

マイコンとゲームエンジンによる組込システム

sorasora-sora3

2023年1月製作完了

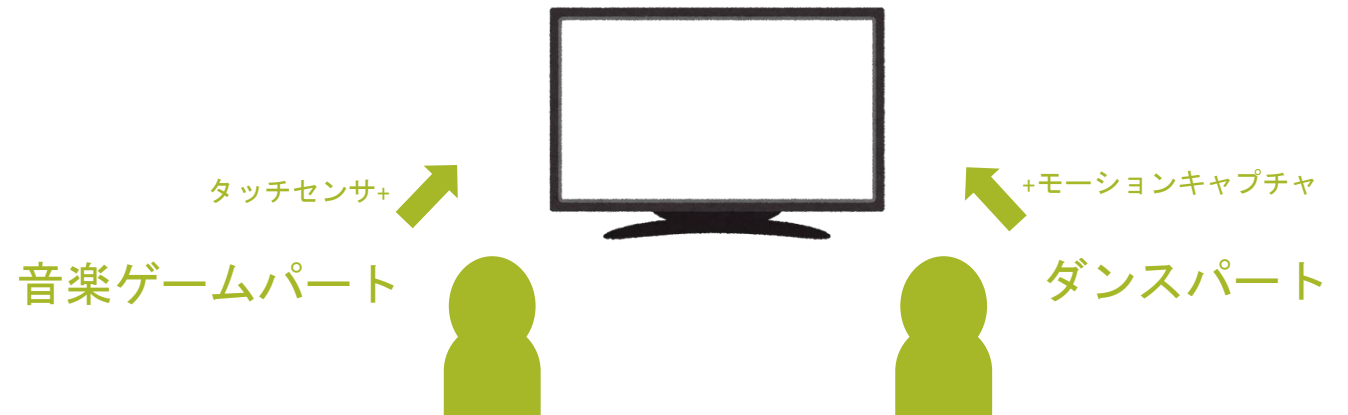
テーマ：MMDモデルを用いた協調型音楽ゲーム

コンセプト：

音楽ゲームパートとダンスパートに分かれる
新感覚な協調型音楽ゲーム

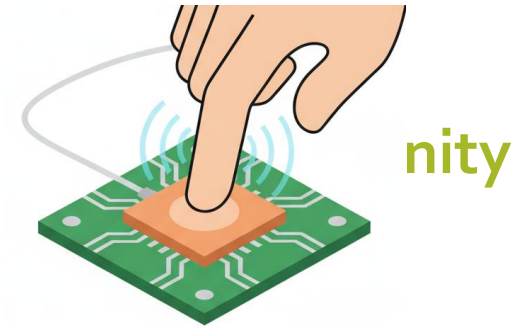
手本モデルとプレイヤーモデルにMMDモデルを適用

→システムとしての質を高め、没入感の向上を期待



製作条件について

- ArduinoとUnityを使用
- Arduinoによりタッチセンサからの入力を収集
内のノーツに関する処理に反映
- 各レーンに対応するタッチセンサに接触
→入力情報をシリアル通信でUnityに送信、ノーツを叩く動作を実現
- Unityによりオブジェクトの操作を容易に



開発の流れ

(3ヶ月間)

10月

Arduino、Unityにおける環境構築

11月

→Unity-Arduinoまわりの環境構築が上手くいかず製作が難航

→ノーツを叩くスクリプトの改良を根気強く行ったところ成功

12月

(1週間)

1週間

MMDモデルの導入、オリジナル譜面の作成、ダンストラッキングのためのmocopi導入、UI・グラフィックの改善を急ピッチで遂行

用いたArduinoのスケッチ

touch-sensor.ino

```
26 int pin2 = 2;
27 int pin3 = 3;
28 int pin4 = 4;
29 int pin5 = 5;
30
31 void setup() {
32   Serial.begin(9600);
33 }
34
35 void loop() {
36   int state2 = digitalRead(pin2);
37   int state3 = digitalRead(pin3);
38   int state4 = digitalRead(pin4);
39   int state5 = digitalRead(pin5);
40
41   if (state2 == HIGH) {
42     Serial.println(pin2);
43   }
44   else if (state3 == HIGH) {
45     Serial.println(pin3);
46   }
47   else if (state4 == HIGH) {
48     Serial.println(pin4);
49   }
50   else if (state5 == HIGH) {
51     Serial.println(pin5);
52   }
53   else
54   {
55     Serial.println(0);
56   }
57
58   delay(10);
59 }
60
```

4つのタッチセンサを使用！

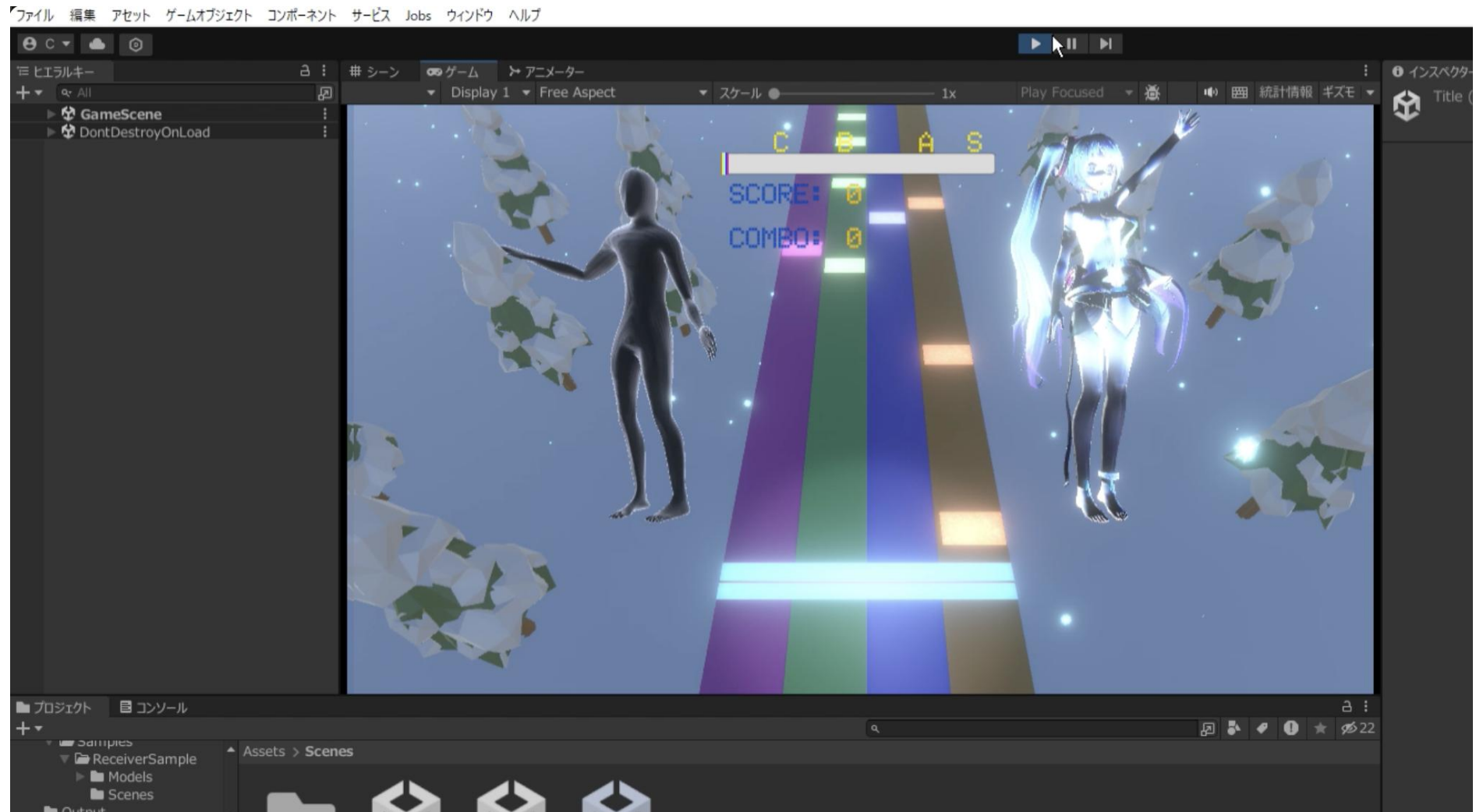
押すとピン番号が出力
→シリアル通信でUnityに伝達！

出力 シリアルモニタ ×

メッセージ ('COM4'のArduino Mega or Mega 2560にメッセージを送信するにはEnter)

3
3
3
3
3
3

踊る側



製作したものの対する価値

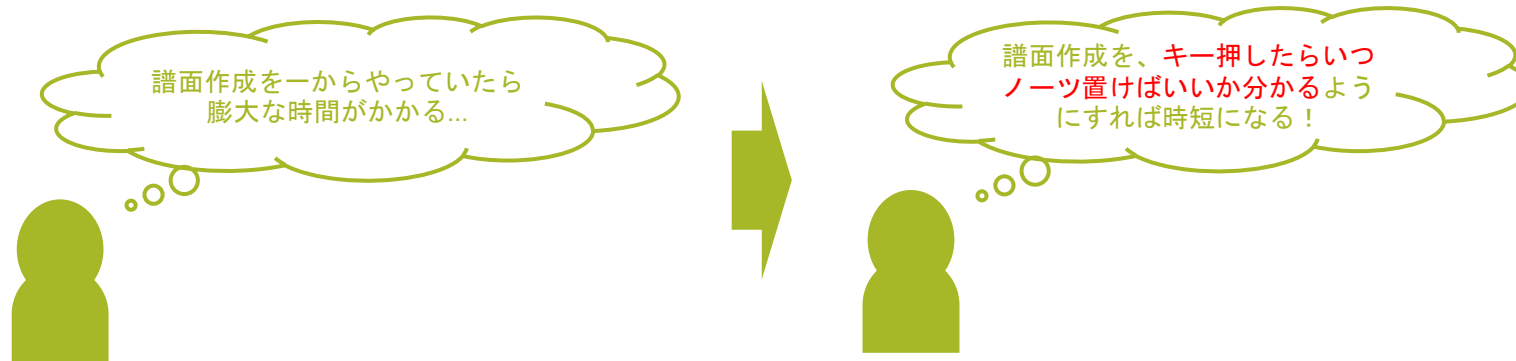
- 従来の音楽ゲームとは異なり、タッチセンサという、扱いやすい素子をコントローラとして適用
→老若男女問わずより**アクティブ**に音楽譜面に合わせてノーツを叩くことが可能
- mocopiの導入により現実空間内の自分を反映（アバタ）
→更なるゲームへの**没入感**向上の期待
- MMDのモデルの導入により楽曲に応じたモーション（ダンス）を本システム内に組込
→アバタと手本モデルが**一体感**を得られるような感覚を提供

没入しすぎている様子（余談）



特に工夫した点

- Unity内で、キーを押したときに
「押したキー」 & 「ゲームの経過時間」 & 「経過時間に対応する設定すべきノーツのz座標」をcsvに記録するスクリプトをその場で作成
- **雪**をイメージしたオブジェクトをパーティクルシステムにより再現
→グラフィックスの効果により**没入感**の向上



導入したかったこと

- mocopiによるアバターの動きも入れた高得点化

→ダンスパートの没入感の維持・向上が課題

- マルチで他の人も一緒に踊ったり観客として見れたりできるシステム

: 音楽ゲームを空間として多人数のユーザに提供すること



「プレイするゲーム体験」から「共に音楽ゲームを楽しむ場としての体験」



遊ぶ人・踊る人・見る人が一体となるライブイベントのような協調音楽ゲーム体験

音楽に応じてノーツが現れるように
キーボードで押下してタイミングを記録
→譜面作成を効率化

	A	B	C	D	E	F
0	H	4.070776	46.17093			
1	J	4.64162	51.87828			
2	F	12.95291	134.9754			
3	Key	ElapsedTi	CalculationResult			
4	Initial	0	5.4709			
5	green	58.67211	592.0806			
6	blue	59.42	599.558			
7	pink	60.16311	606.9877			
8	orange	60.85271	613.8824			
9	orange	61.61274	621.4813			
0	green	62.27122	628.0648			
1	pink	62.3768	629.1204			
2	blue	62.63799	631.7318			

↑ z座標=9.9981×経過時間+5.4709
* 傾きは経過時間とz座標の関係を調べて線形近似

ノーツ譜面の原案

●はノーツを表す→

Ah Whipしたケキのめいめい街は好き! 雷! 本気 Magic

空飛ぶくわん こんフェッティは 冬の魔法をう天使の羽よ Woo

白銀の国へ 出かけるの Go Go Go

雷の汽車が 街を駆け Choo Choo Choo

絵本みたいな 景色を 君に見せてあげるから

願っ込めて 吸気 土の(土) 鳴いて (ヤム!!!)

A B C D ドドレシ ラウ シシ マジカルミラクルクル Pop!

参考にしたサイトなど

- 「Unity C# 音ゲーの作り方 初心者でもスマホで遊べる音ゲーが作れる！」

<https://3dunity.org/game-create-lesson/music-game/>

- 「Sonyのモーションキャプチャー「mocopi」をつかってみた（前編）～Unityで動かすまで」

<https://note.com/thedesignium/n/n2c14e47b2acd#b15633df-d6f0-4570-b3e3-1f48be0a91d4>

- 「[Unity]MMDのモーションをUnityに入れて鑑賞会をしよう」

<https://qiita.com/saba383810/items/c55043bdbc47dd4cb6f4>

- 「mocopiをUnityで使ってみた（Windows10）」

<https://tatsuya1970.com/?p=17310>

- 「UnityとArduinoをシリアル通信」

<https://qiita.com/yjiro0403/items/54e9518b5624c0030531>

※楽曲としてMitchie M 様の「好き！雪！本気マジック feat. 初音ミク」のカラオケ・バージョンを使用、MMDモデルとしてTda 様の「Tda式初音ミク・アペンドVer1.10」を使用、自身のアバターとしてSONY 様の「RAYNOSちゃん Avatar」を使用、効果音としてOtoLogic 様の「Tambourine04-01(Hit-Hand).mp3」等を一部加工して使用しました。