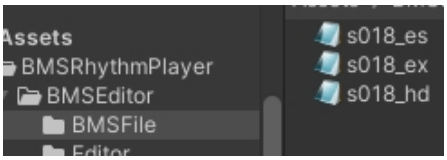
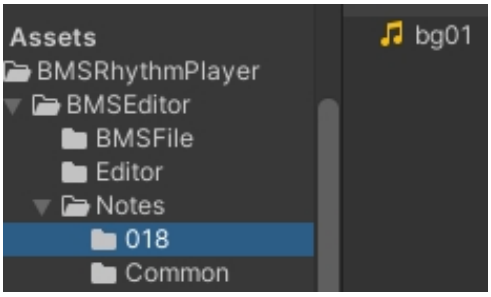
Summary:

BMSRhythmPlayer is an extension tool that allows you to play BMS files exported from iBMSC in Unity, and convert BMS file information into Asset data for storage

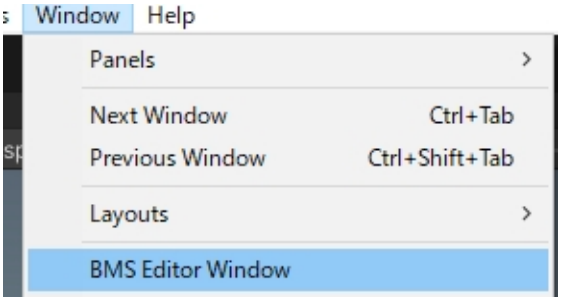
Use the Editor extension to convert BMS files into Asset data.
Place BMS files inside the **BMSFile** folder



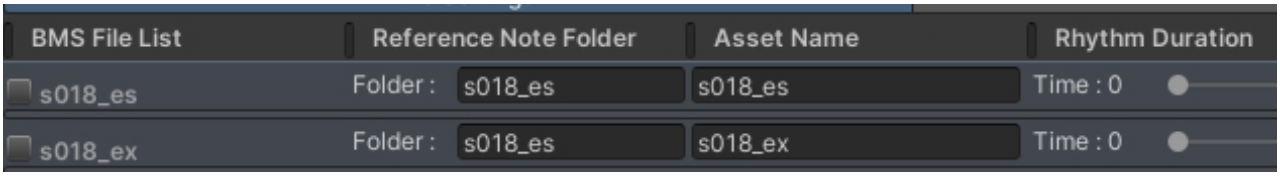
Place audio files and other resources inside the **Notes** folder (for example: Notes/018/018bgm.ogg).



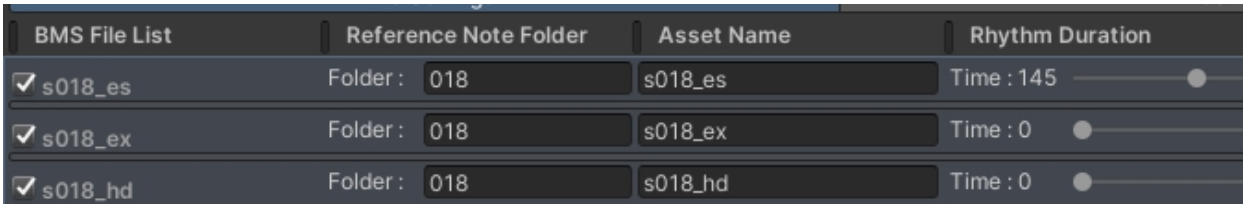
Open the extension window from Window - > BMS Editor Window



BMS File List：Displays the names of the BMS files
Reference Note Folder：Specifies the folder containing the audio files
Asset Name：Specifies the name used when saving as Asset data.
Rhythm Duration：Specifies the rhythm duration. If set to **0**, the system automatically calculates the optimal duration based on the last note. (If the duration does not match, please adjust manually.)
Reference：Before change



Reference：After the change



Select the **Track Settings** panel to configure tracks.

Use **[+ Add Track]** and **[- Delete Track]** to add or remove tracks.

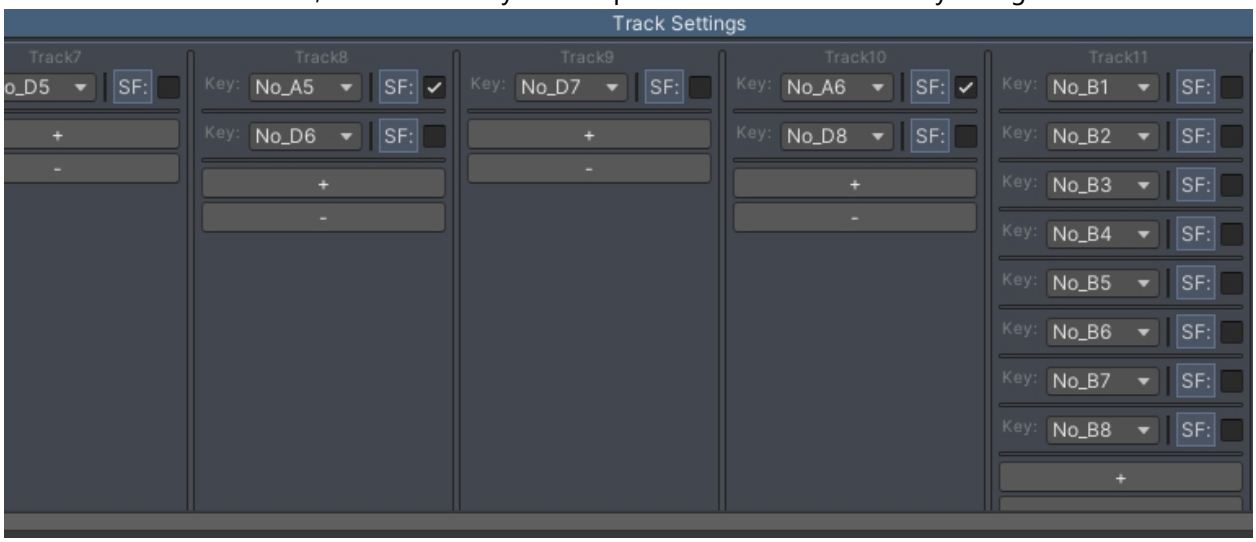
Key:

Select the key number you want to assign to the track.

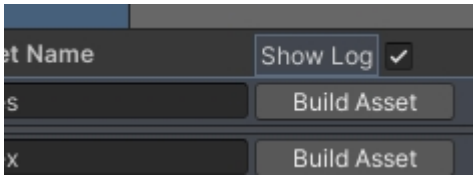
Key numbers correspond to iBMSC's A1–B15. (B1–B15 are BGM tracks.)

SF (Simultaneously Falling)

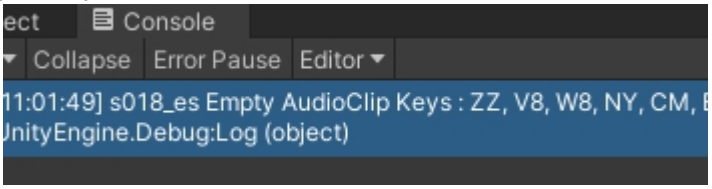
When enabled, the selected keys will be processed as "simultaneously falling."



Show Log：After building the Asset, missing audio keys will be displayed. If the keys do not use audio files, you may ignore them



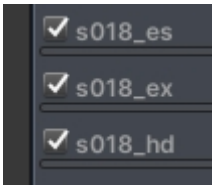
Sample diagram for Log display：



Save Settings：After completing the build settings and track settings,
click **[Save Settings]** to save the current configuration.
This will generate:BMSEditorPlayer/Resources/BMSEditorReference.json.

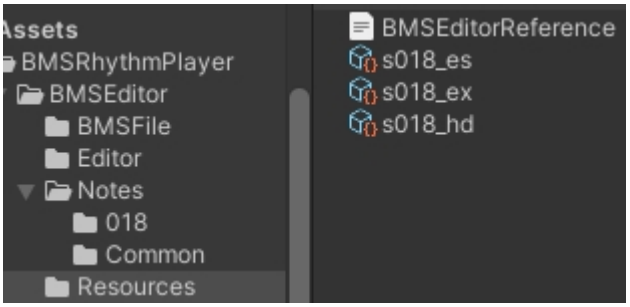
Build a single Asset： Click **[Build Asset]**.

Build multiple Assets： Check the BMS files you want to build → click **[Build Assets]**.

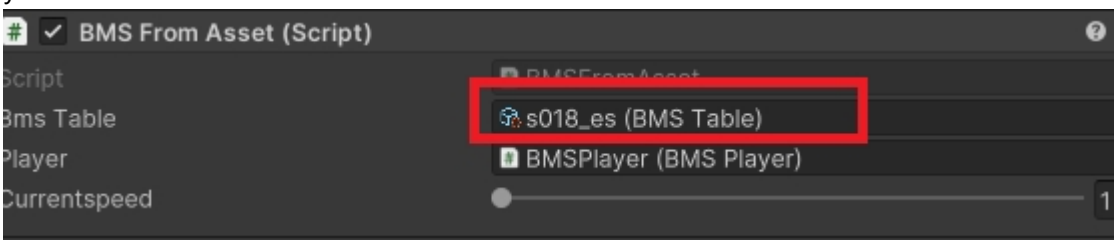


↓

The generated Asset data will be saved to：「BMSRhythmPlayer/BMSEditor/Resources」.

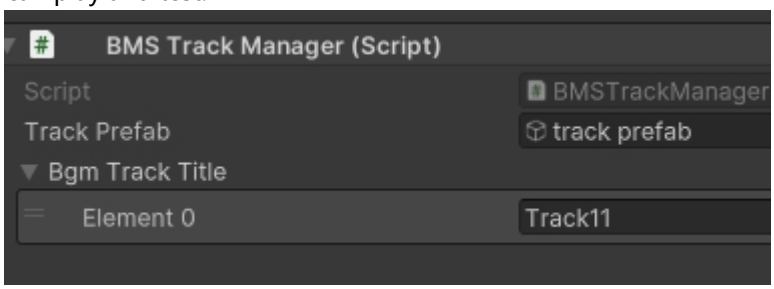


Using Built Assets for Playback
Use the Demo BMSFromAsset.
Specify the Asset in the Bms Table.



↓

Bgm Track Title：Specify the BGM track title.
After setting up, you can play and test.



BMSTable Class Reference

Get total number of notes (Long Notes count as two)

public int GetNoteCount

Get basic note speed

public float GetBasicNoteSpeed

Get rhythm duration

public float GetRhythmDuration

Get BPM track information (used for variable speed)

public List GetBpmTrackInfo()

Get all track packages (includes BGM, excludes BPM track)

public List GetTrackPackage()

Get variable speed information (used per frame)

Parameters:

- **index**: BPM track list index starting from 0; returns the next index
- **time**: delta time per frame
- **totalTime**: total elapsed time
- **basicVariableSpeed**: current base speed

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

Refer to the **BMSTrackManager** class in the demo.

Since the BGM track does not generate or display notes, it must be handled separately.

Generating Notes

Refer to the **BMSTrack** class in the demo.

When simultaneouslyFall is true, simultaneously falling notes will be generated.

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

ShortNote (short notes)

Position generation:

```
var notePoint = (info.noteInfo.point[0] - totalDistance) * gearSpeed;

transform.localPosition = new Vector3(transform.localPosition.x, notePoint, 0f);
```

LongNote (long notes)

Position generation is the same as ShortNote.

Length calculation:

```
var width = info.noteInfo.noteLong * gearSpeed;

widthRect.sizeDelta = new Vector2(width, widthRect.rect.height);
```

Movement distance

Refer to **BMSPlayer**.

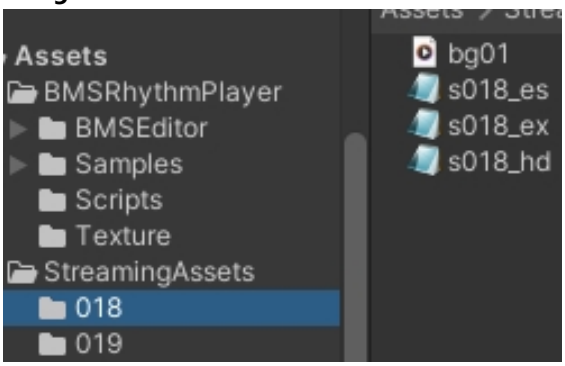
```
var f_time = Time.deltaTime;
totaltime += f_time;

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);
```

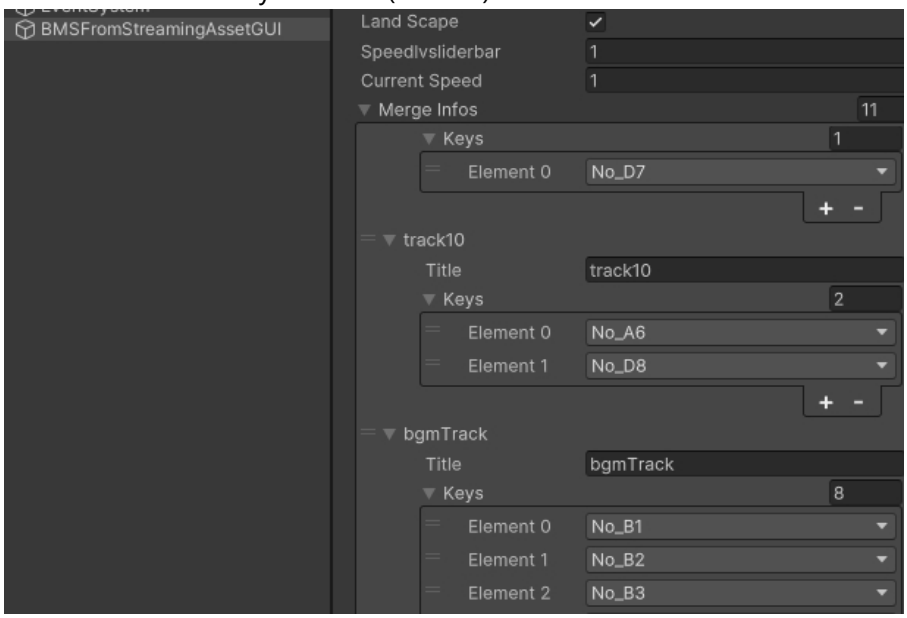
Playing BMS from StreamingAssets

Use the demo **BMSFromStreamingAssets**. Place BMS files under



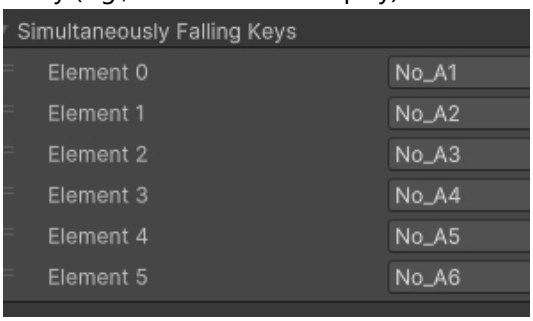
MergeInfos

Edit track titles and key numbers (A1–B15).



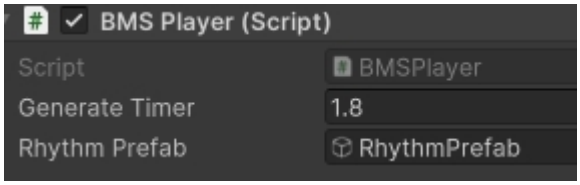
Simultaneously Falling Keys

Handles keys that fall simultaneously (e.g., for same-color display).



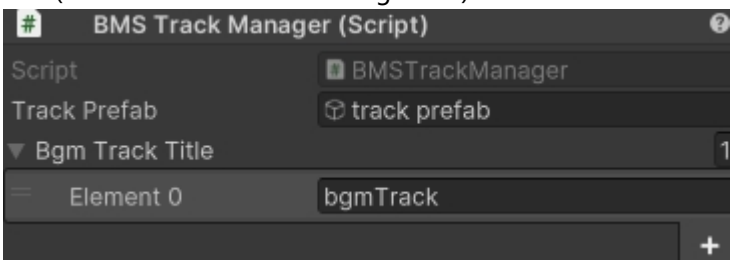
GenerateTimer

Specifies the timing when notes are generated and start falling



Bgm Track Title

Specify the BGM track title (must match the title in MergeInfos).



Playing BMS from StreamingAssets

Select a BMS file → click **Play**

Error:

Displayed when audio files are not specified or cannot be found.

If the key does not use audio, you may ignore it.

Example:

Empty AudioClip Keys: ZZ, UP...