Shuffle - Reflective Statement

What have I achieved?

Developing my Shuffle brief I have created a Shuffle Program included with a track player, and a Test Program to stress-test its capabilities using a 2D Unity project, keeping track of my progress in a production diary.

Coding my Shuffle program, I produced a public function that can be called outside of its script responsible for clearing up previous data to prevent overfilling the list, generating a shuffle that checks and prevents repetition, populating it into a list with a dynamically adjustable size via an adjustable variable in the inspector panel, throwing errors and exceptions to break the code execution if the script was handled incorrectly preventing application freezes, for example missing Audio Clips or component references, and finally outputting the shuffle list with the Audio Clip names to the console. Moreover, I set the shuffle list, Audio Clips array, and shuffle list length integer with public access modifiers so they could be accessed and modified by auxiliary scripts for external functionality.

For my Test program, I developed a function that stress tests my Shuffle program, catching errors to output to the console, checking for repetition in the shuffle list to make sure it meets its requirements logging its results, and finally, an exception catch demonstration that breaks the shuffle program to show the test script can catch errors and repairing it afterwards. Also including a public function callable outside of its script to run the test program.

To add to my Shuffle program, I included an optional track player feature that uses a referenced Audio Source via the inspector panel to playback the shuffled list. Using 3 public functions handling whether to play or pause the Audio Source, one to skip to the next shuffled Audio Clip providing it to the Audio Source, another to go back to the previous Audio Clip again providing it to the Audio Source, and finally a private function to control whether the Audio Source should move to the next shuffled Audio Clip after the previous had finished and if it should play or pause if it was set to previously after changing the Audio Clip, calling this function in the update method to ensure correct operation every frame.

Furthermore, I set up a UI system using a canvas with 4 buttons referencing the game object with a shuffle script attached, ‘Shuffle Tracks’ calling the shuffle function to generate a new shuffle list, another 3 buttons calling the track player functions, ‘<<’ that goes back a shuffled audio clip, ‘Play Pause’ for playing and pausing the audio source, and ‘>>’ which skips to the next shuffled audio clip. In Addition, I incorporated the 5th button at the bottom of the UI referencing the game object with a test script attached to call the function that enables and runs the test script.

To summarize I finished up by exporting the required assets into a unity package, then I wrote out clear documentation explaining the correct usage and set up of the Shuffle and Test program.

What have I learnt?

Whilst constructing my Shuffle brief, I have learnt many things with various Unity and C# systems. Firstly, utilizing C# lists I have gained more of an understanding of how they work compared to an array, discovering many methods, such as .Clear() and .TrimExcess(), and properties, for instance .Count and .Capacity, that I have not previously known of. Grasped further knowledge of certain C# loops work knowing where and when to use them, other types of invoking methods, extra canvas settings for responsive scaling. Comprehended how I should reference components and game objects by finding them with code or connecting them via the inspector to make specific scripts more universal. I have found out how to throw exceptions to break code execution, preventing application freezes, when to use them and how to catch them with try-catch statements printing the exceptions with Debug.LogException(). Gained further experience with Audio Sources finding out about additional methods and properties to control their functionality, Audio Clip properties with useful information that can be used, and in the Unity Engine, I have unearthed the profiler used to track performance statistics.

What would I do differently next time?

Next time developing my scripts I ought to remember starting simple with my code, rather than over complicating causing me to have many issues whilst testing trying to get it to work, also making sure I test my implementations more frequently and using debug logs to track code execution stages, helping me find any problems easily. Furthermore, I would instead use lists instead of arrays to hold information as they are easier to use and less problematic being dynamic compared to statically sized arrays, possibly even making a custom data class used in the list to hold information for the shuffle and tracklist, as well as advancing the shuffle function's repetition prevention to refrain from having too many of the same Audio Clip generated into a list instead of it only avoiding the last shuffled track being the same as the next random pick, and added features to the UI I included displaying the current shuffle list with a highlight over the playing clip including a controllable playback time bar with timings.

How has what I have created assisted me through my studies?

From the experience and awareness, I have obtained whilst making my Shuffle brief, I believe will help me become more efficient in programming my scripts, possibly reducing mistakes I have made throughout, using better methods and practices to create optimised functionality.

In deeper detail using lists for future projects will improve my scripts data holding capabilities with fewer restrictions, a wider range of methods and properties to make use of compared to arrays. Greater sense of Audio Source and Audio Clip functionality and attributes, C# loop and use cases and behaviour, exception throws and handling, debug functionality, additional Unity tools such as the profiler to track performance information, all things now knowing of I can see will assist me in later work and studies.