**CoderDojo Blogs**

I volunteered at CoderDojo every week since the 2nd of October, attending in-person sessions fortnightly and attending and contributing to planning meetings in off weeks. In most sessions, I played the role of a mentor, but I did run a session during the semester. I only helped out with planning and mentoring the Python students but did interact with the Scratch students when they wanted to show off their game. After most of the in-person sessions, I wrote down a few notes about some key things I observed or discussed with other mentors, and in this document, I’ve written up two of the more important ones that are referenced in my report.

*CoderDojo 16/10/23: Trialling teach back.*

There were reasonably small numbers at this session and a good number of mentors. This meant that I was only supervising a few ninjas so could spend much more time supporting each of them. As there was a small group, I thought it would be a great time to try some techniques, primarily teach back. During the session, I tried to put much more of an emphasis on why something was used rather than just what was the correct construct to use for each scenario. An example of this was casting an input in Python to an integer, I tried to explain how they are different types so had to be treated in different ways, and how we needed an integer but were given a string. This was a good opportunity to use teach-back, so I asked the student to repeat an explanation in his own words back to me. I also tried to use semantic waves that I had used in previous weeks. To another ninja I used the classic example that everyone references for explaining variables, I used teach-back for this too, in particular to get a mutual understanding of why variables and functions were useful, when to use them, and how they make your life easier. As well, I tried to associate the code they were writing with the terminology that experienced programmers would use, such as calling functions and throwing errors. This seemed to be very successful as the ninjas seemed very encouraged at the end of the session, and in the following weeks, they seemed to have mostly retained the knowledge of why these key concepts are useful.

*CoderDojo 13/11/23: Leading a session (rocket launch simulation).*

In the build-up to this session, I made notes about the learning objectives, and the questions included in the quiz to present what concepts we were about to learn to the students. I also made note of where these things were introduced to make sure that I paid special attention to them and got the students to understand them.

In class we had covered PRIMM around this time so during the lesson I also trialled pulling up the terminal and writing little code snippets of for loops and getting students to predict what would happen when I ran them, I think this worked okay but was maybe a bit too difficult for a beginner to follow, and perhaps I should’ve written them in the standard Raspberry Pi Code Editor they are familiar with. I tried to make an extra effort to encourage students to make more sophisticated modifications to take ownership of their code beyond just choosing which planet image they wanted to use in their program. An example of this is when I encouraged students to change the spaceship trail graphics by editing the for-loop parameters used. I was happy with this although it ate into the lesson time, making it more difficult to complete in our 1 ½ hour slot.

Again, I made a special effort to explain jargon to the whole class, rather than on an individual level, namely “calling” functions. I wanted to encourage learners to understand the language of Computer Science, as I think this is something that usually isn’t taught, rather people around you use it enough to the point you start to use it and you forget that not everyone knows what it means!

Finally, the biggest thing that I encountered during the session was difficulty managing different work rates. In our group that week we had one advanced ninja (whose parent works in tech) and another who is quite young, so it was difficult to decide whether to wait till everyone was on the same page, or to leave the slower learners behind to work at a comfortable pace. Luckily in this session, the more advanced ninja was also working on a side project in Unity, so was not left idle and instead could work on that while the other ninjas caught up. I also tried to open a discussion with the advanced ninja about computational thinking involving his application, in particular designing a mechanic to stop a player from jumping mid-air and extending this to allow for double jumps. I was very satisfied with the outcome of these conversations, as I felt that he gained something from the sessions that he wouldn’t have gained if he completed the unity tutorial at home.