

Feedback Summary on Cookie Learn

Students

Moth Dalziel (18, They/Them)

Chris' youngest sibling, ex-Higher computing student

Chris supervised a one-on-one lesson with Moth over a call, where Moth screenshared and could ask questions. Hints were provided from the teachers sheet in general, but some hands-on assistance was required where our materials still had some kinks needing worked out.

```
setInterval(() => {  
    level = parseInt(document.getElementById("auto-clicker-level-2").textContent)  
    autoClickerPower2 = (clickedLastSecond + autoClickerPower1) * level  
    score += autoClickerPower1+autoClickerPower2;  
    refreshCookieCount();  
}, 1000);
```

They had a particularly creative idea for an additional autoclicker, where they based the score increase on the score currently being collected from other sources. Their golden cookie design was also very creative.



Moth also kindly helped us to design the first slide of our presentation — thanks Moth!



They answered our feedback questions as follows:

What did you like about the materials? They managed to keep me engaged through the use of an enjoyable premise that encouraged me to play around with the bounds of what could be adjusted.

What did you not like? N/A (I'm struggling!!!!)

Was there anything you found confusing about the materials? I felt that the coding constructs behind optional tasks weren't explained as clearly as the rest of the material due to their nature of being optional, but this did heavily deter me from even attempting these tasks.

Did you like the design (website, slides, materials) I think the simplistic visual design of the website & slides is charming and not over-complicated which could run the risk of adding unnecessary confusion.

Do you think this kind of learning using games is a useful way to understand computing concepts? Absolutely, engaged and/or happy pupils are far more likely to want to try harder to understand concepts in which they are struggling.

Do you think teaching debugging more directly is a useful way to approach computing education? Yes, debugging specifically is an area of computing that requires a fair bit of practice to become proficient in and having a teacher lead a pupil through the early stages of that learning process is highly important.

Matthew Bastiman (21, He/Him)

Aeronautical physics (UOG)/Computer Science A-Level student

Matthew was asked to complete the workbook tasks and give feedback about the lesson as a whole from the perspective of a student, having previously studied Computer Science at A-Level. He commented positively on the use of a familiar game made the lesson feel approachable and commented on how the workbook was well designed with a simple/linear nature making it easy to follow. However, he did bring up some concerns about the lack of support/hints when a student didn't fully understand an early concept and, without just looking at the model solution, they were limited in how far through the workbook they could progress individually. Lastly, he did recognise that there could be some merit to teaching coding through debugging, however he mentioned that he thought there is a chance it could be too ambiguous for some learners if not set up in the right way. Overall, though, Matthew found the game-based approach motivating and useful.

What did you like about the materials? I liked how the lesson used something I know and enjoy (the Cookie Clicker game) to teach the ideas behind what makes a good (and well structured) HTML page/game. The simplicity of the design of the page helped when I wanted to look through it by myself to make sure I understood all the different components. The workbook content was manageable because of its mostly linear structure.

What did you not like? If I struggled to understand an early task, there wasn't much I could do (besides seeing a model solution). The summary tied the task to the technique, but didn't really help a huge amount if I still didn't understand the point of the technique.

Was there anything you found confusing about the materials? The actual materials were straightforward but maybe the distinction between `setTimeout` and `setInterval`. The definitions were fine but could have benefited from a more explicit comparison.

Did you like the design (website, slides, materials) Yes, it all looked clean and professional without being too dull. I liked the consistency of the sort of colour themes used in the slides and the css (maybe the workbook could have had some more effort put into how it looked).

Do you think this kind of learning using games is a useful way to understand computing concepts? I think so. Being able to take some time to explore and play around with the code and see how it changed the webpage (and what breaks it) did encourage me to really try to understand the reasons behind different parts of code.

Do you think teaching debugging more directly is a useful way to approach computing education? Code breaks a lot (especially when I write it) and I think being able to find and fix errors is a really important part of computing education because it helps develop the skill of fixing errors which everyone in computer science will need at some point. I do think that, for some people, debugging might seem like a bit of an ambiguous or even off-putting way of learning about coding.

Teachers

Christopher Robertson (He/Him)

Chris' Nat 5/Higher/Adv. Higher Computer Science Teacher

Mr Robertson ran through the materials from a teacher's perspective, analysing the materials from the perspective of a teacher about to run it for a class. He gave his feedback through a series of messages, giving suggestions for what we could change and more general criticisms of the project from the perspective of somebody who's run many real Higher classes.

He thought positively of the project as a whole, saying "The main bones of the project are good". He had some suggestions about how we could adapt the materials, e.g. including a prior learning section and linking in to previous learning by the class (though he recognised that this wasn't likely to be possible in a standalone piece of work)

His major criticism was that “a lot of less able pupils really struggle with this kind of self-directed work”. This is one of the major concerns we had about the materials before we had even fully finished them, and represents the largest flaw in our work

Other criticisms were about specific complicated wording, or inconsistent punctuation, which he said addressing could “really help improve literacy, especially in ESL pupils”. In response to these criticisms we’ve tried to soften the wording in places to make it less overly technical, and have attempted to make our formatting more consistent

He felt overall however that the materials were good, even saying “I’d quite like to do the project with a higher class and see how they get on”. Due to upcoming study leave for his pupils however this wasn’t possible in the timeframe we had left before the deadline, however we have given him full access to the materials should he wish to use or adapt them for his classes

His responses to our feedback was as follows:

What did you like? Materials were easy to follow and used a context that most pupils would be familiar with (At one point they found an unblocked version of cookie clicker and I had to ask the council to ban it because it was infuriating)

What did you not like? Material would be difficult for less able pupils to follow, perhaps there should be a second more scaffolded version for these pupils.

Was there anything you found confusing about the materials? Nothing

Did you like the design (website, slides, materials) Material were well designed and laid out, easy to follow

Do you think this kind of learning using games is a useful way to understand computing concepts? Yes, there’s a lot of merit in edutainment and tying difficult programming concepts to a familiar context is one way to bridge the gap in understanding

Do you think teaching debugging more directly is a useful way to approach computing education? Yes, it’s proven that pupils learn well by making mistakes and fixing them themselves

Natalie Dodd (She/Her)

Adam's mum, GCSE Religious Education Teacher

Ms Dodd also ran through the materials from a teacher's perspective, however she looked at them from the perspective of a general teacher (not as someone who knows the field of computer science). She gave feedback through voice messages, and provided some suggestions for improvements based on experiences teaching and covering other teachers' lessons.

Generally, she thought the materials were clearly written and had an engaging premise. In particular, she notes that the design was easy to follow and, from the perspective of a non-specialist, she felt the lesson could be used by any teacher, if there was some more scaffolding in place (for both teacher and student benefit).

To what extent do the materials illustrate the CS Engagement Principles^[1]?

Looking at the CS Engagement practices, I instantly see that the constant push for discussion does indeed 'encourage student interaction' in the lesson. This does bleed into the 'use well-structured collaborative learning' principle, however more work could be done to promote collaboration (outside of just discussion, group tasks of some sort).

To what extent are these materials suitable for the intended audience?

For an older class of students, the language seems to be appropriate. While I can't comment necessarily on the level of the material, the language and tasks seem to be a good balance of professional and approachable.

Are the materials suitable from an equality and diversity perspective?

There are no obvious stereotypes or assumptions made based on gender, race or other protected characteristics. One thing to bear in mind is that some students may not have played many/any games for one reason or another and the barrier this could create. Maybe other examples in the presentation (from non-game contexts) are something to be considered.

Please indicate any other points of feedback or suggestions you have for improving the materials

The lesson structure, on the whole, is a good start with some solid ideas. If a teacher were to use this lesson, maybe some suggested timings or pacing notes could be beneficial as I am unsure how long this would be expected to take on the whole (and individually task-wise). Some sort of glossary for key terms students might have common questions about may also help both the teacher and students with understanding.

Bex Godfrey

Adam's mum's friend, GCSE Mathematics/Physics Teacher

Bex also reviewed the materials from the perspective of the teacher, but, with prior coding experience, they were able to engage with and comment on the technical level a little bit more (in comparison to a non-specialist, like Natalie). She provided written feedback with a focus on its approachability, and its balance of challenge across the lesson/workbook. In a similar vein to Mr Robertson, they expressed how a prior learning section could be beneficial to the lesson.

Notes made from the feedback given by Bex are as follows:

- The materials were clearly written, and the premise of developing and fixing a game was engaging. The HTML template looked nice, and the design of all documents was well made and user-friendly. It might not have been the intent of the workbook, however a more thorough introduction of HTML and CSS along with clearer explanations throughout the tasks (especially those concerning the HTML and CSS explicitly) could help to provide a stronger scaffolding for the students before they look at the more complex tasks.
- Finding the right level of difficulty is definitely a challenge and this lesson could provide some more support for the students with less experience, and also those who might find it more obvious (and end up rushing through the tasks and not getting anything from it). The extensions and discussions are a good start to this. For the students who might struggle, along with the hints you have in the teacher notes, some step-by-step guidance could help the teacher gain a better understanding of the workbook/tasks and allow them to provide better guidance for those students.
- On the whole, the use of a game to teach computing is fun and could definitely be effective if done in the right way. It encourages students to problem solve actively. Some more consideration should be given to the students who might struggle to grasp the concepts, even those they might already be expected to understand.

Computing in the Classroom Students

Blythe Wray (They/Them)

D&D Accomplice and close friend

Blythe was given a functionally complete set of resources and asked to give as much feedback as possible from the perspective of another Computing in the Classroom student. They looked at the lesson as a whole (the presentation, the workbook, the teacher notes, the code files etc.) and worked through some of the tasks as if they were a student before looking back on the tasks from a teacher's perspective.

On the whole Blythe praised the materials for being engaging and well-structured, with a specific nod to the Learn-Do-Discuss model used throughout. They commented on how, having the project be related to a game would help with motivation, but also with understanding (as tying computational concepts with game mechanics provides a solid bridge between theory and practice of the ideas).

However, they did also provide some areas for improvement, especially with regards to the support provided for students struggling in the more complex tasks. They also discussed the lack of visual elements in some of the *Learn* segments of the presentation, commenting that it was somewhat text-heavy.

They were given the provided form to answer, as we had yet to finalise our final design questions and knew that they had the prerequisite knowledge to answer these questions (which we could not guarantee for other students).

Review of learning and teaching approach

To what extent do the materials illustrate the CS Engagement Principles^[1]? The Learn-Do-Discuss methodology and hands-on coding tasks complement principles such as Building Student Confidence. Think-pair-share group discussions help with peer learning (Grow Inclusive Student Community). Making it a game (Make It Matter) makes abstract concepts like event listeners more understandable. However, they could add scaffolding for complex tasks like golden cookie implementation. Consider including reflection prompts to tie coding tasks to real-world applications.

To what extent are these materials suitable for the intended audience? The tasks are appropriate for students familiar with HTML/CSS/JS basics as is taught in Higher. However, some tasks, for example JSON objects for upgrades, may require more materials and time.

Review of CS content

To what extent are the CS concepts covered in the materials clear and accurate? Very clear instructions for tasks and good visual aids as well. Covers a very wide range of topics as well as practical application of debugging. However there are some ambiguous terms that could use more clarity.

Review of inclusion

Are the materials accessible for the intended audience? Text-based code examples are easy to read and also screen-reader friendly. Could add some flow charts or maybe video explanations.

Are the materials suitable from an equality and diversity perspective? Gender neutral language is used throughout the materials, and there are no names used as the nature of the lesson doesn't make them necessary.

Recommendation

Adding more visual elements aside from just code examples may be beneficial and the PowerPoint is quite text-heavy, especially since you have the textbook sample to work with.

[1] <https://ncwit.org/resources/engagement-practices-framework/>