# Final report for eSTEeM project:

# Developing programming problem-solving skills using individualised screencasts

Keywords: programming, problem-solving, TMA feedback, screencasts

Project lead: Sarah Mattingly

Report submission date: 11th May 2020

Project members: Christine Gardner, Richard Walker

Contact: sarah.mattingly@open.ac.uk

## Executive Summary

This project set out to investigate whether tutor feedback via video screencasts could help novice computing students develop skills in programming and in problem-solving, and whether creating such screencasts was feasible for tutors in the normal course of correspondence tuition. TM111 tutors provided short screencasts providing feedback on students' TMA answers, tailored to each individual's misconceptions, mistakes and areas for improvement.

The project was intended to address two main issues.

1. Problem-solving is increasingly recognised as a key skill required of programmers (Loksa, Ko, Jernigan, Oleson, Mendez and Burnett, 2016). Yet novices often find problem-solving difficult, and educators find teaching problem-solving difficult, not least as different people may solve problems in different, but equally valid ways. TM111 was the first OU computing module explicitly to embed teaching of problem-solving techniques. Still, some students struggle to develop successful problem-solving strategies, as evidenced by TMA results. This project explored how tutors might aid such students to become better programmers, more able to continue successfully through a computing-related degree, by providing them with audio-visual insight into how an experienced programmer (the tutor) would solve a problem, following the lines of the student's own initial thought processes.
2. As the OU's introductory computing module TM111 is the first point at which novice programmers meet key programming concepts such as iteration and selection. These 'threshold' concepts are well-known areas of difficulty for novice programmers as has been found on predecessor computing courses such as TU100, and more widely (Rountree & Rountree, 2007). This project addressed students' misconceptions about such concepts.

## Findings

Screencasting individual TMA feedback was found to be useful for students and viable in terms of time and effort for tutors in specific circumstances, notably where the screencast:

- focusses on developing students' transferrable programming concepts and skills
- exploits the visual aspects of code creation
- is created for solutions where student has invested time and effort and got somewhere, that are somewhat correct but sub-optimal or wrong in some respect.

We found that whilst screencasting technologies themselves are freely available and easy to use, delivering screencasts to students is not straightforward due to file size and security concerns. The optimum delivery route currently seems to be YouTube; however care must be taken to ensure screencasts are unlisted for privacy, and even so some tutors and students may have other concerns about using YouTube. The development of a new eTMA system may offer additional options.

## Aims and scope of the project

The overarching aim of this project was to increase retention and progression within computing, aligning with University and STEM strategies. TM111 is a key level 1 computing module, taken by approximately 4600 students per year, mostly as their first computing module. Approximately a third of the module focusses on introductory programming ideas, and on problem-solving in programming, using the bespoke OUBuild visual (drag and drop) programming environment. The next module for most TM111 students is TM112, whose focus is almost entirely programming, in the text-based language Python but building on students' prior experience of programming in TM111. Hence it is vital that students get a good grounding in basic programming and problem-solving ideas and techniques in TM111.

As well as contributing to the wider body of knowledge in teaching problem-solving and programming, this project built on previous innovative problem-solving teaching undertaken at the OU, notably the TU100 Programming Bootcamp (winner of a 2017 OU Teaching Award). It also connects with previous and current eSTEeM projects in this area e.g. (Thomas, Kouadri and Jefferis 2018).

The project drew on video feedback research in other disciplines e.g. in the OU's WELS faculty (Harper, Green, and Fernandez-Toro 2012); and on research in other HE institutions (e.g. Atfield-Cutts, Ollis, Coles and Mayes 2016). Typically, such research in the context of computing has focussed on understanding concepts rather than problem-solving; and has also involved more experienced students.

We expected that the project might also address recognised issues with students interacting ineffectively with TMA feedback (e.g. Walker 2007): student engagement with TMA feedback OU-wide has been shown to vary with some students not reading tutors' comments at all. This project aimed to explore whether some students prefer feedback via screencasts.

The project aimed specifically to address the following.

- Do students understand concepts more deeply and are they more skilled at programming problem-solving, as a result of the screencasts?
- What kind of screencast content is most beneficial?
- How time-consuming for tutors is creating individual screencasts?

The final point was a key one. We wanted to assess whether screencasts might eventually be provided voluntarily by tutors in the normal course of their correspondence tuition, in which a TMAs should take on average around an hour each to mark. This meant that we focussed on pedagogical approaches that were expedient for the tutor as well as effective for the student. It also meant we had to consider carefully the technologies involved in order that both the tutor's initial investment in learning to create screencasts and their ongoing use of the technologies were minimally demanding.

### Constraints

On TM111 there is only one programming TMA, so improvement in student performance within TM111 could not readily be measured. Hence the project was evaluated qualitatively. If screencasting were deemed to be viable on a larger scale on TM111 we intended further down the line to see if impact might be detected on student performance on follow-on modules, notably TM112.

## Activities

We initially reviewed (a) relevant literature and (b) technologies for screencasting production and delivery. On this basis we drew up initial guidance in the form of a briefing document for Stage 1 tutors (Appendix 1).

The project then ran in two main stages: Stage 1 in the 19D presentation and Stage 2 in the 19J presentation of TM111.

Stage 1 was a 'free-rein' exploration of screencasting individualised TMA feedback, in which five selected, highly-experienced TM111 tutors were funded to create screencasts in response to their students' submissions for TMA02 (the programming TMA). The tutors decided for themselves which solutions to provide screencasts for, on the content and structure of their screencasts, and on the screencasting technologies. The aims of Stage 1 were to assess the potential for screencasting individualised TMA feedback on a larger voluntary scale, and (if that were thought viable) to form guidelines for a wider cohort of tutors on pedagogical and technological screencasting methods.

At the end of Stage 1 the project team reviewed all screencasts and related student TMA answers; surveyed all students who had received screencast feedback; and held a debriefing session with the tutors involved. On this basis we drew up guidelines for Stage 2.

In Stage 2 two volunteer tutors, both of whom had participated in Stage 1, used these guidelines in producing screencast TMA feedback for their students. The aims were to assess whether, using the experience and guidelines from Stage 1, tutors might usefully and practically incorporate screencasts in the normal course of their correspondence tuition.

At the end of Stage 2 the project team reviewed all screencasts and related student TMA answers; surveyed all students who had received screencast feedback; and sought feedback from the tutors involved. We amended the guidelines slightly in accordance with student and tutor feedback; and selected two screencasts as illustrations of what could be achieved.

## Changes from initial plans

- Stage 1 was initially planned to run in 18J, however SSRP approval was withheld pending resolution of concerns relating to our original intent to interview students. Hence Stage 1 was delayed until 19D.
- The survey of Stage 1 students produced useful data in response to multiple choice questions but no free text comments. For Stage 2 we revised the survey with LDS assistance, to include fewer multiple-choice questions and more scope for free text expansion of answers.
- We also hosted the surveys differently. The Stage 1 survey was hosted on the Jisc online surveys website; in an attempt to increase the response rate, we ran the Stage 2 survey on the VLE, linked directly from the TM111 Study Planner.

## Activities going forward

The guidelines, along the two illustrative screencasts, will be made available on the TM111 tutors' website and tutors in every presentation will be encouraged to use them in providing individualised screencasts in the normal course of their correspondence tuition.

We will monitor the adoption of screencasting by TM111 tutors and assess whether and how screencasting individualised TMA feedback might be adopted by other computing modules.

## Findings

### Initial exploration

Many screencasting technologies exist. TM111 tutors are typically technologically literate, and some have used screencasting in other contexts, and we didn't want to constrain them unnecessarily. So we did not mandate any particular screencasting technology but instead after an initial exploration suggested some freely available options which we determined would be sufficiently functional but straightforward to use for tutors who happened not to have prior experience (see Appendix 1).

A secure mechanism for delivering screencasts, outside of the eTMA system, was required, due to the file sizes involved. ZendTo, an OU hosted secure file transfer system which allows the transfer of files up to 4 GB, was the obvious option at the time. Using ZendTo involves the sender uploading a file to a secure server whereupon the receiver is notified by email and must follow a link to retrieve the file within a 14-day time limit.

### Stage 1

#### *The screencasts*

55 screencasts were created for 31 students. Approximately 75% of these students collected their screencasts from the ZendTo system. Screencasts ranged in duration from 5 minutes to 40 minutes.

The screencasts varied greatly in style and content. Their focus could broadly be classified as follows:

- interpreting the question (for example, showing the question text and highlighting aspects that the student had overlooked);
- creating an algorithm (starting with a student's algorithm and amending it to an appropriate level of detail or to correct it; or talking through a sample algorithm);
- building and testing code (for example showing a student's incorrect code and demonstrating using test data why it was wrong, then re-building it correctly).

Some tutors focussed on a single aspect in each screencast, providing written TMA feedback for other aspects. Others created screencasts feeding back on a student's entire answer. One tutor created screencasts for all solutions even those that were entirely correct in which case he showed the code and talked through it, confirming that it was correct, congratulating the student, emphasising the importance of certain aspects (e.g. use of variables, variable names), and suggesting minor improvements/extensions. One tutor produced a generic screencast showing a correct solution being designed and created, which he inserted into each individual screencast after initial discussion of the particular issues/merits of the student's own solution.

Tutors were in general congratulatory, friendly and supportive. They were informal, countenanced imperfections (small slips of the tongue etc) and interruptions (barking dog, ringing phone etc).

#### *Student feedback*

All 31 students for whom screencasts had been created were asked to complete a short (5-10 mins) online survey a week or so after their marked TMA was made available to them.

- 6 students responded to the survey, a fairly small response rate despite reminder emails. One possible factor was that TMA timing in the 19D presentation required the survey to be conducted during the summer holidays.
- Of these 6, 1 had not collected their screencast before the 14-day time limit imposed by ZendTo.
- The remaining 5 were all highly positive in response to screencasts. They all agreed or strongly agreed that the audio-visual nature of screencasts made their tutor's feedback easier to understand compared to written feedback; that screencasts made the TMA feedback more interesting; and that their time watching the screencasts was well-spent. 4 of the 5 felt that watching the screencasts increased their confidence in tackling programming tasks.
- None of the respondents provided free text comments.

### *Tutor feedback*

- Tutors who focussed their efforts rather than providing screencasts for all solutions reported that they chose solutions:
    - for which they felt audio-visual input would add value;
    - where their feedback lent itself to the visual (e.g. amending an algorithm or re-constructing code);
    - which were very confused and where written feedback would have been lengthy and laborious to create compared to showing and talking.

  There was consensus that the visual nature of TM111's OUBuild programming environment lent itself to screencasting feedback more readily than might other programming environments.

- Not unexpectedly screencasting took longer than written feedback on some solutions especially to begin with, but got quicker with practice. In some cases, for example with solutions in which multiple underlying misconceptions were evident, screencasting was considerably quicker. Time could be saved by producing generic screencasts, or partially generic screencasts, but these were felt to be of less value than properly individualised feedback.
- Using ZendTo was sometimes frustrating and could be time-consuming. The time limit imposed by the system was unhelpful. Some students did not pick up their screencast within the time limit and tutors had to remind them and sometimes resend. Although files could be compressed sizes were still large in some cases and uploads took time. Some of the tutors were concerned that having to collect screencasts posed an additional barrier for students.
- The production of screencasts imposed more practical constraints on tutors than writing feedback, due to the desire to be in a quiet environment free from interruptions.

### *Stage 1 conclusions*

Based on student and tutor feedback, and on our observations, the project team drafted guidelines for screencasting in Stage 2.

In summary, we concluded and reflected in the guidelines that for screencasting to be do-able by tutors in a reasonable time, and to target students most likely to benefit from them, screencasts should:

- focus on developing students' transferrable programming concepts and skills;
- be created for solutions where visual and audio feedback might most readily add value – where it is easier to talk and show what is meant, where in an ideal world the tutor might like to sit alongside the student and talk them through their work;
- exploit the visual nature of OUBuild, using it as a vehicle for demonstrating problem-solving;
- be created for solutions where it appears the student has invested time and effort and got somewhere, that are somewhat correct, but not perfect. In such situations we believed students were likely to be interested in feedback and most likely to benefit from it.

We recommended screencasts should not:

- be created for solutions in which students got largely nowhere, in which case the tutor would essentially be creating the solution program from scratch. Though this might be useful for some students it would be time consuming for the tutor and not really individualised; also we concluded that students who struggle to this extent were less likely to be inclined to access the feedback;
- be created for solutions that were largely correct, in which case screencasts added little value for the student;
- focus on showing students how to use specific features of OUBuild that did not relate to transferrable skills.

We also recommended that tutors try providing screencasts to students as YouTube unlisted videos rather than using ZendTo. Having explored this mechanism, it was found to offer both simplicity and privacy (such a video can only be accessed via a link provided directly from tutor to student).

## Stage 2

### *The screencasts*

Four screencasts were created for four students. They focussed on the following:

- two on improvements to working code (6 and 10 minutes long respectively);
- one on identifying and correcting a logical error in code involving selection (10 minutes);
- one on identifying and correcting arithmetic errors in code (2 minutes).

Analysing the screencasts and the associated TMA solutions confirmed that the tutors had adhered to the guidelines (though one tutor used ZendTo rather than YouTube – see below). The solutions for which screencasts were provided were those where students had evidently invested in achieving working solutions but where there were significant errors or improvements that could readily be illustrated audio-visually. It was interesting to note that unlike Stage 1, all of the Stage 2 screencasts focussed on the code rather than the written algorithms, reflecting (we felt) the affinity of screencasting with the visual nature of code creation in OUBuild.

### *Student feedback*

All four students for whom screencasts had been created were asked to complete a short (5 mins) online survey a week or so after their marked TMA was made available to them.

- 2 students responded to the survey. These were the two students whose screencasts had demonstrated ways to improve working but suboptimal code.
- Both students had received their screencast via YouTube. Both confirmed that they experienced no problems with this delivery medium.

- Both students said that their screencast clarified their understanding of at least one idea or technique. For one student the idea/technique was how to construct code in OUBuild; for the other it was how to write an algorithm (i.e. the 'idea' underlying the code).
- Both students said that it was It was easier to understand screencast feedback than written feedback.
- Free text comments included:
    - *'[the screencast] was addressed to me, easy to understand and I found it very useful.'*
    - *'I liked that it showed a visual representation to feedback rather than written. I also liked that I was easier to follow the visual feedback within OU Build than just being presented with screen captures and written explanation.'*

### *Tutor feedback*

Both tutors reported enjoying the process of creating and providing screencasts and said they intended to continue to do so for selected TMA solutions in future.

The tutor who used ZendTo did so because of concern about YouTube (being doubtful about: security, making a link between working and personal online presence; and asking students to use a service that may track users).The student for whom a screencast was delivered via ZendTo did not respond to the survey.

### *Stage 2 conclusions*

Student and tutor feedback along with the project team's evaluation of the screencasts, indicated that the screencasting guidelines drawn up as a result of Stage 1 were largely appropriate, with only small amendments needed. The technical guidance was refined to enable tutors to choose between ZendTo and YouTube. See Appendices 2 and 3. Two Stage 2 screencasts were selected to be used as examples for tutors in future presentations (https://youtu.be/QX5tDYC0Iaw and https://youtu.be/M_ehmI_bKug).

## Impact

At the outset of the project we felt that the findings may have the potential to inform other modules in which problem-solving and/or programming feature. However actual feedback was that tutors and students found the greatest benefits of screencasting to be closely allied to the visual nature of the programming environment OUBuild, which is unique to TM111. It is not clear how that would translate into text-based programming languages such as Python used on TM112.

Hence, we intend to continue to promote and develop individualised screencasting for TMA feedback on TM111 and explore more cautiously the possible extension of screencasting into other modules.

We presented a poster on this project at the 2020 eSTEeM conference. Discussion with participants indicated interest in pursuing individualised screencasts for TMA feedback on other computing modules; and in assessing whether students with text disabilities (dyslexia etc) might particularly benefit from screencasts. We intend to discuss and pursue these ideas further as appropriate

## References

Loksa D., Ko A., Jernigan W., Oleson A., Mendez C. and Burnett M. (2016). Programming, Problem Solving, and Self-Awareness: Effects of Explicit Guidance. Proceedings of 2016 CHI Conference on Human Factors in Computing Systems. Available from https://dl.acm.org/citation.cfm?id=2858252.

Rountree J. and Rountree N. (2007). Issues Regarding Threshold Concepts in Computer Science. Proceedings of the Eleventh Australasian Conference on Computing Education - Volume 95. Available from http://crpit.com/confpapers/CRPITV95Rountree.pdf

Harper, F., Green, H. and Fernandez-Toro, M. (2012). Evaluating the integration of Jing screencasts in feedback on written assignments. In: 15th International Conference on Interactive Collaborative Learning, 26-28 Sep 2012, Villach, Austria https://ieeexplore.ieee.org/abstract/document/6402092

Atfield-Cutts, S., Ollis, G., Coles, M. and Mayes, H. (2016). Blended Feedback II: Video feedback for individual students is the norm, on an undergraduate computer programming unit. Paper presented at the Psychology of Programming Interest Group 2016 27th Annual Conference http://www.ppig.org/biblio?f[author]=796

Walker, M. (2007). How could TMA feedback support students better? Paper presented at The Open University CTSS Conference, The Open University, Milton Keynes: http://www.open.ac.uk/opencetl/colmsct-cetl/activities-projects/assessment/improved-learning-through-improved-feedback-tmas

Thomas, E., Kouadri S. M. and Jefferis H. (2018). Visualising the code: are students engaging with programming at level 1? Current eSTEeM project. http://www.open.ac.uk/about/teaching-and-learning/esteem/projects/themes/technologies-stem-learning/visualising-the-code-are-students-engaging-programming

## University approval processes

- SRPP/SSPP – Approval from the Student Research Project Panel/Staff Survey Project Panel was obtained according to the Open University's code of practice and procedures before embarking on this project. Application numbers **2018/121** and **2020/017.**

- Data Protection Impact Assessment/Compliance Check – A Data Protection Impact Assessment/Compliance Check was obtained according to the Open University's code of practice and procedures before embarking on this project. Data Protection registration number **4297** and **28-04-27**.

## Appendices

Appendix 1 – TM111 screencasting project briefing notes for tutors 19D

Appendix 2 - TM111 screencasting – pedagogic guidelines

Appendix 3 - TM111 screencasting – technical guidelines

# Appendix 1

## TM111 screencasting project briefing notes for tutors 19D

(Stage 1 of *Developing programming problem-solving skills using individualised screencasts*)

### Background

This project is funded by eSTEeM, the OU's centre for STEM scholarship into pedagogy and teaching. The project team consists of TM111 Module Team members Christine Gardner, Sarah Mattingly and Richard Walker.

### Outline

The project involves tutors providing short screencasts as feedback on student solutions to Questions 2 and 3 of TMA02. A screencast is an audio-visual recording of what happens on a computer screen, along with voiceover (the tutor speaking, in this case). It can capture whatever happens on a screen e.g. 'handwriting' (if the user has a digital pen), highlighting or creation of text (e.g. in Word) and of course creating programs in OUBuild.

We envisage the screencasts in this project providing personalised feedback to individual students, tailored to their misconceptions, mistakes and areas for improvement, aimed at helping them develop skills in programming and in problem-solving.

The project aims to address the following.

- Do students understand key programming concepts more deeply and are they more skilled at programming problem-solving, as a result of screencast feedback?
- What kind of screencast content is most beneficial?
- How time-consuming is creating individual screencasts?

### Pilot stage

The pilot stage of the project runs in 19D and involves five tutors trialling the use of screencasts and then, along with their students, feeding back on the experience. We will ask you afterwards about such things as what you think useful in terms of screencast content, about any technical issues, and about how demanding/challenging/enjoyable producing such screencasts was for you. Depending on those results we will then decide whether and how the use of screencasts in TMA feedback might be pursued in future presentations.

### Timings

| **Date** | **Activity** |
|---|---|
| April 2019 (approx 1 week after presentation start) | Project team send student PIs from all 5 groups to Student Research Project Panel (SRPP) for checking. SRPP tell us if any students may not be included in the project |
| Early June 2019 (*date tba late May*) | Online tutor briefing |
| Early July 2019 | Students contacted, informing them about the project |
| 18 July 2019 | TMA02 cut-off, marking begins |
| End July 2019 | Most TMA02s marked and returned. Tutors inform project team which students received |

| | |
|---|---|
| | screencasts. Project team distribute questionnaires to students; tutors follow up by encouraging students to complete questionnaires.<br>Project team start looking at solutions and screencasts |
| Early September (*date tba late July*) | Online tutor debriefing |
| September-October 2019 | Project team analyse and write up Stage 1 results |

## Your screencasts

In creating the screencasts we want you to draw on your experience and imagination. There are no limits and no prior expectations. We want to know what you think works well and what might usefully be suggested to other tutors in future. However we aren't asking you necessarily to modify your fundamental pedagogy or to do anything startlingly novel. And we're definitely not asking you to spend vast amounts of time on screencasts (though inevitably they will take longer to produce to begin with). For us to pursue this further we have to be sure that screencast feedback is achievable for tutors in the normal course of correspondence tuition, so anything that takes huge amounts of time isn't going to be useful in the end.

For each of your students' Q2 and Q3 solutions, ask yourself how you might usefully give the student feedback in audio-visual form. Perhaps imagine they were sitting alongside you – how would discuss with them their algorithm/program/tests? How might you help them to correct problems or to improve a working solution? Might you talk them through fleshing out an initial algorithm? Might you describe how to correct and then implement their algorithm? Might you show them how carrying out different tests would reveal problems with their code?

Just a few guidelines:

- We recommend that screencasts are short, perhaps 5-10 minutes each. That way (a) if you make a serious mistake (though see below) you might more readily start over and (b) students may be more likely to watch them through.
- We recommend a 'quick and dirty' approach. Have a rough plan for what you want to get across, but don't aim for perfection and don't be tempted to edit! Screencasts aren't expected to be perfect – just as if you were explaining something at a tutorial there are bound to be infelicities, errors that you correct, cats walking across your keyboard etc. etc. In fact students tend to take kindly to such things – they learn from seeing other people recover from mistakes, and imperfection can make tutors seem more approachable and human.
- Please *offer* to give each student a written version of your feedback, in case they can't or don't want to access screencast(s). For example, you might write on the PT3: 'I hope you will find my screencast feedback on your Q2 helpful. But if you would prefer feedback in written form, let me know and I will provide it.'
- Please ensure it is clear to students, for example via comments on their script, how to access any screencast you provide.
- You are not expected to provide screencast feedback on all scripts. Certainly if a few words are all you need to say (e.g. 'Well done! That's exactly the program I would have created.')

then text is probably fine. Concentrate your screencasting efforts on those students you think will most benefit from it.

## Making screencasts

For making the screencasts you are welcome to use any system of your choice, as long as it generates mp4 files. However Apowersoft offer a free online screen recorder which seems quite good.

https://www.apowersoft.com/free-online-screen-recorder

There is a "get going" video available at

https://www.youtube.com/watch?v=KzD8YKJDAs0

## Sending screencasts to students

The feedback screencasts are confidential in the same way any other form of TMA feedback.

The method we ask you to use for sending them to students is the university's secure file transfer system ZendTo, which is entirely OU hosted and allows the transfer of files up to 4 GB. Instructions are here

http://intranet6.open.ac.uk/it/main/secure-file-transfer

## Sharing issues and solutions

The project is trying something fairly new, so to some extent we will be building up knowledge about the technical aspects as we go along. Please share any issues you meet and/or any solutions you come up with on the TM111 eSTEeM Project Forum.

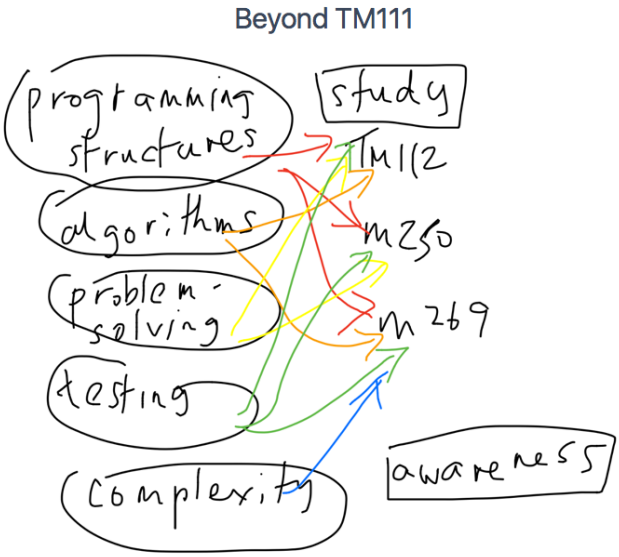
# Appendix 2

## TM111 screencasting – pedagogic guidelines

We suggest you create screencasts for TMA02:

- that are short (around 5-10 minutes) Otherwise files sizes are likely to be a problem; the process may become too time consuming for you; and we expect students will be less inclined to view them
- that provide personalised feedback tailored to individual misconceptions, mistakes and areas for improvement
- that focus on developing transferrable concepts and skills – things that will be valuable to students when programming in future, so feeding forward. Transferrable knowledge might be (a) to underpin further study or (b) something that will contribute to the student's general awareness of a topic. For example:
    - a program implements incorrect logic;
    - a program does not use a loop when it should;
    - a program involves an unnecessarily lengthy or complicated combination of selection constructs;
    - a program hard-codes what should be variable input (for example, in Q2 of TMA02, some students work with fixed rather than random numbers);
    - a student who does nothing more with algorithms will still benefit from having a basic grasp of the fact that it is not enough for an algorithm to merely work; if it takes an impractical number of operations it will be of no practical use.

  The sketch below shows the big themes of TM111 Block 2 and how they individually feed into later modules. They also all contribute to awareness and so are important to all students whatever they go on to study.



- that are imperfect! Have a rough plan for what you want to get across, but don't aim for perfection and don't be tempted to re-record or edit unless you make a complete hash of something. It is very tempting to aim for a polished performance but do resist: just as if you were explaining something at a tutorial there are bound to be infelicities, errors that you correct, cats walking across your keyboard etc etc. In fact students tend to take kindly to such things – they learn from seeing others recover from mistakes, and imperfection can make tutors seem more approachable and human
- for solutions where visual and audio feedback might readily add value – where it is easier to talk and show what you mean, where (in an ideal world) you might like to sit alongside the student and talk them through their work and your feedback
- in which you interact with a student's OUBuild code, exploiting the visual nature of OUBuild and using it as a vehicle for demonstrating problem-solving. You might show the process of testing it; you might talk through amending it (if it is incorrect), or improving it (if it is inelegant or inefficient)
- for solutions where you believe the student has invested time and effort and got somewhere, that are somewhat correct, but not perfect. In such situations we believe students are more likely to be interested in your feedback and more likely to benefit from it;
- for solutions which are very confused and where written feedback would be lengthy and laborious to create compared to showing and talking.

We suggest you do not create screencasts for:

- solutions in which students have got largely nowhere, in which case you would essentially be creating the solution program from scratch. Though this might well be useful for some students it would be time consuming for you and not really individualised.
- solutions that are largely right. For example, if a student has simply used inappropriate variable names, a written comment on their work may be all that's required
- showing students how to use specific features of OUBuild (how to manipulate blocks, for example) – instead concentrate on transferrable skills, as described above. Remember most students will never use OUBuild again after TM111.

# Appendix 3

# TM111 screencasting – technical guidelines

## Capture

Many applications can capture screencasts. If you have a favourite you are using already we recommend continuing with it, as long as the end result is an mp4 file.

If not, then PowerPoint on Windows and QuickTime on macOS are suitable, but there are a considerable number of others, many free. Have a browse, do some quick trials and choose one you like.

Screencast files are usually quite large, so they take a while to upload and download, and might also eat into a student's data allocation. If sending an mp4 file (see below for options) then compressing it can often achieve a worthwhile size reduction.

## Distributing screencasts

TMA feedback screencasts should be confidential as with any other form of TMA feedback, so each should only be made available to the individual student. Two forms of distribution have been found to work, though each has drawbacks as well as advantages.

### ZendTo

This is the University's secure file transfer system, which allows the transfer of files up to 4GB. Your TMA comments tell the student to expect a screencast; you upload the screencast file; the student is alerted via email to its existence and has 14 days to collect it. Instructions are here http://intranet6.open.ac.uk/it/main/secure-file-transfer.

### YouTube

You make your screencast available as an unlisted video, essentially accessible only to anyone with the resulting link, which you insert within your comments on the student's TMA. The Appendix details how to put up an unlisted video.

## Monitoring

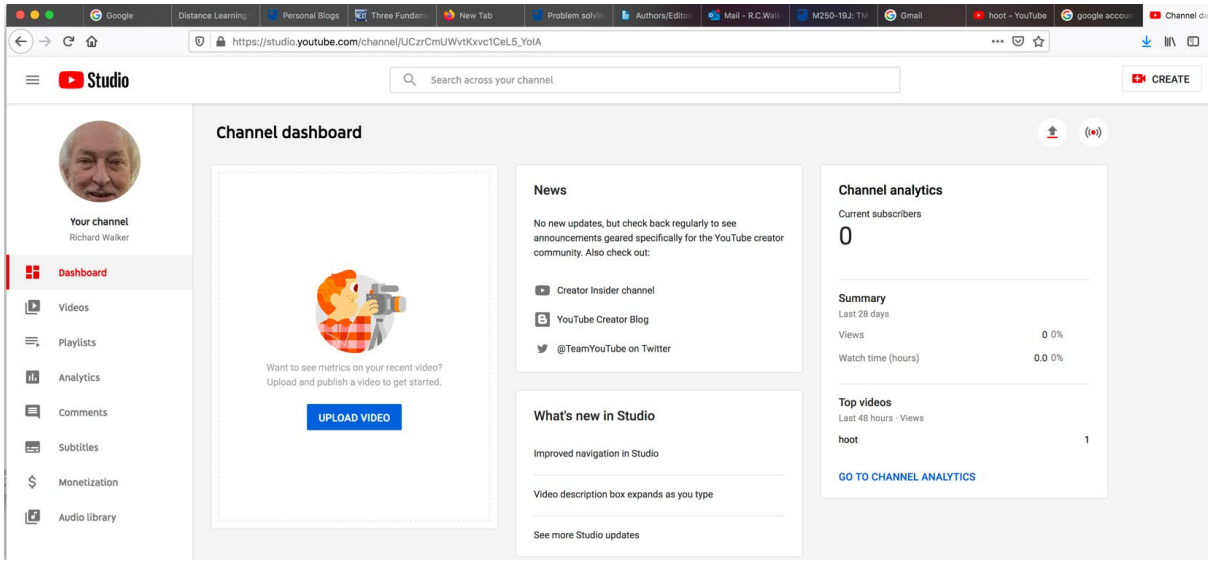
Please keep hold of (a link to) any screencasts you provide, in case the person monitoring your TMA marking needs to see them.

# Appendix: Uploading a screencast to YouTube

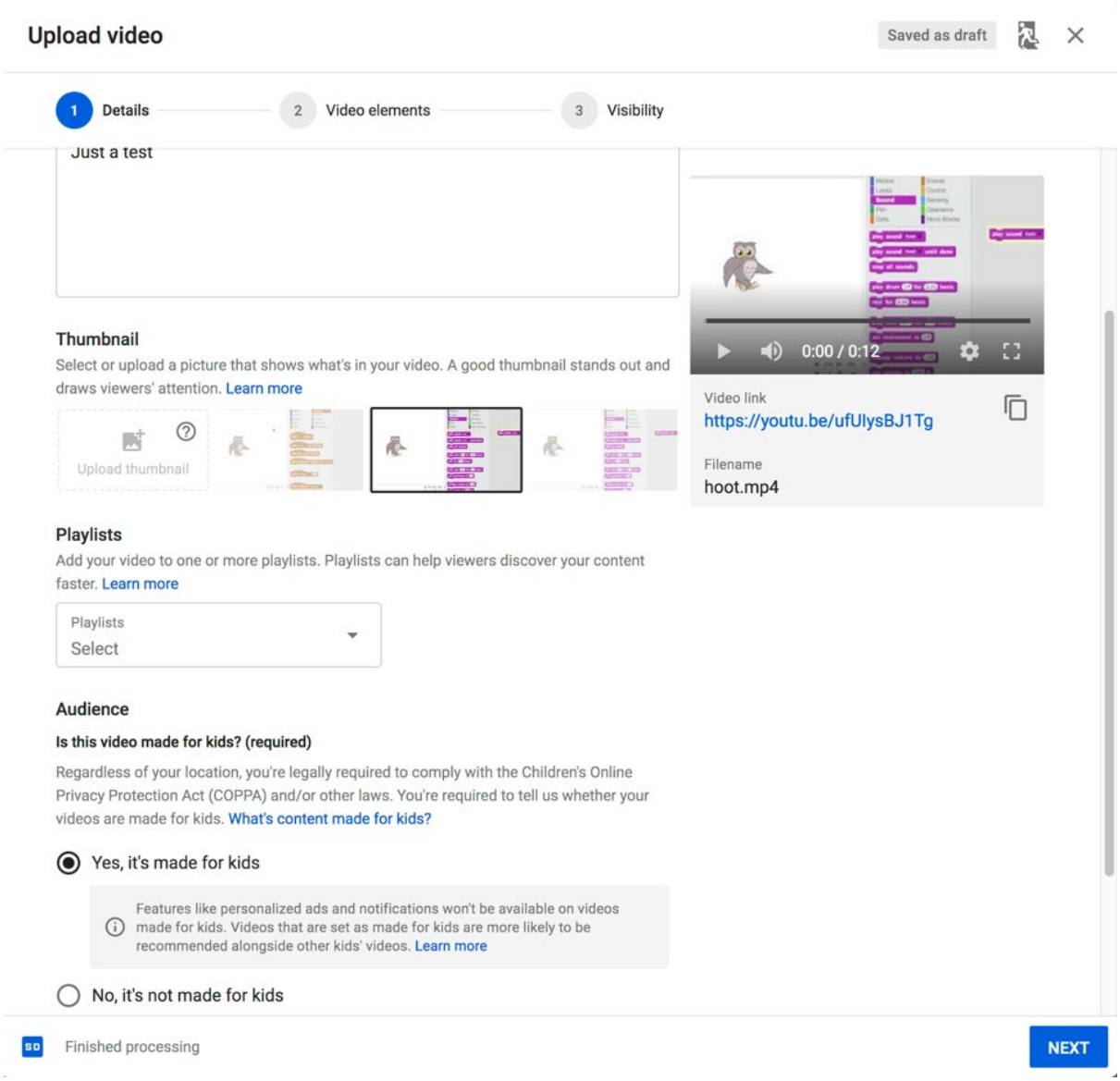
Sign in to Google

Go to [https://studio.youtube.com](https://studio.youtube.com)

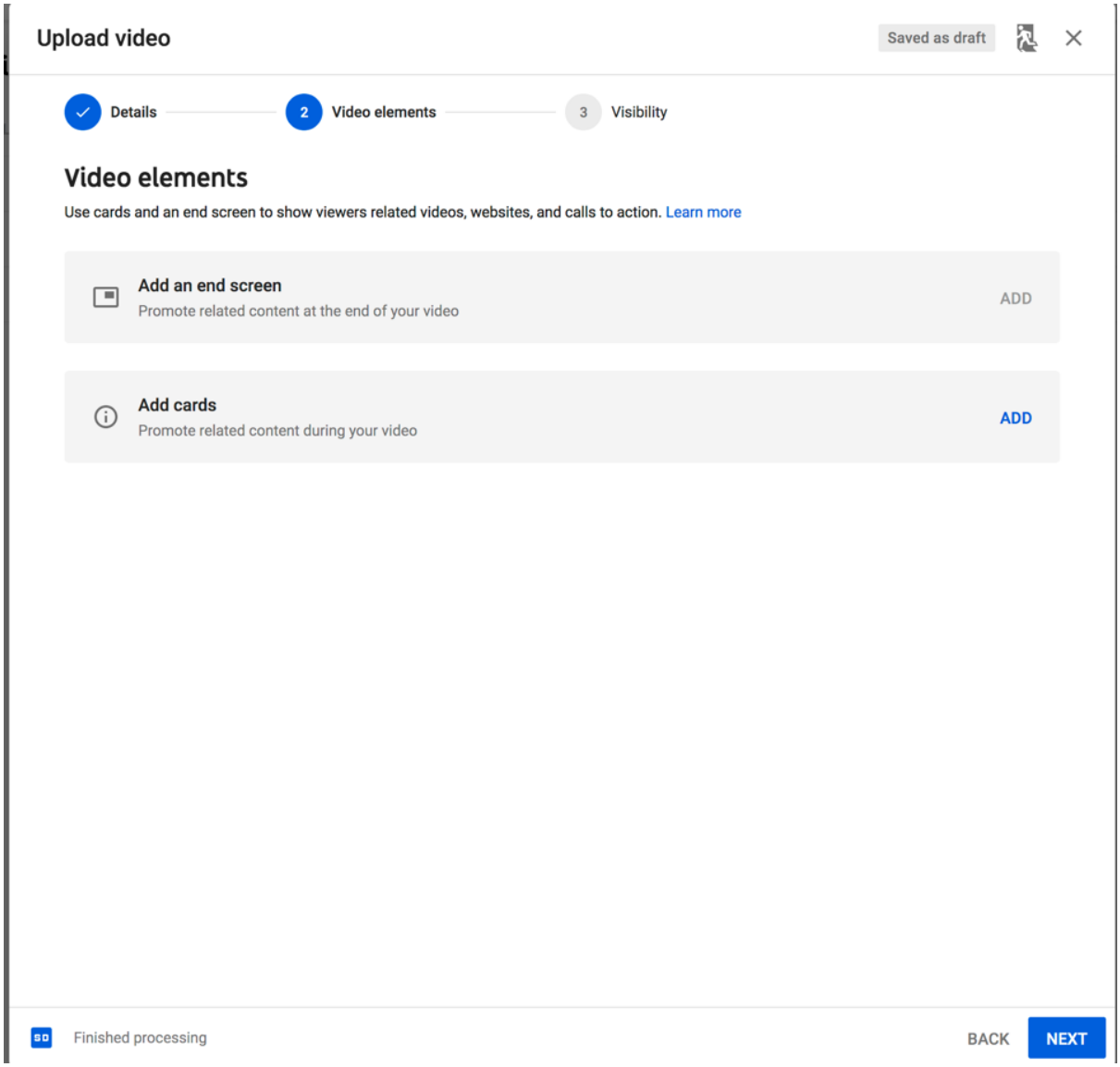
You should see this screen (you may need to 'create a channel' first):

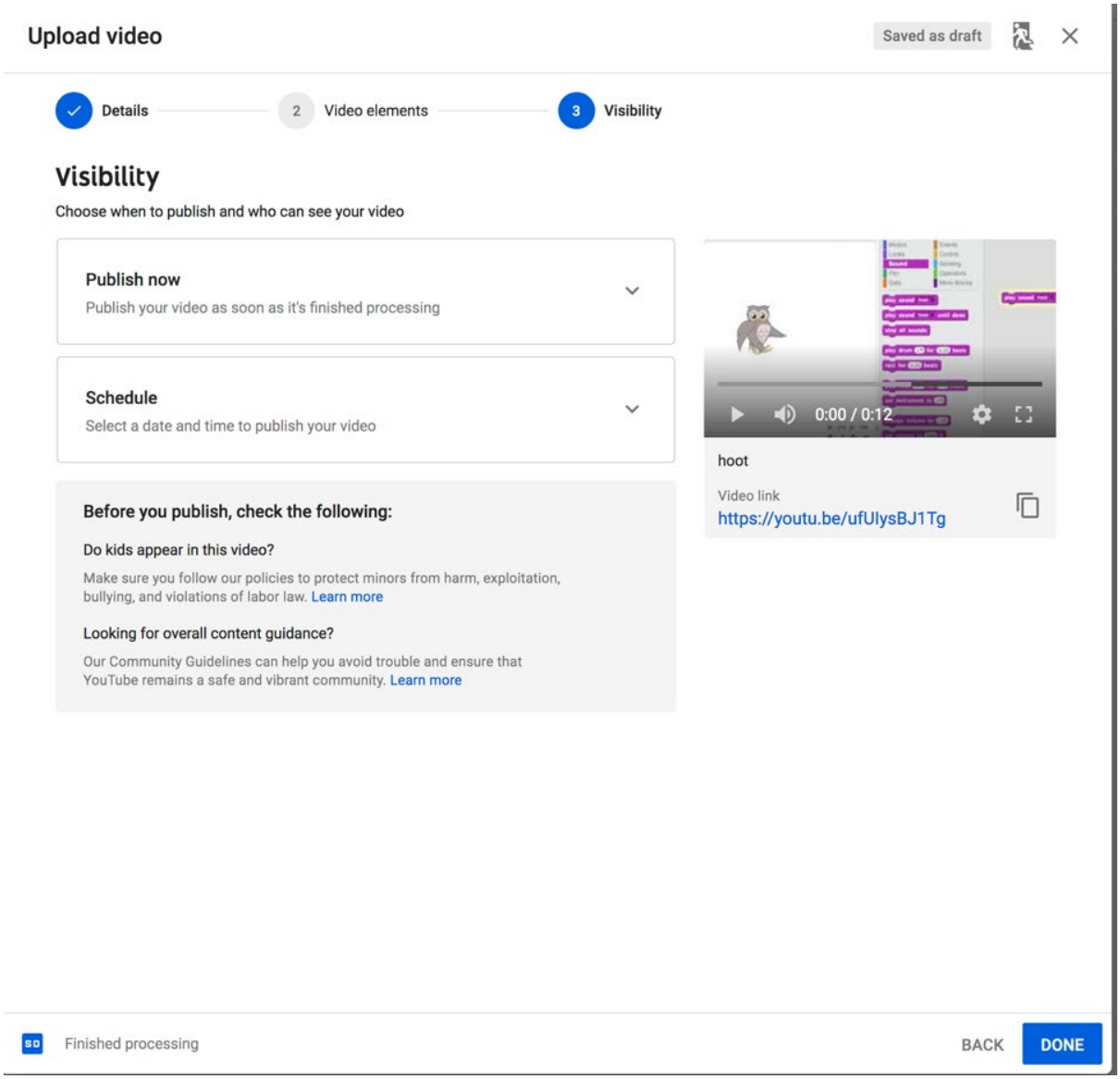



Click on Upload video, follow the steps and adopt the settings shown below.

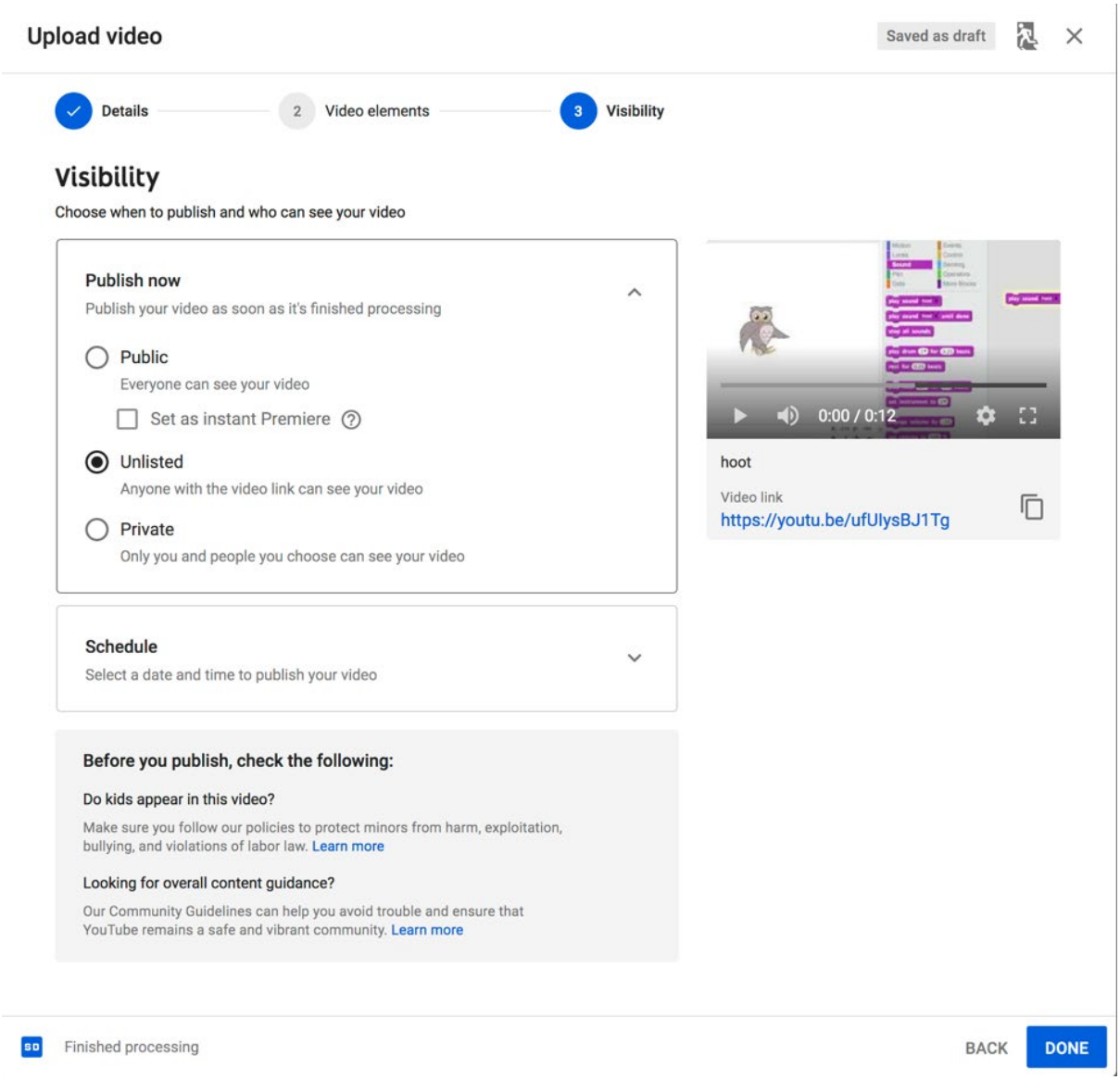



Labelling as ‘made for kids’ cuts out some of the more annoying adverts.


Upload video
Saved as draft
Details
2
Video elements
3
Visibility
Video elements
Use cards and an end screen to show viewers related videos, websites, and calls to action. Learn more
Add an end screen
Promote related content at the end of your video
ADD
Add cards
Promote related content during your video
ADD
Finished processing
BACK
NEXT




Mattingly, S. (2020) Developing programming problem-solving skills using individualised screencasts. eSTEeM Final Report.




https://youtu.be/ufUIysBJ1Tg

The video will be unlisted but visible to anyone with the link. Put the link within your comments on the student's TMA.