Graphics Libraries and 3D Printing as a Tool for Increasing Student Motivation in Introductory Programming Courses

Cameron Adams
Department of Computer Science
University of Cape Town
Cape Town, South Africa
ADMCAM003@myuct.ac.za

Maria Nanabhai
Department of Computer Science
University of Cape Town
Cape Town, South Africa
NNBMAR003@myuct.ac.za

CCS CONCEPTS

• **Computer Science education** → **Program visualization**; • **3D printing** → *Graphics libraries*;

KEYWORDS

Graphics libraries, visualization, 3D printing, web development

1 PROJECT DESCRIPTION

Computational skills are increasingly relevant across all fields in modern society. The importance of these skills has become globally recognised [10], but the number of individuals with these skills has not kept up to pace with this growth. This makes it essential to promote and improve introductory programming courses. However, teaching and learning programming has been found to be difficult, and students often lose interest in pursuing it [14]. The computer science (CS) major has some of the highest drop-out rates across all disciplines [16]. These challenges can be attributed to a variety of causes, including the difficulty of teaching dynamic concepts using static materials, preconceived notions of what a programmer looks like, and students perceiving CS as irrelevant, boring, or pointless in their lives [4, 9].

Several interventions have been proposed to address this challenge through making programming more practical, creative and engaging. These include visualisation tools and games. One such approach is to use programming, through graphical libraries, as a means for students to create a physical artefact. This can be done using digital fabrication machines such as 3D printers. Similar approaches have been found to achieve significant success in increasing student motivation towards programming, as we will discuss in section 5. However, the particular usage of 3D printing has not been studied in much detail.

The aim of this project is to investigate the effectiveness of 3D printing student-created models as a tool to increase student motivation and engagement in programming.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 PROJECT STATEMENT

Graphics libraries such as p5.js, Processing.js and Logo Turtle were created to motivate new programmers, which made drawing and creating program visualizations easy. Extending this principle, a proposed tool will be developed which allows students to create and print 3D artefacts. This tool will be designed for students who are not complete beginner programmers, and have some understanding of programming fundamentals. However, due to the COVID-19 pandemic, research will be conducted remotely. This means surveying students and using a 3D printer will be substituted with surveying educators and using 3D printing simulation.

The research questions are now:

(1) Is it possible to create a usable 3D graphics library for use in programmable 3D printing in novices?
(2) Is it possible to create a usable 3D printing simulator for use in programmable 3D printing in novices?
(3) Do educators consider a 3D graphics library a useful tool to increase student motivation in learning introductory programming?
(4) Do educators consider a 3D printer simulator a useful tool to increase student motivation in learning introductory programming?

3 PROCEDURES AND METHODS

In order to investigate the research questions, we first develop a user-friendly tool for beginner programmers to programmatically design and simulate the printing of 3D models. We then evaluate this tool among educators for suitability in an educational context.

3.1 System Design

Overview: The process of 3D printing an object consists of 4 stages: design, compilation, printing, and iteration [15]. The proposed tool, implemented as a web application, is designed to facilitate this whole process.

3.1.1 Design. Designing a 3D model programmatically requires a development environment, graphical libraries to specify the objects and parameters forming the model, and a previewing function. The core application provides a typical IDE (Integrated Development Environment) with a text editor and the ability to run the implemented code. The code3D API renders the code on-screen as a 3D visualisation of the model. It also offers additional features to make javascript programming easier for novices, such as simplified graphical libraries and more understandable error messages. The code3D API is built upon OpenJSCAD, an open source javascript-based language for computer-aided design. The visualisation features are described in more detail in 3.1.5.

3.1.2 Compilation. Javascript code cannot be used directly by a 3D printer. The code3D API converts this code into an STL (Surface Tessellation Language) file which is a mesh made up of triangles, which describe a 3D model, and is widely used in CAD (Computer-Aided Design) programs. The user can specify a number of 'printing parameters' in their Javascript code, parameters such as infill and filament width. The print3D API uses these printing parameters, to slice the STL into a gcode file. A gcode file is a list of instructions for a 3D printer to print a physical 3D object.

3.1.3 Printing. In lieu of a physical 3D printer, our system provides a 3D printing simulation. Note that the compiled code is compatible with and could be used for physical printing as well, but this is outside the scope of our project. With the compiled code, the print3D can now execute the gcode which simulates and allows interaction with a 3D printing.

3.1.4 Iteration. Research has found that a smooth transition between software design and the creation of an artefact improves motivation and user experience [5, 11] . The system is designed to be fully integrated within a single web application, making it easy for a user to make modifications to their code in the editor immediately upon finding flaws in their printed artefact. They can then reprint it if desired.

3.1.5 Visualization. Figuire 1 shows is a sequence diagram of the two visualizations and how they presented to the user.

A user implements javascript code that specifies a 3D artefact. The client sends this javascript file to a web server which runs the implemented code with the code3D API. The server returns a graphical representation of the artefact to be rendered in the user interface. The user can now interact with their 3D artefact by rotating and zooming into it. The second part of the tool is the 3D printing simulation. As before, the client sends this javascript file to a web server, which converts the user's implemented javascript code with the code3D and print3D APIs to a gcode file. The server now executes the gcode with the code3D API and returns the graphics to be rendered in the client's user interface. The user can now interact with the 3D print simulation by scrolling through the timeline of the print and as well as by rotating and zooming into it.

3.1.6 Architecture. An architectural diagram is shown in Appendix B. It shows a client-server model, which is typical of a web application. The front-end/user interface will be developed in React, and will use the Ace text editor. The back-end will be developed in Express and Node. These four are very popular javascript libraries used to create web applications. The project APIs, code3D and print3D, encapsulate open source graphics and 3D printing APIs. They also add extra functionality to make it as simple as possible for users to create and visualise 3D artefacts and prints.

3.2 Evaluation

3.2.1 Research Design. Due to the constraints imposed by the pandemic, we have chosen to use expert evaluation for assessing our system. On account of the small available sample size, we employ a mixed methods approach, combining both quantitative and qualitative methods. This includes surveys regarding system usability and motivation, as well as interviews and focus groups with participants to discuss in greater depth their experiences using the system. The usage of multiple measurement approaches means that results are more likely to be due to underlying phenomena than the particular method used, reducing bias and error and improving validity [12].

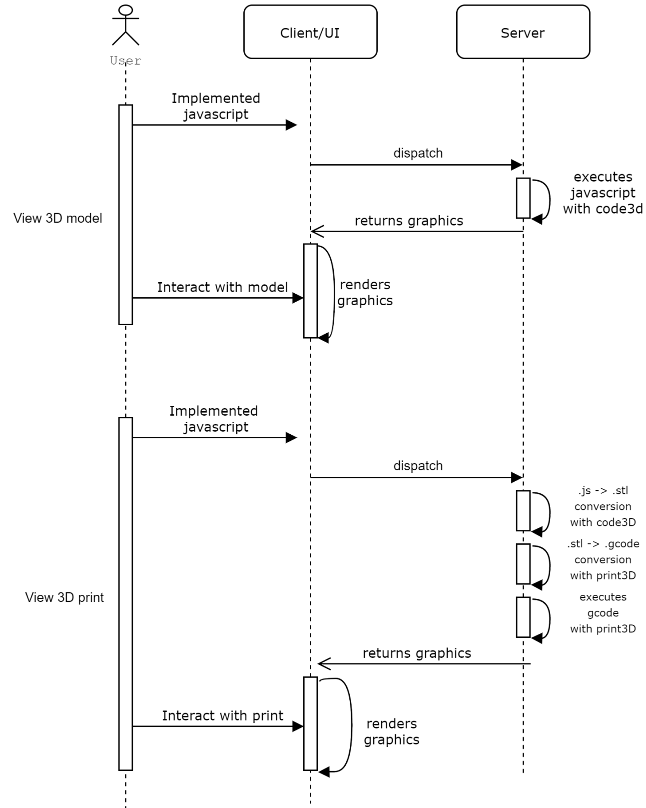


Figure 1: Sequence diagram of the two visualizations

3.2.2 Participants. All participants will be staff members from the Computer Science department at the University of Cape Town, recruited using the purposive sampling method through our supervisor, Gary Stewart. This means that participants are selected on the basis of the experience and knowledge they possess, which is an efficient method for collecting expert data [8]. In this case, we aim to recruit at least 5 educators and teaching assistants with experience teaching introductory programming courses.

3.2.3 Materials/Measures. The instruments discussed below have been widely applied and found to be reliable and valid, with reliability scores between 0.8 and 0.97.

Post-Study System Usability Questionnaire (PSSUQ). [13] Assessment of system usability will be done using the PSSUQ, a 19-item questionnaire with 7-point Likert items. This questionnaire provides four subscores: overall satisfaction, system usefulness, information quality, and interface quality.

Modified Survey About Educational Robotics for Instruction (M-SERI). [7] Instructor attitudes towards the system as a tool for teaching will be assessed using the M-SERI, an 18-item questionnaire with 5-point Likert items. Three subscores can be calculated: ability to use the tool, perceived utility, and intent to use it in the future. We modify the original survey by interchanging "educational robots" with "3D printing" and altering questions to concentrate on utility towards motivation.

3.2.4 Procedure. Evaluation will be done in two stages: initial testing and final assessment. An initial pilot study will be run on the system to garner feedback regarding obvious issues. This will consist of informal interviews where participants explore various features. Interviews will be conducted through a video call. Educators will visit the web app and try to complete a task using the tool in real-time, while the researchers ask and answer questions. For the final assessment, participants will be asked to perform assigned tasks, such as creating a specified object. First, they will use the provided libraries to create a 3D model of the object. They will then send their models to the virtual printer and observe the creation of the object. A task example could be: create a four sided pyramid with the box() function and the for loop, create a cone with the cylinder() function and the while loop, etc. An advanced task could be to use the box() function and recursion to create a tree fractal.

Once the tasks are complete, they will be given the two surveys to complete: the PSSUQ and M-SERI (see Appendix E for M-SERI survey), reflecting on their experiences using the system. After, interested participants will be invited to a focus group where they can discuss the tool, its features, and its applications in the classroom with fellow educators.

4 ETHICAL, PROFESSION LEGAL ISSUES

This study uses human participants and thus needs to consider associated ethical issues. Before participants will be allowed to partake in any activities relating to our study, they will be required to read and sign a consent form (Appendix D). There are no physical risks involved in the study. Ethical clearance will also be obtained from the faculty as well as the university. No identifiable participant information will be released with the results, and all data will be kept on a password protected computer only accessible to the researchers and supervisors on this study. All software identified for use has been published under open source licenses. The software developed in this study will also be released open source, under the GNU General Public License [1].

5 RELATED WORK

The use of graphics libraries to engage and motivate students in learning introductory programming has consistently shown to be useful. In the early days of CS education, Logo Turtle, which was used in over 250 classrooms, was found to have dramatic improvements in student engagement, confidence, problem-solving abilities, and motivation, among others [2]. Since then, a variety of newer tools have been found effective. A recent study [3] used Java TurtleGraphics to teach a fourteen-week introductory programming course at the University of Malaysia. Results showed students felt satisfied, confident and most importantly, motivated when doing assignments with the TurtleGraphics library.

The Processing Project [18] was created with the aim to facilitate an exploratory and design-oriented approach to teaching programming. The Processing programming language is based on Java and provides a flexible set of code elements to create visual designs in both two and three dimensions. Processing-based CS [20] report great success in improving student interest and retention through this approach, which they studied across two universities over six years.

A similar approach has also been extended to physical models. Processing-based libraries for creating designs to be forged by a laser cutter, for example, have also been successful, finding that almost all participants in workshops using them are interested in programming again in the future [11]. Moreover, reviews find that in general, sessions of 'making' improve participants' competence and engagement, as well as expanding their perceptions of the subject. Creating and interacting with objects increases experiences of autonomy, involvement and places learning into context. Students are more motivated to put effort into their work, and more interested in pursuing CS in the future. They also feel more comfortable and confident with programming. Almost all studies find positive results in this regard [17].

Limited research has been done on the applications of 3D printing, specifically, to introductory programming. Only one thorough study could be found on this, but it also found that students were motivated and engaged by the opportunity to manufacture their designs, making more effort than in previous workshops consisting only of digital design [6]. The students designed their models in the OpenSCAD language, of which OpenJSCAD, the language to be used in this project, is a derivation.

6 EXPECTED OUTCOMES

6.1 System

The completed system will consist of a minimal IDE, deployed on the web, with an integrated easy-to-use graphics library for the creation of 3D models, and a 3D printing API to preview the models and simulate their printing.

6.2 Expected Impact

There is a shortage of usable novice-oriented programmatic computer aided design tools. With 3D printing being a rising technology with widespread applications, this system, if satisfactorily usable, will provide an accessible entry point for beginners interested in 3D printing. It also provides an engaging and unique tool for teaching and learning programming. If our hypotheses are correct, this study will also provide evidence for an effective intervention to the problem of student motivation in computer science education. This is tied to improved student retention as well as better performance in programming courses [19].

6.3 Key success factors

- Successful implementation of environment and libraries that can display and print programmed models
- Successful implementation of printer simulation interface for modifying and visualising printed objects
- As assessed through evaluation:

- System being user-friendly and participants being able to use it to complete the assigned tasks.
- Participants finding the tools viable as a teaching tool, and having the intention to apply them in the future.

7 PROJECT PLAN

7.1 Risks

The risks for this project are outlined in a risk matrix that describes strategies for risk management and mitigation, as presented in appendix B. Risks observed with highest probabilities are ones that involve integrating the two tools together. Testing and designing the integration are ways of monitoring and mitigating these risks

7.2 Timeline

Our project runs from 2 June until 19 October, where we submit our web page. See appendix A for a Gantt chart detailing the full project schedule.

7.3 Resources required

Software: The web application will be created using Javascript libraries – React, Ace, Express and Node. The code3d and print3d APIs will build upon the OpenJSCAD, jscut, and Slic3r APIs. Skype will be used as a video conferencing tool for remote interviewing, and Google Forms for remote surveying. Data analysis will be done in Microsoft Excel.

Hardware: The researchers will use their own personal computers for development.

Human resources: Educators will be recruited from the computer science department; our supervisor Gary Stewart will assist in this process.

Web hosting: Google's Firebase offers free hosting. Amazon Web Services can be another option.

7.4 Deliverables

The most central deliverable of the project will be an academic paper presenting the results of our research. This will include detailing if, and for which visualization tools are usable and if educators consider them to increase student motivation in learning introductory programming. A summarised list of all notable deliverables (some already completed) is presented below:

- Two literature reviews
- Project Proposal
- Revised project proposal
- Draft of final paper
- Final paper
- Implemented code and 3D print visualization Web App
- Project Poster
- Web page

7.5 Milestones

Notable milestones include the revised project proposal to be submitted on the 29th of June, initial feasibility to be demonstrated on the 3rd of August (after which user testing will be run) and final code to be submitted on the 25th of September. The paper should be worked on from the end of June until a final draft is submitted on the 11th of September. The final paper will be submitted on the 21th of September. The project will then be demonstrated on the 5th of October. Additional milestones can be found in appendix A.

7.6 Work Allocation

Work will be split amongst the team members in the following way: Cameron Adams will focus on developing the print3D API which simulates a 3D print of an implemented javascript program by using and extending open source 3D slicing and 3D printing simulation APIs. The code editor and code3D API, which visualises a 3D artefact, will be developed by Maria Nanabhai by using and extending previous graphics libraries. Both members will collaborate to integrate these two tools by developing a web app. This combined tool will be used to survey and test educators to answer our research questions.

REFERENCES

[1] 2007. GNU General Public License. (2007). http://www.gnu.org/licenses/gpl.html

[2] 2015. Logo history. (2015). https://el.media.mit.edu/logo-foundation/what_is_logo/history.html

[3] Marini Abu Bakar, Muriati Mukhtar, and Fariza Khalid. 2020. The Effect of Turtle Graphics Approach on Students' Motivation to Learn Programming: A Case Study in a Malaysian University. *International Journal of Information and Education Technology* 10, 4 (2020).

[4] Maureen Biggers, Anne Brauer, and Tuba Yilmaz. 2008. Student perceptions of computer science: a retention study comparing graduating seniors with cs leavers. *ACM SIGCSE Bulletin* 40, 1 (2008), 402–406.

[5] Leah Buechley, Mike Eisenberg, and Nwanua Elumeze. 2007. Towards a curriculum for electronic textiles in the high school classroom. In *Proceedings of the 12th annual SIGCSE conference on Innovation and technology in computer science education*. 28–32.

[6] Christos Chytas, Ira Diethelm, and Alexandros Tsilingiris. 2018. Learning programming through design: An analysis of parametric design projects in digital fabrication labs and an online makerspace. In *2018 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 1978–1987.

[7] Todd I Ensign. 2017. Elementary educators' attitudes about the utility of educational robotics and their ability and intent to use it with students. (2017).

[8] Ilker Etikan, Sulaiman Abubakar Musa, and Rukayya Sunusi Alkassim. 2016. Comparison of convenience sampling and purposive sampling. *American journal of theoretical and applied statistics* 5, 1 (2016), 1–4.

[9] Anabela Gomes and António José Mendes. 2007. Learning to program-difficulties and solutions. In *International Conference on Engineering Education–ICEE*, Vol. 2007.

[10] Steve Hodges, Sue Sentance, Joe Finney, and Thomas Ball. 2020. Physical computing: A key element of modern computer science education. *Computer* 53, 4 (2020), 20–30.

[11] Jennifer Jacobs and Leah Buechley. 2013. Codeable objects: computational design and digital fabrication for novice programmers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1589–1598.

[12] R Burke Johnson, Anthony J Onwuegbuzie, and Lisa A Turner. 2007. Toward a definition of mixed methods research. *Journal of mixed methods research* 1, 2 (2007), 112–133.

[13] James R Lewis. 1992. Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 36. SAGE Publications Sage CA: Los Angeles, CA, 1259–1260.

[14] Louis Major, Theocharis Kyriacou, and O Pearl Brereton. 2012. Systematic literature review: teaching novices programming using robots. *IET software* 6, 6 (2012), 502–513.

[15] Chandrakana Nandi, Anat Caspi, Dan Grossman, and Zachary Tatlock. 2017. Programming language tools and techniques for 3D printing. In *2nd Summit on Advances in Programming Languages (SNAPL 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

[16] Matthew W Ohland, Sheri D Sheppard, Gary Lichtenstein, Ozgur Eris, Debbie Chachra, and Richard A Layton. 2008. Persistence, engagement, and migration in engineering programs. *Journal of Engineering Education* 97, 3 (2008), 259–278.

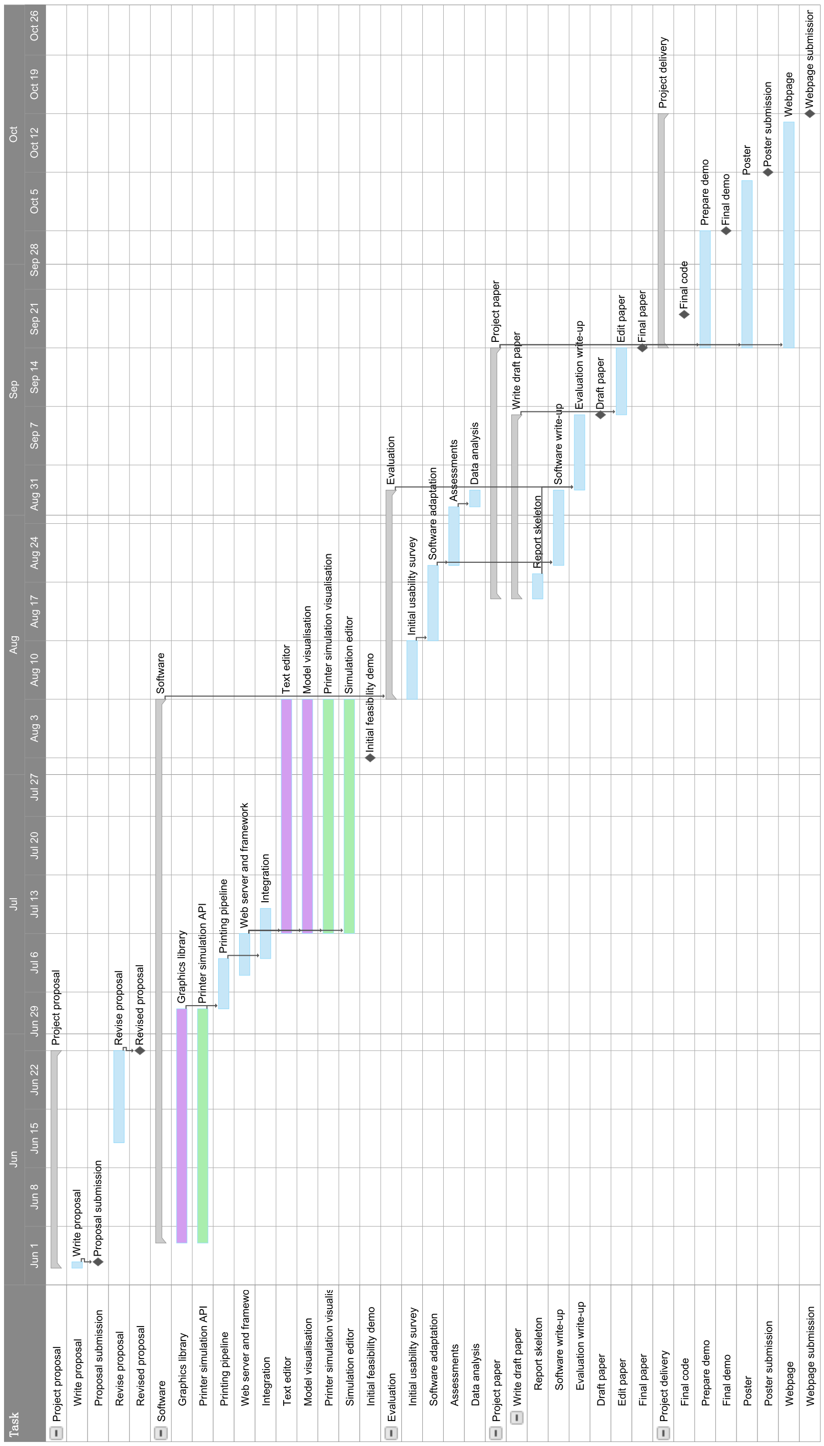
[17] Sofia Papavlasopoulou, Michail N Giannakos, and Letizia Jaccheri. 2017. Empirical studies on the Maker Movement, a promising approach to learning: A literature review. *Entertainment Computing* 18 (2017), 57–78.

[18] Casey Reas and Ben Fry. 2006. Processing: Programming for the Media Arts. *AI Soc.* 20, 4 (Aug. 2006), 526–538. https://doi.org/10.1007/s00146-006-0050-9

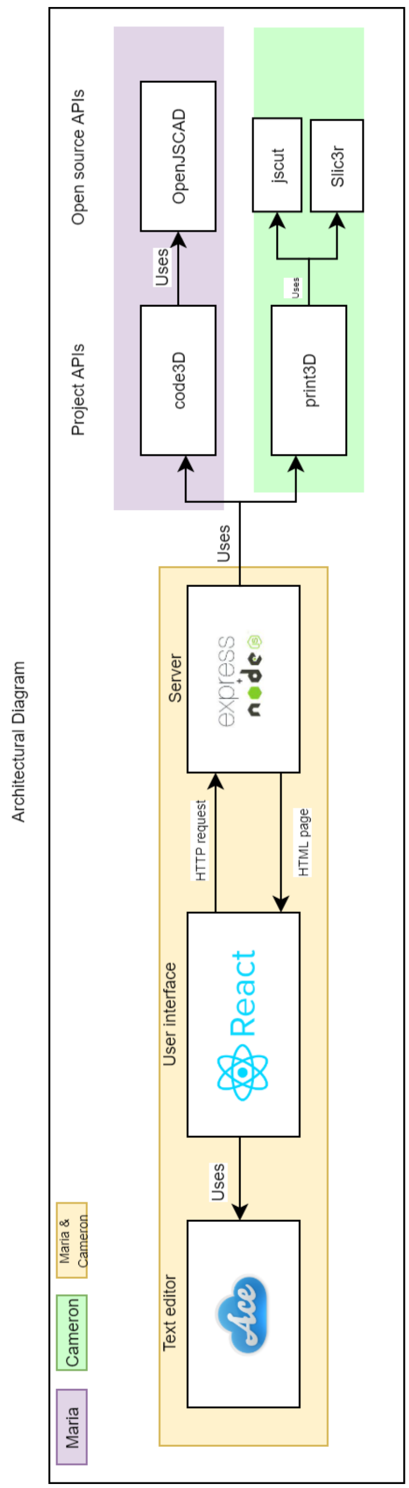
[19] Richard M Ryan and Edward L Deci. 2000. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist* 55, 1 (2000), 68.

[20] Dianna Xu, Ursula Wolz, Deepak Kumar, and Ira Greenburg. 2018. Updating Introductory Computer Science with Creative Computation. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 167–172.

A GANTT CHART



B ARCHITECTURAL DIAGRAM



C RISK MATRIX

#	Risk	Probability	Impact	Monitoring	Mitigation	Management
1	Integration challenges between various APIs	8	8	Frequent testing	Design system so that changes can be effected more easily	Reconfigure components to integrate
2	Unexpected schedule changes	7	6	Keep up to date with current developments; Keep close track of progress along timeline	Ensure sufficient buffers in timeline	Allocate more resources (work time)
3	Difficulty obtaining committed participants	3	10	Check in with participants periodically	Contact participants in advance, offer incentives	Consult supervisor for advice. Consider alternative participants or revise methodology.
4	Not meeting user requirements	3	9	Pilot test	Design system so that changes can be effected more easily	Update system to meet requirements
5	Conflict between team members	2	7	Check up on each other	Maintain open communication channels	Have a mediation session
6	Web architecture decreasing render time	9	6	Frequent testing and comparing	Use more optimized libraries, try different web services	Keep a report of render time test to see if optimization have worked

INFORMED CONSENT FORM

Dear Participant,

<u>Graphics Libraries and 3D Printing</u>

This study looks to investigate the applications of 3D printing to improving student motivation in introductory programming courses.
We are looking for staff in the computer science department who have experience teaching such courses.

Choosing to partake in this study will require you to:

1. Complete some sample programming exercises to create and print a model
2. Fill out some surveys regarding your experience working with the 3D printing interface.

If you are interested, we may invite you to an interview to expand on your experiences. There are no physical risks in the study, and you will not be required to partake in any physical activities.

You may choose not to participate in the study and your decision has NO negative effects on you or your academic career at UCT.
You are also free to stop participation in the study at any time without consequences. Participation is completely voluntary and there will be no repercussions for those wishing to stop partaking in the study.

All the information collected will be kept confidential and only known to the researchers on this study. We will not publish your name with our findings, and we will be the only people to handle this information.

Any study-related questions, problems, or emergencies should be directed to the researchers at the following email addresses:

Cameron Adams: ADMCAM003@myuct.ac.za
Or
Maria Nanabhai: NNBMAR003@myuct.ac.za

Questions about your rights as a study participant, comments or complaints about the study also may be presented to:
Gary Stewart: gstewart@cs.uct.ac.za

Survey About Educational 3D Printing for Instruction

Directions: Below are 18 statements about the use of 3D printing (specifically the Code Print 3D interface) for instruction.
For each statement, there is a scale of responses that range from "strongly disagree" to "strongly agree". There are no correct or incorrect responses. For each statement, please circle the number (only one) that corresponds to **your** response.
Please be sure to answer each question.

Statement		Strongly disagree	Disagree	Not sure	Agree	Strongly agree
1	3D printing sounds like problematic instructional technology to me.	**0**	**1**	**2**	**3**	**4**
2	I don't know how to use 3D printing (i.e. 3D printing is unfamiliar to me)	**0**	**1**	**2**	**3**	**4**
3	Some day, I will use 3D printing in my classroom.	**0**	**1**	**2**	**3**	**4**
4	If given the opportunity, I would like to learn to use 3D printing for instructional activities	**0**	**1**	**2**	**3**	**4**
5	Children should be introduced to 3D printing in school.	**0**	**1**	**2**	**3**	**4**
6	I am confident that I could learn to use 3D printing for teaching.	**0**	**1**	**2**	**3**	**4**
7	The challenge of using 3D printing for instruction does not appeal to me.	**0**	**1**	**2**	**3**	**4**
8	I think 3D printing is not interesting.	**0**	**1**	**2**	**3**	**4**
9	I am unsure of my ability to integrate 3D printing into my classes.	**0**	**1**	**2**	**3**	**4**
10	I will do as little work with 3D printing as possible.	**0**	**1**	**2**	**3**	**4**
11	I think that integrating 3D printing with teaching would take too much time.	**0**	**1**	**2**	**3**	**4**
12	I am willing to spend time setting up 3D printing for instruction.	**0**	**1**	**2**	**3**	**4**
13	I believe 3D printing will help motivate students.	**0**	**1**	**2**	**3**	**4**
14	I would enjoy using 3D printing in my classes.	**0**	**1**	**2**	**3**	**4**
15	I think working with 3D printing in class would be enjoyable and stimulating.	**0**	**1**	**2**	**3**	**4**
16	I think using 3D printing in class would not work for me.	**0**	**1**	**2**	**3**	**4**
17	3D printing can create more learning opportunities for students.	**0**	**1**	**2**	**3**	**4**
18	3D printing is a valuable educational tool.	**0**	**1**	**2**	**3**	**4**