

Shuffle - Diary

- Created a 2D unity project
- Made a folder to hold scripts
- Added 2 new scripts the 'Shuffle' program and a 'TestScript'

Shuffle Script – initial programming

- Declared a list of type AudioClip called 'songs' to hold the tracks in the Shuffle script with SerializeField to make it adjustable in the inspector, I set this as a list so it can be dynamically changed during runtime
- Created a function called "ShuffleSongs"
- Declared a list of type int called 'currShuffle' to hold the indexes of the shuffled tracks
- Declared an int variable called 'shuffleListLength' with a default value of 10 to control the size of the Shuffle list with SerializeField to make it adjustable in the inspector
- Declared an int variable called 'lastIndex' to hold the last index added to the Shuffle list to prevent repetition
- Declared an int variable called 'shuffledIndex' to hold the current shuffled index
- At the start of the 'ShuffleSongs' an if statement checks if the capacity of the 'currShuffle' list is bigger than 0, if it is it clears the values from the list and Trims the capacity back to 0
- Next a bool variable 'forLoopDone' is declared with a value of false, then a for loop repeats it code by the size of the 'shuffleListLength' int variable
- Inside the for loop a do while loop randomly picks an index in the 'songs' list between 0 and the list's count value and sets it in the 'shuffledIndex' variable, then an if statement checks if the 'shuffledIndex' is not equal to the 'lastIndex' int, if true it sets the 'shuffledIndex' to the 'lastIndex' and adds the 'shuffledIndex' to the 'currShuffle' list. Lastly the while condition checks if the 'shuffledIndex' is equal to the 'lastIndex', if true it repeats its do loop code to reshuffle the next song to be added to the list to prevent repetition
- At the end of the for loop a ? operator checks if the for loop 'i' value is equal to the 'shuffleListLength', if true it sets the 'forLoopDone' bool to true
- Lastly at the end of the 'ShuffleSongs' function an if statement checks if 'forLoopDone' is true, if so "***New Shuffle List**" is printed to the console, then a foreach loops through the 'currShuffle' list printing the name of the AudioClip relating to the index provided by the foreach to the console

Test Script – Initial programming

- Declared a bool variable called 'testScriptEnabled' to control when to use the Test Program
- Declared a 'Shuffle' class variable called 'shuffleScript' to reference the object containing the 'Shuffle' script program
- Created a 'TestShuffle' function with a try catch statement that runs the 'ShuffleSongs' function in the 'Shuffle' script to catch any Exceptions
- In the update function I added an if statement checking if 'testScriptEnabled', if so it run the 'TestShuffle' function

Initial Scene setup

- Created two Empty Game Objects one called 'ShuffleProgram' and another called 'TestProgram'
- Added the 'Shuffle' script to the 'ShuffleProgram' Game Object
- Added the 'TestScript' script to the 'TestProgram' Game Object
- Left the 'songs' list in the 'ShuffleProgram' game object empty to test if the 'TestProgram' would catch any exemptions from this

Testing

- Enabled the 'testScriptEnabled' bool on the 'TestProgram' Game Object

- Played the scene on Unity but the whole program froze leading me to kill the Unity process in task manager
- Restarted the Unity Editor and redid my Initial scene setup as I forgot to save before playing and crashing the engine
- Saved the scene, left the 'testScriptEnabled' bool disabled and played the scene without it crashing this time
- Enabled the 'testScriptEnabled' bool with the scene in play mode only for it to freeze up again
- Restarted the engine again, I assume the issue was down to having the 'TestShuffle' function continuously being called in the update function, so instead I called the function inside the Start function of the 'TestScript' so it only gets called once
- Ran the scene again with 'testScriptEnabled' disabled so the start function would only call the 'TestShuffle' function once, but Unity freezes up again
- Decided to call the 'ShuffleSongs' function straight from the start function this time instead of the try catch statement in the 'TestShuffle' function, still unity freezes up so there's something obviously wrong with my 'ShuffleSongs' function
- Added a debug log to the 'ShuffleSongs' function that prints "Shuffle start" to see if the function would run at all from the beginning of it being called, but after testing this unity would still freeze up before it could even print the message
- Imported some tracks and added them to the 'songs' list of the 'ShuffleProgram' Game object to see if the list being empty was the problem but again the engine crashes
- Commented out the first if statement in the 'ShuffleSongs' function that checks if the 'currShuffle' list is not empty to clear and reset it, unity still crashes
- Moved the 'forLoopDone' bool declaration to the start of the 'Shuffle' class instead of being inside the 'ShuffleSongs' function and set it to false at the start of the function instead, finally unity stop freezing but I get a OutOfMemory Exception from the 'ShuffleSongs' function when it goes to add an index to the 'currShuffle' list
- After checking my task manager I see my RAM usage is at 85% and assume this is what is causing the exception throw so I kill some beefy processes taking up the space
- Again, I get the same exception, now I feel like having the 'songs' variable as a list is a bad idea for memory so I decide to turn it into a fixed array and change any relating code to array compatible methods
- Same exception gets thrown, I see in task manager unity is taking up 5GB+ of RAM whilst playing and 3GB+ when stopped so I restart the program
- Restart the engine and run again with the same exception thrown, looking at my code in the 'ShuffleSongs' function it seems the while loop is over filling the list taking the ridiculous amount of RAM usage whilst running
- I replace the while loop with an if statement checking if 'shuffledIndex' is equal to 'lastIndex', if so it randomly selects an index between 0 and the length of the songs array setting this to 'shuffledIndex', then an else if statement checks if 'shuffledIndex' is not equal to 'lastIndex', if true the for loop 'i' gets increased by 1 instead of the for loop doing this itself in its 3rd statement, 'lastIndex' is set to 'shuffledIndex' and the 'shuffledIndex' is added to the 'currShuffle' list
- Finally, I get no exceptions and the shuffle list is printed to the console and is working as expected
- Next I uncomment my code at the beginning of the 'ShuffleSongs' function that resets the 'currShuffle' list to 0 capacity every time the function is called to prevent the list from being overfilled
- In the 'TestScript' class I remove the 'ShuffleSongs' function call from the start function, set the Update function to a FixedUpdate to slow down the 'TestShuffle' call repetition, enable 'testScriptEnabled' in the 'TestProgram' inspector properties
- Play the scene again and everything runs as expected with the 'TestShuffle' function inside the 'TestShuffle' class calling the 'ShuffleSongs' function inside a try catch statement to catch errors every FixedUpdate call, outputting a new Shuffle list to the console
- I set the 'songs' array size to 0 to check if it outputs an error for having an empty array for the 'TestShuffle' function to catch, after playing unity freezes up again

- I add an if statement to the beginning of the 'ShuffleSongs' to check if the 'songs' array size is equal to 0, if true the script logs an error to the console stating "Songs array is empty, please add tracks to the songs array" and throws an InvalidOperationException to stop further execution, unity freezing up and an to provide an exception to the 'TestShuffle' function's try catch statement
- After testing everything works as expected with the error and exception printed to the console and unity no longer freezing

Test program – Adding repetition check functionality & Exception catch demonstration

- I set the 'currShuffle' list and 'songs' array to public so it can be accessed by the 'TestScript'
- I add a return statement inside the catch statement of the 'TestShuffle' function to break further execution
- Under the try catch inside 'TestShuffle' I declare an int 'lastIndex' with a -1 default value so the first check of the index value is not findable
- Then I add a for loop that loops up to the 'currShuffle' list's count, starting with an if statement checking if the value of each 'currShuffle' index is equal to the last index, if true a log warning outputs "There 2 tracks in a row in the current shuffle list! - Current shuffle list does not meet requirements!" to the console and returns out of the function, if false the value of each 'currShuffle' index is set to the 'lastIndex' variable ready for the next for loop check. Then at the end of the for loop an if statement checks if 'i' is equal to the 'currShuffle' list's count, if true a log outputs "No repetition found - Current shuffle list meets requirements" to the console
- Next I add some code to demonstrate further error catching in the test program, I do this by having the 'TestShuffle' function set a random value in the 'songs' array to null, and to prevent the test program repeating to call 'ShuffleSongs' after an exception catch I set the 'testScriptEnabled' bool to false inside the catch function
- After testing the above I find the exception demonstration code works throwing a NullReferenceException, but my repetition check code is not printing its results to the console
- I change the last if statement in the for loop that checks for repetition to check if 'i' is equal to the 'currShuffle' count minus 1
- Test again and everything works as expected with the repetition check printing its result to the console and the Exception catch throwing an error to the console after a random value of the 'songs' array is set to null throwing a NullReferenceException
- Added a debug log error "Test Script: Exception found" above the debug log exception in the catch statement in the "TestShuffle" function to make it clearer when an exception is found and printed by the Test program

Extras – Adding a GUI and audio player with controls

- In the 'Shuffle' script I added a public AudioSource variable 'aS' for connecting a Game Object with an Audio Source component
- Added an int variable 'nextShuffleIndex'
- Inside the 'ShuffleSongs' function I set the 'nextShuffleIndex' to 0 inside the last if statement checking 'forLoopDone' is true (Meaning a new shuffle list has been calculated)
- I create a function 'PlayPauseShuffle', inside first an if statement checks if the 'aS' AudioSource variable is not null, then inside that first an if statement checks the 'currShuffle' list if its count is equal to 0, if true it calls the 'ShuffleSongs' function to generate a new shuffle list to populate in the 'currShuffle' list
- After the previous if statement I add another if checking if the 'aS' AudioSource is playing audio, if true it pauses it
- Next an else if checking if the 'aS' AudioSource is not playing and if its clip is not null, if true it starts playing audio
- Lastly another else if checking if the 'aS' AudioClip is null, if true first the 'aS' AudioSource clip is set to the 'songs' array clip value at the index found in the 'currshuffle' list at the index int in the 'nextShuffleIndex' variable, then the 'aS' AudioSource is played, +1 is added to 'nextShuffleIndex',

finally an if checks 'nextShuffleIndex' is bigger than the 'currShuffle' list's count -1, if true setting 'nextShuffleIndex' to 0 preventing it from going outside of the 'currShuffle' list's index range with values

- I add another public function 'NextShuffleTrack' that first checks if the 'aS' AudioSource is not null, inside that an if checks if the 'currShuffle' list's count is 0, if true calling the 'ShuffleSongs' function to generate a shuffle list to populate the 'currShuffle' list
- Next +1 is added to 'nextShuffleIndex', then an if checks if 'nextShuffleIndex' is bigger than 'currShuffle' list's count -1, if true setting 'nextShuffleIndex' to 0, finally, under the if, the 'aS' AudioSource clip is set to the 'songs' array clip value at the index found in the 'currshuffle' list at the index int in the 'nextShuffleIndex' variable
- I create another public function 'PreviousShuffleTrack' first with an if checking 'aS' AudioSource is not null, inside that an if checks if the 'currShuffle' list's count is 0, if true calling the 'ShuffleSongs' function to generate a shuffle list to populate the 'currShuffle' list
- Next -1 is taken away from 'nextShuffleIndex', then an if checks if 'nextShuffleIndex' is smaller than 0, if true setting 'nextShuffleIndex' to 'currShuffle' list's count -1, finally, under the if, the 'aS' AudioSource clip is set to the 'songs' array clip value at the index found in the 'currshuffle' list at the index int in the 'nextShuffleIndex' variable
- After coding the above into the 'Shuffle' script I attach an Audio Source component to the 'ShuffleProgram' Game Object and link itself inside the 'aS' AudioSource value on the inspector
- I create a canvas with 'Scale with Screen Size' scale mode, adding 4 buttons 'ShuffleBTN' connecting the 'ShuffleProgram' with the 'Shuffle' script's 'ShuffleSongs' function to its OnClick behaviour, 'PreviousTrackBTN' connecting the 'ShuffleProgram' with the 'Shuffle' script's 'PreviousShuffleTrack' function to its OnClick behaviour, 'PlayPauseBTN' connecting the 'ShuffleProgram' with the 'Shuffle' script's 'PlayPauseShuffle' function to its OnClick behaviour, and 'NextTrackBTN' connecting the 'ShuffleProgram' with the 'Shuffle' script's 'NextShuffleTrack' function to its OnClick behaviour, also setting their relevant text for each button
- Testing the buttons work, playing audio and reshuffling, but I need to make the Next and Previous buttons play audio after changing the clip
- Added code in 'NextShuffleTrack' and 'PreviousShuffleTrack' functions after they change the 'aS' AudioSource clip to play the AudioSource
- Testing the Next and Previous buttons now play the audio after changing the song, but still play if the 'aS' AudioSource is paused so I need to add a bool that tracks if the audio is meant to be playing
- At the top of the 'Shuffle' script I add a bool 'audioPlay' with a false default value, setting this to true when audio is played in the 'PlayPauseShuffle' function and false when its paused, then I add an if around the code that plays the audio to check if the 'audioPlay' bool is true in both the 'NextShuffleTrack' and 'PreviousShuffleTrack' functions
- Testing the Next and Previous buttons only play the track if the audio was previously set to play, but when a track reaches the end, the next track does not play
- Added a new function to the 'Shuffle' script called 'AudioPlayerCtrl', first with a if checking if the 'aS' AudioSource is null, if true the function returns, then an if checks if the current playback position is bigger or equal to the current clip's length and the 'aS' AudioSource is not playing, if true +1 is added to 'nextShuffleIndex', an if checks 'nextShuffleIndex' is bigger than 'currShuffle' list's count -1, if true setting 'nextShuffleIndex' to 0, after the previous if the AudioSource clip is set to the 'songs' array clip value at the index found in the 'currshuffle' list at the index int in the 'nextShuffleIndex' variable, finally, at the end of the function, an if checks if 'audioPlay' is true and the AudioSource is not playing, if true it plays the AudioSource, and an else if checks 'audioPlay' is false and the AudioSource is playing, if true pausing the AudioSource
- At the top of the 'Shuffle' script I add a Start method, inside an if checks the 'aS' AudioSource is not null, if true with InvokeRepeating I call the 'AudioPlayerCtrl' function every .5 seconds
- Testing I get a NullReferenceException from the 'AudioPlayerCtrl' trying to check for the 'aS' AudioSource clip length

- I add an if that checks if the current clip in the AudioSource is not null around the if that tries to check for the current clip's length
- Testing the same track plays again
- I move ifs checking audioPlay is true or false, playing or pausing the AudioSource into the if checking if the clip is not null and turn them into else ifs, after checking if the current track has ended to play the next track and remove the check if the AudioSource is playing while checking if its ended, I also replace the Start method with an Update with it calling the AudioPlayerCtrl every frame if the 'aS' AudioSource is not null
- Testing the next track is loaded into the AudioSource after the previous is playing, continuing to keep the AudioSource playing if it was set to previously using the 'audioPlay' bool, but I notice in the Inspector debug the Play/Pause button adds to the 'nextShuffleIndex' after playing the first track, skipping a track in the list when the first track finishes playing
- I remove the code adding to the 'nextShuffleIndex' and the if checking its still in the list's index range in the 'PlayPauseShuffle' function, I also remove the code playing the AudioSource after changing the track in the 'NextShuffleTrack' and 'PreviousShuffleTrack' functions as the 'AudioPlayerCtrl' function now does this automatically
- Testing the Play/Pause button no longer adds to 'nextShuffleIndex' after playing the first song and the AudioSource still continues to play after Next and Previous buttons changing the AudioSource's clip
- Added a public function to the 'TestScript' script 'RunTestProgram' that sets the 'testScriptEnabled' to true to run the test, creating a new button on the UI 'Run Test Program' with an OnClick function connected to the 'TestProgram' Game Object, calling the 'RunTestProgram' function
- Testing the 'Run Test Program' button works, but it leaves the Shuffle program broken
- Added code in the 'TestScript' to save the removed values used for the exception demonstration, applying them back in the correct position once 'testScriptEnabled' is disabled after an exception and 'errorCaused' is true after being set by the demonstration code, disabling this after the fix is applied to prevent repetition
- Testing the Test program fixes the Shuffle program after its demonstration
- Changed the 'Shuffle' script reference to be connected in the 'TestScript' script's inspector properties, adding an if to check if it is not null before running the 'TestShuffle' function to prevent errors, and adding an if to check if its reference is not null in the 'RunTestProgram' function before setting the 'testScriptEnabled' to true, if it is null an error log prints to the console "Shuffle Script not referenced in the Inspector! Please connect the Game Object the script is attached to in the Test Script component"
- Attached the 'ShuffleProgram' Game Object to the 'TestScript' script's inspector reference on the 'Test Program' Game Object
- Testing the Test Program works as expected
- Added a debug log warning to 'PlayPauseShuffle', 'NextShuffleTrack', 'PreviousShuffleTrack' and 'AudioPlayerCtrl' telling the user "No AudioSource found! Please reference an AudioSource inside the Shuffle component's inspector properties to play the shuffled AudioClips..." if the 'aS' AudioSource is null. I also set the 'shuffleListLength' int variable to public so it can be accessed by other scripts.
- Testing the warning outputs as expected if the AudioSource has not be referenced in the inspector