

# Bump&Scrape

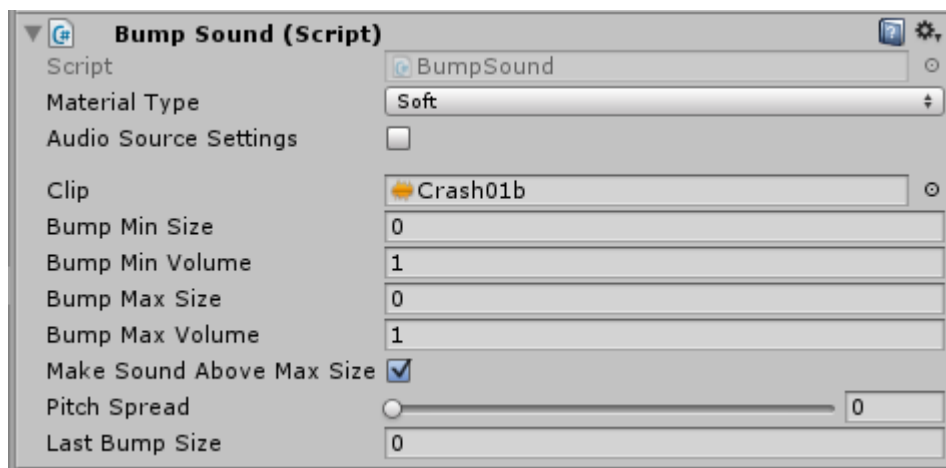
Bump&Scrape is a small but useful library of scripts handling bumping or scraping sounds when a 3D object collides with or slides across another.

All scripts are in c#, well documented and easy to use and understand.

## BumpSound

A script/class playing a one-timed bump sound when its object collides with another.

The user can set the clip to be played and how should the volume depend on the collision strength.



## Parameters

**MaterialType** – Types of materials it should react to – see SoundMaster

**AudioSourceSettings** – opens settings for created audiosource (Output, Priority, StereoPan, SpatialBlend, ReverbZoneMix, RolloffMode, MinDistance, MaxDistance) and sets them over the default. For exact meaning of these settings please check Unity documentation.

**Clip** – rather selfexplanatory, the sound clip to be played in time of collision

**BumpMinSize** – The minimum impact which will cause sound to be played. The value is the collision impulse (see unity documentation) divided by mass of the object. That means that the value shouldn't depend on the mass (light or heavy box). This should make configuration rather simple and in 99% of cases you should be just fine with the default (or your favorite) values.

**BumpMinVolume** – the sound volume used for **BumpMinSize** sized impact

**BumpMaxSize** – impact at which the sound reaches the full volume (**BumpMaxVolume**)

**BumpMaxVolume** – The full volume of the sound

**MakeSoundAboveMaxSize** – If true, the sound is played even at collisions stronger than **BumpMaxSize**. The volume is clipped at the **BumpMaxVolume** value.

If false, no sound is played. That is useful in case you want to “stack” the scripts and have different sound play at different contact sizes (e.g. one clip for small contacts and one for big ones).

**PitchSpread** – pitch multiplier variation. Value of 0.2 will cause the pitch (play speed) to be randomly value between 0.8 and 1.2 times faster than the clip’s standard value.

**LastBumpSize** – not a parameter for you to set. This value shows the size (severity) of the last impact. This means that should you be unsure about what values to use, you simply drop or bump into the object in the Unity Editor and have a look what value shows up. This should make fine-tuning the values rather easy.

## Usage

The usage is very easy. You drop the script onto the object, preferably an object with a Rigidbody. If the script can’t find a Rigidbody it will try to get collision impulse value from the other colliding Rigidbody. If that fails it can’t establish contact severity and a default value is used.

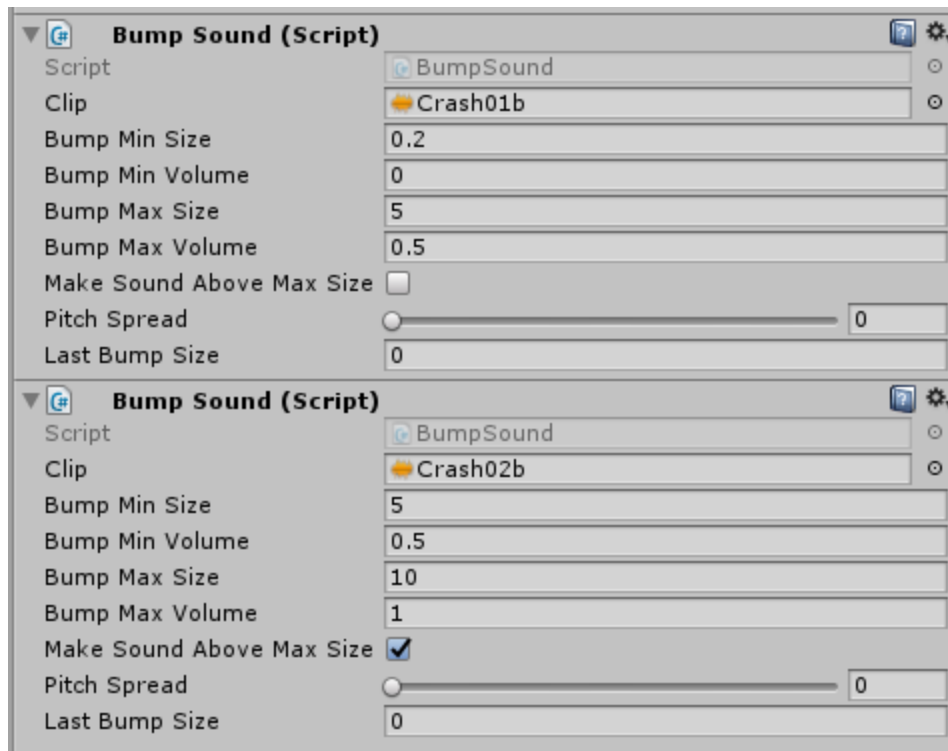
The default setup is that even the smallest contact will cause full volume sound. This is a robust and useful if not very sophisticated option.

If you want more control. You can tweak the parameters to serve your needs better. The parameters are described in previous chapter.

For example, the script in previous picture is setup to make sounds at collisions with size 0.2 and above. The full volume is reached at and above the value of 10. The collisions between those values have the volume interpolated.

**Tip 1:** Don’t forget to use the **LastBumpSize** value. Do the hit you want to make a sound in the editor and check the **LastBumpSize** value. Adjust the **BumpMinSize** and **BumpMaxSize** accordingly.

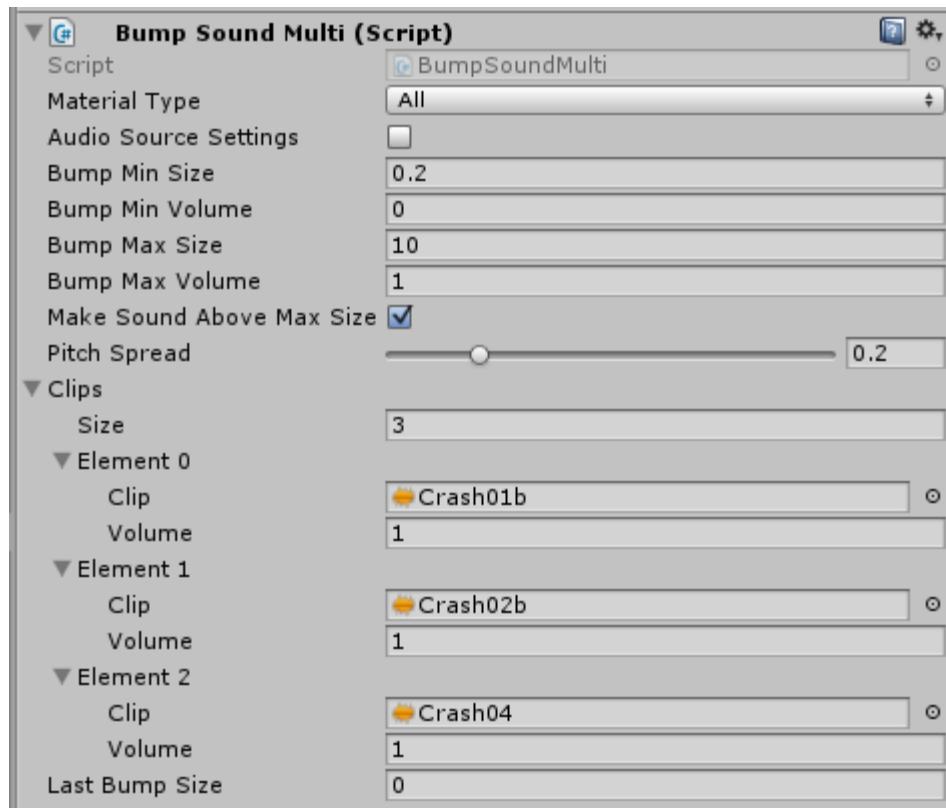
**Tip 2:** Use the stacking capability. Use more than one script per object to sound different clips depending on the strength of the collision. Following picture shows such a setup. Note that the first script setup to sound at lower contact sizes has **MakeSoundAboveMaxSize** parameter false so that it doesn’t sound above the contact size 5 together with the second script.



## BumpSoundMulti

Another script/class playing a one-timed bump sound when its object collides with another. This one is an extension of the BumpSound class. The user can set several different clips. At each collision one of them will be selected at random.

The user can set the clip to be played and how should the volume depend on the collision strength.



## Parameters

**MaterialType** – Types of materials it should react to – see SoundMaster

**AudioSourceSettings** – opens settings for created audiosource (Output, Priority, StereoPan, SpatialBlend, ReverbZoneMix, RolloffMode, MinDistance, MaxDistance) and sets them over the default. For exact meaning of these settings please check Unity documentation.

**BumpMinSize** – The minimum impact which will cause sound to be played. The value is the collision impulse (see unity documentation) divided by mass of the object. That means that the value shouldn't depend on the mass (light or heavy box). This should make configuration rather simple and in 99% of cases you should be just fine with the default (or your favorite) values.

**BumpMinVolume** – the sound volume used for **BumpMinSize** sized impact

**BumpMaxSize** – impact at which the sound reaches the full volume (**BumpMaxVolume**)

**BumpMaxVolume** – The full volume of the sound

**MakeSoundAboveMaxSize** – If true, the sound is played even at collisions stronger than **BumpMaxSize**. The volume is clipped at the **BumpMaxVolume** value.

If false, no sound is played. That is useful in case you want to “stack” the scripts and have different sound play at different contact sizes (e.g. one clip for small contacts and one for big ones).

**PitchSpread** – pitch multiplier variation. Value of 0.2 will cause the pitch (play speed) to be randomly value between 0.8 and 1.2 times faster than the clip's standard value.

**Clips** – an array of clips with their specific volume multipliers among which the script selects when collision occurs.

**Clip** – rather selfexplanatory, the sound clip to be played in time of collision

**Volume** – Clip’s specific volume multiplier

**LastBumpSize** – not a parameter for you to set. This value shows the size (severity) of the last impact. This means that should you be unsure about what values to use, you simply drop or bump into the object in the Unity Editor and have a look what value shows up. This should make fine-tuning the values rather easy.

## Usage

The usage is very easy. You drop the script onto the object, preferably an object with a Rigidbody. If the script can’t find a Rigidbody it will try to get collision impulse value from the other colliding Rigidbody. If that fails it can’t establish contact severity and a default value is used.

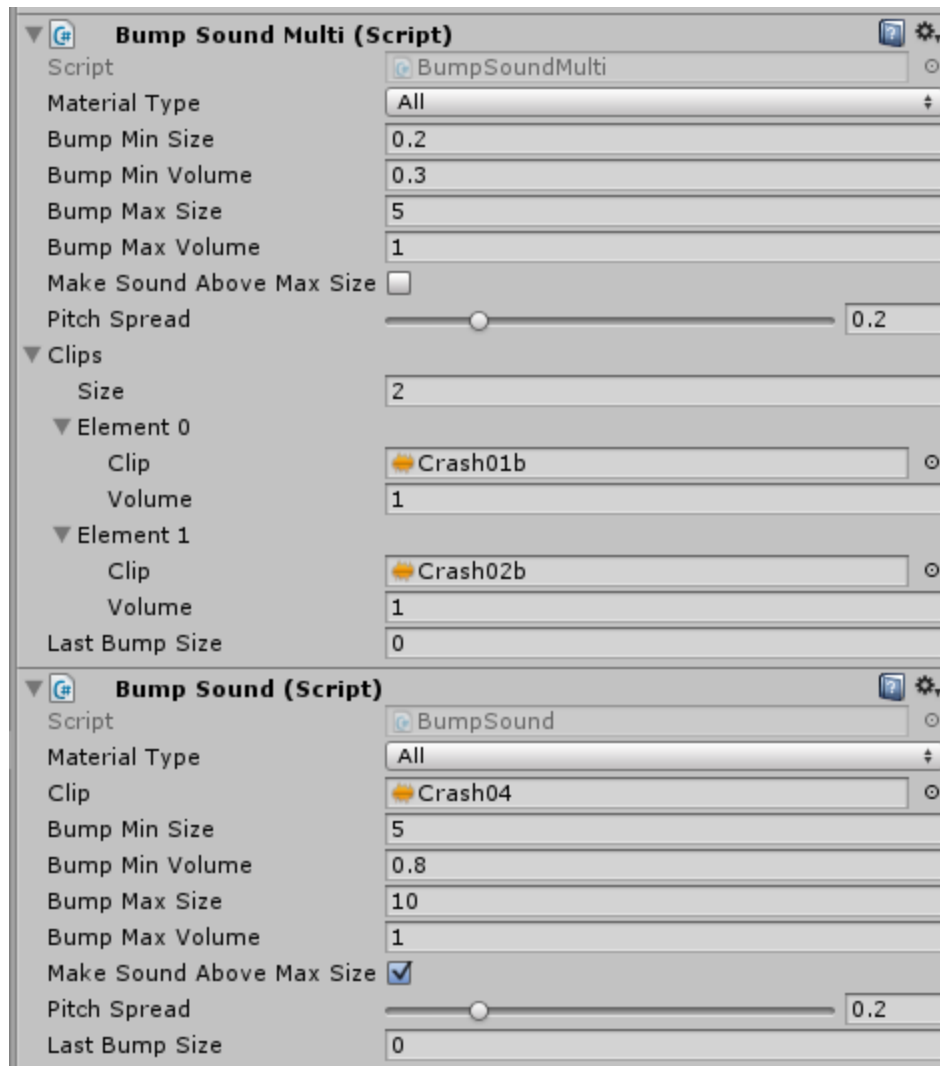
The default setup is that even the smallest contact will cause full volume sound. This is a robust and useful if not very sophisticated option.

If you want more control. You can tweak the parameters to serve your needs better. The parameters are described in previous chapter.

For example, the script in previous picture is setup to make sounds at collisions with size 0.2 and above. The full volume is reached at and above the value 10. The collisions between those values have the volume interpolated.

**Tip 1:** Don’t forget to use the **LastBumpSize** value. Do the hit you want to make a sound in the editor and check the **LastBumpSize** value. Adjust the **BumpMinSize** and **BumpMaxSize** accordingly.

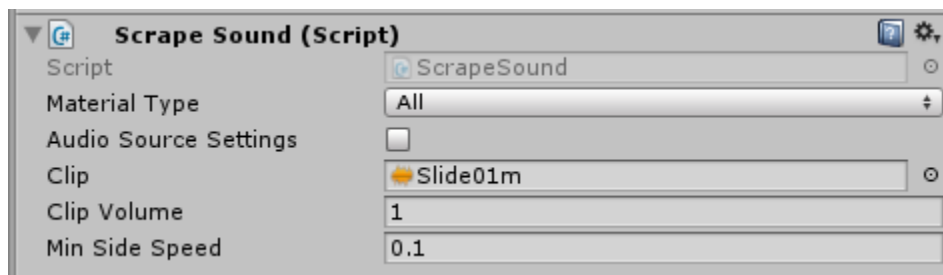
**Tip 2:** Use the stacking capability. Use more than one script per object to sound different clips depending on the strength of the collision. Following picture shows such a setup. Note that the first script setup to sound at lower contact sizes has **MakeSoundAboveMaxSize** parameter false so that it doesn’t sound above the contact size 5 together with the second script.



## ScrapeSound

A script/class playing a continuous (e.g. scraping) sound when its object slides along a ground or another object.

The user can set the script to be played, its volume and the minimum sliding speed which still makes a sound



## Parameters

**MaterialType** – Types of materials it should react to – see SoundMaster

**AudioSourceSettings** – opens settings for created audiosource (Output, Priority, StereoPan, SpatialBlend, ReverbZoneMix, RolloffMode, MinDistance, MaxDistance) and sets them over the default. For exact meaning of these settings please check Unity documentation.

**Clip** – rather selfexplanatory, the sound clip to be played in time of collision

**ClipVolume** – the sound volume to be used

**MinSideSpeed** – and the minimum sliding speed which still makes a sound in meters per second

## Usage

The usage is very easy. You drop the script onto the object.

You can set the volume at which the clip is played. If you want the volume to depend of the severity of the contact you can use the ScrapeSoundPlus script.

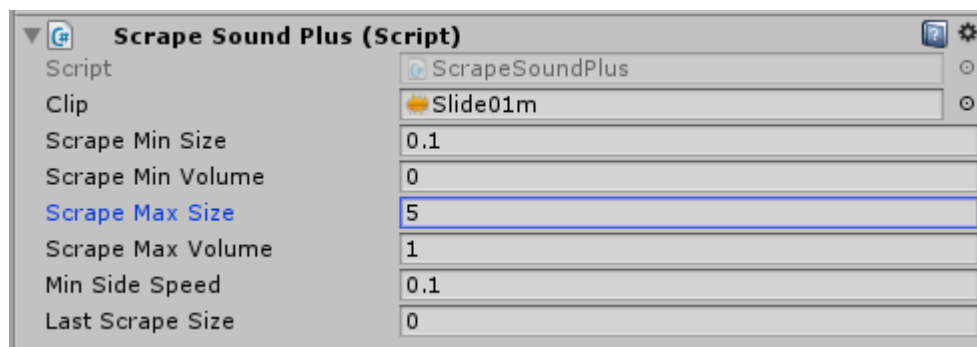
You can adjust the threshold speed for sliding. The default of 0.1 m/s should be fine for 99% of cases. Maybe only if you make some kind of microworld where speeds below that value are already substantial you might want to make the value smaller.

So in short, in most cases it's just dropping the script onto the object, choosing a clip and adjusting the volume in case the clip is too loud. That is it.

## ScrapeSoundPlus

A script/class playing a continuous (e.g. scraping) sound when its object slides along a ground or another object. The difference from the ScrapeSound is that the volume can depend on the strength of the contact.

The user can set the script to be played, some volume controlling parameters and the minimum sliding speed which still makes a sound



## Parameters

**MaterialType** – Types of materials it should react to – see SoundMaster

**AudioSourceSettings** – opens settings for created audiosource (Output, Priority, StereoPan, SpatialBlend, ReverbZoneMix, RolloffMode, MinDistance, MaxDistance) and sets them over the default. For exact meaning of these settings please check Unity documentation.

**Clip** – rather selfexplanatory, the sound clip to be played in time of collision

**ScrapeMinSize** – The minimum impact which will cause sound to be played. The value is the collision impulse (see unity documentation) divided by mass of the object. That means that the value shouldn't depend on the mass (light or heavy box). This should make configuration rather simple and in 99% of cases you should be just fine with the default (or your favorite) values.

**ScrapeMinVolume** – the sound volume used for **ScrapeMinSize** sized impact

**ScrapeMaxSize** – impact at which the sound reaches the full volume (**ScrapeMaxVolume**)

**ScrapeMaxVolume** – The full volume of the sound

**MinSideSpeed** – and the minimum sliding speed which still makes a sound in meters per second

**LastScrapeSize** – not a parameter for you to set. This value shows the size (severity) of the last impact. This means that should you be unsure about what values to use, you simply drop or bump into the object in the Unity Editor and have a look what value shows up. This should make fine-tuning the values rather easy.

## Usage

The usage is very easy. You drop the script onto the object, preferably with a Rigidbody on it. If the script can't find a Rigidbody it will try to get collision impulse value from the other colliding Rigidbody. If that fails it can't establish contact severity and a default value is used.

You can set the volume at which the clip is played. If you want the volume to depend of the severity of the contact you can use the ScrapeSoundPlus script.

You can adjust the threshold speed for sliding. The default of 0.1 m/s should be fine for 99% of cases. Maybe only if you make some kind of microworld where speeds below that value are already substantial you might want to make the value smaller.

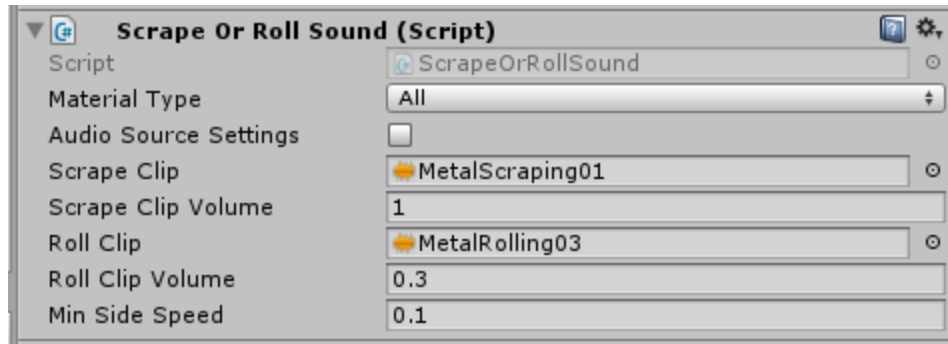
So in short, in most cases it's just dropping the script onto the object, choosing a clip and adjusting the volume in case the clip is too loud. That is it.

**Tip 1:** Don't forget to use the **LastScrapeSize** value. Do the hit you want to make a sound in the editor and check the **LastScrapeSize** value. Adjust the **ScrapeMinSize** and **ScrapeMaxSize** accordingly.

## ScrapeOrRollSound

A script/class playing a continuous sound when its object moves along a ground or another object. The user sets clips and volumes for both scraping (sliding) and rolling movement. The script detects which type of movement occurs and plays the appropriate sound.





## Parameters

**MaterialType** – Types of materials it should react to – see SoundMaster

**AudioSourceSettings** – opens settings for created audiosource (Output, Priority, StereoPan, SpatialBlend, ReverbZoneMix, RolloffMode, MinDistance, MaxDistance) and sets them over the default. For exact meaning of these settings please check Unity documentation.

**(Scrape/Roll)Clip** – rather selfexplanatory, the sound clip to be played in time of collision

**(Scrape/Roll)ClipVolume** – the sound volume to be used

**MinSideSpeed** – and the minimum sliding speed which still makes a sound in meters per second

## Usage

The usage is very easy. You drop the script onto the object.

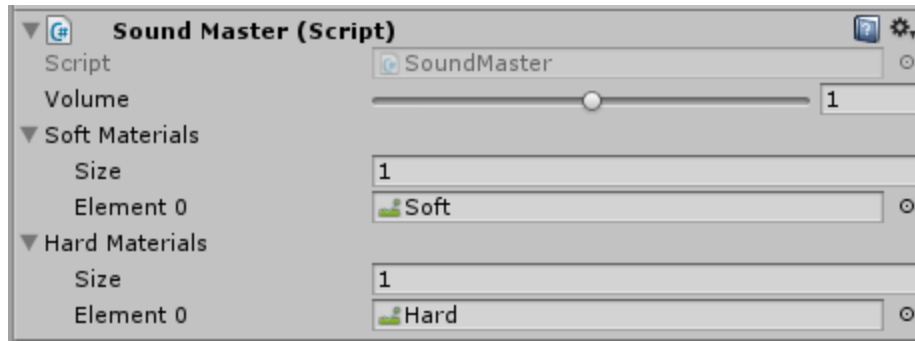
You can set the volume at which the clip is played. You can adjust the threshold speed for sliding. The default of 0.1 m/s should be fine for 99% of cases. Maybe only if you make some kind of microworld where speeds below that value are already substantial you might want to make the value smaller.

So in short, in most cases it's just dropping the script onto the object, choosing a clip and adjusting the volume in case the clip is too loud. That is it.

## SoundMaster

The SoundMaster can be used to control the sound volume of all the previously mentioned scripts at the same time. Its only parameter is Volume which increases or decreases the volume of all Bump&Scrape scripts.

This script is not mandatory. All the sound scripts work perfectly fine without it. All scripts try to find a SoundMaster component in the scene. If there is none, they simply go on without it.



## Parameters

**Volume** – increases or decreases the volume of all Bump&Scrape scripts

**SoftMaterials** – Physics materials considered “soft” as far as the contact sounds are concerned.

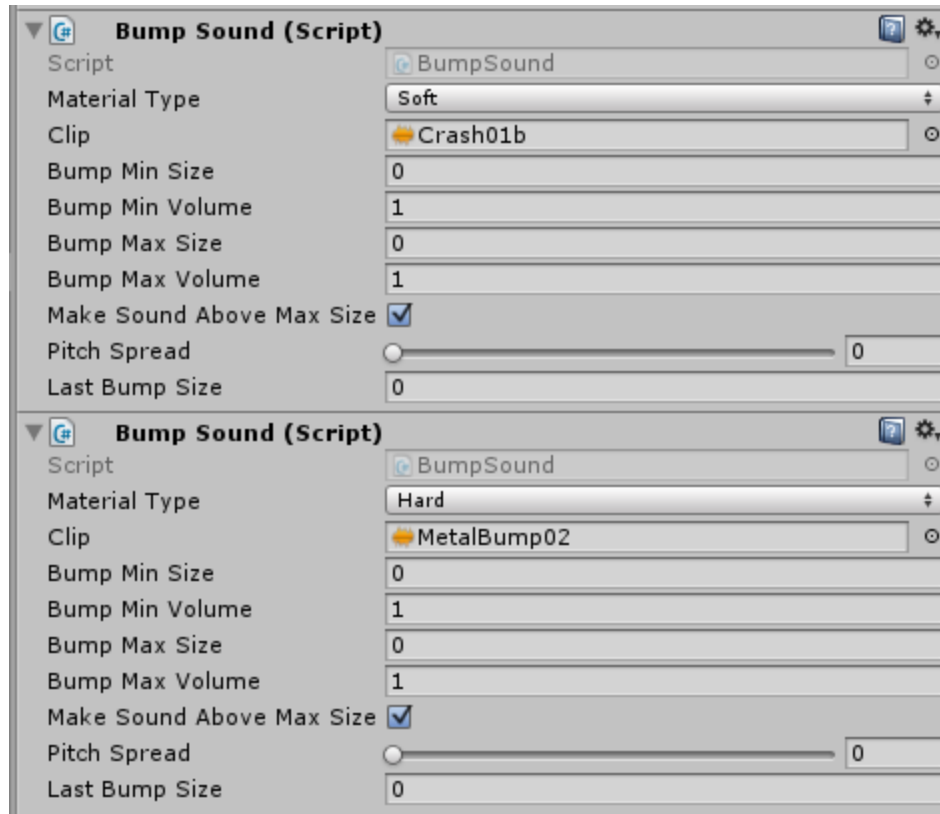
**HardMaterials** – Physics materials considered “hard” as far as the contact sounds are concerned.

## Usage

The usage is very easy. You drop the script onto an object in the scene. You set the Volume or leave it at the default 1.0 value and that is it.

The Volume is a normal public float variable so you can also set it at runtime from your own scene control script or sound manager or whatever your game/app uses.

Each Bump&Scrape script can be set to react to All/Soft/Hard/NonSoft/NonHard/Other materials. This way you can have, for example, two BumpSound scripts on a single object. One will sound for contacts with hard materials (concrete), one for soft ones (grass).



And once again, the script is not necessary. All other Bump&Scrape scripts are perfectly fine without it. It just provides the convenience of adjusting the volume of all the Bump&Scrape sounds at once.

## Tips and Tricks

- If you want to play bump/crash/hit/impact sounds go for BumpSound or BumpSoundMulti components
- If you have just one sound clip BumpSound is a good choice. If you have more clips which you want to alternate BumpSoundMulti can handle that.
- Off course, you can use BumpSoundMulti with just one clip as well!
- Bump scrips can be stacked. Use more than one script per object to sound different clips depending on the strength of the collision. Note that the first script setup to sound at lower contact sizes has MakeSoundAboveMaxSize parameter false so that it doesn't sound above the contact size 5 together with the second script (unless that is what you want).
- The volume of bump sounds depends on the severity/size of the impact. Don't forget to use the **LastBumpSize** value. Do the hit you want to make a sound in the editor and check the **LastBumpSize** value. Adjust the **BumpMinSize** and **BumpMaxSize** accordingly.
- If you want to make scraping sounds when the object slides across the ground or another object use the ScrapeSound or ScrapeSoundPlus components.

- If you have let's say a box sliding on the ground the ScrapeSound is the right choice. In such case the pressure in the contact depends only on the weight of the objects and is not likely to change.
- If you have situation like a car scraping along guard rails, the sound volume might depend on the severity of the contact. In this case using ScrapeSoundPlus might be worth the additional setup parametrization.
- Any two (or more) Bump&Scrape components can be used on the same object. Typically you will have one (or more) bump components and one scrape component.
- The SoundMaster is not necessary, but can be useful to control the sound volume of all the Bump&Scrape components from one place.
- Each Bump&Scrape script can be set to react to All/Soft/Hard/NonSoft/NonHard/Other materials. This way you can have, for example, up to three BumpSound scrips on a single object. One will sound for contacts with hard materials (concrete), one for soft ones (grass) and one for the rest/others. Bump&Scrape scripts don't care about the physics values in the materials so consider the soft and hard just groups of materials. Hard list might contain even a strawberry jelly for all the scripts care. They are just names chosen based on the most commonly used criteria.
- If you have some special requirements for the AudioSources (other than clip and volume) you can tick the **AudioSourceSettings** field. Additional parameters show up in the inspector (Output, Priority, StereoPan, SpatialBlend, ReverbZoneMix, RolloffMode, MinDistance, MaxDistance) and overwrite the unity default. For exact meaning of these settings please check Unity documentation.

*I hope you find the Bump&Scrape library useful. Enjoy.*

*Kajaman*

*kajamansup@gmail.com*

*<http://kajaman.is-best.net>*

Credits:

Brecht Lecluyse ([www.brechtos.com](http://www.brechtos.com)) for the ConditionalHideAttribute.