Performing engineering digital literacies in context

T. D. Drysdale and J. Scoles
The University of Edinburgh
Edinburgh, Scotland

Conference Key Areas: Integrated learning environments for the digital native learners, Open and online teaching and learning

Keywords: coding, computational thinking, digital literacies, integrated learning environments

ABSTRACT

In this paper, we present the results from an intervention that is based on engineering students being introduced to coding using a programmable sun-tracker simulation. Our study is underpinned by two arguments. Firstly, that it is helpful to introduce engineering students to digital literacies - such as computational thinking, automated control, and coding - at an early stage in their degree programme. This will better prepare them to respond to the complex data acquisition, analysis and reporting and machine control tasks that they are likely to face in their later professional work. Secondly, if educators reduce the time between teaching students coding concepts and then applying these to a learning activity clearly and contextually linked to their specific discipline, then this will increase student engagement with programming and digital literacies skills. An intended side-benefit of the intervention was to reduce anxiety in the target cohort, a group of first year students in their first semester, who had been tasked with producing a physical two axis sun-tracking solar panel, without any formal training in electronics or programming (aside from the intervention). The intervention was conducted by a morning briefing on sun-trackers (50 minutes) and an afternoon session (90 minutes) of group work with a purpose-built webapp that combined an interactive 3D animation of a two-axis sun-tracking solar panel, including simulated sensor data, Blockly graphical programming language, an output graph, and a scoring system. The evaluation was by survey before and after the intervention, and results indicated a reduction in anxiety by nearly two-thirds.

1 INTRODUCTION

In a recent survey of 22,000 UK post-compulsory education students, 82% said digital skills were important to their career, but only half of the higher education students thought their courses would "*prepare them well for the digital workplace*" [1]. The study also reported low usage of interactive digital media, such as simulations, that "*provide rapid intrinsic feedback*." Although these results were not specific to engineering, it can be argued that there is an even higher need for digital skills in engineering graduates. Therefore, one can expect that engineering subjects will be among the earlier adopters of new and improved approaches to developing digital skills in their students. Consequently, developments in engineering education in this area can potentially have a greater breadth of impact than just within engineering.

Yet what digital skills are most important to our engineering graduates? Confusion has arisen in the past due to a trend of lumping together everything 'digital' under the singular term '*digital literacy*', and then focusing only on a subset of the possible literacies. Goodfellow and Lea [2] highlight the distinction between 'digital literacy' as a range of skills or capabilities located 'within' an individual, and 'digital literacies' as a more complex engagement with a shifting, dynamic 'range of socially and culturally situated practices'. Therefore, we prefer the plural '*digital literacies'*, with an explicit indication of which specific literacy or literacies are being considered, for example, the current study on coding relates to digital creation literacy, see [3].

In this Paper, we present and evaluate an intervention intended to reduce the anxiety around the introduction of coding to a cohort of first year engineering design students. The cohort were predominantly interested in mechanical engineering, and their course culminated in needing to control machinery using code. In this study, 65% of participants reported that programming was important or very important to their future career.

We argue it is important to reduce students' anxiety and negative connotations towards digital creation (coding) due to coding's inherent value in affording good practice in professional work. For example, a widely used introductory tool to numerical analysis are spreadsheets. Yet data can easily be skewed through inadvertent errors which are difficult to spot because of the hidden-by-default nature of formulas. For example, in the Austerity paper by Reinhart and Rogoff [4], they made a simple error in setting the range of a summing formula that markedly skewed the public debt-to-GDP ratio by excluding five countries, Australia, Austria, Belgium, Canada, and Denmark, from the analysis [5]. That example is not an engineering analysis, but similar pitfalls await an engineer conducting calculations in a spreadsheet. On the other hand, using code for data analysis is much safer. Recorded data can be kept in a read-only file where it cannot be inadvertently altered, data-manipulations are not hidden, and there are additional internal consistency checks that can be performed on code or scripts that help alert a user to potential errors. A code-based numerical analysis can be applied to different data sets, and it can be applied to test data to validate the operation. By comparison, re-doing a spreadsheet to suit different data may introduce new errors. Students capable of performing numerical analysis in code are therefore better prepared for professional practice where it relates to data handling. Examples of useful languages for numerical analysis include Python, R, and Octave. For Python, the `numpy`, `scipy` and `pandas` libraries provide numerical, scientific and data handling respectively.

We also argue that the current teaching of coding and programming is often decontextualised from the social, political, cultural, and material practices of everyday engineering work. This can lead to students, who are unfamiliar with the concept of partitioning problems, to fail to intuitively grasp the relevance of a skill presented in the abstract. For example, an engineering student who is not studying electronics or computing may not appreciate the relevance of digital literacies to their future career, and a traditional approach to teaching programming is so abstract that it is unlikely to correct that deficiency until later in the course. This approach delays students' gratification in seeing an immediate benefit to understanding a seemingly complex skill, and thus disengages them from learning.

The rest of the paper is structured as follows. In Section 2, we argue for the reframing of skills to encourage the sector in moving beyond treating them unhelpfully as *cognitive knowledge to acquire* to better describe them as *sociomaterial enactments*. We describe how this reframing leads to subtle but important differences in the design of interventions that relate to developing digital skills that better prepare students for professional practice. In Section 3, we explain the methodology of our intervention. In Section 4, we evaluate the results and reflect on the wider impact for the engineering education field.

2 REFRAMING SKILLS IN THE DIGITAL ERA

The metaphors we use in our everyday language are extremely important, but often over-looked, in how we construct our ontologies about what 'knowing' and 'learning' is and does. The dominant rational, cognitive, and human-centred perspectives in engineering education position 'knowledge' as a reified and de-contextualised outcome, and metaphors of transfer encourage educational practices to simplify, codify and commodify knowledge. Engineering students are commonly treated as rational problem-solvers, those who "learn as individuals, largely by applying formulas and rule to the solution of structured, 'right-answer' problems" [6] and fail to take into account the "human social performance" in engineering [7].

This cognitivist model has been criticised for ignoring the social, material and cultural dimensions of knowledge and learning processes. This is also the case for most policy and research into digital literacies [8, 9] which fall short of highlighting the ways in which these dimensions are integral to how literacy practices are achieved and performed in work and learning. Gourlay and Oliver [8] argue that, in the translations of recent empirical work, "*the nuanced nature of the data has been* *rendered less visible, which results in framing digital literacies as 'quantifiable, relatively stable, generic and transferable entities', which have been 'abstracted away from any specific, situated instance [and] whose defining features can be identified as residing in the individual*."

Along with Gourlay and Oliver, we argue that a sociomaterial perspective can help to position digital literacies as successful, co-constitutive social *and* technical achievements, which foreground the more nuanced, messy and materially-mediated aspects of learning, and foreground the 'specific, situated instances' [8]. Sociomaterial approaches aim to de-centre the traditional emphasis on the individual human subject, which positions 'knowledge' as a static and abstract idea that exists independently 'out there' to be acquired. Instead of placing the human at the centre of inquiry, metaphors of relationality, situatedness and emergence are favoured [10]. These help to conceptualise knowledge and learning as being *performed*, or enacted, into reality, through relationships and connections with various artefacts – computers, teachers, webapps, observatory data sets, classroom spaces, timetables, rendering software, and so on. Therefore, learning is firmly situated in action, and emerges as a performance through different practices and processes. Importantly, the focus on *relationality* means that the theoretical gaze does not separate humans from non-humans, which troubles more established educational theories that may frame, for example, the classroom as a 'container' for educational practices and the learner as a separate 'free-floating' individual [11].

This has important consequences when thinking about designing simulation-based learning activities to teach coding and programming. If we shift from trying to design learning activities that are based solely on the concept of abstracted knowledge transfer and acquisition, to a focus on the material affordances of the simulation and learning activity, then students enter into learning spaces that invite tinkering, playfulness and surprise. Here, 'right answers' are not necessarily predetermined but can emerge through the material and social relations unfolding in the learning activity.

3 METHODOLOGY

The intervention was framed as a problem-based learning session [12], in which students study an open-ended problem in small groups. Our groups were typically of 6 – 8 students working together at tables equipped with a computer and large, wall-mounted monitor. Each group had 90 minutes to work on trying to control a virtual solar panel track a moving sun, to capture as much energy from the sun as possible.

3.1 Overview of the intervention

The students were in the first semester of an engineering degree, taking a newly developed course that had not run before. Approximately 100 students were in the class, of which N=35 consented to take part in surveys associated with the intervention. It was stressed that non-participants would experience the same education experience, with the same free and open access to the web application.

Students attended a briefing during a 50-minute lecture in the morning. Before revealing the intervention, consenting participants were surveyed to determine their attitudes to engineering, their studies, and the physical project that they were working on in their groups (see section 3.2). During the briefing, the students were introduced to the apparent motion of the sun relative to the earth, the atmospheric model in the simulation (that leads to lower light levels and red light at sunrise and sunset), and the cosine law for calculating the relative energy extracted from a solar panel, based on its angle relative to the sun.

Then the webapp was introduced, and a link given so that students could follow along in class if they wished. The webapp is described in Section 3.3, and combined a model of a solar panel, with student-editable code to control it. The location of the virtual solar panel was shown to the students (a neighbouring observatory, whose GPS coordinates had been entered in to the sun almanac controlling the virtual sun position). This real-life connection to the observatory acted as 'specific, situated instance' [8] and was intended to increase student engagement as it reduced the abstraction of the task. Then the use of the Blockly graphical programming language was demonstrated to the students, by using it to show that the virtual data obtained from tilting the solar panel matched the cosine law, as shown on the graph.

Next, the task for the afternoon was introduced, including a description of how to use light sensors and baffles to cast a shadow. An example tracking solar panel was shown, and the representation and function of the virtual light sensors was explained and demonstrated by plotting sensor data on the graph. Students were alerted to the fact that this was an embodiment of an important class of control problem (tracking a moving object, such as a ship targeting an aircraft, or a ground station targeting a satellite) and because it is an example of classic control problem that has much more immediate relevance for students, because it is a visible, familiar phenomenon (motion of the sun) than tracking a satellite which they cannot see. This allowed the students to concentrate on understanding the technical aspects of controlling a tracker, rather than worrying about what was being tracked.

The afternoon session explicitly did not have a traditional lab sheet to encourage exploration. Instead, verbal assistance was given as appropriate to each group. The cohort was split into two, with back-to-back, 90-minute sessions, due to the limited capacity of the group-working room. The students in the first session did not have the opportunity to brief the incoming students for the second session.

The main goals of the session were to:

a) Understand that a solar panel that tracks the sun will collect more energy;
b) Recognise that the magnitude of the difference between two sensor values indicates the magnitude of the pointing error to the sun, in one rotational axis;
c) Recognise that four sensors can track the sun in both azimuth and elevation;
d) Appreciate that the strength of the corrective drive signal applied to the machinery needed to depend on the size of the error and the characteristics of the machinery;

e) Appreciate that useful solutions do not have to give perfect behaviour, rather that meeting specification can be enough in the early stages of development.

At the end of the afternoon session, students completed another survey on their experiences, relating to satisfaction of use, perception of quality of the webapp, and their attitude to the project following the intervention. Given the longer-term goal of having the students perform well in their physical project, no formal collection or accounting of the results was to be taken, to avoid giving the impression that the session was anything other an exploratory experience intended to enhance their preparation for the physical project.

3.2 Survey design

Two surveys were conducted. The first survey was administered before the briefing, to gauge interests, experience of and attitudes towards technology and engineering. It was intended to reveal whether students were operating altruistically (malleable intelligence mindset [13]; focused on learning for its own sake) or strategically (fixed intelligence mindset; aiming to maximise performance), although results were within norms expected and not further explored in this paper. The second survey was offered at the end of the intervention, and asked the students to reflect on their experiences, as well as to indicate their emotions before and after the intervention, using the 21-factor epistemic emotion scale [14].

3.3 Webapp design

The user interface to the webapp is shown in Figure 1, with annotations showing where the WebGL 3D model, Blockly code, sensor model, graph data and menus are. The

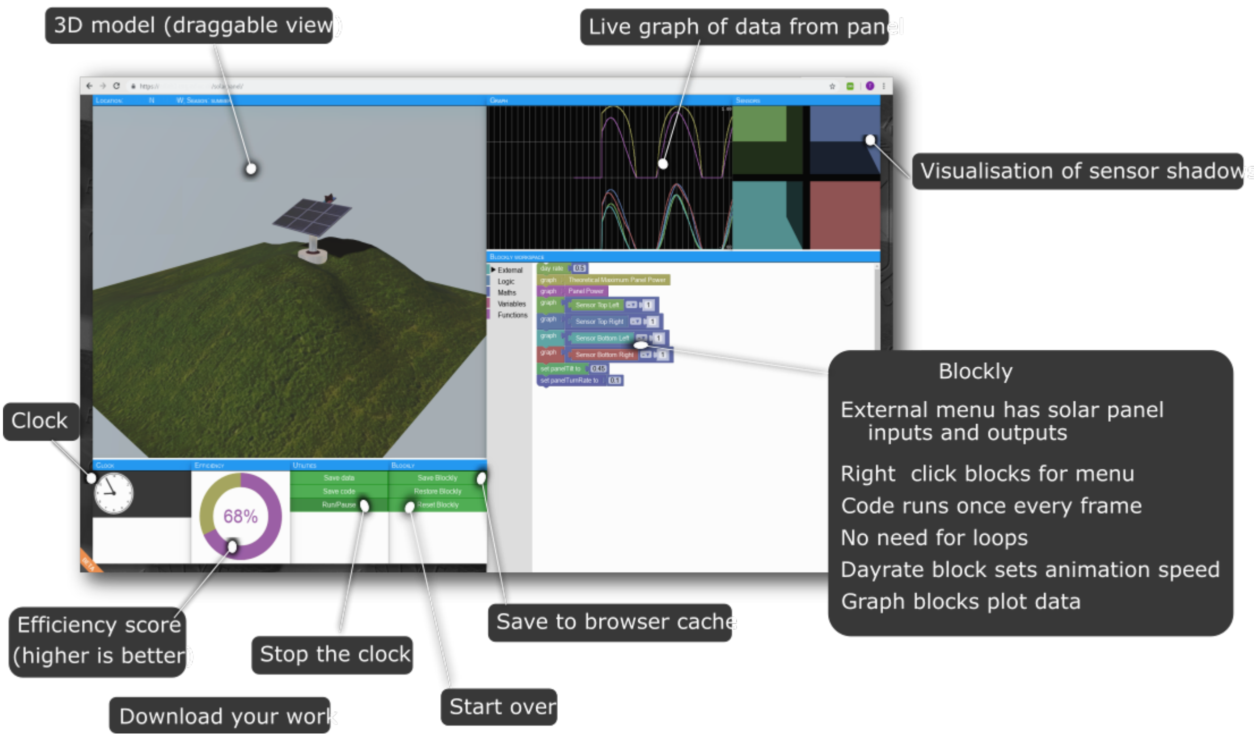

Fig. 1. Webapp interface (annotated)

Webapp was implemented in Javascript and served from a static HTTPS server. The WebGL model was based on a 3D model created from scratch in Blender, using public domain textures. The animation and control were achieved using THREE.js. The sensor shadow view was not rendered well by WebGL, so it was implemented as a canvas so that well-resolved shadows could be produced efficiently. The standard Blockly blocks were supplemented by blocks that could read and write to and from the global variables controlling the animation, and reporting on the state of the simulation (e.g. intensity of sun, power generated by the solar panel, sensor signals from the sensors), inspired by integrations of Blockly with virtual and remote laboratories [15,16].

4 EVALUATION

4.1 Usage

Figure 2 shows the usage of the webapp, via the count of analytics logging messages A small number of students followed along in the class. The WebGL simulation is known to be intensive on battery life, and since this was the first lecture of day, it was only fair to warn students of this. As a result, it is likely some students chose not to follow along so as to preserve battery life in their laptops. Here, the social and material relations in the students' learning practice is highly evident: the ability for students to access power points in interactive teaching sessions will shape the outcome of the learning process. Most students first interacted with the webapp in the afternoon. A number of students returned to the webapp the next day, showing that interest in the problem had been piqued. This is also consistent with the sentiment recorded in the Jisc survey that over 70% of HE learners agreed that with digital resources [1], *"I can fit learning into my life more easily",* and *"I am more independent in my learning."* Again, a sociomaterial perspective highlights that the classroom is indeed not the container in these instances of learning; different learning enactments are afforded when the student and simulation can work together remotely via the webapp and internet.

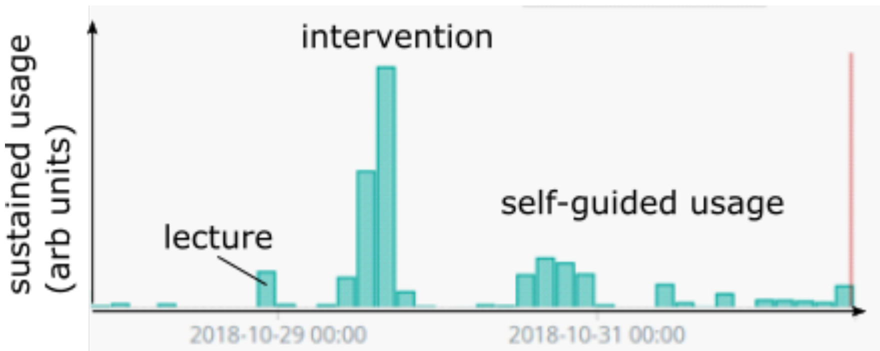

Fig. 2. Usage of the webapp

4.2 Attitude

Epistemic emotional data was collected on a 21-point scale, and collapsed into the seven main groups for analysis, as shown in Figure 3. The main outcomes of the session were that curiosity fell slightly (compensated for by an increase in surprise), while anxiety fell significantly. Overall, positive emotions reported remained constant, while there was a reduction in negative emotions (Fig. 4), as intended.

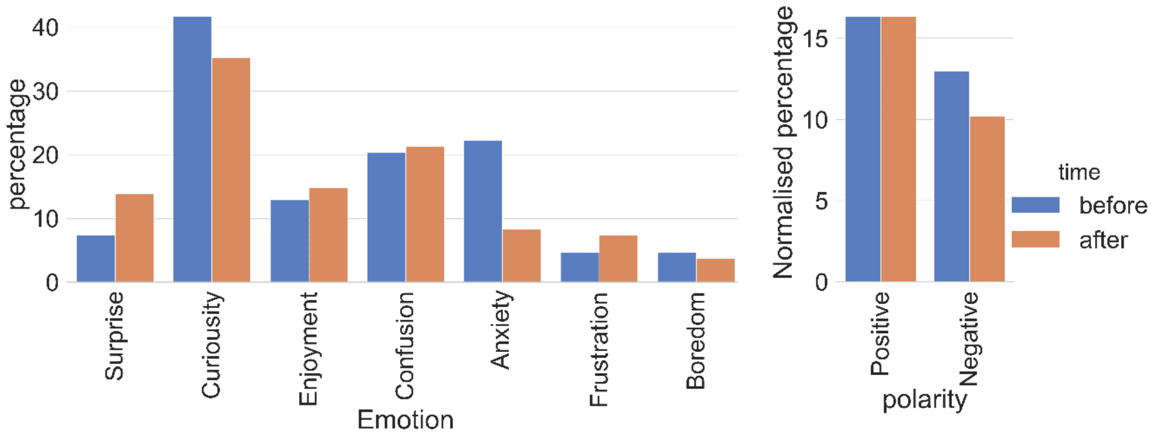

Fig. 3. Epistemic emotions before and after intervention

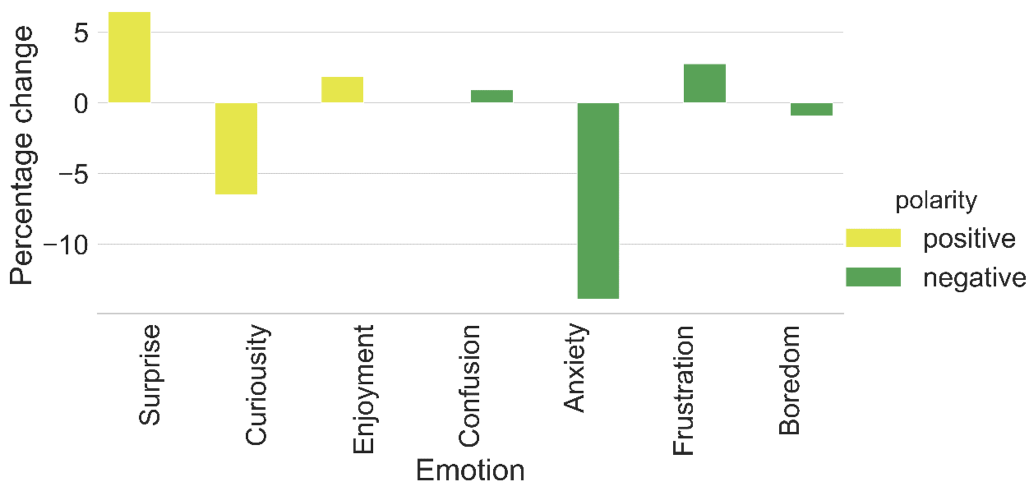

Fig. 4. Percentage change in epistemic emotions before and after intervention

4.3 Achievement

The outcomes from the session were not explicitly assessed. Inspection during the sessions revealed that the majority of students achieved a partial solution, with some achieving near-complete solution. One group did not proceed far beyond the initial screens. One of the most creative solutions came from an individual who came late and was working alone, who used a stochastic rather than deterministic approach. They pointed out the specification and scoring did not require an efficient solution!

For their final physical deliverable at the end of the course (a working solar panel tracker with two-axis control), we suggested to groups that they either use an Arduino microcontroller or an analogue control circuit (of which an example circuit diagram was supplied). Over two thirds of the groups chose to program a microcontroller to control their solar panel tracker, which was a higher percentage than we anticipated given the complexity entailed. This was not inconsistent with the reduction in anxiety that occurred after the intervention.

4.4 Improvements to the intervention

Some students were slowed down by a flaw in the interface. Looping constructs were inadvertently made available to the students via the Blockly toolbox. If they used them, the single-threaded Javascript interpreter became stuck in evaluating the code in an endless loop and could no longer able to update the rest of the user interface. This forced the Author to reveal to the affected students some details of the code's architecture. This prompted useful conversations that opened up the black-box of programming for the students, which showed them how the effects of human and non-human's relations were "constitutively entangled" in these coding practices [17].

An improvement would be to run the Blockly evaluation in a separate thread (via using a Web-worker). Explicit synchronisation with the animation would be required, although the option could be retained to do this in one or more ways. For example, synchronisation could be based on a notional computational speed for the Blockly controller made relative to the animation speed, so that performance is invariant between different computers or smartphones that the students are working on (important if solutions are being submitted to a server for automatic marking). Alternatively, an `interrupt` model could be introduced that triggers on every animation frame, although this requires additional explanation to the student. Both models are representative of real-world programming practices, but consideration needs to be given to ensuring the initial experience is relatively straightforward while also allowing deeper exploration.

A related issue is stopping the evaluation function cleanly when the code needs updating. This can potentially be handled by decorating the implementation of loops with boiler plate code that will cause any loop to exit when the code is flagged as being stale. The ability to write one's own implementation of the code underlying each Blockly block is of great help in solving these sorts of issues.

A relatively straightforward improvement to propose is automated testing of both the simulation, the data it provides, and the running of various Blockly programmes, so that modifications can be checked to ensure no new bugs are created. This becomes more relevant when the core code forms the basis for a remixable starting point for other activities.

A final improvement is to create an equation parser block so that more experienced programmers are not frustrated by the cumbersome Blockly syntax. This is important for allowing other staff to more fully enjoy the experience and probe for weaknesses and inconsistences in the underlying simulation model.

4.5 Wider impact

The intervention can be customised to suit other engineering or science specialisms by modifying the model, simulation engine, and user interface. This requires some technical input. It is not necessarily expected that academics themselves will already have, or wish to develop, the skills required to adapt the example code. We argue that education institutions should consider employing in-house resource to handle this type of digital content creation in support of curricula, analogously to the way that staff are already employed in machine and electronic workshops to design, develop and test physical artefacts in support of practical work. There is no suggestion that an academic must personally machine the artefacts used in their laboratories, although they may very well have designed, specified or helped evolve them. Admittedly the skill-sets are entirely different for digital content creation, but in our view, it is equally valid to consider a mixture of academics and non-academic staff working together to support the digital side of teaching delivery. Finally, we have started to show that a sociomaterial approach to understanding digital literacies shifts how we conceptualise learners as individuals who rationally acquire knowledge to understanding learning as a social and material performance, which should focus on specific, situated instances that are highly contextualised. In these properly supported spaces, experimentation and playfulness can emerge, which we expect to lead to a reduction in student anxiety, and increase graduate preparedness.

5. CONCLUSION

A sociomaterial approach to teaching digital literacies, such as coding, motivates the creation of interventions that all students to contextualise engineering practices that are new to them. We created an online simulator to allow students to tinker with the concept of controlling a solar panel. The exercise was not assessed, but out survey results indicated a reduction in student anxiety relating to their overall task of producing a real solar panel at the end of the course.

Acknowledgements

We acknowledge the staff, technicians, and tutors involved in the wider preparation and delivery of the design course in which this intervention took place.

REFERENCES

[1] Newman, T., and Beetham H. (2017), "Student digital experience tracker 2017: the voice of 22,000 UK learners", *JISC,* UK, June 2017.

[2] Goodfellow, R. & Lea, M. (2007) Challenging E-learning in the University: A Literacies Perspective, Open University Press, Maidenhead.

[3] JISC Building Digital Capabilities Framework, online: http://repository.jisc.ac.uk/6611/1/JFL0066F_DIGIGAP_MOD_IND_FRAME.PDF, Last Accessed 24 April 2019

[4] Reinhart, Carmen M. and Rogoff, Kenneth S. (2010): "Growth in a Time of Debt". *American Economic Review: Papers & Proceedings*, nr. 100 (May).

[5] Herndon, Thomas, Ash, Michael and Pollin, Robert (2013): "Does High Public Debt Consistently Stifle Growth? A Critique of Reinhart and Rogoff". *Working Paper Series* nr. 322 (May). Political Economy Research Institute, University of Massachusetts Amherst.

[6] Sheppard, S., Macatangay, K., Colby, A., & Sullivan, W. (2009). *Educating Engineers: Designing for the Future of the Field*. San Francisco, CA: Jossey-Bass, p 12.

[7] Trevelyan, J. (2010). Reconstructing engineering from practice. *Engineering Studies*, *2*(3), p187

[8] Gourlay, L., and Oliver, M. 2016. It's Not All About the Learner: Reframing Students' Digital Literacy as Sociomaterial Practice. In Research, Boundaries, and Policy in Networked Learning. Eds T. Ryberg, C. Sinclair, S. Bayne and M. de Laat. Cham: Springer International Publishing, pp. 77-92.

[9] Knox, J., & Bayne, S. (2013). Multimodal profusion in the literacies of the Massive Open Online Course. Research in Learning Technology, 21. doi: http://dx.doi.org/10.3402/rlt.v21.21422.

[10] Fenwick, T., & Edwards, R. (2010). Actor-Network Theory in Education. Abingdon, Oxon: Routledge.

[11] Fenwick, T., Edwards, R. & Sawchuk, P. (2011) Emerging Approaches to Educational Research: Tracing the sociomaterial, Routledge, Abingdon.

[12] Perrenet, J.C., Bouhuijs, A.J. & Smits, J.G.M.M. (2000) The Suitability of Problem-based Learning for Engineering Education: Theory and practice, Teaching in Higher Education, 5:3, 345-358

[13] Dweck, Carol S. (2002), Messages that Motivate, in “Improving Academic Achievement,” Joshua Aronson (Ed)

[14] P. Reinhard, Vogl, E., Muis, K. R. & Sinatra, G. M. (2017) Measuring emotions during epistemic activities: the Epistemically-Related Emotion Scales, Cognition and Emotion, 31:6, 1268-1276

[15] Galan, D., Heradio, R., de la Torre, L., Dormido, S., & Esquembre, F. (2018). Web experimentation on virtual and remote laboratories. In *Online Engineering & Internet of Things*(pp. 205-219). Springer, Cham.

[16] Galan, D., de la Torre, L., Chaos, D., & Aranda, E. (2018, March). Combining Virtual and Remote Interactive Labs and Visual/Textual Programming: The Furuta Pendulum Experience. In *International Conference on Remote Engineering and Virtual Instrumentation* (pp. 100-109). Springer, Cham.

[17] Orlikowski, W. J. 2007. “Sociomaterial Practices: Exploring Technology at Work,” Organization Studies (28:9), pp. 1435-1448.