

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

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。

2. 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。

3. 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。

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

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する。

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環境にインストールして使用した。

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。

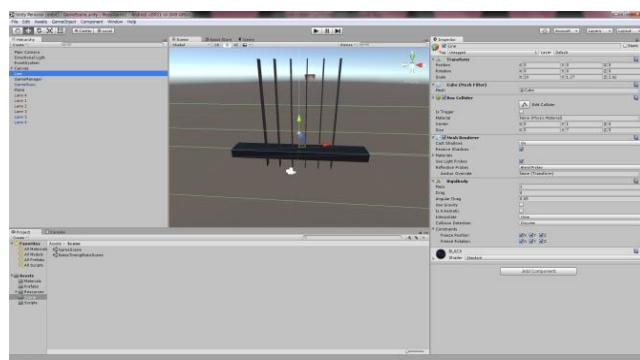


図 2. Unity の画面

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。

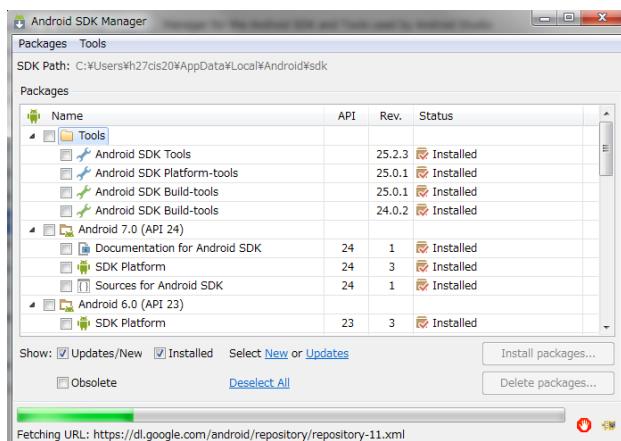


図 3. AndroidSDK マネージャーの画面

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。

4. 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 2 つのシーンを用意することにした。

4.2 ゲームシーンの作成

ここではゲームをプレイするために図 4 のようなシーンを作成した。

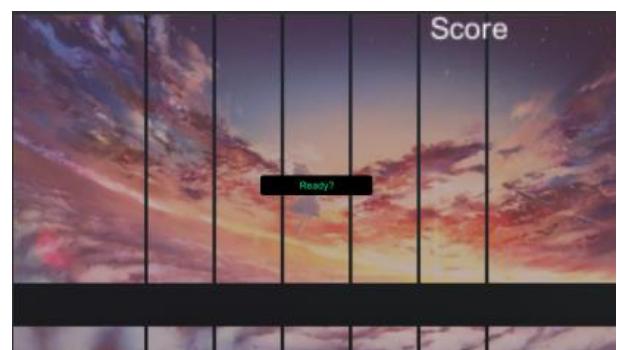


図 4. ゲームシーンの画面

このシーンで行う処理については次の通りである。

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を生成
4. オブジェクトが降ってくるようにする
5. タイミングよく押したらスコアを加算する

① 譜面ファイルを読み込む

譜面となるファイルはシーンとは別に作っておく必要がある。ファイルの形式としてはテキストファイルを使用するが、ただのテキストファイルではデータの区切りがわからず、編集も難しいと思い CSV 型のデータを使うことにした。CSV ではオブジェクトを生成するタイミングのデータとオブジェクトの落ちてくる位置のデータを 2 つ扱うようにした。

* CSV ファイルは comma-separated values の略で項目をカンマで区切って列举したテキストファイルでメモ帳や Excel で編集ができる。

	A	B
1	1.46	○
2	1.86	1
3	2.26	2
4	2.66	3
5	3.06	2
6	3.46	1
7	3.86	2
8	4.26	3
9	4.66	2
10	5.06	2
11	5.46	3
12	5.86	4
13	6.26	3
14	6.66	2
15	7.06	3
16	7.46	4
17	7.86	2
18	8.26	3
19	8.66	2
20	9.06	3
21	9.46	4
22	9.86	3
23	10.26	2
24	10.66	3
25	11.06	4
26	11.46	2
27	11.86	2
28	12.26	3
29	12.66	4
30	13.06	4
31	13.46	3

図 5. Excel で開いた CSV の

中身(左)とメモ帳で開いた CSV の中身(右)

この CSV ファイルをスクリプトにより上から一行ずつ読み込み、オブジェクトを生成するためのデータにしたため、作ったファイルは Unity のプロジェクトフォルダの中に入れておき、スクリプトで読み込めるようにした。

② BGM を再生

音楽ファイルは譜面ファイルと同様に、プロジェクトのファイルの中に入れておく必要がある。音楽ファイルに関しては拡張子の指定は特にない。(たとえば〇〇〇.mp3 や〇〇〇.wav 等)

図 6 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。

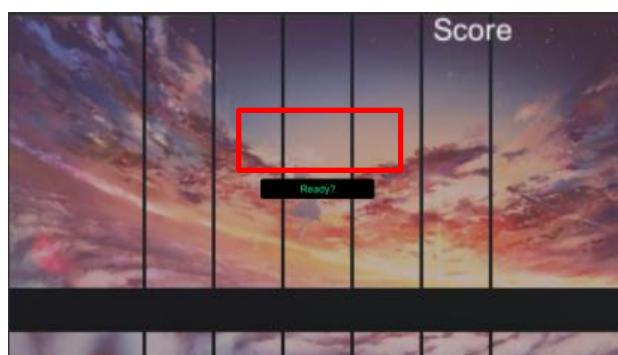


図 6. ボタンを押すと BGM が再生される

③ タイミングに合わせてオブジェクトを生成

図 5 の赤い枠線で囲まれた部分のデータを一つずつ読み込み、開始してからの時間と赤い枠線のデータが一致したとき緑の枠線のデータを読み込み、オブジェクト（白い四角のパーツ）を生成する。



図 7. プレイ中の画面

④ オブジェクトが降ってくるようにする

図 5 の緑の枠線は図 7 のオブジェクト（赤枠）が落ちてくる位置を示す。たとえば図 5において 2 という数字が読み込まれた場合、図 7 の赤い字の 2 の位置にオブジェクトが降ってくるような動きにしている。

⑤ タイミングよく押したらスコアを加算する

図 7 の黒い横長のライン（判定ライン）にオブジェクトが接触した時、そのオブジェクトをタッチすると図の右上の数字（ここでは 25 と表示している位置の数字）が変わる。接触していない時にタッチしてもスコアは変動しない。

4.3 譜面作成用のシーンを作成

ここでは譜面ファイルとなる CSV 型データを BGM に合わせてキーボード入力で作ることができます。

このシーンで行う処理については以下の通り。

1. BGM を流す
2. オブジェクト生成の時間とオブジェクトの降下位置を CSV に書き込む
3. 入力したキーによってオブジェクトの降下位置を変える



図 8 . 譜面作成シーンの画面



図 10 . 画面に表示されているテキスト

① BGM を流す

ゲームシーンと同様に Unity のプロジェクトフォルダの中に音楽ファイル（今回は既存のゲーム BGM を使用）を入れておく。

図 9 の画面の真ん中のボタンを押すと BGM が再生される。



図 9 . ボタンを押すと BGM が再生

③ 入力したキーによってオブジェクトの降下位置を変える

図 11 のように CSV の譜面ファイルが作られると中にはオブジェクト生成の時間（赤枠の部分）とキーに対応した数字（緑枠の部分）が書き込まれる。このキーに対応した数字がオブジェクトの降ってくる場所を示しており、例えばキーの F を入力した場合 CSV には 2 という数字が書き込まれる。

	A	B
1	1.398478	2
2		
3		
4		
5		
6		

図 11 . CSV ファイルの中身

② オブジェクト生成の時間と降下位置を CSV に書き込む

ボタンを押して BGM が再生されたら画面に表示されている文字をキーボードの該当するキーから入力してみる（図 10）。するとプロジェクトフォルダの中に譜面ファイル（CSV ファイル）が自動的に作られ、キーを入力した時間とそのキーに対応した数字が書き込まれる。

4.4 全体的な処理の流れ



① 譜面作成シーンを開く



② 真ん中のボタンを押す



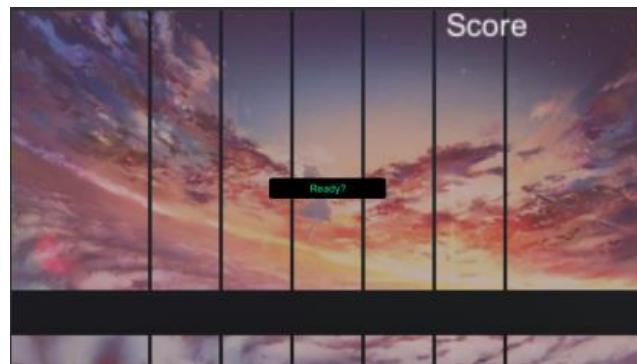
③ キーを入力



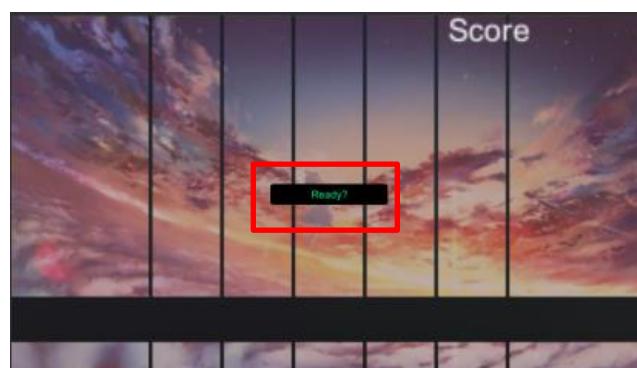
④ CSV ファイル(譜面ファイル)が作られる

	A	B
1	1.398478	2
2		
3		
4		
5		
6		

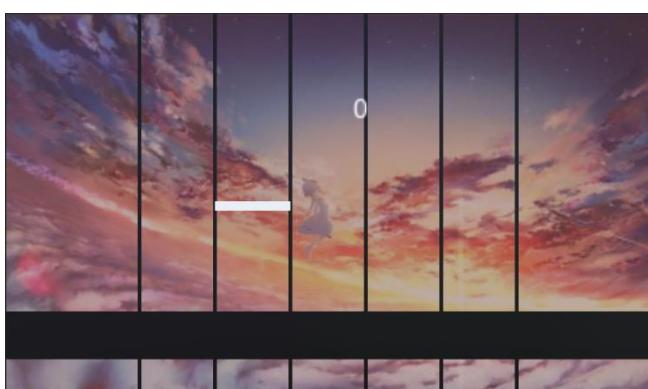
⑤ CSV の中身ができる



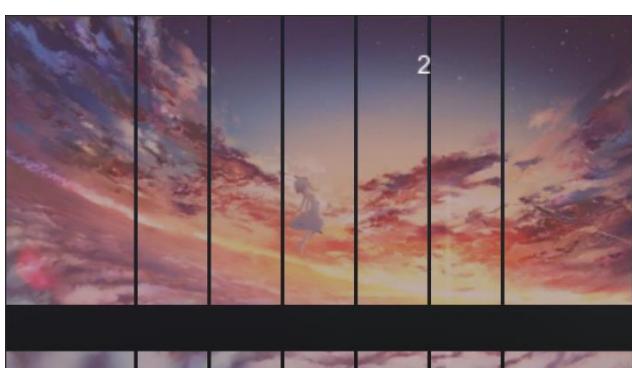
⑥ ゲームシーンを開く



⑦ 真ん中のボタンを押す



⑧ ゲーム開始



⑨ タイミングよく押すとスコア加算

4.5 現時点での改善点

- ゲーム画面のデザインの改善
操作の仕方が一目でわかりにくいため、ユーザーが一目で理解できるような画面を作る。
面白さが伝わるような画面を作る。
- ユーザーに理解しにくい機能の改善
譜面の作り方やプレイ中の操作等をユーザーが理解しやすいように作る。
チュートリアル（操作説明）シーンや readme ファイルを作る。
- Android のタッチパネルに対応した機能の追加
ゲームとして十分に遊べるようにする。
タッチパネルならではの操作を追加する。

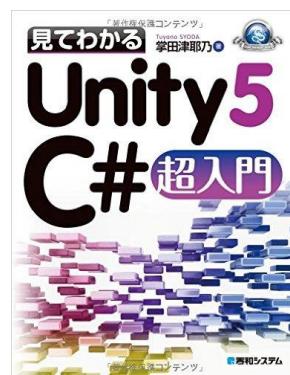
4.6 今後の作業予定

月	予定
1 月～1 月	機能の追加と改善
2 月	ゲーム画面のデザインの改善 アプリの完成
3 月	卒業論文 発表 産技短展

5. まとめ

音楽ゲームとして最低限の動きだけは出来ているがまだ改善の必要があり、誰にでも触ってもらえるようなゲームとしての完成には程遠い。今後の作業で完成、またはそれに近づけるようにし、必要であれば更に機能の追加をしたいと考えている。

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 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>