

BGK - Bence's Game Kit

Modules:

- Save System
- Audio
- Utility

Namespaces:

- BGK: *Main namespace*
 - SaveSystem: *Save system scripts*
 - Serialization: *Custom serialization classes*
 - Audio: *Audio scripts*
 - Utility: *Utility functions*
 - Demo: *Demo contents*

Save System

Namespace: BGK.SaveSystem

Main features:

- Manage multiple save type.
- Manage multiple save file.
- Manage multiple save location.

Save methods:

1. Save a single variable or list:

1. Import the Save System: using BGK.SaveSystem;
2. Save the variable or list using one of the following commands:

```
SaveSys.Save(<variable>, <name of file>);  
SaveSys.Save(<variable>, <path of file>, <name of file>);  
SaveSys.Save(<variable>, <custom SerializationSurrogate[]>, <name of file>);  
SaveSys.Save(<variable>, <custom SerializationSurrogate[]>, <path of file>,  
<name of file>);
```

2. Save using a class:

1. Import the Save System: using BGK.SaveSystem;
2. Create a class:

First option:

```
[System.Serializable]  
public class <name>  
{  
    public <variables>  
}
```

//Then use as a variable in another script

Second option:

```
[System.Serializable]
public class <name>
{
    public <variables>

    public void <function name>(<import script>)
    {
        <variables> = <import script>.<variables>
    }
}
```

3. Save the class one of the following commands:

```
SaveSys.Save(<class>, <name of file>);
SaveSys.Save(<class >, <path of file>, <name of file>);
SaveSys.Save(<class >, <custom SerializationSurrogate[]>, <name of file>);
SaveSys.Save(<class >, <custom SerializationSurrogate[]>, <path of file>, <name
of file>);
```

Load methods:

The load method must be the same as the save method.

All methods are available with return or output values (always first value). For example:

- <type> <var name> = SaveSys.Load(<name of file>) as <type>;
- SaveSys.Load(out <type> <var name> as <type> (From now on, refer to it as <out>), <name of file>);

If you used a class in the save, use it as a variable.

1. Import the Save System: using BGK.SaveSystem;
2. Load the variable or list using one of the following commands:
 - SaveSys.Load(<out>, <name of file>);
 - SaveSys.Load(<out>, <path of file>, <name of file>);
 - SaveSys.Load(<out>, <custom SerializationSurrogate[]>, <name of file>);
 - SaveSys.Load(<out>, <custom SerializationSurrogate[]>, <path of file>, <name of file>);

Other functions:

- GetSaveFileList(): *return the list of all save file locations.*
- DeleteSaveFile(<path>, <name of file>): *delete save file.*
- DeleteAllSaveFile(): *delete all save file.*
- CheckSaveFile(<path>, <name of file>): *check if save file exist.*

Audio

Namespace: BGK.Audio

Main features:

- Automatic soundtrack and ambient playback
- Sort sounds to 3 different category
- Easy sound library management

Setup:

1. Create an Audio Library
 1. Go to the project folder
 2. Right click
 3. Select Create > BGK > Audio > AudioLibrary
2. Add sounds to the Audio Library
 1. Select the Audio Library
 2. Add each audio file to the Audios array
 1. Add a new element to the array
 2. Give it a name (Used later to play the sound)
 3. Choose the Audio Clip
 4. Adjust the volume
 5. Select the type of sound (Music, Ambient, Effect)
 6. Adjust the other variables (Recommended: (Pitch: 1; Space: 0))
 3. (Optional) Turn off Auto Generate Soundtrack and Ambient and define it yourself
3. Create an Audio Mixer with one master and 3 subgroups (Ambient, Music, Effect)
See more: <https://docs.unity3d.com/Manual/AudioMixer.html>
4. Setup GameObject
 1. Create a new empty GameObject in the scene
 2. Add an Audio Manager Script
 3. Set up the variables
 4. *Keep in mind that this GameObject will not be destroyed when switching scenes. This means that if it's on Scene A and you switch to Scene B, even if you didn't put the GameObject on Scene B, it will be transferred to Scene B with exactly the same data as it had on Scene A.*

Usage:

Soundtrack: Audio Manager automatically plays the soundtrack playlist you have set.

Ambient: Audio Manager automatically plays the ambient playlist you have set.

You can start any sound in the library at any time with one of the following commands:

- `AudioManager.instance.Play(<name of the audio>, <parent GameObject>, <Continue after scene change?>);`
- `AudioManager.instance.Play(<name of the audio>, <Continue after scene change?>);`
- `AudioManager.instance.Play(<name of the audio>, <parent GameObject>);`
- `AudioManager.instance.Play(<name of the audio>);`

Utility Functions

Namespace: BGK.Utility

List of functions:

- `ObjectManagement.FindObjectWithInstanceID(<InstanceID>)`: *Find scene object or asset with matching Instance ID*
- `ArrayManagement.Add(<array>, <element>)`: *Add element to array*
- `ArrayManagement.Replace(<array>, <old element>, <new element>)`: *Replace the first matching old element with a new element*
- `ArrayManagement.ReplaceAll(<array>, <old element>, <new element>)`: *Replace all matching old element with a new element*
- `ArrayManagement.Remove(<array>, <elemet>)`: *Remove the first matching element*
- `ArrayManagement.RemoveAll(<array>, <elemet>)`: *Remove all matching element*
- `ArrayManagement.ConvertToArrayArray(<multidimensional array>)`: *Convert multidimensional array to an array of array*
- `ArrayManagement.ConvertToMultidimensional(<array of array>, <Skip empty?>)`: *Convert an array of array to a multidimensional array*
- `Sort.BubbleSort(<array>, <order>)`: *Sort array*
- `Sort.InsertionSort(<array>, <order>)`: *Sort array*
- `Sort.SelectionSort(<array>, <order>)`: *Sort array*
- `SearchSorted.BinarySearch(<array>, <value>, <order of sorting>)`: *Search a value in a sorted array*
- `SearchUnsorted.LinearSearch(<array>, <value>)`: *Search a value in any array*