１２．Unityを使った音楽ゲームアプリの開発

20番 福山　将樹 3番 尾形　祐紀

指導教員　佐々木　建

1. はじめに

|  |  |
| --- | --- |
| OS | Windows7 Ultimate sp1 |
| IDE | Unity5.3 |
| 使用言語 | C# |
| 使用機材 | Android 4.4(Nexus7) |
| その他ツール | AndroidSDK、AndroidNDK |

　私たちは普段から音楽ゲームをやっており、時間があればアーケード（ゲームセンター）の音楽ゲームを、時間があまりない授業の合間や昼休みなどでもスマートフォンやタブレット等の音楽アプリゲームを開いて日々音楽ゲームに励んでいる。

図１．iPadで音楽ゲームをやっている様子

　本研究で音楽ゲームをテーマとしたのは音楽ゲームがどのようにして作られているのか。また、AndroidやiOSといったモバイル端末向けの音楽ゲームというのは非常に数が少なく、自分たちだけでどれだけクオリティのあるゲームを作ることができるか挑戦してみたかったのが理由である。

1. 研究の目的

　この研究ではAndroidの端末で動かせる事を前提とし、ゲームとして必要な機能の実装と見栄えの良い画面、そして最終的には音楽ゲームに興味があるなしにかかわらず、誰にでも楽しんでもらえるようなゲームを作ることを目的とする。

1. 研究概要

　まず始めに今回の研究で使用する開発環境を表１にまとめた。

表１．開発環境

　Androidに対応したゲームを作るということで、ゲーム作成に適していて、なおかつAndroidに移植できるような開発環境を探したところUnityがもっとも使われているようだったので本研究ではこれを使って開発を進めていくことにした。

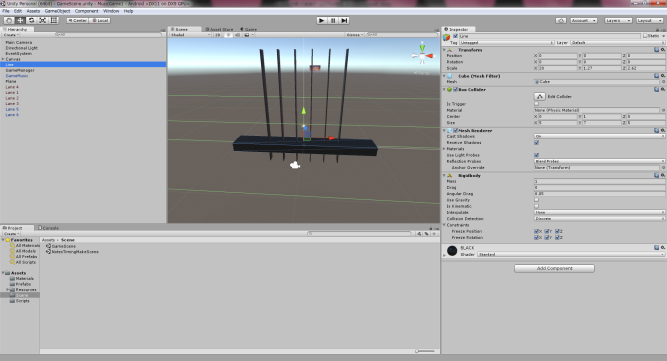
　また使用言語のC#はオブジェクト指向が主なスタイルの言語で、感覚としてはJavaがある程度使えるのであれば習得するのはそれほど難しくなさそうなので本研究で使うことにした（個人差はある）。

　デバッグとして使うNexus7(Android)は学校から貸し出されているものを使用した。

　AndroidSDKとNDKについてはUnityを使ってAndroidにビルドするときに必要になるのでそれぞれのツールを併せてインストールし環境を整えた。

3.1　Unityについて

　UnityはC#でスクリプトが書けることから多くの開発者が利用している。主にブラウザ、モバイル向けのゲームを開発するために用いられておりAndroidやiOS向けにビルドすることもできるのでここ数年は特にユーザー数が伸びている。

図２．Unityの画面

3.2　AndroidSDKについて

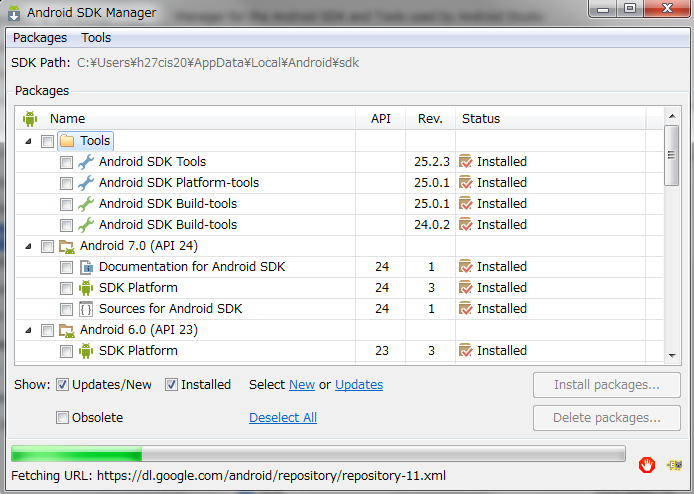
　パソコンなどでAndroid向けソフトウェアを開発するための開発環境であり、Google社が無償で公開している。Androidアプリを作るうえで必要なツールはSDKマネージャーからインストールができる。

図３．AndroidSDKマネージャーの画面

3.3　AndroidNDKについて

　Android Native Development Kitの略称で、Androidアプリケーションの一部または全部をC/C++言語で開発するためのキットである。

　Unity5.2からC#のコードをC++に変換できるIL2CPPというものが利用できるようになったためAndroidにビルドする際はこれが必要になる。

1. 研究の詳細

4.1　シーンの作成

　Unityでは一つのゲームの画面をシーンという単位で作成し、それぞれのシーンに別々のオブジェクトやスクリプトを追加することができる。

　今回の研究では4つのシーンを用意することにした。

1. ゲームスタートシーン
2. セレクトミュージックシーン
3. ゲームシーン
4. 譜面作成シーン

4.2　ゲームスタートシーンの作成

　ここではゲームを起動したときに一番最初に表示する画面を作成した。（図４）



図４．ゲームスタートシーンの画面

　図４赤枠の部分をタッチすると選曲画面となるセレクトミュージックシーンに移動することができる。

4.3　セレクトミュージックシーンの作成

　ここでは曲とその難易度を選択する図のようなシーンを作成した。

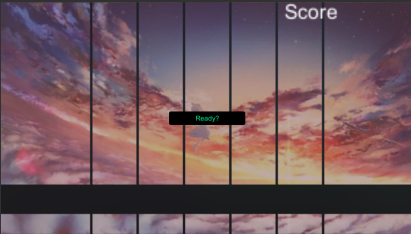


図５．セレクトミュージックシーンの画面

　画面左側に表示されている項目(赤枠)が曲のリストである。このうちのどれか１曲を選択すると画面右側の難易度(黄枠)の選択ができるようになる。

　難易度は簡単な順からEasy,Normal,Hardと設定しており、それぞれ別の譜面が用意されている。

　難易度を選択すると自動的にゲームシーンに移動する。



4.4　ゲームシーンの作成

　ここではゲームをプレイするための図６のようなシーンを作成した。

図６.ゲームシーンの画面

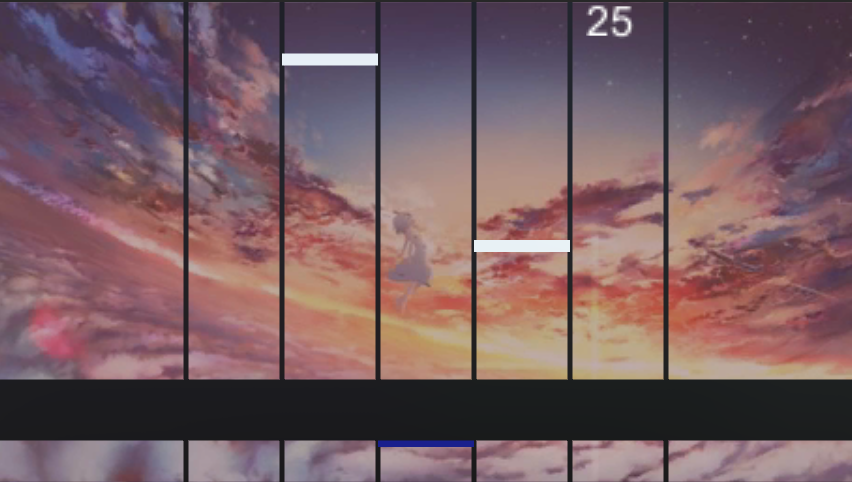
　このシーンでは、次の４つの処理が行われる。

1. 譜面ファイルを読み込む
2. BGMを再生
3. タイミングに合わせてオブジェクトが降ってくるようにする
4. タイミングよく押したらスコアを加算する

　以下、４つの処理について具体的に説明する。

1. 「譜面ファイルを読み込む」

譜面ファイルは譜面作成シーン(後述4.5で説明)で作ってあるので、作った譜面ファイルを基にゲームに必要なデータを取得していく。



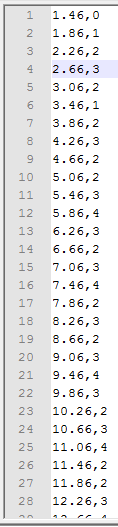


図７.譜面ファイルの内容

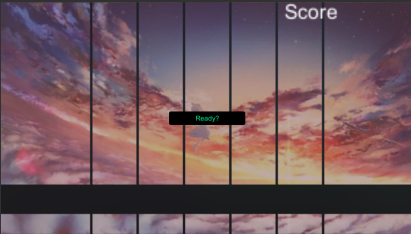
1. 「BGMを再生」

　図７の画面の真ん中のボタンを押すとBGMが再生されゲームが始まる。

図８.ボタンを押すとBGMが再生される

　Unityでは様々なフォーマットの音楽ファイルがサポートされているが、ファイルサイズの原因でゲームに影響を及ぼすこともある。低容量化のため今回はoggファイル\*というものを使用している。

\*oggとは

　オーディオファイルの圧縮形式としては比較的新しい。ビットレートを下げても音質の差があまり変わらない為、ファイルサイズを小さくしやすいのが特徴である。主な用途としてはPCゲームに使われることが多いようである。

1. 「タイミングに合わせてオブジェクトが降ってくるように表示する」

　前述の①で取得したデータを基にオブジェクト(図９の赤枠)を生成していく。以降の説明では、このオブジェクトをノーツと呼ぶことにする。

図９.オブジェクト（ノーツ）を生成

　BGMに合わせてタイミングよくノーツを落としていくのがこの研究での鬼門でもあったがほぼ満足のいく動作ができているのではないかと思う。



1. 「タイミングよく押したらスコアを加算する」

　ライン(図１０の青枠)にノーツが接触した時、そのノーツをタッチするとスコア（図１１の緑枠）が変わる。

図１０.スコアが０の状態

図１１.スコアが２になる

4.5　譜面作成シーンの作成

　このシーンでは譜面ファイルを作成することができる。

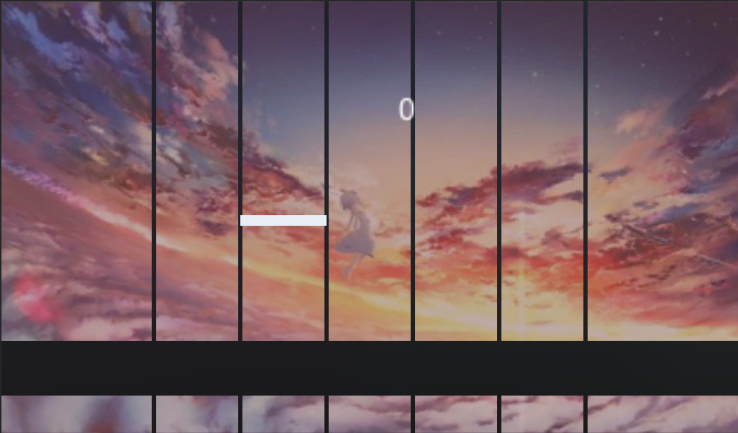


図１２.譜面作成シーンの画面

　特に目に見えるようなオブジェクトは置いていないがCSV\*に譜面のデータを入力するための処理を行っている。

　このシーンでは次の２つの処理を行っている。



1. midi\*\*ファイルを解析（読み込み）
2. midiファイルの情報を基にノーツのタイミングデータを入力

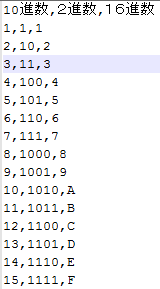
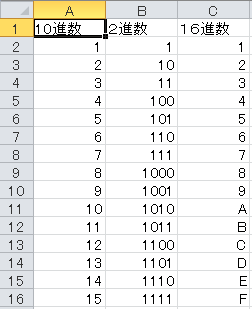
\*CSVとは

　Comma Separated Valueの略でカンマで区切った文字列が入っているファイルである。

　Excelのような表計算ソフトでデータを開くのに適しており、またデータベースソフトや電子手帳、ワープロソフトなども含めた様々なアプリケーションで開くことが出来る利点がある。

\*\*midiとは

　MIDI(Musical Instrument Digital Interface)と呼ばれる信号を扱い、デジタル楽器やコンピュータや音楽機材などを演奏（操作）するために作られたデータ転送規格。簡単に言うと、電子ピアノやシンセサイザーといった電子楽器の鍵盤一つ一つの音のデータを外部の機器（PC等）でも扱えるようにしたものである。主にDTMと呼ばれるコンピュータ上の作曲ソフトで使用されている。



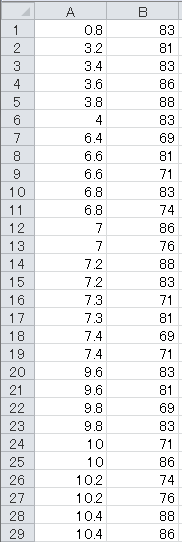
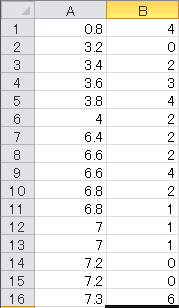


図１３.CSVファイル

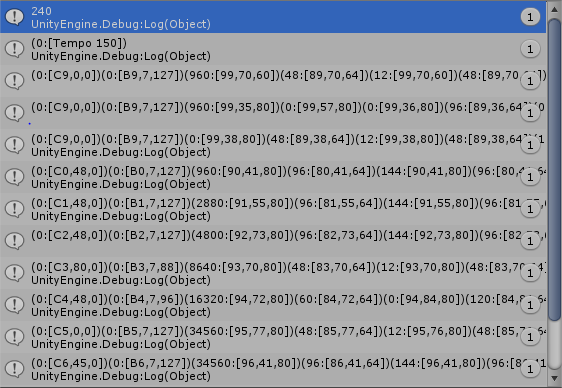
左.メモ帳で開いた画面

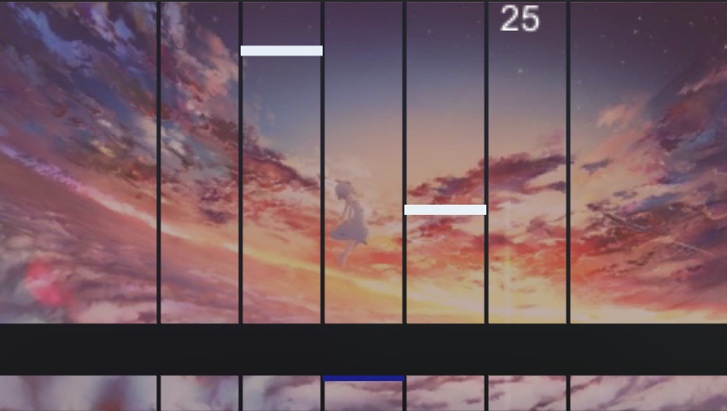
右.Excelで開いた画面

　以下、譜面作成シーンの２つの処理について説明する。

1. 「midiファイルを解析（読み込み）」

　midiファイルを読み込んで音楽の情報を取得していく。ここで取得した情報の一部が後程譜面データを作成するうえで必要になる。ちなみにmidiファイルを解析した時の情報を出力したものが以下の図１４である。





1

2

0

3

4

5

6

図１４.midiファイルを解析

1. midiファイルの情報を基にノートのタイミングデータを入力

　midiファイルから取得した情報を基にしてCSVファイルにノートのタイミングデータを入力していく。

　出来上がったCSVファイルが図１５である。

図１５.CSVファイルの内容

左.入力されたデータ

右.0～7の数字に変えた後

　緑枠がノーツが降ってくるタイミングを示しており、赤枠がノーツを落とす位置となっている。(図15右)

　出来上がったCSVファイルの図１５の赤枠の部分は0～6の数字に変える。(図１５右)

　するとゲームシーンでは図１６のようにノーツが降ってくるように表示する。

　図１5の左の赤枠のデータはスクリプトによって自動生成されたもので本研究には全く関係ないものである。

図１６．ゲームシーンの画面

4.6　改善点 課題等

* ノーツをタッチした時の判定の精度

タッチした時の処理のアルゴリズムが複雑なほどゲームの動作に影響がある。現在、対策を模索中である。

* 譜面の作り方をわかりやすくする

このゲームはもともと自分たちで譜面を作ってそれをプレイする前提で研究を進めていたが、誰でも譜面を作れるようにすればさらに面白いのではないかと思った。



1. まとめ

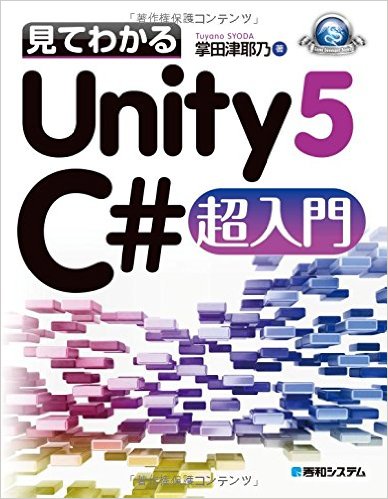
　音楽ゲームとしてはある程度出来上がっているとは思う。しかし自分たちの思い描いていた理想の形には程遠く、実装したかった機能がいくつか残っているのが心残りである。

1. 最後に

　研究を通して、勉強になったことがたくさんあった。特にC#を使ったプログラムの書き方を習得できた事は今後仕事で使う機会があった時にアドバンテージとなるであろう。

　Unityはスクリプトの書き方さえ解れば簡単にゲームが作成できるが、物理演算や衝突判定の機能がデフォルトでUnityに備わっているため突き詰めていけば高いクオリティのゲームも作ることができる。この資料を読んでUnityに興味を持った方がいたら軽く触るだけでもいいので是非Unityをインストールしてほしい。

**参考文献**

秀和システム　出版

「見てわかるUnity5 C#　超入門」

[掌田　津耶乃](https://www.amazon.co.jp/%E6%8E%8C%E7%94%B0%E6%B4%A5%E8%80%B6%E4%B9%83/e/B004L5AED8/ref=dp_byline_cont_ebooks_1)　著

SB Creative　出版

「ゲームの作り方　改訂版　Unityで覚える遊びのアルゴリズム」加藤　政樹　著

**参考webページ**

過去バージョンのUnityをインストール

<https://unity3d.com/jp/get-unity/download/archive>

Android SDKのインストール

<https://developer.android.com/studio/index.html?hl=ja>

[Unity]シンプルな音ゲーの作り方

<http://qiita.com/Teach/items/12a2e3b2f8c09dbbc5bd>