

Curriculum Development for Computer Science at Second-Level

Aaron Maher

1 BJC Work

Working with Dr Clare McInerney and Dr Chris Exton in Lero/CSIS, I was introduced to a collaborative project with Dr Keith Quille & team in TU Dublin. The goal of the project was to create a version of the "Beauty and Joy of Computing (BJC)" Curriculum for Irish second-level students. This content will be hosted in a national learning platform CSLinc (<http://csinc.ie/Home/CSLINC>). My role was to create activities and tasks for students to complete.

Task 2A

Now you've seen a visual cue, however in this case an audio cue might be more useful. In your CSinc course are 2 audio files, "clicked.wav" and "notclicked.wav". Click on these files to download them. Go to your Snap! window and click the "Sounds" tab near the "Scripts" and "Costumes" tabs (not the "Sound" tab with code blocks). Drag the downloaded files "clicked.wav" and "notclicked.wav" into the scripting area to add them to your project. Click the "Scripts" tab to return to your code.

Next go to the "Sound" tab containing the code blocks and grab the "play sound" code block. It should have an arrow on it that brings down a list when clicked. You should now see "clicked" and "notclicked" in this list. These are the music files you added to the project. Figure out where to place these in your script. Test your game to make sure it's working as intended.

When creating activities, I had to think about the best approach that would appeal to the students while also being good learning experiences for them. I tried to make tasks that would be practical and engaging. From my own learning experiences, I figured the best approach would be a clear and concise one. I explained what the student would be doing, why they would be doing it, and laid out the steps to accomplish it as clearly as possible.

I recognise the importance of giving students the freedom to figure out things by themselves, but I wanted to make sure that they had the foundation to do so. Therefore, I left any creative tasks towards the end of the lessons to be completed optionally.

2 Student Feedback:

During the time I was creating the materials, my sister was in Transition Year. Since this was the target audience of our materials, I decided to conduct a 1-on-1 evaluation with her. I asked her to go through the material by herself and tell me if she needed help with anything. Afterwards I asked her a series of questions I had prepared beforehand.

```
Q1) Were the instructions clear enough to get the work done? (Yes? No? Why?)
A1) "Yes, very much." (Edit: This is likely due to me helping her though, there were definitely some unclear moments.)
Q2) Did you have adequate time to get the work finished?
A2) "Yes."
Q3) What did you like best about the work?
A3) "Making my own costume."
Q4) What did you like least about the work?
A4) "The difficult words describing the actions." (I asked for some examples.) "Graphic effects."
Q5) Have you learned a bit more about the process of programming?
A5) "Yes."
Q6) Having participated in this evaluation, has your view on computer science and/or programming changed?
A6) "Yes, it's not as hard as I thought."
Q7) Would you recommend the use of this module in Transition Year for your friends?
A7) "Yes, if we can do it."
```

I focused on creating questions that would give me an insight into the positives and negatives of the material. Her answers, in addition to me watching her go through the work, gave me a lot of understanding as to what could be improved.

After the evaluation, I went through all the material we had created with my newfound knowledge. I put myself in the shoes of a student and made note of anything that could be confusing or wasn't coherent enough. I then presented my findings to our team so we could amend them.

```
Lesson Confusions/Inconsistencies:
>> Week 1 | Extension Activity #1
- Does not tell/show the student what block moves Alonso. As this is their first time experiencing programming it might be a good idea to show them an image of the blocks they need and just have them find them and recreate the image. It is shown in the solutions, but I feel it's a bit mean to have them wonder around if they're not sure what they're doing yet.
```

3 Sphero Work:

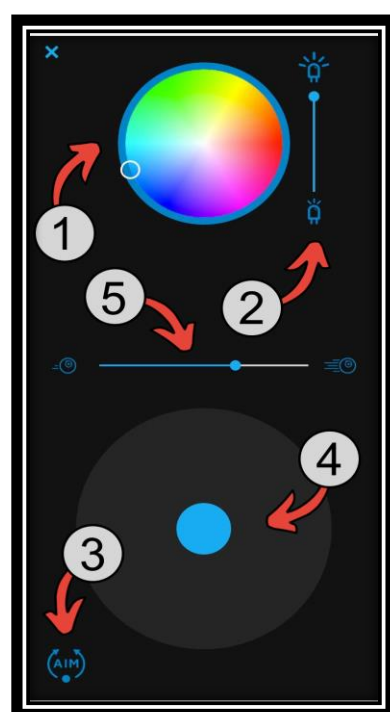
When I came back to university in September, I started working on a new initiative. Using what I had learned from the BJC work, I began creating educational materials for introducing students to programming, but this time using Sphero SPRK, a robot created by Sphero, Inc.

While I believe "Snap!" is a more realistic tool for introducing people to programming, Sphero SPRK is much more appealing as it is a tangible object that you can affect in real time.

When creating the materials, I decided to start off by letting them play around with the robot as much as possible. This lets students garner interest and understanding of how the robot works and is a much more fun approach than having them build programs they don't yet understand.

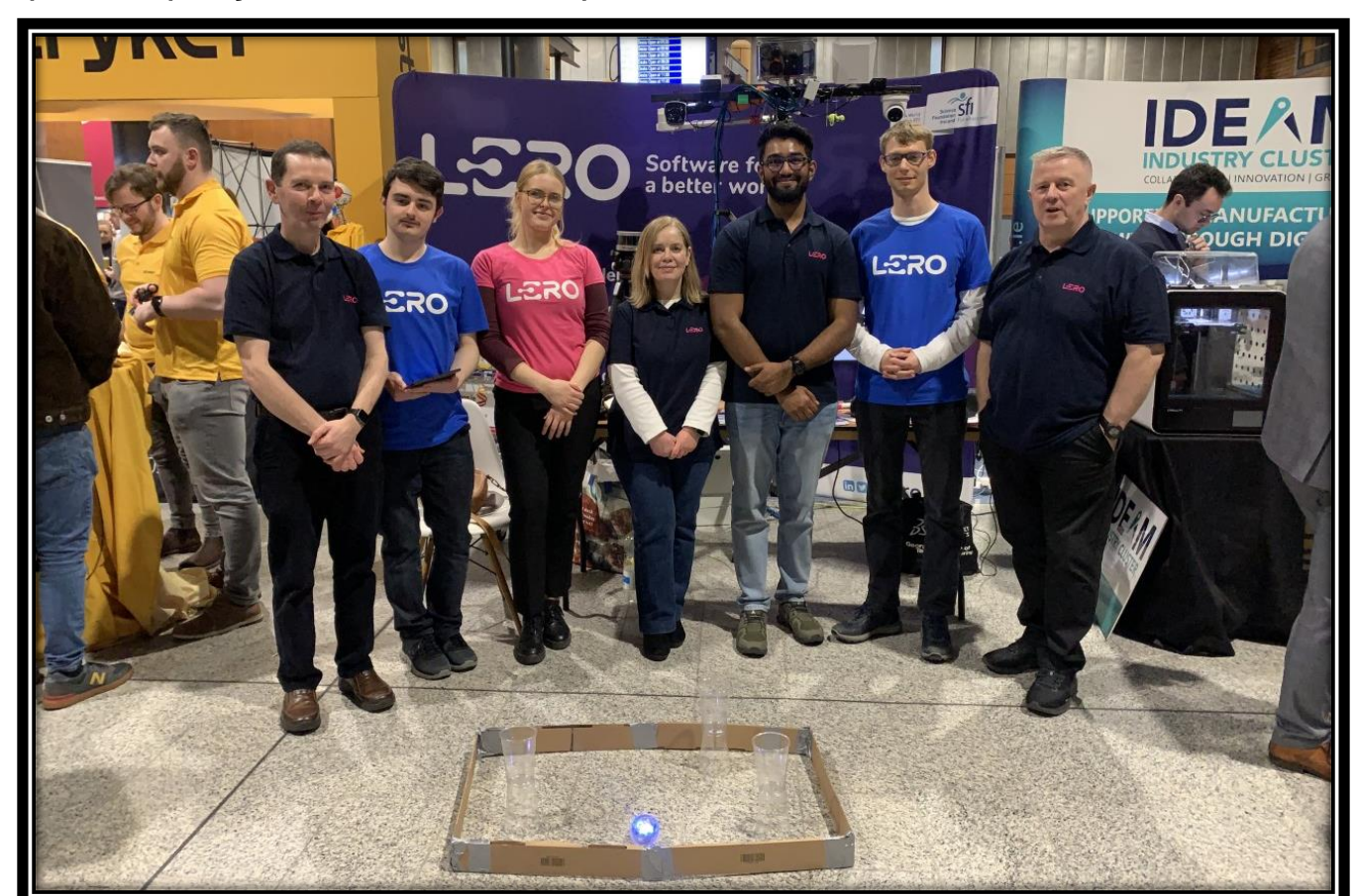
I also experimented with the format and style of my documents, trying to figure out the best way to communicate through paper how to interact with the app. I ended up with a figure heavy design, using a lot of pictures to demonstrate what I was talking about.

Control 3 is your Sphero's aim. When you start moving your Sphero, it needs to know which direction it will be travelling first, so you need to aim it toward the direction you plan to start moving towards. When you are aiming your Sphero, it will emit a blue light. This represents the back of your Sphero. Try aiming your Sphero until the blue light is facing directly towards you, then move onto the next control.



4 Sphero Demonstration:

On March 2nd, I attended the "Explore Engineering" event held in Shannon Airport with other Lero team members. This event was held to demonstrate and introduce technology-based fields to people (mainly students). I was asked to set up a small area for people to play around with Sphero.



Using a D.I.Y. budget playpen, I set up a small arena containing plastic cups for people to knock over using the Sphero robot. It turned out to be a big success and a lot of people were eager to try it out. There were few moments when the Sphero wasn't being played with by someone.

Due to the nature of a pop-up booth like this, it wasn't feasible to introduce anyone to the programming aspect of Sphero, so I just explained how to operate the manual controls. It was nonetheless a fun experience and at the least it gave me some verification that the Sphero robot was an appealing gateway into the field.

HOST INSTITUTION



PARTNER INSTITUTIONS



FUNDED BY:

