**Computing in the Classroom – Creating my Python Teaching Materials.**

*Originally intended to go into the main report but moved here as I wanted to focus on my CoderDojo experiences there.*

Motivation

My main motivation for wanting to produce a Python exercise is because of my experience delivering the Intro to Python Pathway to my CoderDojo. After using this resource several times, there were a few frustrations I identified that I wanted to address by producing my own coding exercise. I wanted to produce an exercise that fits the format of CoderDojo better and fixes some of the identified issues with the RP Pathway while capitalising on its merits.

What is my exercise?

For my exercise I made a simple game using Python and the P5 package called “LookBeforeYouLeap”, the game consists of a rectangle representing the player and the user needs to input the number of jumps and amount of jumps they want to character to complete. The character then spawns in and bounces along some platforms to try and reach the flag at the end of the level. There are some screenshots of the game included in the Appendix. I implemented this in the Raspberry Pi Code Editor so that mentors would not have to install Python on the different OSes of students before they could begin the lesson. The exercise is meant to be delivered in combination with in-person mentors who can assist students if they get stuck, the tutorial could become too long if students had to debug their own code with no assistance, so I think mentors are a necessity for this material. Because it’s meant to be delivered this way I’ve made 2 sets of instructions, one for students, and one for mentors with more straightforward instructions, and some information on how to use the material and who it’s aimed at.

The Raspberry Pi (RP) Pathways use something they call the 3,2,1… Make! Framework. This shares some similarities with the Use-Modify-Create framework (as well as PRIMM which we will discuss later). My assignment is designed to be delivered to the ninjas in my dojo who have completed the Intro to Python Pathway so it follows a similar format to one of the Explore Pathway lessons, intending to teach global variables, as well as getting more experience using if statements and functions, which I felt like the original course either didn’t cover or I felt that scaffolding for that concept was removed a bit too quickly, resulting in learners not really knowing what the code did and instead just copying it out.

Key Design Decisions and Inspirations

The biggest influence in the creation of my materials was the Use-Modify-Create (UMC) framework. I wanted to produce an assignment that first let the students get to grips with what they were going to create as they do in the Raspberry Pi materials, then leave some opportunities for modification with various levels of sophistication. Finally, I wanted to have some slightly more open-ended tasks where the student had to create their solution. Of the examples of UMC I’ve seen, some split these across multiple lessons and did them sequentially, but for my materials, I aimed to include them all.

Following the Use principle, one thing I’ve noticed is that many Python tutorials, and coding tutorials at large, fail to teach how to use other people’s code and how to effectively utilise websites like Stack Overflow. This is becoming increasingly important with the rise of technologies like GitHub CoPilot that can actively suggest code as you write it. I would like to try to tackle teaching this skill, although the learners in my CoderDojo would be too young for this, I think. I tried to add this into my exercise by providing some predefined “skeleton” code in my exercise for students to understand the purpose of and use, even if they don’t fully understand the fine details of the implementation.

Sweller’s cognitive load theory (Sweller, 2011) explains that our working memory can only handle a limited amount of load. There are two main categories: intrinsic and extraneous cognitive load, intrinsic load refers to tasks with a heavy cognitive load by nature, and it can only be changed by changing what is learned or the knowledge of the learners. While extraneous cognitive load refers to a heavy load due to the way material is presented. A researched effect of extraneous cognitive load is the Split-Attention Effect, this considers examples where a cognitive load is imposed because learners must split their attention between different sources, for example, if a colleague is presenting a slideshow filled with words, members of the audience are going to have more trouble remembering what they say as their attention is divided between the presenter’s speech and the words on the slides. In my assignment different programming constructs and concepts are mixed, rather than separated, this is because to produce a game you have to build on top of other features, meaning that to have a working product at each stage we need to mix concepts. This has a chance of causing extraneous cognitive load because while a student is learning to use and apply global variables they are also expected to learn how to use If statements. To alleviate this in my assignment the instructions should be given to the students either printed or digitally so that they can refer back to them as they work. This should reduce cognitive load as it is easier for students to consider each new topic in isolation and refer back to previous examples when they return to continue learning a topic.

Another very positive aspect of the RP Python pathways is that at the beginning of each lesson, they will provide an end version of the project we are trying to achieve in that session. This is very engaging for students as the final product is normally interactive, so they enjoy running the code because they get excited about the final product this pushes them to get stuck in with the lesson to create their own version. Not only this but these examples also work well as an objective, it is much harder for a student to produce a specified application if they have no idea what it should look like, this encourages them as they complete the steps of the session as they can see the application take shape and they can understand if something is working as it is untended. Following this though, it is helpful as a mentor to encourage a learner to make the application their own by making some changes from the example program. I wanted to incorporate this into my project, so ideally learners should run the finished project before they begin to fill out the skeleton code.

PRIMM is a framework for teaching programming and stands for Predict, Run, Investigate, Modify, Make and it uses some of the ideas of UMC. These are the stages used for planning lessons to teach programming to all levels of learners. While my exercise does not use this framework, it does take some inspiration from it, namely the Predict and Run stages. I chose to add a few tasks in my assignment that follow the structure of predicting a change and then running the code to observe the actual change. I chose to do this because of the research showing that the predict and run stage is effective at engaging students (Sentence, 2019). In the Prediction stage of PRIMM it is important to keep the stakes low to aid confidence (Meer, 2014), to help do this I have categorised the tasks in my assignment, with tasks with a square box being mandatory “writing” tasks to get the code running, while the standard bullet points are “thinking” tasks to help to learn, the prediction tasks fall under these bullet points so while they are still should be completed, it should hopefully be clear to the learner that this task is not essential to getting a working final product so the stakes are low.

I also added some extension tasks, marked with a ‘!’ this was inspired by the adaptive teaching I had been implementing in regular sessions. I wanted the materials to change depending on who was completing them, so for a high achieving learner they could still learn rather than ending up in a situation where they need to wait for other learners to catch up. This should help create an inclusive session environment where all learners can benefit from the assignment.

One aspect of the RP pathways that I thought worked quite well was the quizzes they included at the end of each lesson; we didn’t always have time to complete them, but it normally worked as a good opportunity to bring the class together to discuss the options as they were suitably challenging. They also helped review the material we had covered to observe how effective the session was and what the weak areas of the class were. Thus, I added a quiz at the end of my own task and tried to make it sufficiently challenging to allow for collaboration between students. It focuses on the key aspects of computational thinking from the lesson, so should be used by mentors to evaluate how well the lesson worked, and perhaps provide ideas for topics that may need to be recovered in the future. This draws on social constructivism that was mentioned in the main report.

The term “Scaffolding” was first coined by Vygotsky in 1978 and refers to strategically adding support to a lesson. It has decades of research showing it produces confident independent workers (van de Pol, 2010) who defer to teachers or mentors less. But I have also seen it be discouraging to learners when completing the RP Intro to Python Pathway when they come across a task where scaffolding is removed too soon so they don’t feel equipped to tackle it and lose confidence in themselves for the remaining tasks. I tried to remove some scaffolding during the assignment by no longer providing code for tasks that I thought were suitably easy to solve. Also, if the students can’t solve the problem initially, they should have access to the instructions and so should be able to refer to the previous task code and find a way to fit it into the current task. This should avoid discouraging students as the scaffolding is removed within the lesson rather than across lessons as is the case in the RP pathways.

Evaluating My Exercise

I am satisfied with the predict and run elements of my exercise as well as the more “thinking” questions at large, as I think they encourage not just copying out code line by line, but understanding the mechanics of a programming construct better. Along these lines I have also removed some scaffolding during the task, this should not only encourage independence and confidence in students’ skills but also gives mentors a good way of gauging how well a student can apply a construct. This is useful because if they notice one student is struggling then extra explanations can be provided to support them.

I think my assignment does not have a big enough focus on Create. You could argue that tasks where I have excluded code snippets or allowed a student to choose a colour are creation, but I think this is a bit weak. I found it difficult to find an area to let students do some creating of their own within the context of my assignment, particularly as the aim is to get them a finished application in time. Create is particularly hard to implement into an exercise when I’m unsure if they have strong enough foundations to be able to create something of their own, this gave me an appreciation of 3,2,1…Make! as this aims to directly tackle this issue by having 3 lessons dedicated to Exploring, and Design before they come to the Invent task which is the most Create heavy lesson.

Also, I’m unsure how easy my assignment is to follow, the graphics package I use is quite finicky, so there are some lines that if written in the wrong place will break the app with unintuitive error messages. I ran through the instructions twice to make sure that I could get a working result out of it by following them, but this was difficult as I already knew what the code should be, so may have been lenient in how I interpreted the instructions.

Another big concern is timing, I gained lots of appreciation for the RP pathways during my creation of this material, in particular, managing to create an engaging project within a short time. I tried to make my game simple to avoid this, but I still ended up with a very long set of instructions. Ideally, the exercise would be able to be completed within 1 ½ hours, but as of now, I’m not sure if it will go over time. This will need to be verified with the actual group of students, and if it is over time perhaps it could be split across multiple sessions in some way.

Future Work

Because I was not fully satisfied with the Make aspect of my assignment, I have briefly thought of another lesson using similar code, in which I plan to build upon the current game to give more control over a player. This would mean a user would choose a few movements before the game began and then the player would perform these to try and navigate a level, to try and stimulate computational thinking (ideally this would include a mechanism to loop over some actions). After they have completed a basic level, I then planned on letting the students design their own levels to have each other or a mentor complete. From experience, I believe some of the ninjas would particularly enjoy making very difficult levels that are impossible to complete. In the future, I will most likely implement this to go along with the first lesson.

Finally, the most obvious future work would be to test this on a group of students to see how easy to follow the instructions are and how long they take to complete. This should help me refine the usability of my materials and leave less room for students to make critical errors that they need a mentor’s assistance to fix. Along these same lines, the RP pathway occasionally included common bugs and how to fix them, I would quite like to add this to my project after observing the common problems that the students run into.

References

1. Tomlinson, C. A. (2001). How to Differentiate Instruction in Mixed-Ability Classrooms. 2nd Edition. Chapter 1.
2. Boulden, D C. et al. (2021) Supporting Students’ Computer Science Learning with a Game-based Learning Environment that Integrates a Use-Modify-Create Scaffolding Framework. ITiCSE '21: Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V.1. pp129-135. DOI: https://doi.org/10.1145/3430665.3456349
3. van de Pol, J., Volman, M. & Beishuizen, J. (2010) Scaffolding in Teacher–Student Interaction: A Decade of Research. *Educ Psychol Rev* **22**, 271–296. DOI: <https://doi.org/10.1007/s10648-010-9127-6>
4. Sweller, J. (2011). Chapter two – Cognitive Load Theory, Psychology of Learning and Motivation. Volume 55, p37-76, DOI: <https://doi.org/10.1016/B978-0-12-387691-1.00002-8>.
5. Sue Sentance, Jane Waite, and Maria Kallia. 2019. Teachers' Experiences of using PRIMM to Teach Programming in School. In Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19). Association for Computing Machinery, New York, NY, USA, 476–482. DOI: <https://doi.org/10.1145/3287324.3287477>
6. Meer, N. M., Chapman, A. (2014) Assessment for confidence: Exploring the impact that low-stakes assessment has on student retention. The International Journal of Management Education, Volume 12, Issue 2. p186-192. DOI: https://doi.org/10.1016/j.ijme.2014.01.003.

Usage of AI tools

I didn’t use any AI tools personally, but I did use an analogy of a chef using containers of ingredients when describing variables in my assignment. This idea was partly inspired by another student from class, I can’t remember exactly but I’m fairly sure they had got the idea from ChatGPT, but the explanation was entirely my own.

Appendix

A screenshot of a video game

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated