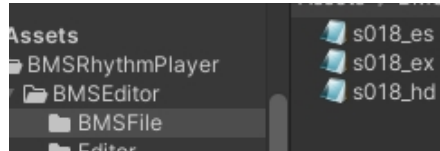


概要:

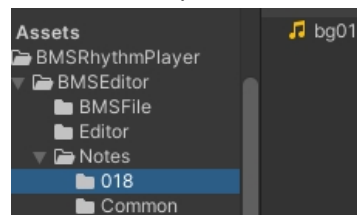
BMSRhythmPlayerは、iBMSCで書き出したBMSファイルをUnity上で再生したり、BMSファイル情報をassetデータに変換して保存できる拡張機能です。

Editor拡張機能を使って、BMSファイルをアセットデータに変換して使う。

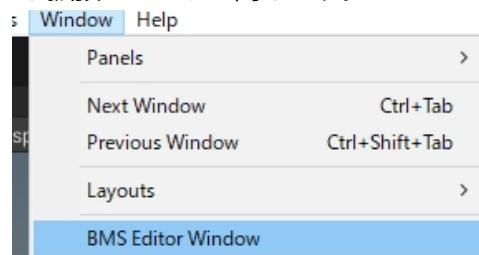
BMSFileフォルダにBMSファイルを配置します。



Notesフォルダに音源ファイルなどを配置します。(例Notes/018/018bgm.ogg)



Tool -> BMS Editor Window から拡張ウィンドウ開きます。



BMS File List: BMSファイルの名前が表示されます。

Reference Note Folder: 音源ファイル参考フォルダを指定します。

Asset Name: アセットデータとして保存する時に使う名前を指定します。

Rhythm Duration: 時間長さを指定します。0の場合、自動的に、最後のキーボタンに合わせて最適化した時間の長さを指定します。

(時間の長さが合わない場合手動で調整してください。)

参考図: 変更前

BMS File List	Reference Note Folder	Asset Name	Rhythm Duration
<input type="checkbox"/> s018_es	Folder : s018_es	s018_es	Time : 0
<input type="checkbox"/> s018_ex	Folder : s018_es	s018_ex	Time : 0

参考図: 変更後

BMS File List	Reference Note Folder	Asset Name	Rhythm Duration
<input checked="" type="checkbox"/> s018_es	Folder : 018	s018_es	Time : 145
<input checked="" type="checkbox"/> s018_ex	Folder : 018	s018_ex	Time : 0
<input checked="" type="checkbox"/> s018_hd	Folder : 018	s018_hd	Time : 0

Track Settingsパネルを選択して軌道設定します。

[+ Add Track]と[- Delete Track]を使って軌道を追加、削除します。

Key: 決まった軌道上で移動させたい、キー番号を選択します。

キー番号はiBMSのA1～B15までの番号です。(B1～B15はBGM軌道)

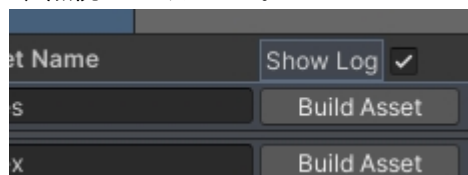
SF (SimultaneouslyFalling) :

チェック入れたキー番号のみの間で、同時に落ちるキーをチェック処理します。

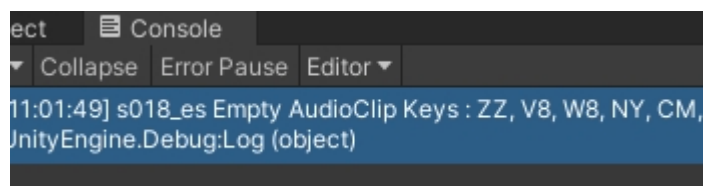


Show Log: アセットビルドが終わった後、見つからない音源ファイルのキーを出力します。

音源ファイル対応しない時は、無視してください。



Log表示参考図:



Save Settings: ビルド設定、軌道設定が終わったら、現在の設定を保持したい場合、

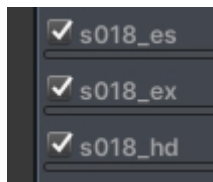
[Save Settings]ボタンを押して保存します。

すると、BMSEditorPlayer/Resources/BMSEditorReference.jsonが作成されます。

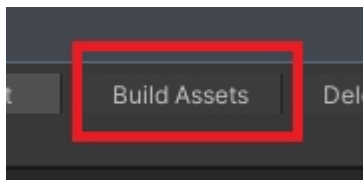
アセット個別にビルド: [Build Asset]ボタンを押します。



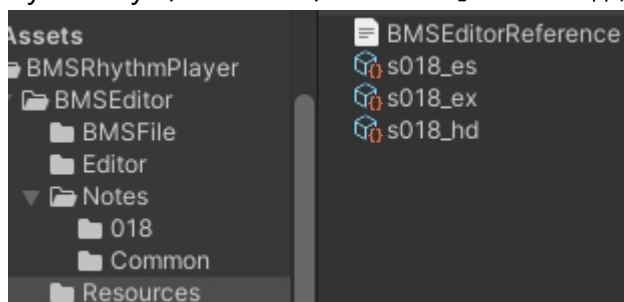
アセット複数同時にビルド：ビルドしたいBMSにチェックを入れます。



[Build Assets]ボタンを押します。



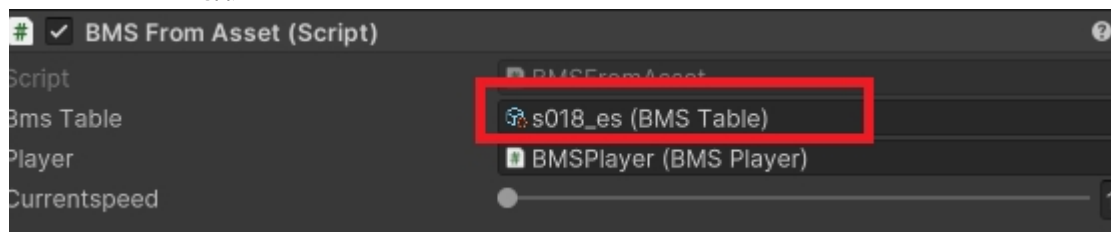
アセットデータは「BMSRhythmPlayer/BMSEditor/Resources」のしたに保存されます。



ビルドしたアセットを参照して、リズム再生する

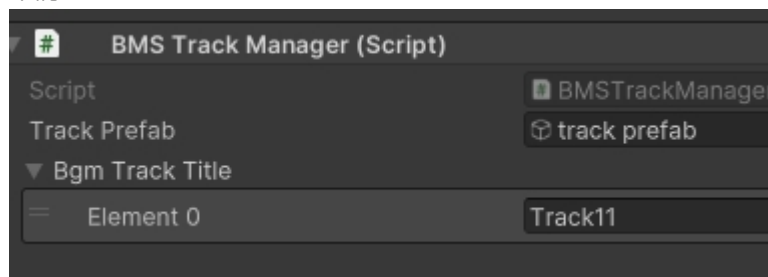
Demo BMSFromAssetを使います。

Bms Table：アセット指定します。



Bgm Track Title：BGM軌道タイトル名指定します。

終わたらプレイして確認できます。



アセットデータClass BMSTableを参照操作する。

参照:

キーの合計数 (Longキーのみ2回にカウントします。)

```
public int GetNoteCount
```

基準スピード

```
public float GetBasicNoteSpeed
```

リズム時間長さ

```
public float GetRhythmDuration
```

BPM軌道情報リストをかえします。変速 (ソフラン) 機能に使用します。

```
public List GetBpmTrackInfo()
```

BGM軌道含む全ての軌道情報パッケージリストを返します。(BPM軌道をのぞく)

```
public List GetTrackPackage()
```

変速 (ソフラン) 機能にこのメソッドが使用します。変速なくても使用します。

index: 0から始まるBPM軌道リスト参照用index、次のindexを返します。

time: フレーム毎の時間

totalTime: 開始してからの時間

basicVariableSpeed: 現在の基準速度

```
public BPMVariableInfo GetBPMVariableSpeedInfo(int index, float time, float totalTime, float basicVariableSpeed)
```

軌道を作成する: DemoのBMSTrackManagerクラスを参考してください。

BGM軌道はキーボタンの作成、表示、位置設定がないため、個別で処理しています。

キーボタン作成する: DemoのBMSTrackクラスを参考してください。

同時に落ちるキー作成: simultaneouslyFallがtrueの時に作成します。

```
if(_track.noteInfo.keyType == KEY_TYPE.Short_Note)
{
    if (_track.noteInfo.simultaneouslyFall)
        noteOb = GameObject.Instantiate(shortRedPrefab);
}
```

ShortNoteクラス参照: 普通のキーボタン

生成座標指定:

```
var notePoint = (info.noteInfo.point[0] - totalDistance) * gearSpeed;
transform.localPosition = new Vector3(transform.localPosition.x, notePoint, 0f);
```

LongNoteクラス参照: 長いキーボタン

生成座標指定:

```
var notePoint = (info.noteInfo.point[0] - totalDistance) * gearSpeed;  
transform.localPosition = new Vector3(transform.localPosition.x, notePoint, 0f);
```

長さ指定:

```
var width = info.noteInfo.noteLong * gearSpeed;  
  
widthRect.sizeDelta = new Vector2(width, widthRect.rect.height);
```

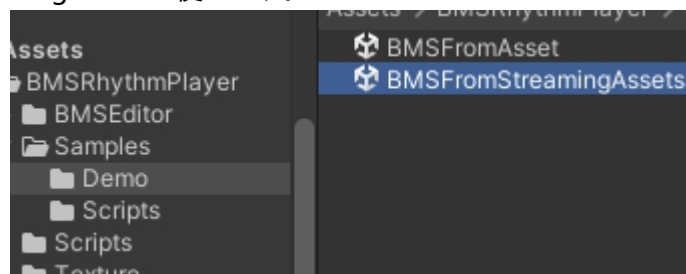
キーの移動距離: DemoのBMSPlayerクラスを参考してください。

BMSTable の GetBPMVariableSpeedInfo() メソッド使って移動距離と基準速度を取得します。

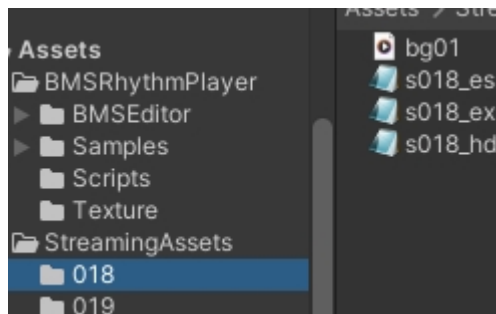
```
var f_time = Time.deltaTime;  
totaltime += f_time;  
  
var variableInfo = table.GetBPMVariableSpeedInfo(variableIndex, f_time, totaltime, basicVariableSpeed);  
variableIndex = variableInfo.index;  
basicVariableSpeed = variableInfo.basicVariableSpeed;  
totalBasicDistance += variableInfo.distance;  
var moveDistance = variableInfo.distance * gearSpeed;  
  
trackmanager.Run(generateTimer, f_time, totaltime, gearSpeed, moveDistance, totalBasicDistance);
```

BMSファイルをStreamingAssetsから参照して、リズム再生する。

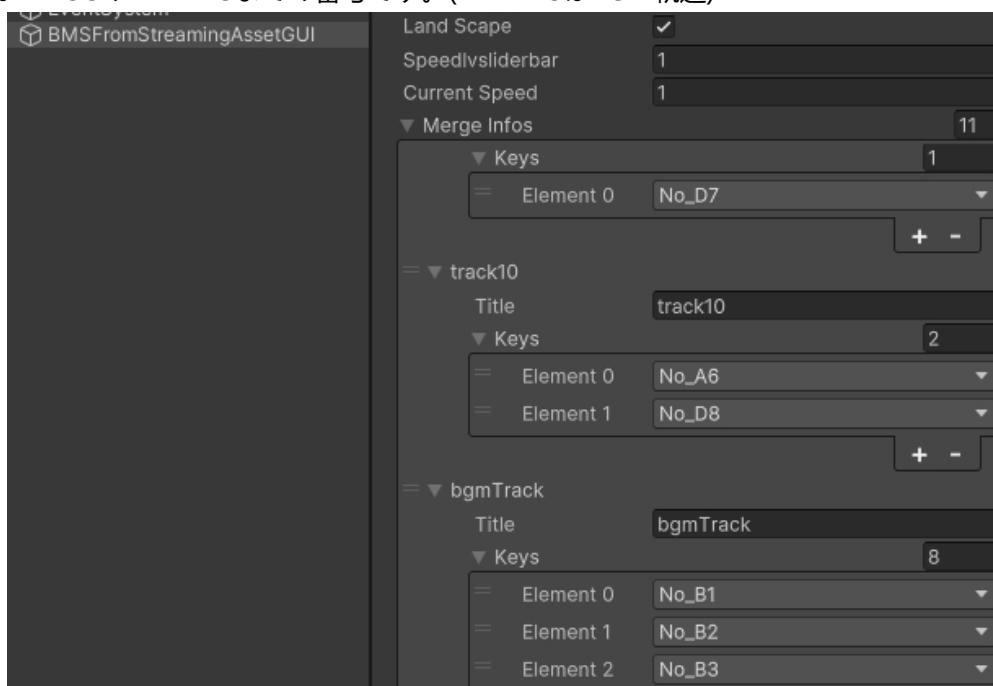
Demo BMSFromStreamingAssetsを使います。



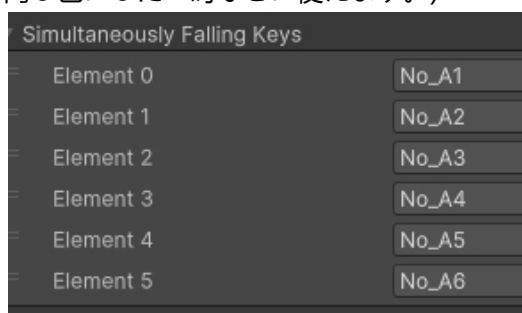
図を参考にStreamingAssetsの下にbmsファイルを配置します。



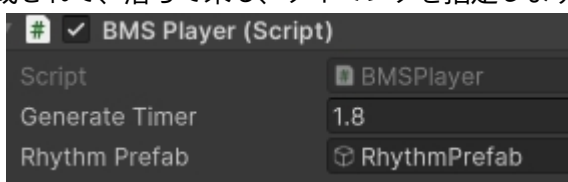
BMSFromStreamingAssetGUIから、MergeInfosを編集して、軌道を作成します。
 軌道titleと、軌道上で移動させたい、キー番号を入れます。
 キー番号はiBMSCのA1～B15までの番号です。(B1～B15はBGM軌道)



Simultaneously Falling Keys: 同時に落ちるキーを処理してくれます。
 (例えば: 同時に落ちるキーを同じ色にしたい時などに使えます。)

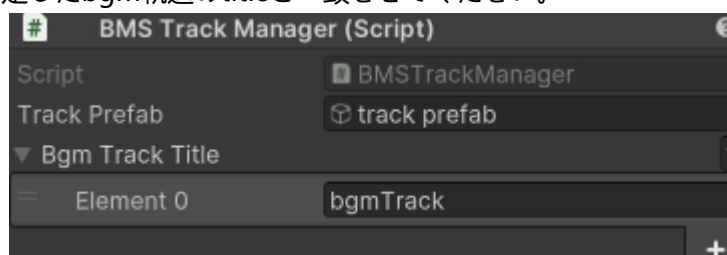


GenerateTimer: キーが生成されて、落ちて来る、タイミングを指定します。

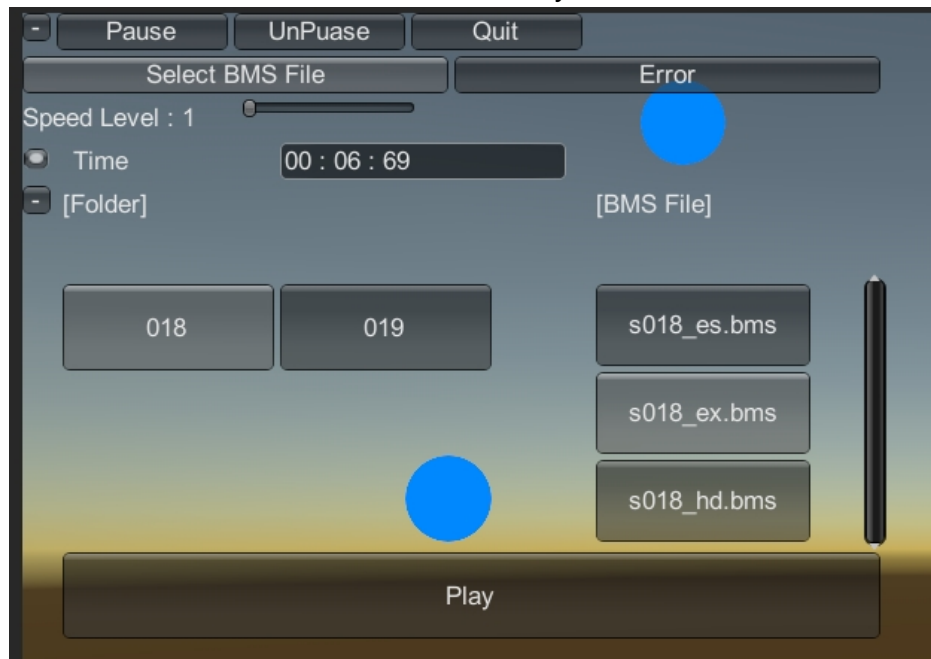


Bgm Track Title: bgm軌道のtitleを指定します。

MergeInfosで指定したbgm軌道のtitleと一致させてください。



StreamingAssetsに保存したbmsファイルを選択して、Playボタンクリックして再生します。



Error: これらのキー音は、音源ファイルが指定されていない、もしくは、音源ファイルみつからない時に表示されます。(音源ファイル使っていなかったら、無視してください。)

Empty AudioClip Keys: ZZ, UP...

