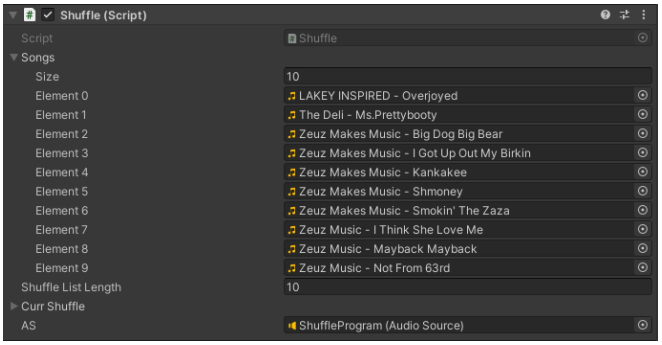
Shuffle – Documentation

Initial Setup



First, the 'Shuffle' script must be connected as a component to any Game Object, then for it to work AudioClips need to be populated under the 'Songs' array input inside the inspector, if none are provided the 'Shuffle' script will fail to work throwing an error and exception, it's size can only be adjusted outside of play mode.

Underneath the 'Songs' array, you can adjust the 'Shuffle List Length' inside or outside edit mode to choose how many tracks you want to be generated in the Shuffle list; this will default at 10.

Additionally, you can connect a Game Object with an Audio Source attached to it for playing the tracks provided by the Shuffle list, but this is not required for the script to work generating a shuffle list.

To make setup easier in the provided Unity Package, there is a pre-setup Scene, and or a Game Object 'ShuffleProgram' prefab with the 'Shuffle' script component populated with Audio Clips supplied in the package, with an AudioSource component attached and pre-referenced in the 'Shuffle' script's component properties.

There is also a Canvas Prefab 'ShuffleUI' with buttons to control the 'ShuffleProgram' Game Object, to set this up fully each button's OnClick function must have the Game Object with the 'Shuffle' script connected (**'ShuffleProgram' prefab**) referenced, with the 'Shuffle Tracks' button calling 'Shuffle' -> 'ShuffleSongs()', '<<' button calling 'Shuffle' -> 'PreviousShuffleTrack()', 'Play Pause' button calling 'Shuffle' -> 'PlayPauseShuffle()', and '>>' calling 'Shuffle' -> 'NextShuffleTrack()'. If the Test Program is being used the 'Run Test Program' button must have the Game Object with the 'TestScript' script attached (**'TestProgram' prefab**) referenced with it calling 'TestScript' -> 'RunTestProgram()'.

In Addition, a 'TestProgram' Game Object prefab can be found inside the Unity Package to stress test the Shuffle program, for this to work a Game Object with the 'Shuffle' script component attached must be referenced in the 'TestScript' component inspector panel.

Shuffle Script Usage

To control the 'Shuffle' script component it must be set up properly (***Check Initial Setup***) and referenced in other scripts to be able to use its functions, or a Game Object with it as a component can be used with an OnClick function of a button, calling the 'Shuffle' script's desired function. *(The 'ShuffleProgram' prefab in the Unity Package has this script connected already)*

To create a new shuffle list, a public function 'ShuffleSongs()' can be called from the 'Shuffle' script which generates a shuffled list 'currShuffle' of indexes relating to the 'songs' array values holding AudioClips, finally outputting a list of shuffled AudioClip names to the console.

The size of the list generated can be adjusted by changing the value of 'Shuffle List Length' in the 'Shuffle' script's inspector panel, although a new Shuffle list must first be generated to use the new size if adjusted.

Shuffle Script - AudioSource Control

If you decided to use the Shuffle Player control functions below, an AudioSource must be referenced in the 'Shuffle' script's inspector panel otherwise the functions will not work throwing a warning to the console. *(The 'ShuffleProgram' prefab in the Unity Package has this script connected with an AudioSource component attached and referenced already)*

To play or pause an AudioSource referenced in the inspector with an AudioClip provided by the shuffle list, a public function 'PlayPauseShuffle()' can be called from the 'Shuffle' script to do so. If the current shuffle list has not been populated this function will call the 'ShuffleSongs()' function to generate one and play the first shuffled AudioClip automatically.

To go back through the shuffle list providing the previous shuffled AudioClip to the referenced AudioSource, a public function 'PreviousShuffleTrack()' can be called from the 'Shuffle' script, this will go to the end of the shuffle list if it is

currently at the beginning of the list. The changed AudioClip will only play if the AudioSource was set to play previously. If the current shuffle list has not been populated this function will call the 'ShuffleSongs()' function to generate one.

To go forward through the shuffle list providing the next shuffled AudioClip to the referenced AudioSource, a public function 'NextShuffleTrack()' can be called from the 'Shuffle' script, this will go to the start of the shuffle list if it is currently at the end of the list. The changed AudioClip will only play if the AudioSource was set to play previously. If the current shuffle list has not been populated this function will call the 'ShuffleSongs()' function to generate one.

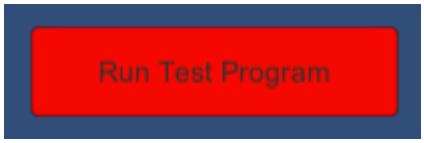
Test Script Usage

To control the 'TestScript' script component it must be set up properly (***Check Initial Setup***) and referenced in other scripts to be able to use its functions, or a Game Object with it as a component can be used with an OnClick function of a button, calling the 'TestScript' script's desired function. *(The 'TestProgram' prefab in the Unity Package has this script connected already)*

To activate the Test Program, a public function 'RunTestProgram()' can be called from the 'TestScript' script, this will stress test the referenced Shuffle Program, catching any exceptions and printing them to the console. It will also purposely break the Shuffle Program to demonstrate it can catch exceptions but will repair it and disable the Test Program after it has done so.

Shuffle UI usage

A canvas prefab 'ShuffleUI' can be easily used and set up (***Check Initial Setup***) to control the 'ShuffleProgram' Game Object prefab. 'Shuffle Tracks' in yellow will call the 'ShuffleSongs()' function to generate a new shuffle list, '<<' will call the 'PreviousShuffleTrack()' function to go back through the shuffle list providing the previous shuffled AudioClip to the referenced AudioSource, 'Play Pause' will call the 'PlayPauseShuffle()' function to play or pause the referenced AudioSource, and '>>' will call 'NextShuffleTrack()' to go forward through the shuffle list providing the next shuffled AudioClip to the referenced AudioSource.



An extra button can be found at the bottom of the Canvas 'Run Test Program' calling the 'RunTestProgram()' function from the 'TestProgram' Game Object's 'TestScript' script component.

Advanced code usage

The shuffled list 'currShuffle' list can be accessed and changed outside of the 'Shuffle' script. The 'currShuffle' list consists of int values for the Index to the related AudioClip value stored in the 'songs' array. Values and size of the 'currShuffle' list can be adjusted during runtime.

The 'songs' array can also be accessed and set outside of the 'Shuffle' script. The 'songs' array consists of AudioClip values which can be changed during runtime/play mode, but only at a set array size so only values can be changed. Leaving any value set to null (empty) will result in the 'Shuffle' script throwing errors and exceptions when provided with a null (empty) value.

To access the related AudioClip shuffled in the 'currShuffle' list you must use a 'currShuffle' value as an index for the 'songs' array to obtain AudioClip information, for example, to grab a shuffled AudioClip value, the code would look like "***Shuffle.songs[currShuffle[i]]***" *('Shuffle' referencing the script, 'songs' referencing the array holding the AudioClips, using 'currShuffle' as an index referencing the list holding the shuffled Indexes of the 'song's array, and 'i' being the index of the desired int index value kept in 'currShuffle').*

The 'shuffleListLength' int value can be accessed and adjusted outside of the 'Shuffle' script. This is used to determine the size of the shuffle list 'currShuffle' when it is next generated.