

## **Nathan Diill MDU112.3 Postmortem**

What worked well on MDU112.3? One of the key factors that helped in my project was my organisation. By organisation I mean the management of the files in the project. I constantly made sure that everything was sorted into the appropriate folder. The moment a new item was made it was put into a folder, rather than let the default storage accumulate over time. When it came to finding certain items for use during the project it was easy and saved a lot of time. Another key factor that led to the success of my project was my knowledge of Unity. I found only on a rare occasion where I was needing to search online with help for Unity related issues. This was great because it left me with even more time to focus on what was important, the code. GitHub was a huge success in the project. On several occasions I would find myself referring to it for code. Especially after editing a file for a long period of time and finding what I was trying to do didn't work and needed to go back to what I had before. Without GitHub I can't imagine how painful that would have been.

What didn't work well on MDU112.3? Time management was a somewhat smaller issue this time round. I was never unsure about whether I could complete the project by the due date/time. This took a lot of pressure off, however, the times I was working on the project weren't desirable. Most of the project was completed at hours that were late during the night. While I work well at those times, it won't work in the future and isn't a professional approach to work. Creating major issues from smaller issues was a constant problem. I would just run with it and write heaps of code then I would go to run the code and it wouldn't work. After spending a long time searching for the problem, the solution was often simple, either I wasn't calling the function or the function wasn't located properly. While with practice I will remember to include those sorts of things, it did happen more often than I care to admit.

My planning work on MDU112.3. I jumped into it first by having the player moving inside a box. Before continuing I did some minor planning. I wrote down some concept ideas on a page with a list of the important tasks to complete. After that the planning was non-existent, I focused all my attention to the hands on work for the project. Could this have affected the outcome of the project? Quite possibly, however I was confident that I knew what needed to be done. I knew what items needed to be set up with inheritance and exactly how that could have been done.

How did my work on MDU112.3 improve/not improve from MDU112.2? In MDU112.2 I mentioned about updating and practicing my knowledge of Unity for the last

assignment. I believe this goal was a huge success due to only one reason. Attending class each week and going through the tutorials ensured that my knowledge of Unity and Unity principles was never lacking. I also briefly touched on my issue with attaching code to GameObjects. This was another success, as I am confident with managing GameObjects with scripts attached to them. This is due to constant practice at home and in class lessons. GitHub continued to be a huge success and even an integral part of my project. Nothing but success with the use of GitHub. Structuring and commenting my code was another topic that needed to be carried over to MDU112.3. I feel as though it went well, but not perfect. The structuring of my code was good, aside from the few times where I would place a function in the wrong spot and sometimes the wrong script. Commenting my code was done well but not perfect. I would get too caught up in writing code and completely forget to comment and I found myself coming back later and adding in the comments. I was unsure on several occasions whether or not to comment, debating whether it was too simple to add a comment to. Lastly, as I did in MDU112.2 I continued all through MDU112.3 to save my work at a compulsive rate. I made commits to GitHub every few hours, maybe more if I lost track of time.

How will I ensure what worked well on MDU112.3 continues to do so? The organisation of my project will be imperative heading forward into Scripting 2. While I believe it will never be an issue because of my need for everything to be organised, after seeing how much it helped on this project, it will never be forgotten. My knowledge and understanding of Unity will only get better. I'm sure that I will be using Unity in the future. To ensure that I'm always learning new things to do in Unity I will be working on my own projects during the break. Aside from the stress of needing to complete the project, MDU112.3 was a lot of fun and will only get more fun as I learn more. This is all the incentive I need to continue working in Unity. I will undoubtedly be using GitHub in the future. Aside from the fact that it would be compulsory, GitHub was a huge success and I can't imagine my project without it. Something as imperative as GitHub will always be on my mind in the future.

How will I ensure what didn't work well on MDU112.3 doesn't happen again? My time management of this project probably won't be sufficient enough heading into Scripting 2. While I was always sure I would complete the project on time, as the projects get bigger and the due dates get shorter, my current approach won't be enough. Adapting to some sort of study calendar or work plan is my only chance. Making sure that constant attention is given to future projects is a must, I cannot be leaving them for several days untouched. Working during the late hours of the night needs to be corrected. I have hopes that creating an exact calendar with specific times to study will correct this issue. It is a habit I have fallen into, I'm confident that as the projects get

bigger it will snap me out of this habit as working during the night will not be favourable. Creating major issues out of smaller issues is a very fixable problem. Continuing to work on my coding skills during the break and during future projects will ensure that I will know to include the call for a function and where exactly to put everything. Practice makes perfect.

Concepts covered on MDU112.3. Working on several scripts at a time. Practical use of inheritance. Arrays, classes, OnTriggerEnter, SendMessage, reusable functions. Issuing and managing C# files in Unity. Unity, being able to edit project settings and various others within the Unity UI. Prefabs, sprites and other important objects in the hierarchy. Audio, adding and setting event triggers to audio.

Resources used on MDU112.3. Unity tutorials helped on occasion. Incompetech was used for all background audio. Unity version 5.3.5 in 64-bit was used. Visual Studio was used for all code. GitHub.

## Maze

Various item pick-ups in the game that only give xp, xp levels the player up and allows them to move quicker and grab pickups from further away.

Maze will feature various keys that are required to progress. 1 key at a time!

Maze police are patrolling the maze. If touched by them you reset to the start - will be easier to avoid as you level up. But should you grab the xp?

At the end of the maze is a big police boss blocking the exit directly. The player must find and equip a gun (in the key slot) and defeat the boss to escape.

## UI

Main menu & credits screen

Win/Lose screen

UI for level and speed

## Audio

Music in main menu, credits and in game.

Sound effects for each in game event

## Art

Art as sprites, materials & models.

GitHub: <https://github.com/nathandiill/16T3-MDU112.3>