**CoderDojo Blogs**

I’ve included 2 blogs from sessions that I thought were quite formative, or interesting to talk about. Both were written the day after the session, and just detail some of content as well as some of my thoughts, however I’ve reformatted them, so they are a bit less informal.

**CoderDojo 12/2/24: Python Turtles and Fractals**

I was mentoring 4 students in the lesson, and the lesson was a series of challenges regarding python turtles. The aims of the lesson were to get python installed on individual computers like a “real” programmer would use it, and then focus on for loops which they have seen before, but I was not convinced they fully understood so wanted to cover again. I think running this session it was clear that they did not particularly remember the for-loop syntax and when to use one, so I was glad I did recover them.

First, I helped students install python, this was straightforward with a small group even though they used different OSes. Then I showed them all how to make a turtle move forward and turn left, then had them go and use this to draw a square. Next, I had them convert their sequence of moves into one for loop, then extend their code to draw a window of four squares and again had them convert this into a for loop. After this I had them change the parameters of these for loops and change the size of the squares to start making cool patterns. After this I had them change their line lengths and angles to produce some unique fractal patterns. For learners who completed this I had them change the colours of their fractal patterns, having them use random numbers that they had seen in previous weeks to make a nice aesthetic pattern.

Because of the room layout it made this lesson especially hard as the group was a bit more spread out than usual so had to move between learners to provide instructions of what I wanted them to do next. It would have been useful to have written the tasks down so that I did not have to jump between them, but I had not anticipated that the group would be split up.

Two of the learners were siblings and have a bit of a rivalry, this kept them very engaged, but I felt a bit conflicted about their learning being driven by being competitive rather than self-motivated. The older of the two is a bit more tech savvy, so helped the other when small technical issues came up, these issues slightly discouraged the younger learner. Overall, I was really impressed with the younger sibling, as they did very well because this task was more focused on computational thinking than on knowing python well, even if they were a bit discouraged at the beginning.

Helped another learner visualise their code using a toy car they brought to the session, where the car was their turtle, they needed to program. This worked well as it was after this that the code clicked for them.

Turtles were a good medium for learning about loops, but I wonder how useful they could be to learn about conditions and if statements.

**CoderDojo 11/3/24: ImagiLabs Rorschach Generator**

This session I was supervising 8 learners (aka ninjas!) as I was covering for another group of 4 python learners alongside my regular group. I had covered for them the previous session as well, and used ImagiLabs, the online editor for an 8x8 led screen (we didn’t have the physical screens just used the online version) as this is what the previous group had been using and I thought the transition for my students would be easier as they have a little more python experience.

For this session I made a Rorschach image generator in Imagilabs, then because I had a large group I got to use the TV, which I put my laptop onto to walk the students through step by step. Stopping to ask questions, thinking about how we would do something programmatically, before showing and reviewing the concepts that I wanted to cover during the lesson such as why you would use a nested for loop. This session ran much smoother with the board, as it allowed me to get everyone’s attention at once rather than trying to manage 8 individuals separately.

At the beginning of the session, I showed them all the output of the program and explained briefly what a Rorschach image is and had the students interpret some of the patterns it produced. This worked very well as got the students engaged in what they were about to complete and gave them a good mental image of what they were aiming for.

I also let them make the project their own through the colours for their patterns. Letting them experiment with different colours and randomness to make something that was personal to them. For weaker learners I assisted them in actualising their designs while for the stronger ones I encouraged them to do some experimentation to find a design that they liked.

After completing the tasks, I again did had every student create and interpret a pattern from their program and the students all examined their peers picture and gave their own interpretations which was very fun.

I thought ImagiLabs was a good medium although I felt that it was reaching the end of its usefulness as python can do so much, so constraining it to just a 8x8 grid feels a bit limiting as students’ progress.

Other things that were mentioned in the sessions were random numbers, coordinate systems, RGB and a quick explanation of a Kibibyte vs. a Kilobyte which I can’t remember how but came up from one of the questions asked by a learner. Most of these have been covered in the past so I wasn’t too concerned with overloading the students with information, more just recapping old stuff and only introducing new concepts to the students I thought could handle it well.

Overall, I thought the sessions worked really well, students were more engaged and quite curious which gave me lots of opportunities to provide further explanations to more advanced students who wanted to know more. I also had dedicated time for students to play around and fiddle with the code which I think really helped them to understand the effect of the bits of logic in the program on the resulting image.

A screenshot of a computer program

Description automatically generated

Figure 1: My Original Program

A screenshot of a computer

Description automatically generated

Figure 2: The Student's Programs

A screenshot of a computer program

Description automatically generated

Figure 3: Student A's program