript という Python 言語に近いスクリプトでプログラミングができる。Python 言語は学びやすく簡潔であるため、高等学校のコンピュータ教育の言語として試用されている [1]。また C# でのプログラミングも準備されている。併せて UnrealEngine のブループリントのようなノード接続によるビジュアルプログラミングもできる。

ネットワークリソースを含んで基本的なことは網羅されているが、ゲームエンジンの基本を押さえたコンセプトであり、高度なことはプログラミングで補う必要がある。少なめであるがサンプルやアセットも準備されている。

プログラミングで一番大事なことは成果物が作れることや、どんな環境にも対応できる能力を身につけることである。本書では Godot でのチュートリアルを通じて 2D・3D のプログラミングを一通り学べるよう心がけたつもりである。公式のチュートリアルになじめなかった読者も気軽

※1　「Godot ドキュメント」https://docs.godotengine.org/ja/stable/index.html

※2　「Developing for Oculus Quest」https://docs.godotengine.org/ja/stable/tutorials/vr/oculus_mobile/developing_for_oculus_quest.html

に手に取って欲しい。本格的なゲームエンジンに取り組む前に、このようなサブセット的な軽い環境を学習し、基本を押さえて学びをすすめてもらいたい。

また本書は、神奈川工科大学の情報学部情報メディア学科の 2021 年度ゲームプログラミング科目の提示資料を改稿したものである。この科目を受講した全ての学生諸君がいなければ、本書は完成していない。

また講義をすすめるにあたって、質問が多く寄せられたことも本書の礎となっている。また本書で扱っているサンプルデータは神奈川工科大学情報学部情報メディア学科の酒井研究室の有志「書籍データを作るのだチーム」から厚意で提供して頂いたものである。この場を借りて御礼を申し上げる。

対象読者

- ゲームエンジンを理解したい高校生、専門学校生、大学生。
- プログラミングに関しては初心者でも構わないが、スクリプト言語を含めて何らかのコンピュータ言語の体験があると望ましい。Python 言語をおすすめする。
- チュートリアルを通じて学ぶ方式が、自分に合っていると感じる読者。
- Blender などの 3DCG を一通り（例：簡単な操作をマスターしてモデリングできる）操作できることが望ましい。

また Godot は MIT ライセンスであり、かなり自由度は高い。詳細は https://godotengine.org/license を参照して欲しい。

```
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

-- Godot Engine <https://godotengine.org>
```

またGodot3.4ブランチのドキュメントのライセンスは以下のようになっている。本書も時折参照する。

```
Copyright(c) 2014-2022, Juan Linietsky, Ariel Manzur and the Godot community
 (CC-BY 3.0). Revision 17842f5c.
```

第 4 章、第 5 章、第 6 章、第 7 章、サンプルの忍者ゲームのデータのデータに関しては、Yuka SHIMIZU によって、第 8 章のデータに関しては、Masahiro SAKAI によって、第 9 章、第 10 章、第 11 章のデータに関しては、Akika TAKADA によって CC0（クリエイティブコモンズ）によりライセンスされている。

また、Godot のバージョンは、3.4.2 を本書を通して使用する。

目次

付 録 …… 197

1

ゲームエンジンを学ぶ目的：得点類型の実装

1.1 ゲームプログラミングとは

とても乱暴かもしれないが、ビデオゲーム、もしくは電子ゲームとは「ハードウエア、もしくはソフトウエアでプレーヤが得点する様々な方法を提供する」こととしてみよう。

もちろんビデオゲームのジャンルは非常に多く、そのジャンルの中で大まかなプレーヤの得点方法の「類型」があって、それを踏襲してゲームデザインが決まってくる。例えばジャンルの一つに「Platformer」がある。今となってはとても簡単なシステムである。

横スクロール型のステージがあり、プレーヤをジャンプさせつつ、穴などに落ちないようにして、迫ってくる敵を倒して得点し、ステージをクリアしていく。この部分が得点の「類型」にあたる。得点のベースにはキャラクタの接触（コリジョン）がある。プレーヤが敵を踏みつけるコリジョンは敵の消滅と得点を生む。敵に正面から当たれば、プレーヤは死んでしまう。この見慣れたタイプの得点システムは、一つの「類型」として、プレーヤに受け入れられ、応用デザインが生まれてきた。よくゲームのプログラミングにはこの platformer が例題で出てくる。本書も同様である。またゲームには、直接得点に結びつかなくても、条件達成によるレベルのアップや、ガチャによるアイテムの獲得などもある。これも大きく俯瞰して考えれば「得点類型の拡張」と考えることができるだろう。

ゲームはそもそも「マニュアルレス」なものである。筆者の記憶では 1990 年代初めには、ゲームは開始直後から 1 タームは「チュートリアル」が実施されるようになった。その中でゲームの操作方法をはじめとする「システム」を理解させる必要がある。その際に「バトル」等の「得点の方法」を理解させた上で、ユーザを引きつける必要がある。この場合の「引きつける」をもう少し細かく説明すると、そのチュートリアルの中で「ストーリ」や「ミッション」を伝え、得点を重ねることでレベルが上がりミッションを達成するという経験を通じてゲーム継続の「興味」を抱かせることにある。であれば、短いチュートリアルの中で重要なことは、できるだけプレーヤのゲーム体験「得点類型」に寄り添って「攻略の展望」を認識させることでもある。

当然この得点類型には難度も含まれる。ゲームバランスは得点類型と難度のバランスである。得点の方法が理解できても、スピードが速くなればクリアができない。敵がうじゃうじゃ出てきてよけきれないようではクリアはできない。難度が高くてもクリアできるポイントが作ってあれば「無理ゲー」にはならない（人生もクリアできるポイントが作ってあってそれが見えればいいのに）。その点でみると「クソゲー」の定義もはっきりする。いわゆるクソゲーとは突然難度を厳しくし、クリアしにくくして、ゲームバランスの逸脱が明らかに見えてしまうと、ユーザは流石にやる気をなくしてしまう。この点はクソゲーの一要素でしかないが、その点がゲームづくりのポイントとも言えるだろう。

いささか脱線気味になったが、ゲームエンジンとは「得点類型」を設計し実装するためにあるとは言える。操作性のテストも、その得点類型をいかにスムーズにストレスなく体験させるかに重点を置くべきであろう。

1.2 読者のみなさんは……

読者のみなさんは非常に多くのゲームを体験してきて「得点類型」の経験を多く積んでいるはずである。ゲームを作ってみたいという動機は、自分なりの得点システムをこしらえてみたいと多少なりとも思っているはずである。でなければゲームをプログラミングすることを目的としたこの本を手に取るはずがない。みなさんは小さな時から「デジタルネイティブ」（英語圏ではGen-Z：Z 世代）であり、非常に高度で完成されたコンテンツを消費し目が肥えている。また、みなさんはゲームを由来とするいくつもの愛でるべき物語があって、その世界を楽しんでいるはずである。

単純な「得点類型」の体験であったはずなのに、高度なコンテンツを大量かつ長時間消費しゲームの経験が蓄積されることで、みなさんの体験は考え方を変えるまでになっているはずである。詳細で美麗なゲームグラフィックスやモチベーションを高めるゲームサウンドの効果も相まって、内的なゲームの経験は好みの「世界観」と名を変えてしまっているかもしれない。その世界観はなかなか親の世代や筆者のような大学教員には理解されないことが多い。みなさんはきっと、ある程度の年齢がはなれたその世界に嗜みのない人たちには、理解できないと思っているだろうし、おそらくそれは正しい（なぜならその人達はみなさんと同じようなゲームの体験がないからである）。

ちょっと、みなさんが考えているその世界を図にしてみた（図 1.1）。

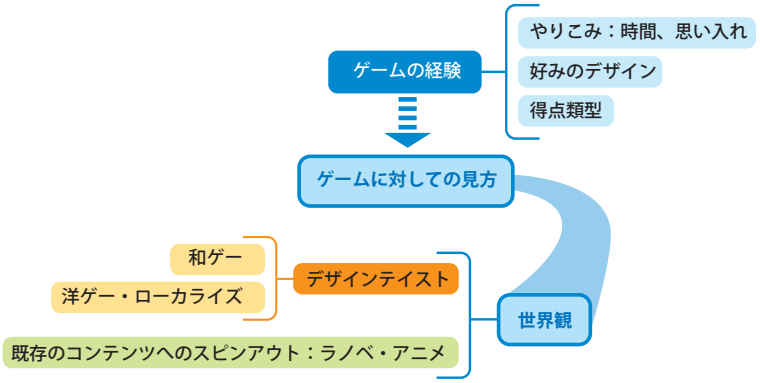


図1.1●ゲームによる世界観の生成

この図では、みなさんは既にゲームコンテンツ（得点類型）の消費を通じて、世界観に関しては非常に深い知識と経験を積んでいる点を描いている。みなさんの経験は「世界観」に昇華してしまっている。例を示せば、みなさんは古いコンテンツ（アニメや映画）からのゲームのフィードバック（移植）や、ゲームコンテンツから他のメディアへのエクスポート（例：ドラマ化・アニメ化・映画化）もよく知っていると思うし、その世界観もゲーム企画に活かせるだけの知識や、ひょっとしたらコンテンツの保有会社にプレゼンできる知識と熱量を持っているかもしれない。

1.3 スキルの第一歩はなにか

この図から考えると、みなさんのゲームへの情熱は既に大きな概念に昇華してしまっている。しかし出発点は「得点類型の体験」であったはずで、みなさんが立ち返って身につけるべきスキルは、沢山の得点類型のデザインとその実装方法であるはずだ（デザインやサウンドは本書の範疇から外れるので取り扱えない）。とすればプログラマ（が第一志望のみなさん）が、まず目指すべきスキルは得点類型の実装である。そのための道具を得て、実験をしてプロトタイプを作り、それがプレーヤに受け入れられるか試し続けなくてはならない。つまり、みなさんがゲームエンジンを学ぶ目的はどんなエンジンであれ「得点類型実装」の練習をし続ける道具を得ることだ。

もう一つ言及をしなければならないことは、IT の世界の代謝が非常に早いということだ。2008 年頃に世界的にモバイルの普及が開始されて、十数年経つがほぼ毎年 OS はリプレスされてきた。そのたびに機能は増え、古くからある機能も大きく改まり複雑化して、API の学び直しが必要に

なった。つまりゲーム分野に限らずプログラマを続ける限り「勉強のやり直し」は起きて、その情報は実装方法に限定すると、教科書であることは少なくほぼネットにある（学問では違うことは起きるが、IT の実装はググる→ Qiita[※1] などをあたるはずだ）。そこで正しい道を選ぶのは自分で、それをミスすると膨大な時間を無駄にする。この正しい道の鑑識眼の獲得は慣れと訓練が必要である。何度も繰り返し作品を作って慣れていってもらいたい。

また、沢山の試作や作品を作ることは就職活動には有利である。大学生の就職活動にはいつの頃からか、経験者の転職の際に求められる「職務経歴書」にあたる「ポートフォリオ」があると活動に有利になるとされてきた。Godot がみなさんの経験を支えて、試作をストックし「作品帳：ポートフォリオ」を整備する手助けになれば幸いに思う。

※ 1　「Qiita」https://qiita.com/（エンジニアに関する知識を記録・共有するためのサービス）

2

Godot のインストールとデプロイ

この章で学べること

- Godot のダウンロードとインストール
- Godot で作成したゲームのデプロイ

2.1 ダウンロードとインストール

Godot のインストールは特別な ID 管理も無く簡単である。ブラウザは日本語翻訳機能が標準化されている、Google Chrome を推奨する。まず Google で検索をする（図 2.1）。

図2.1●Godotの検索

Godot のホームページがわかったら、そこにジャンプする（図 2.2）。

図2.2●Godotの結果

Godot の公式ホームページを図 2.3 に示す。Godot のアプリケーションダウンロード先は「download」で示されている。

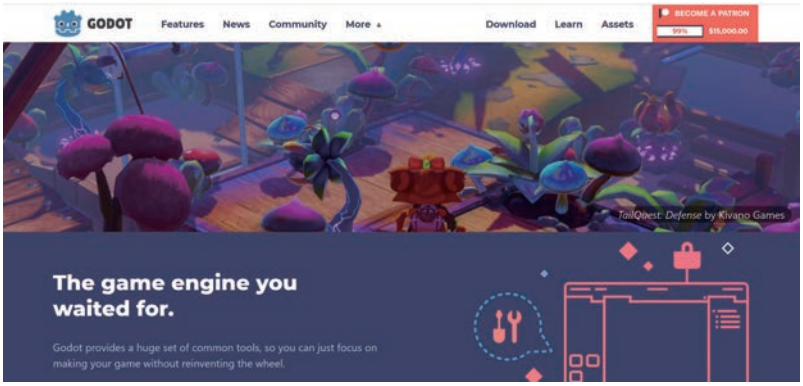


図2.3●Godot公式ホームページ

ダウンロード先で自分の環境にあったものを選択しよう（図 2.4）。本書では WindowsOS 向けの Godot について解説する。

図2.4●ダウンロードの選択

ダウンロードフォルダに Godot が入れられる（図 2.5）。

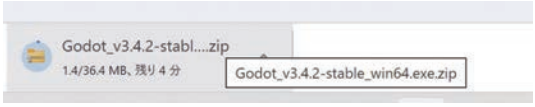


図2.5●ダウンロード中

ダウンロードファイルは zip ファイルである。このファイルをダブルクリックで開き、本体を確認する。本体は Godot-v3.x.x-stable-win64.exe（環境やバージョンによって異なる）（図 2.6）。

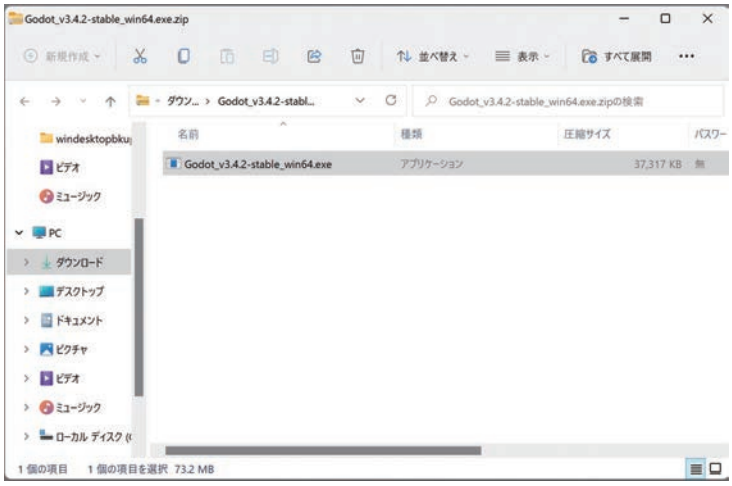


図2.6●ダウンロードファイル

exe 本体をデスクトップにファイルをドラッグアンドドロップする（図 2.7）。セキュリティを強化している場合などは、インストールができないなどのエラーが出るかもしれない。そのような場合はセキュリティを調整して対処しよう。

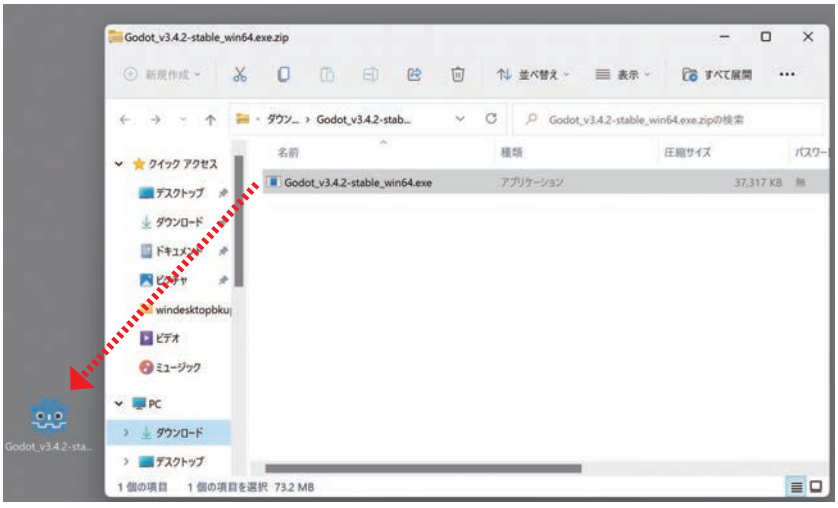


図2.7●ファイルの移動

インストールは以上である。アプリケーションは Windows は 10、11 いずれでも動作する。