Enriching programming student feedback with audio comments

Author:
Renzella, J; Cain, A

Publication details:
Proceedings - International Conference on Software Engineering
pp. 173 - 183
9781450371247 (ISBN)
0270-5257 (ISSN)

Event details:
Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering Education and Training
ELECTR NETWORK
2020-06-27 - 2020-07-19

Publication Date:
2020-06-27

Publisher DOI:
https://doi.org/10.1145/3377814.3381712

Enriching Programming Student Feedback with Audio Comments

Jake Renzella
School of Information Technology
Deakin University, Geelong
Geelong, Victoria
jake.renzella@deakin.edu.au

Andrew Cain
School of Information Technology
Deakin University, Geelong
Geelong, Victoria
andrew.cain@deakin.edu.au

ABSTRACT

Introductory programming is challenging for many students, requiring them to engage with a deep approach to learning concepts in order to succeed. These challenges compound for online students who do not have direct face-to-face interactions with teaching staff. With the growing demand for online education, we need to examine approaches that assist in building supportive learning environments for these students. A growing body of work from other education disciplines indicates that audio feedback provides an opportunity for developing stronger relationships with students. Further studies recommend an integrated implementation of audio recording into the virtual learning environment. To evaluate audio feedback for use in programming education, we developed an integrated, cross-browser audio feedback feature into the open-source Doubtfire learning management system. Doubtfire is used to support and scale a task-oriented teaching and learning system built upon the principles of constructive alignment and has been shown to help students engage with programming concepts in campus-only units. Our findings from experimental and observational activities indicate that programming tutors can use a blended approach of audio and text feedback via the learning management system to better support student learning. The richer, more nuanced feedback delivery communicates personality to students while retaining the benefits of written feedback for code-specific issues.

CCS CONCEPTS

• **Applied computing → Learning management systems**; **Distance learning**; *E-learning*; • **Human-centered computing →** *Human computer interaction (HCI).*

KEYWORDS

Learning management system, introductory programming, formative feedback, audio feedback, online students

ACM Reference Format:
Jake Renzella and Andrew Cain. 2020. Enriching Programming Student Feedback with Audio Comments. In *Software Engineering Education and Training (ICSE-SEET'20), May 23–29, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3377814.3381712

ICSE-SEET'20, May 23–29, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7124-7/20/05…$15.00
https://doi.org/10.1145/3377814.3381712

1 INTRODUCTION

Many students find learning to program challenging, as evidenced by the high attrition and failure rates commonly reported in undergraduate introductory programming units [3, 23, 30]. To support students engaging in programming education, educators utilise a wide range of approaches in curriculum and assessment design & format [17]. Regardless of approach, providing students with high-quality, formative feedback is integral to ensuring students achieve learning outcomes [10].

Growing global online student enrolments [21] has driven work investigating considerations for online learner issues. Investigations of student retention through to course evaluation highlight a need for teaching and learning systems to change in order to support online student cohorts [1], as online cohorts have been observed to have higher student attrition in contrast to campus students [22]. In the United Kingdom, Simpson [26] observes that specific intervention techniques, including tutor intervention, have been shown to increase student retention. We have seen similar online enrolment growth in our Australian programming unit.

Task-oriented portfolio assessment [8] provides a structured approach to achieving constructive alignment, and has been shown to be effective in teaching programming [6, 7]. The model provides processes and business rules that build upon constructive alignment [4], a student-centred approach to teaching and learning stemming from constructivist learning theory, combined with aligned curriculum. The use of constructively aligned unit design results in students undertaking deeper situational learning approaches across a number of domains [14, 29].

Previous approaches to task-oriented portfolio assessment have focused on offerings that were entirely campus-based, where staff and students interact in person. These interactions assist with the formative feedback process, central to supporting students in these units. With the growth in online and mixed campus students [27], we looked to better support tutor-student interactions through the tight integration of audio comments into a virtual learning environment. Audio feedback has been observed to improve student-tutor relationships, and impart a greater sense of caring on behalf of the tutor towards the student [12, 31]. Doubtfire [32] is an open-source learning management system originally designed to support and scale a task-oriented teaching and learning approach. The inclusion of audio comments into Doubtfire aimed to help address the disparity of discussion opportunities between online and campus-based students and their tutors.

In this paper, we present our experiment and results of tutor experiences with audio feedback and on their perceptions of how this supports feedback for programming students. We conducted a series of experimental activities with programming tutors, taking participants through two series of simulated introductory programming task submissions and having them provide formative feedback with and without audio comment support. Each series of submissions contained eight typical programming tasks designed to cover a representative selection of student programming submissions. Also included are our reflections after using audio feedback for programming students for two teaching periods.

To ensure we included all the different forms of feedback programming tutors are typically required to provide, and thus develop the appropriate simulated programming tasks for our experiment, we developed an introductory programming task feedback taxonomy, which is presented and explained in this paper.

This paper begins with an exploration of related work and background, followed by details of our audio comment recording additions to the Doubtfire repository. Proceeding sections describe the research method, taxonomy development, results, discussions, and concluding remarks.

2 RELATED WORK

2.1 The Task-Oriented Approach

Biggs & Tang [5] claim that students are motivated to engage with deep approaches in constructively aligned courses, with evaluations since reinforcing the claim. One qualitative and quantitative analysis of documents, interviews, and surveys found that while students adapt to the teaching and learning environments, "students in more constructively aligned courses were more likely to adopt deep learning approaches and less likely to use surface learning approaches in their study of a particular course" [29].

Constructively aligned approaches to teaching and learning underpin the task-oriented approach described by Cain and colleagues [8, 25]. The task-oriented portfolio model [8] (TOPAM) depicted in Figure 1, is thus a supportive teaching and learning system designed to encourage and reward students for engaging with deep approaches to learning.

Luxton-Reilly et al.'s 2018 structured literature review on approaches to teaching introductory programming [17] reports on the wide range of research associated with supporting the teaching of programming. Methods for assessment reported in their literature review include paper-based exams [13], group assessment [19], and portfolio assessment. The task-oriented model described in this paper is portfolio-based. In this system, student results are associated with holistic assessment of a final portfolio that is a compilation of smaller, weekly, tasks that students develop with the support of teaching staff across the teaching period. The rationale behind using portfolio assessment is that students construct their own learning through relevant learning activities [5].

Frequent formative feedback is central to the TOPAM approach. This feedback process is illustrated in Figure 1. In TOPAM, students review and engage with learning materials, either in the form of class activities or through online resources. These resources support students as they engage with the unit tasks, in which they demonstrate usage and understanding of the concepts and skills covered. Once the student believes they have successfully demonstrated the required learning, they submit associated work to their tutor for feedback. The tutor reviews the submission and either provides feedback to help the student achieve the learning outcomes, or indicates that the work successfully demonstrates the required learning outcomes. Where the task has not been signed off, students are encouraged to incorporate feedback, address any misconceptions, and resubmit. This feedback process may repeat a number of times if necessary, in order to help ensure that each student demonstrates the required standard for each task before it moves into their portfolio.

2.2 Audio Feedback

Audio feedback refers to the provisioning of formative or summative feedback, via a digital recording of the tutor's voice, sent to the recipient. Audio feedback is becoming widely available in popular learning management systems (LMSs) in some capacity. We conducted an informal review of audio recording availability in large LMSs via first-party documentation or help pages, our findings indicated that the Canvas LMS, Moodle, Blackboard and Desire2Learn all support audio feedback via attachment out of the box. In some cases, audio can be recorded directly via the browser with the use of third party plug-ins.

In this section, we will present some results of studies that have analysed the use of audio feedback in higher education.

Wood et al. [31] performed a pilot study comparing written and audio feedback. In this study, more than 70% of students indicated a clearer understanding of tutor feedback via audio, and 80% of students rated audio feedback as more personal than written feedback. Similarly, Tom Lunt & John Curran [16] surveyed students to examine experiences receiving audio feedback. 88% of students indicated that audio feedback would help them improve their coursework, and 75% of students reported more detailed feedback compared to written feedback.

A study on student perceptions of audio feedback provided via the popular WhatsApp messaging client in a distance education environment, showed a majority of students believed they could better understand how their grade was awarded, with 93% of students indicating they wish to receive more audio feedback in the future [11].

Ice et al. [12] introduced audio feedback into an asynchronous unit, providing approximately half of the individualized feedback in a text-based format and the other half via audio, providing snippets of audio via the unit's student forum. Benefits identified in this study include the increased ability for students to understand nuance, feelings of increased involvement, content (feedback) retention, and an increased sense of instructor caring.

Rawle et al. [24] report that future use of audio feedback will continue following their study which revealed high student and teaching assistant satisfaction. These results were obtained through a survey. They mentioned that future teaching assistants were being trained to provide footnote written feedback alongside audio.

Aligned with the results reported in these studies, a 2019 qualitative literature review analysed ten studies of audio feedback to determine student and tutor perceptions [15] concluded that alternative feedback modes, such as audio feedback, promote a better

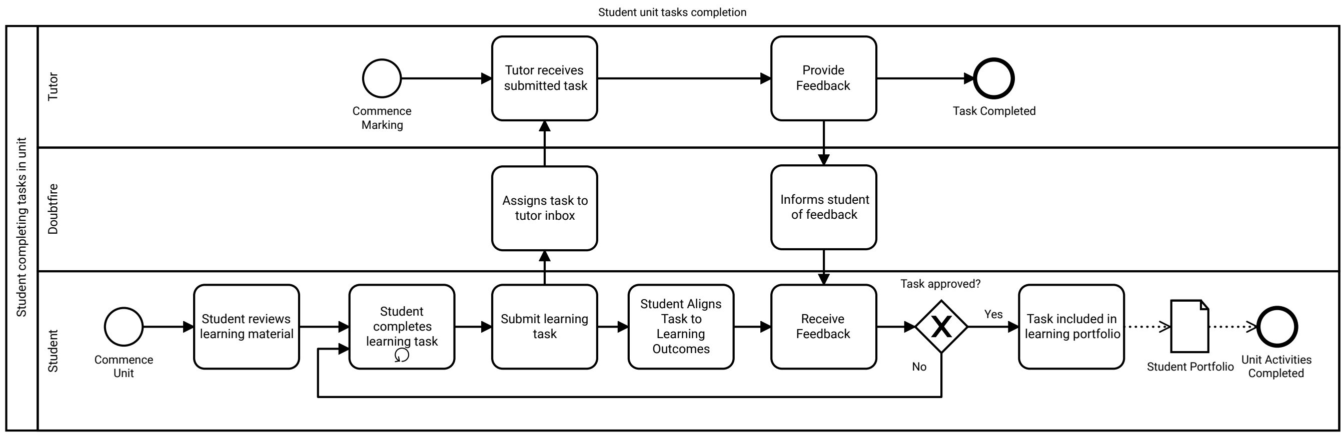

Figure 1: Business Process Modelling Notation Diagram (BPMN) of the Task-Oriented Model.

sense of belonging for students, and can help address the sometimes vague nature of written feedback.

Stephen Merry & Paul Orsmond's 2008 paper [18] investigated student and tutor experiences after introducing audio feedback in a biology unit that previously included only written feedback. Their findings were in line with Wood et al [31] and other studies mentioned above. In this study, students submitted their choice of past assignments to participant tutors who would then provided audio feedback. Thirteen of the fifteen student participants said they would like to receive more feedback in this format, with two students showing no preference between written and audio feedback. Tutors also reported positive experiences, highlighting the ability to provide more detailed feedback in a similar time scale.

Carruthers et al. [9] evaluated a plug-in to BlackBoard Learn+ which enabled support for media feedback within the browser, citing increased time savings compared to attachment-style audio feedback commonly used in other studies.

Morris et al. [20] randomly allocated 50% of the 68 students in a science laboratory to receive either written or audio feedback throughout the unit. Average time spent providing feedback was lower for the audio feedback overall; however this particular study used a file attachment approach, resulting in no overall time savings. In this respect, Renzella et al. [25] explore the technology and systems used to facilitate formative feedback (written) in support of the task-oriented portfolio approach at scale. The work describes the importance of low-overheads when performing teaching activities such as providing feedback, or when switching between submissions due to the significant number of small tasks typically used in the TOPAM approach. These results indicate that any initiatives associated with improving feedback for online students needs to minimise overheads, and work seamlessly within the online environment.

3 IMPLEMENTING AUDIO COMMENTS IN DOUBTFIRE

In preparation of introducing audio feedback support in an introductory programming unit, we explored different implementations of web-based audio recording features before deciding to integrate audio comments closely within the commenting system of Doubtfire. In this section we describe the context of our study and the implementation of audio comments in Doubtfire.

3.1 Context

As we use Doubtfire [32] to deliver task-oriented portfolio assessment, the inclusion of audio comments into Doubtfire aimed to help address the disparity of discussion opportunities between online and campus-based students and their tutors. This was essential to support the feedback process that is such an important part of the task-oriented portfolio assessment approach.

In fact, critical to the feedback process is the close interaction between the tutor and student. Students need to be able to accept and act upon feedback, that may require them to reexamine their understanding of topic concepts. While this is central to effective learning, it is not standard assessment practice. More commonly, students submit work and receive a summative grading that may indicate issues, or provide feedback, but not *require* students to incorporate or review what they have learnt. In fact, in on-campus offerings, tutor-student interactions allow tutors to explain and highlight issues identified in the student work, before submission, and suggest strategies to improve student learning to enable them to successfully complete the task. These feedback loops are facilitated through in class discussions and thus, cannot be easily applied to online students. Additional tools are needed to help facilitate these interactions online.

Task discussions are also used in our unit as a means to help mitigate potential issues with academic integrity. The close tutor-student relationship helps ensure tutors are able to provide effective feedback, but can also help identify cases where work submitted appears to be uncharacteristic for that student. Embedded within the task sign off process is a requirement for students to discuss tasks with their tutor before final sign off is given. This formalises the requirement for student-tutor interaction, and provides additional challenges for the adoption of this approach for online students.

As the Learning Management System DESIRE2LEARN[1] is accessible to every unit at our institution and includes an audio recording

[1] https://www.d2l.com/en-apac/

attachment tool we analysed this before implementing audio feedback in Doubtfire. While it is encouraging that a recording feature is available, the tool is nested within menus, introducing overheads to the feedback process. With the number and frequency of feedback in the TOPAM approach, this overhead would make the feature prohibitively time consuming. In addition, given we are targeting introductory programming units, which may include several hundred students each offering, scalability and time efficiency were also concerns that need to be considered.

Previous studies of audio comments examined the use of audio snippets recorded in a separate program on the computer, then emailed as attachments to students [16, 18]. While this approach is suitable for experiments or small cohorts, it is not appropriate for larger units, as the overheads introduced for recording and emailing the feedback would prove prohibitive. Instead, we chose to integrate audio comments closely within the commenting system of Doubtfire. We reviewed other audio recording implementations, such as Facebook's Messenger web app [2], and took inspiration from their implementation, including UX design patterns of recorder visualisation.

3.2 Audio Comments in Doubtfire

Doubtfire is a web application designed to support the task-oriented approach at scale [32]. Figure 2 shows the the main tutor view of the Doubtfire system broken into the three primary panels. Panel 1 labelled "Task Inbox" provides each tutor with an inbox style list of task submissions awaiting feedback, along with existing submissions containing new text or audio comment notifications. When a submission is selected in Panel 1, Panels 2 and 3 are updated to show the details related to that submission. Panel 2 shows the submission as a single PDF document that combines together all submitted images, documents, and code file.

Panel 3 shows the task comment panel, where students and tutors can send and receive threaded text or audio comments is the focus of the changes described in this work. This shows comments from the current user with their blue highlighted user icon on the left of the comment, and those from other parties listed with grey icons on the right. In Figure 2, the current user is to a tutor so the tutors messages have the user icon on the left, and the student responses have the student user's icon on the right. This pattern would be familiar with users of many modern commenting and messaging applications.

Users can open the audio recorder panel (shown open), and access their computer's microphone. After recording, users can optionally playback for review before sending to the recipient. Submitted audio appears in the comments panel and notifies the recipient(s). The threaded nature of the comments results in audio feedback which has the context of surrounding textual or audio comments.

There were several technical challenges associated with implementing a modern, web-based audio-recording tool, particularly cross-platform audio recording without using third-party plugins, such as Adobe Flash technology. The modern MediaRecorder API supports microphone access and audio recording in HTML5 browsers, however, as of the time of writing this API is not supported in the Safari browser. Instead, our hybrid approach uses the MediaRecorder API if availability is detected in the browser, otherwise providing a fallback to a custom implementation of the MediaRecorder API. Another significant hurdle was codec compatibility of the major browsers. Codecs are responsible for encoding and decoding the contents of an audio file, and at the time of work. At the time of writing, supported codecs in which each major browser can produce a recorded audio file are incompatible for playback on other major browsers. As a result, our system converts all submitted audio recordings from the originating browser's codec, into a format supported for playback on all major browsers.

[2]https://messenger.com

4 RESEARCH METHOD

To evaluate the efficiency and tutor expectations towards audio feedback, we conducted a mixed research methods experiment, based upon the research design used in Wood's 2011 paper [31], which contained a study of the use of audio comments in nursing education.

Participants were required to have some tutoring experience in university introductory, ideally with experience in delivering task-oriented, portfolio units, however this was not made a requirement as to not overly reduce the already small participant population.

The entire experiment consisted of three sections, first a pre-experiment interview, designed to capture participant background and experience, and expectations towards audio feedback. The main experiment has each participant providing written and audio feedback to a series of simulated student programming submissions. Finally, a post-experiment interview captured the participant's thoughts and experiences during the experiment.

4.1 Pre-Experiment Interview

The pre-experiment interview was designed to collect information on the participants background and experience with learning management systems, and whether or not the participant has experience with the Doubtfire learning management system. The pre-experiment interview also collected participant's thoughts on the proposal of an audio comment feature, and included gathering information on how tutors may present feedback differently for campus-based and online-based students. The questions of the pre-experiment interview include five-point Likert questions as well as open-ended. These included the following questions:

- AQ1) How would you assess your experience with Learning Management Systems in general? (rating 0-5)
- AQ2) How likely do you think it is that you would use audio comments to leave feedback to students if it were a feature of a learning management system which you use? (rating 0-5)
- AQ3) What do you think the advantages or disadvantages would be when considering an audio comment feature?
- AQ4) How might you use Doubtfire and its features, including written and audio comments differently for cloud(cloud) students vs local students who you will see in class? If no difference state so
- AQ5) Do you have any other suggestions for possible features of a Learning Management System?

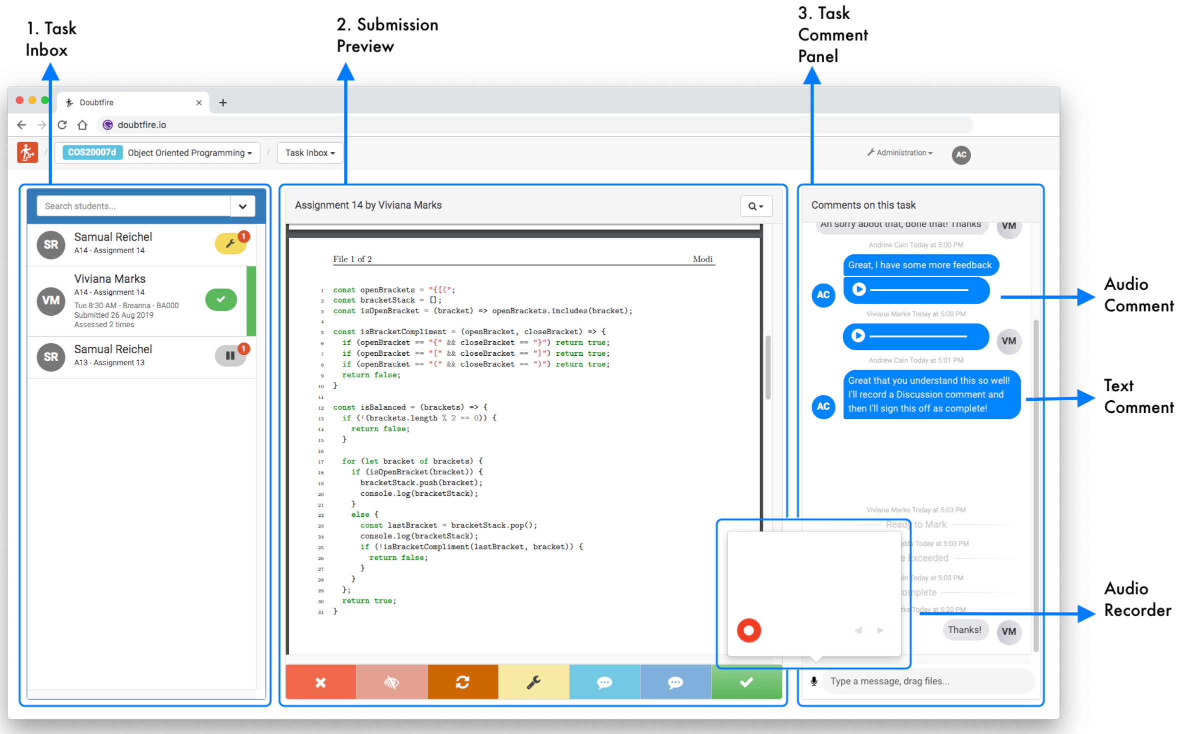

Figure 2: Doubtfire Task Inbox.

4.2 Experiment

After completion of the pre-experiment interview, the main component of the experiment was undertaken. The experiment consisted of participants using an early version of the audio recording feature in the Doubtfire learning management system, to step through a series of eight simulated student submissions and provide formative feedback on each submission. This process was conducted twice with two series of the eight submissions, one time with no access to the audio comment feature, and a second time with access to the feature. 50% of participants were given access to the audio comment feature in the first series of submissions, in order to reduce any ordering threats to validity brought upon by the sequence in which participants used the audio comment feature. Each submission was small, with about 15 to 30 lines of code.

The participants navigated a sandboxed Doubtfire task inbox, see Fig 2. On the left-most panel (panel 1 in the figure), participants clicked on each of the eight tasks in sequence, which would update the preview shown in panel 2. After reviewing the submission and reading any associated questions, they would provide their written or audio feedback in panel 3.

The goal of the experiment was not to evaluate participant's knowledge of programming or tutoring, as such the problems with the simulated submissions were outlined during the experiment. For example, Figure 3 shows one of the code files provided to participants. For this code file, suggested feedback included inconsistent case for constant identifiers, and fixing inconsistencies with brace placement. Participants were not limited to the suggested feedback.

```
#include <library.h>

const float GRAV = 0.08f;
const int max = 5;

void update_velocity(sprite player) {
  set_dy(player, sprite_dy(player) + GRAV);

  if (sprite_dy(player) > max) {
    sprite_set_dy(player, max); }
    else if (sprite_dy(player) < -max) {
    sprite_set_dy(player, -max); } }
```

Figure 3: Example category C3 submission [3].

Throughout the experiment, the participant's usage of the Doubtfire system was screen and audio recorded to facilitate later analysis, and to extract how long participants spent providing the textual or audio feedback for any given submission.

4.3 Post-Experiment Interview

Following the conclusion of the experiment, a series of qualitative questions were asked of the participants:

[3] The following identifiers were shortened for formatting purposes: GRAVITY to GRAV, sprite_set_dy to set_dy.

BQ1) What are your thoughts on audio comments in terms of their speed of feedback?
BQ2) Did you find written feedback or audio feedback to allow you to give more thorough feedback?
BQ3) What are your thoughts on audio comments when considering your ability to potentially connect with your students?
BQ4) What are your thoughts on audio comments when considering your ability to communicate code specific issues with students?

Questions above were supplemented with exploratory questions asked during the interview.

We chose not to analyse the contents of the written or audio feedback. Student submissions provided to participants in the experiment environment were simulated, and as such, may not accurately represent the feedback given in a typical unit. Further, tutors typically experience some formal or informal training on how to provide feedback for a given unit, which we did not offer in our experiment.

5 INTRODUCTORY PROGRAMMING TASK FEEDBACK TAXONOMY

To design a representative set of simulated programming tasks which could be used in the main experiment, a taxonomy was produced to guide what each of the eight submissions should contain, reflecting historical patterns identified in providing and analysing feedback provided in previous introductory programming units.

5.1 Development of the taxonomy

The development method and structure of the proposed Introductory Programming Task Feedback Taxonomy was borrowed from the software engineering field. Usmen et al. [28] provide a structured taxonomy development process, itself inheriting and addressing issues from previous works by Bayona-Oré et al. [2]. The process outlines a series of 13 steps to be followed in the planning, identification, design, testing and deployment phases of the taxonomy construction.

The objective of our *qualitative hierarchical* taxonomy is to represent the range of content and styles of formative feedback tutors provide to students, based upon the individual programming task submission. The taxonomy was able to be utilised in the experiments presented in this research paper, so that a representative range of submissions was evaluated. The taxonomy was developed with historical student programming submissions.

Our taxonomy includes 4 dimensions with 8 categories (sub dimensions):

A. No issues.
B. Conceptual issues.
C. Non-Conceptual issues.
D. Unaccompanied Question or clarification (no code submitted).

Descriptions of items in the taxonomy are described below:

A1) No Problem With Submission: The work submitted is of high quality and meets expectations, would result in no formative feedback from the tutor but would often include a response showing support of the student progress and achievement within the task.

Table 1: Proposed Introductory Programming Task Feedback Taxonomy

Key	Type	Description
A1	-	No problem.
B1	Issue	Conceptual.
B2	Clarification	Conceptual.
C1	Clarification	Non-Conceptual.
C2	Issue	Syntactical.
C3	Issue	Code Quality.
C4	Issue	Code Misuse.
D1	Clarification	No Submission.

B1) Conceptual Issue: Conceptual problem submissions occur when student's work demonstrates a lack of understanding of how to use a particular programming concept. An example would include a student declaring and calling a function, but also repeating the contents of the function elsewhere. This could demonstrate that the student did not understand the concept of functional decomposition correctly.

Typical feedback for this type of submission often includes referring students to course material, or otherwise highlighting the misunderstood concept.

B2) Conceptual Clarification: Concept Clarification Submissions represent occurrences of submitted code accompanied by the the student seeking clarification of how a concept is working. The concept in question may be appropriately implemented, so the tutor relies on the question or comment for prompting. An example could be of a student asking how *cout* and the pipe operator is working in the C++ programming language. As mentioned, if the student does not ask for clarification, the tutor may assume this is not an area which requires discussion or feedback.

C1) Non-Conceptual Clarification: Non-Conceptual Question Submissions wherein a concept is applied appropriately, and submitted to the tutor, but the student requires clarification on non-functional aspects of the code. For example, a student could be unsure of an appropriate name for a function.

Typically in this case, tutors nudge or provide guidance to the student to help them arrive at an appropriate understanding themselves.

C2) Syntactical Issue: Syntax problem submissions represent student submissions wherein a misunderstanding or mistake has occurred regarding the syntax of the programming language in use. An example of such a submission would be a missing semicolon following a statement in C++, which is a requirement of the C++ programming language. Typical feedback for this occurrence could be to highlight the syntax issue, either providing a solution or indicating where the student can resolve their issue.

C3) Code Quality: Code Quality refers to any non-functional code issues, such as formatting, indentation, identifier naming conventions, identifier casing conventions etc. Commonly tutors would indicate the coding convention which is expected.

C4) Code Misuse: This type of code submission usually consists of code which may compile and run, however is bad practice. For example, an initialised variable which is not used in a function.

Typical feedback on this submission could be to ask the student what that variable does, or otherwise indicate to the student that it should be removed.

D4) Unaccompanied Question: Unaccompanied Question Submissions do not contain code in the submission, but aim to seek clarification of a technical nature. Examples of such questions include:

- What is recursion?
- How do I run a block of code many times?

In these examples, the tutor relies entirely on the student's question, and is not necessarily provided context. Because of the openness of these questions, the feedback provided is often longer form.

6 RESULTS

Our call for participants was made across two universities that we know are using the TOPAM (task-oriented, portfolio assessment model) approach for teaching introductory programming. This call targeted staff teaching introductory programming units, and eight participants agreed to participate with representatives from each institution. Participants spent approximately one and a half hours conducting all three stages of the experiment.

6.1 Expectations (Pre-Experiment)

The pre-experiment interviews were designed to evaluate tutor expectations towards audio comments as a mechanism for providing formative feedback to students. Of the eight participants, five expected audio comments would be a more efficient way of providing feedback, however most felt that written feedback would allow them to be more thorough. Seven of the participants felt that audio comments would afford better communication and relationships with their students.

When asked for expected advantages or disadvantages of audio feedback **AQ3**, all participants who provided an advantage indicated the speed or efficiency.

Participant 1: "The advantages would be that I would be able to leave more thorough comments through an audio comment, in a smaller time frame compared to typing as text.", similarly participant 3: ""Speed and clarity, precise nature of communication. Mainly speed."

Responses to **AQ4** were mixed with some participants indicating that they would use feedback tools within Doubtfire differently for online-based or campus-based students, with seemingly 2 schools of thought. Participant 1 notes: "There is a difference, cloud students need something like audio comments as it will be a lot easier to provide clear, concise feedback that they can understand. I would have loved to have had audio comments with previous cloud students (and reduce marking time). I can say a 200 word comment in a minute.", and participant 2: "Reception is different - cloud students humanises the content (feedback) delivery. To have that human element for students you won't see is beneficial." (Participant clarified they are more likely to use audio comments for cloud students than local students).

The second school of thought seems to see different uses of feedback tools as inequitable. Participant 3: "For the purpose of providing an consistent experience to all students, I wouldn't lean towards one feature or the other for cloud vs local students. If I increase audio comments for cloud I'll increase for local as well." and participant 4: "Doubtfire as a tool is a submission/communication, class Doubtfire is not used much, so no difference". While some participants believe they would use Doubtfire features differently for online or campus students, no participants believe that the system should have different features made available for the two categories of students.

Addressing **AQ5**, one participant noted that screen sharing and remote control would be useful, while many participants highlighted the need for a comment repository (reuse previously recorded or written comments).

6.2 Main Experiment

Timing data was recorded for participant's written and audio comment feedback. For written comments, timing began once the participant types the first character on the keyboard until they press send. For audio feedback, timing began once the participant clicked the record button, until they send the recording. There were two occurrences of re-records (where the participant deleted and re-recorded their verbal feedback), and the time recorded for those events include the total time required to provide the feedback, including the time of both recordings.

The results shown in Fig 4 present that on average, participants took less time providing audio feedback compared to written feedback across all eight categories of the simulated programming task submissions.

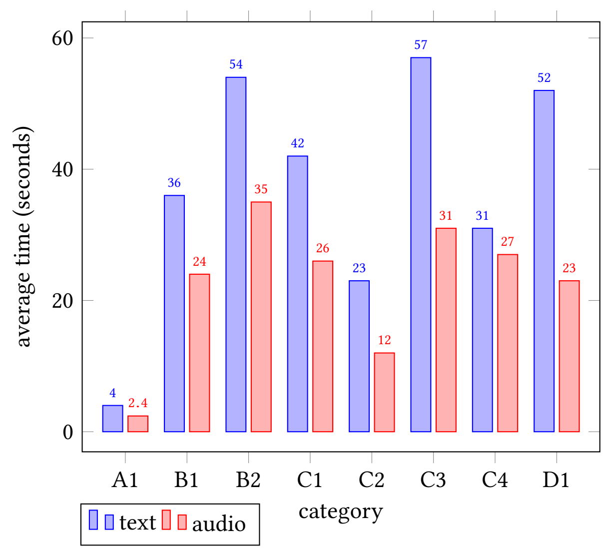

Figure 4: Average time spent providing audio feedback

Observations during the experiments did not indicate that the time reduction was due to detail of feedback, with similar points addressed in each piece of feedback.

The variations of writing and recording times across the eight categories in the Introductory Programming Task Feedback Taxonomy shown in Table 4 were not uniform. Excluding category **A1**, audio feedback was provided in 40% less time on average compared to written.

Outliers include **C2** at a 52% reduction, **C4** at 13% reduction and **D1** at a 55% reduction.

6.3 Experiences (Post-Experiment)

Following the post-experiment interview, seven of the eight participants felt that audio feedback was at least as efficient as written comments, with six finding it to be more efficient, an increase compared to the pre-experiment expectations. All eight of the participants felt that audio feedback would allow them to develop stronger connections with their students.

Considering the efficiency of feedback **BQ1**, participants were unanimous in that audio feedback was more efficient than written, and in many cases, more efficient than they expected. Some participants raised concerns with the efficiency, citing concerns of rambling or audible "umming". Participant 2: "Speed is good, but there is a danger of waffling and rambling.", and participant 5: " When you talk, you do some pausing, umming erring in a recorded a message...". Of particular note was participant's 3 response: "Typing is an impediment for me in terms of time, so I tend to write less and if you're marking lots you tend to be repetitive and start to reduce the complexity of information, whereas audio takes less time to produce, so I give higher quality".

BQ2 concerns the general thoroughness of audio feedback, but not explicitly programming-related issues. While the feedback was observed to be of similar thoroughness in both the written and audio feedback, five of the eight participants felt as if written feedback was more thorough than audio. Participant 1: "There's things I want to say in an audio comment, but it's hard to describe - so it's easier to type so they can see exactly what I'm talking about". Some participants explained this result to be due to lack of experience providing audio feedback. Participant 6 notes: "At least at the moment I think audio comments lead to less thorough feedback, I have less experience with it etc. But I think I would give more concise answers verbally, whereas with written I type a bit more with the intent of having more time to read through it. With verbally, students need to listen to me and relate it at the same time". Other participants indicated that the audio feedback was more thorough. Participant 3: "Ability to provide fast information leads to more likeliness of high-quality thorough feedback".

Regarding tutor-student connection **BQ3**, pre-experiment had seven participants expected audio comments to afford stronger student connection, with this figure raising to eight post-experiment. Participants noted that the audio feedback allowed more expression within the feedback, allowing them to "...add tone, texture or more elements to the information. I have been told that my typing is abrupt, I have to put smiley faces at the end of sentences, but it's just me having too much to do and being brief, audio lets me be more personal". (Participant 3). Participant 2: "I think audio conveys a lot more emotion than written - they might think I'm tearing them apart but I am trying to be general - so my tone, delivery, cadence could lessen the blow of harsh feedback and package it differently. Written is harsher and audio is more "these are the areas of improvement". Written makes your eyes drawn to syntax "bad bad bad" vs listening to the entire message.". All participants made similar comments to participant 2 and 3.

Finally, participants were asked for their thoughts audio comments being used to communicate code-specific issues to students **BQ4**. Results were not consistent; with Likert results had participants 1, 4, 6 and 7 indicating that written feedback was more effective at discussion code-specific issues. Participant 5 saw no different, with the others feeling that audio was more effective. All participants who designated written comments for **BQ4** refer to the inability to present code syntax verbally as the reason. "Audio comments can only be reference specific lines (there's an issue with line number 13 where there's a semicolon) whereas with a written comment I can identify the incorrect syntax right next to the correct syntax, so there's no guessing." (participant 1). and "...with written, you can give a direct example, whether their code with a fixed problem or a similar example, often the students haven't learnt to translate comments to code, a question about naming a function for example, the problem is they don't understand why they wrote it, so purely verbal answers to code problems might not solve that problem...". Participant 2 noted these issues, however highlighted that using audio and written comments in compliment addresses the above issues: "...with writing I can send a link but cannot with audio. If audio is also augmenting comments, then it solves this issue.".

7 DISCUSSION

Of the categories defined in our introductory programming task feedback taxonomy with the most significant time savings (D1, C1 and C2), we can begin to observe some similarities in the resulting feedback. These particular categories often require non-conceptual, open-ended feedback. Our results further indicated specific categories of feedback benefit more from audio feedback than others.

Exploring the differences between the two largest outlier categories, **C4** (very little difference) and **D1** (considerable savings with audio feedback). Category C4 (code misuse) requires tutors to discuss code features, which is seemingly better suited to written comments where syntax, code formatting and programming symbols are naturally represented. Category D1, Unaccompanied Questions, does not have a code submission associated, so participants are talking freely about ideas and approaches. Cases where not much feedback is required (A1), also do not benefit much from the use of audio feedback.

Does this mean audio feedback is not well-suited to programming related units? We don't think so. The major themes of audio feedback criticism collected in our qualitative results mainly indicated difficulty communicating code specific issues such as syntax. As participant 2 identified, audio feedback used together with written feedback provides benefits of the two modes. This idea is also identified in lessons-learnt from literature [24]. Tutors can provide verbal feedback, gaining the benefits of developing better relationships with students, while writing syntax-specific comments as a supplement. The comments do not need to overlap, with verbal feedback directing the student to provided links, references, or example code in a text comment. Overwhelming, all our participants identified audio feedback would help them better connect with students, and all but one participant found audio feedback to be faster.

The simulated code submissions given to participants in our experiment were designed to be small in size, often including only single issues. In real units, submissions are often more extensive, containing several issues ranging the categories identified. While time spent providing feedback and associate time savings afforded by audio recording presented in our results are relatively small in duration, and may not completely represent real-world findings, they still indicate efficiency gains. In practice, these efficiencies are amplified by the repeated marking of multiple weekly tasks undertaken each week throughout a unit. Task-oriented units with several weekly programming tasks naturally benefit further from these efficiency gains.

In our experiences, the relationship-building properties of audio feedback suits challenging fields like programming. The benefits of audio feedback have diffused tense situations in which online students have become frustrated with their lack of progress. We have also had many cases of online students indicating they appreciated and enjoyed receiving audio feedback and responding in turn with audio.

Many studies, including this one, highlight the benefits of audio feedback in higher education, yet it does not appear to be a common practice in most disciplines. While many learning management systems support audio comments via attachment, few support native, cross-browser recording and playback. Time and cognitive overhead of opening external audio recording applications to record feedback, save, and upload may be deterring teaching staff from providing audio feedback. In contrast, browsers natively support textual feedback. Technical hurdles faced during the development of our audio feedback system into Doubtfire may explain the lack of support. Native, cross-browser audio recording and playback without third-party plugins is a significant challenge, as the HTML5 MediaRecorder API is still not available on all major browsers. In our results and discussions with participants, there were no comments raised regarding the time or difficulty in producing the audio feedback. If we want to increase adoption of audio feedback in higher education, making the experience seamless and integrated should be a priority, similar to text comments.

The integration of audio recording into the learning management system is exceedingly important when considering the task-oriented portfolio approach. The nature of attachment-style is not conducive towards the desirable low-overheads, indicated by previous works [25].

An ancillary outcome of this work is the introductory programming task feedback taxonomy, shown in table 1. The taxonomy was designed to guide the series of simulated submissions, and we found both the development and use of the taxonomy provided structure for how to think about the different types of feedback programming tutors deliver.

Finally, we feel it important to discuss the social and accessibility aspects of audio feedback. Several of our participant's indicated that providing audio feedback to students felt "weird", and that they needed more experience to become comfortable. Users who are not native speakers may feel uncomfortable providing audio feedback and require training, encouragement, or allowed to refuse the use of audio recording. Tutors who find it difficult or are unable to record should similarly be able to fall back to an alternative, accessible method of providing feedback.

7.1 Early experiences from deployment

Following a code review process, audio comments have been made available in the Doubtfire system, and an updated version of the software has been deployed for use in our software engineering units. Initial review of Doubtfire server logs indicated users had sent several thousand audio comments in twelve months. Further analysis is ongoing to understand the context surrounding this figure; however, this indicates healthy adoption rates considering no formal awareness push had taken place.

Student responses to audio feedback have been overwhelmingly positive, with many students expressing appreciation when provided audio feedback. Similarly, tutors have reported feeling they could establish better relationships with online students by using audio comments to provide feedback. Particularly notable cases included situations where audio feedback helped ease tensions with students who had misinterpreted the tone of written feedback, leading to defensive and tense interactions.

We did not find audio comments to be a suitable replacement for written comments in all cases. When providing feedback in overly noisy, public or quiet environments, it may be inappropriate to record. In this case, we support pre-recorded audio uploads to Doubtfire, which is suitable in certain situations. Secondly, tutors for which English is not their first language may feel intimidated or concerned regarding their recorded English.

Having now used audio and written feedback for several teaching periods, we recognised a phenomenon we have termed "Perfect Comments". After presenting online students with technical questions either via text or audio comments within Doubtfire, for example asking students to explain what effects the ampersand (&) character has on a parameter in a C++ programming context, online students would often return with high quality, "perfect responses", while campus-based counterparts could not provide equally high-quality responses in person.

While further work is required to explore this further, we believe that given the affordances of asynchronous communication, students would research responses, or work with others before responding. Concerns were raised by the teaching team, as aspects of the discussion within the task-oriented portfolio approach becomes compromised. We outline our initial thoughts for scalable approaches to solving this problem in the future work section below.

8 THREATS TO VALIDITY

The small participant pool size of eight participants is a potential threat to validity. While the data capture undertook the experiment does include quantitative results, we utilised observations and long-form qualitative interview techniques to validate the findings.

Whether participants were native English speakers was not captured, nor was typing confidence or ability. These factors may impact the external validity of the results.

The simulated tasks developed for the experiment were small, around 15-30 lines of code. Small tasks were chosen to avoid overloading participants, as well as aligning with tasks commonly used in our task-oriented portfolio unit. While we expect efficiency to increase with the size of the task and associated feedback, further study of larger tasks and analysis of real-world feedback would be beneficial.

9 FUTURE WORK

The Introductory Programming Task Feedback Taxonomy outlined in section 5 was derived from reviewing historical submissions at one institution, however further work broadening the review to other educators and institutions would improve the validity and generalisability of the taxonomy.

We have not yet undertaken similar experiments or interviews with students to establish student perception of related work. While our experiences and other studies indicate positive student experiences receiving audio feedback, planned future work in this area consists of an action research approach to investigate how audio comment feedback and future iterations of discussion tools impact the teaching and learning outcomes in programming education.

Participants indicated a desire for other alternative feedback modes such as video capture, screen recording (including screen highlighting). We expect these modes of feedback to better support programming educators. Similar technologies to the HTML5 MediaRecorder API used in our implementation for cross-browser audio recording can support these features, and our plans include development and evaluation.

While we achieved good adoptions of the use of audio feedback in introductory programming, the asynchronous nature of written and audio comments does not provide tutor confidence in student acquisition of understanding, or in academic integrity. Our future work aims to design and evaluate more intelligent discussion tools, which can provide scalable, synchronous student-tutor discussions.

10 CONCLUSION

This paper describes our motivations, evaluation and approach to introducing an audio feedback tool into a large, task-oriented, portfolio-assessed programming unit.

We integrated a cross-platform, in-browser audio recording and playback feature into the open-source Doubtfire learning management system, a tool used primarily to support task-oriented portfolio assessment. The feature allows Doubtfire to access the computer's microphone, record and send audio feedback comments to student's for playback.

To review tutor expectations and experiences towards audio feedback, we conducted a three-part experiment with several introductory programming tutors. First, a pre-experiment interview established background, experience and expectations. The main experiment had participants review two series of simulated student programming submissions. Only half of the participants had access to our new audio recording tool for the first series, to mitigate potential ordering threats to validity; flipping access for the second series. Throughout the experiment, we collected timing data to evaluate efficiency differences between the two modes of feedback. Finally, a post-experiment interview established experiences following the use of audio feedback.

While our participant pool was small, our findings indicated that audio feedback is less time consuming than written for providing formative feedback on introductory programming submissions.

Follows is an outline of the practical contributions following our work:

- Different types of programming student submissions benefit differently from audio feedback. For longer-form explanations audio feedback decreases the time required to provide feedback, while enhancing student-tutor connection. For syntax-heavy feedback, using written feedback as a supplement to audio feedback appears to inherit the best properties of both modes of feedback. Even minor efficiency gains from adopting audio feedback in task-oriented, portfolio assessed programming units are amplified.
- To encourage audio feedback, ensure recording and playback capability are integrated seamlessly into the learning management system or environment; especially important in task-oriented approaches where tutors are marking weekly submissions.
- We believe online students will benefit the most from audio feedback, helping to bridge the equity of experience between campus students who see their tutor in-tutorial. Our student responses upon receiving audio feedback confirm this, with reports of positive learning experiences and appreciation.
- More can be done to supplement feedback provided to programming students. In our interviews, participants noted screen recording to highlight example code would be useful, while many participants highlighted the need for a comment repository (reuse previously recorded or written comments).

This work begins to validate findings from other disciplines, exploring how benefits of audio feedback can be applied in programming education. Audio feedback can improve student-tutor connection and relationships, and where required, tutors can provide supplemental textual feedback such as when providing code examples or sending links.

With growing online cohorts, opportunities for in-person communication is decreasing; this is particularly troublesome for challenging domains such as programming. We believe that audio feedback integrated into the learning environment as described in this paper, appears to present one more tool that programming educators can draw upon in supporting students, especially in task-oriented programming units, or units with large, mixed campus and online cohorts.

REFERENCES

[1] Sue D Achtemeier, Libby V Morris, and Catherine L Finnegan. 2003. Considerations for developing evaluations of online courses. *Journal of Asynchronous Learning Networks* (Feb. 2003). http://www.edtechpolicy.org/ArchivedWebsites/Articles/ConsiderationsDevelopingEvaluations.pdf

[2] Sussy Bayona-Oré, Jose A Calvo-Manzano, Gonzalo Cuevas, and Tomas San-Feliu. 2012. Critical success factors taxonomy for software process deployment. *Software Quality Journal* 22, 1 (Dec. 2012), 21–48. https://doi.org/10.1007/s11219-012-9190-y

[3] Jens Bennedsen and Michael E Caspersen. 2007. Failure rates in introductory programming. *ACM SIGCSE Bulletin* 39, 2 (June 2007), 32–36. https://doi.org/10.1145/1272848.1272879

[4] John Biggs. 1996. Enhancing teaching through constructive alignment. *Higher Education* 32, 3 (1996), 347–364. https://doi.org/10.1007/BF00138871

[5] John Biggs. 2011. Biggs' structure of the observed learning outcome (SOLO) taxonomy. *Assessment* (2011), 1–5. www.tedi.uq.edu.au/downloads/biggs_solo.pdf

[6] Andrew Cain. 2013. Constructive Alignment for Introductory Programming. (2013). http://www.ict.swin.edu.au/personal/jgrundy/theses/cain2013.pdf

[7] Andrew Cain and Muhammad Ali Babar. 2016. Reflections on applying constructive alignment with formative feedback for teaching introductory programming and software architecture. In *Proceedings of the 38th International Conference on Software Engineering Companion - ICSE '16*. ACM Press, New York, New York,

USA, 336–345. https://doi.org/10.1145/2889160.2889185
[8] Anrew Cain, John Grundy, and Clinton J Woodward. 2017. Focusing on learning through constructive alignment with task-oriented portfolio assessment. *European Journal of* (2017). http://www.tandfonline.com/doi/abs/10.1080/03043797.2017.1299693
[9] Clare Carruthers, Brenda McCarron, Peter Bolan, Adrian Devine, Una McMahon-Beattie, and Amy Burns. 2014. 'I like the sound of that' – an evaluation of providing audio feedback via the virtual learning environment for summative assessment. *Assessment and Evaluation in Higher Education* 40, 3 (May 2014), 352–370. https://doi.org/10.1080/02602938.2014.917145
[10] John Hattie and Helen Timperley. 2016. The Power of Feedback. *Review of Educational Research* 77, 1 (Nov. 2016), 81–112. https://doi.org/10.3102/003465430298487
[11] Pierre E Hertzog and Arthur J Swart. 2018. Student perceptions of audio feedback in a design-based module for distance education. *Global Journal of Engineering Education* (2018). http://ir.cut.ac.za/handle/11462/1505
[12] Philip Ice, Reagan Curtis, Perry Phillips, and John Wells. 2019. Using Asynchronous Audio Feedback to Enhance Teaching Presence and Students' Sense of Community. *Journal of Asynchronous Learning Networks* 11, 2 (Feb. 2019), 3–25. https://doi.org/10.24059/olj.v11i2.1724
[13] Ase Jevinger and Kristina Von Hausswolff. 2016. Large Programming Task vs Questions-and-Answers Examination in Java Introductory Courses. In *2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE)*. IEEE, 154–161. https://doi.org/10.1109/LaTiCE.2016.25
[14] Sundari Joseph and Charles Juwah. 2012. Using constructive alignment theory to develop nursing skills curricula. *Nurse Education in Practice* 12, 1 (Jan. 2012), 52–59. https://doi.org/10.1016/j.nepr.2011.05.007
[15] Clare Killingback, Osman Ahmed, and Jonathan Williams. 2019. 'It was all in your voice' - Tertiary student perceptions of alternative feedback modes (audio, video, podcast, and screencast): A qualitative literature review. *Nurse Education Today* 72 (Jan. 2019), 32–39. https://doi.org/10.1016/j.nedt.2018.10.012
[16] Tom Lunt and John Curran. 2010. 'Are you listening please?' The advantages of electronic audio feedback compared to written feedback. *Assessment and Evaluation in Higher Education* 35, 7 (Dec. 2010), 759–769. https://doi.org/10.1080/02602930902977772
[17] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A Becker, Michail Giannakos, Amruth N Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. 2018. *Introductory programming: a systematic literature review.* ACM. https://doi.org/10.1145/3293881.3295779
[18] Stephen Merry and Paul Orsmond. 2008. Students' Attitudes to and Usage of Academic Feedback Provided Via Audio Files. *Bioscience Education* 11, 1 (2008), 1–11. https://doi.org/10.3108/beej.11.3
[19] Daniel L Moody and Guttorm Sindre. 2003. Evaluating the Effectiveness of Learning Interventions: An Information Systems Case Study. *European Conference on Information Systems* (2003). https://aisel.aisnet.org/ecis2003/80
[20] Cecile Morris and Gladson Chikwa. 2016. Audio versus written feedback: Exploring learners' preference and the impact of feedback format on students' academic performance:. *Active Learning in Higher Education* 17, 2 (March 2016), 125–137. https://doi.org/10.1177/1469787416637482
[21] Shailendra Palvia, Prageet Aeron, Parul Gupta, Diptiranjan Mahapatra, Ratri Parida, Rebecca Rosner, and Sumita Sindhi. 2018. Online Education: Worldwide Status, Challenges, Trends, and Implications. *Journal of Global Information Technology Management* 21, 4 (Nov. 2018), 233–241. https://doi.org/10.1080/1097198X.2018.1542262
[22] Belinda Patterson and Cheryl McFadden. 2009. Attrition in online and campus degree programs. *Online Journal of Distance Learning Administration* 12 (2009). https://pdfs.semanticscholar.org/db53/a3c7b8ada105ba6789413f1be3df27c770e3.pdf
[23] Arnold Pears, Stephen Seidman, Lauri Malmi, Linda Mannila, Elizabeth Adams, Jens Bennedsen, Marie Devlin, and James Paterson. 2007. A survey of literature on the teaching of introductory programming. *ACM SIGCSE Bulletin* 39, 4 (Dec. 2007), 204–223. https://doi.org/10.1145/1345375.1345441
[24] Fiona Rawle, Mindy Thuna, Ting Zhao, and Michael Kaler. 2018. Audio Feedback: Student and Teaching Assistant Perspectives on an Alternative Mode of Feedback for Written Assignments. *The Canadian Journal for the Scholarship of Teaching and Learning* 9, 2 (Sept. 2018). https://ojs.lib.uwo.ca/index.php/cjsotl_rcacea/article/view/7039
[25] Jake Renzella and Andrew Cain. 2017. Supporting better formative feedback in task-oriented portfolio assessment. In *2017 IEEE 6th International Conference on Teaching, Assessment and Learning for Engineering (TALE)*. IEEE, 360–367. https://doi.org/10.1109/TALE.2017.8252362
[26] Ormond Simpson. 2007. The impact on retention of interventions to support distance learning students. *Open Learning: The Journal of Open, Distance and e-Learning* 19, 1 (Jan. 2007), 79–95. https://doi.org/10.1080/0268051042000177863
[27] UCube. 2019. *uCube - Higher Education Statistics.* Technical Report. http://highereducationstatistics.education.gov.au/
[28] Muhammad Usman, Ricardo Britto, Jürgen Börstler, and Emilia Mendes. 2017. Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method. *Information and Software Technology* 85 (May 2017), 43–59. https://doi.org/10.1016/j.infsof.2017.01.006
[29] Xiaoyan Wang, Yelin Su, Stephen Cheung, Eva Wong, and Theresa Kwong. 2013. An exploration of Biggs' constructive alignment in course design and its impact on students' learning approaches. *Assessment and Evaluation in Higher Education* 38, 4 (May 2013), 477–491. https://doi.org/10.1080/02602938.2012.658018
[30] Christopher Watson and Frederick W B Li. 2014. *Failure rates in introductory programming revisited.* ACM, New York, New York, USA. https://doi.org/10.1145/2591708.2591749
[31] Kathryn A Wood, Cary Moskovitz, and Theresa M Valiga. 2011. Audio Feedback for Student Writing in Online Nursing Courses: Exploring Student and Instructor Reactions. *Journal of Nursing Education* 50, 9 (Sept. 2011), 540–543. https://doi.org/10.3928/01484834-20110616-04
[32] Clinton J Woodward, Andrew Cain, Shannon Pace, Allan Jones, and Joost Funke Kupper. 2013. Helping students track learning progress using burn down charts. In *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2013*. IEEE, 104–109. https://doi.org/10.1109/TALE.2013.6654409