GaPA: An Application to Assist Novice Users With 3D Printing

A Major Qualifying Project

Submitted to the Faculty of Worcester Polytechnic Institute
in fulfilment of the requirements for the degree in

Bachelor of Science
in Computer Science
and Mechanical Engineering

By

Collin Touchette (ME)
James Scherick (CS)
Matthew Gulbin (CS)
Parker Coady (CS)

Advised by

Professor David C. Brown (CS)
Professor Pradeep Radhakrishnan (ME)

May 6, 2021

Abstract

The 3D printing process can be daunting to novices, requiring knowledge of a multitude of settings to determine a part's optimal printing configuration. We have developed a software application that guides beginners through the 3D printing process, which utilizes an intuitive UI for novices. This program inputs a user's .stl file and asks about its purpose through multiple choice questions, then uses a heuristic algorithm to determine the top 4 orientations and associated settings. The user can then generate the G-Code for each orientation. The goal of this program, titled G-Code and Printer Automation (GaPA), is to be used as an introductory slicing software. It is currently being used by large groups of students in Design and Manufacturing courses at WPI, with positive results.

Acknowledgements

We would like to thank all those who made this project possible. Firstly, we would like to thank Professors David Brown and Pradeep Radhakrishnan for their support, vital advice, and consistent guidance, ensuring that GaPA’s development remained on track. We would also like to thank Mitra Anand, Advanced Technology and Prototyping Specialist at the Makerspace and Professor Erica Stults at the Rapid Prototyping Lab for their consultation on 3D printing and slicing programs. Lastly, we would like to thank all the students of the Mechanical Engineering department at WPI who tested GaPA, and whose valuable feedback helped give us a deeper insight into our project.

Contributions

Section	Primary Author(s)	Primary Editor
Abstract	Parker Coady	
Acknowledgements	Parker Coady	
1. Introduction	Collin Touchette Parker Coady	
2.1. 3D Printing and Slicing	Collin Touchette	Parker Coady
2.2. Usability and UI Design	Matthew Gulbin	Parker Coady
2.3. UI Design Principles	Matthew Gulbin	Parker Coady
2.4. Interface Evaluation Principles	Matthew Gulbin	
2.5. Available Slicing Software	Parker Coady	
2.6. Testing and Evaluation	James Scherick	Parker Coady
3. Motivations, Requirements, and Goals	Parker Coady	
4. Methodology	Parker Coady	
5.1. User Flow Diagram	Parker Coady	James Scherick
5.2. Program Technical Details	Matthew Gulbin	
5.3. Evolution of the Interface Design and Rationale	Matthew Gulbin Parker Coady	
5.4. Mode 1: Import File	Parker Coady	Matthew Gulbin
5.5. Mode 2: Determine Settings	Parker Coady	Matthew Gulbin
5.6. Mode 3: Solution Selection	Matthew Gulbin	Parker Coady
5.7. Printer Specification	Matthew Gulbin	
5.8. Similarities and Differences to Currently Available Slicers	James Scherick	
5.9. User Requirements	Collin Touchette	

5.10. Cause and Effect of Settings on the User's Desires	Collin Touchette	Parker Coady
5.11. Other Cases	Collin Touchette	
5.12. Materials	Collin Touchette	Parker Coady
5.13. Printer Information	Collin Touchette	
5.14. IF/THEN Statements	Collin Touchette	
5.15. Survey Questions	Collin Touchette	
5.16. Generating a List of all Valid Orientations	James Scherick	Collin Touchette
5.17. Scoring Each Orientation	James Scherick	Parker Coady
5.18. Selecting a Printer and Applying Settings	Collin Touchette	Parker Coady
5.19. Exporting .stl and G-Code files	James Scherick	
5.20. Scoring Evaluation	Collin Touchette	James Scherick
6. Testing Procedure	James Scherick	Collin Touchette
7. Testing Results and Analysis	James Scherick Collin Touchette	Parker Coady
8. Broader Impact on Mechanical Engineering	Collin Touchette	
9. Team Members' Applied and Developed Skills	Collin Touchette	
10. Conclusion	Matthew Gulbin	Parker Coady
10.1. Future Work	Matthew Gulbin Collin Touchette	
11. References	ALL	
12. Appendices	ALL	

Table of Contents

Abstract ……….……………………………………………………………………ii

Acknowledgements ...…………………………………………………………………...iii

Contributions …..……………………………………………………………………...iv

Table of Contents ………………..…………………………………………………....vi

List of Figures...……………………………………………………………………....ix

List of Tables ………………………………………………………………..……..xii

1. Introduction …………………………………………………………………………1

2. Literature Review …………………………………………………………………...3

2.1. 3D Printing and Slicing……………………………..………………………3
2.2. Usability and UI Design …………………………………………………………11
2.3. UI Design Principles ………………………….………………………………12
2.4. Interface Evaluation Principles ………………………………………………...14
2.5. Available Slicing Software …………………………………………………...16
2.6. Testing and Evaluation …………………………………………………………25

3. Motivations, Requirements, and Goals ……………………………………………..29

4. Methodology………………………..……………………..……………………..31

4.1. A Term Methodology …………………………………………………………....31
4.2. B Term Methodology …………………………………………………………....36
4.3. C Term Methodology …………………………………………………………....42
4.4. D Term Methodology …………………………………………………………....45

5. Design ……………………………………………………………………………...47

5.1. User Flow Diagram……………………………………………………………..47
5.2. Program Technical Details …………………………………………………….50
5.3. Evolution of the Interface Design and Rationale …………………………………51
5.4. Mode 1: Import File …………………………………………………………….60
5.5. Mode 2: Determine Settings …………………………………………………...63
5.6. Mode 3: Solution Selection ……………………………………………………65
5.7. Printer Specification ……………………………………………………………67
5.8. Similarities and Differences to Currently Available Slicers………………………69
5.9. User Requirements ……………………………………………………………..70
5.10. Cause and Effect of Settings on the User’s Desires……………………………71

5.11. Other Cases 87
5.12. Materials 88
5.13. Printer Information 89
5.14. IF/THEN Statements 91
5.15. Survey Questions 92
5.16. Generating a List of all Valid Orientations 96
5.17. Scoring Each Orientation 98
5.18. Selecting a Printer and Applying Settings 108
5.19. Exporting .stl and G-Code files 109
5.20. Scoring Evaluation 110

6. Testing Procedure 118
6.1. Software Testing 119
6.2. User and UI Testing 121
6.3. Introduction and Consent Form 122
6.4. Background Information 123
6.5. Testing Sequence 124
6.6. Follow-Up Questions 124
6.7. Concluding the Test 125

7. Testing Results and Analysis 127
7.1. User Experience 127
7.2. Program Speed Expectations 128
7.3. Orientation Optimization 130
7.4. Areas of Highest Difficulty 132
7.5. Most Disliked Aspects 135
7.6. Question Clarity 137
7.7. Satisfaction With Program 131
7.8. Program Evaluation 140
7.9. Feedback from Professionals 141

8. Broader Impact on Mechanical Engineering 144
8.1. ASME Code of Ethics 144
8.2. Societal and Global Impact 144
8.3. Environmental Impact 145
8.4. Codes and Standards 145
8.5. Economic Factors 145

9. Team Members' Applied and Developed Skills 147

10. Conclusion 149
10.1. Future Work..151

11. References..154

12. Appendices 159
Appendix A: List of If/Then Statements for Settings 159
Appendix B: List of Survey Questions..163
Appendix C: Consent Form 166
Appendix D: List of Survey Free Responses 168

List of Figures

Figure 2.1: 3D Printing market growth (CBRE et al., 2017)…………………………………..4
Figure 2.2: G-Code example ……………………………………………………………….5
Figure 2.3: Infill density visual example (Siber, 2018) ……………………………………….8
Figure 2.4: Platform adhesions (Simplify3D, 2020)………………………………………….9
Figure 2.5: Supports Visualization (Chakravorty, 2020) ………………………………….10
Figure 2.6: Slic3r main screen……………………………………………………………….17
Figure 2.7: Slic3r main screen with an .stl file loaded …………………………………….18
Figure 2.8: Slic3r's print settings window ………………………………………………...19
Figure 2.9: Slic3r Configuration Wizard ……………………………………………………..20
Figure 2.10: MatterControl main screen …………………………………………………….20
Figure 2.11: MatterControl slicer settings …………………………………………………...22
Figure 2.12: Cura main screen ……………………………………………………………….23
Figure 2.13: Cura print settings ……………………………………………………………….24
Figure 4.1: Slicer Analysis Spreadsheet ……………………………………………………..33
Figure 4.2: Platform Decision Matrix ………………………………………………………...35
Figure 4.3: Gantt Chart for B Term work schedule ....……………………………………..36
Figure 4.4: UML class diagram ……………………………………………………………….38
Figure 4.5: System Block Diagram …………………………………………………………..39
Figure 4.6: Gantt Chart for C Term work schedule ……………………………...……....42
Figure 4.7: Gantt Chart for D term work schedule ………………………………………....44
Figure 5.1: First draft of User Flow Diagram …..…………………………………………….47
Figure 5.2: Second draft of User Flow Diagram …………………………………………....49
Figure 5.3: Example of four orientations generated for a model by our system …...…....52
Figure 5.4: Early low fidelity prototype of Mode 1 ………………………………….……....53
Figure 5.5: Early low fidelity prototype of Mode 2 …………………………………………54
Figure 5.6: Early low fidelity prototype of Mode 3 …………………………………………55
Figure 5.7: Final low fidelity prototype of Mode 2 …………………………………………58
Figure 5.8: Final low fidelity prototype of Mode 3………………………………………….59
Figure 5.9: Mode 1, before importing a model………………………………………………...61
Figure 5.10: Mode 1, with File Explorer window open to import a model……………….61
Figure 5.11: Mode 1, with model imported …………………………………………………62
Figure 5.12: Mode 2, with an example multiple choice question………………………....64
Figure 5.13: Mode 2, with example force question ………………………………………...64
Figure 5.14: Mode 3 Interface Design………………………………………………………66
Figure 5.15: Printer Specification Screen …..………………………………………………68

Figure 5.16: Standing, On-Edge, and Flat Orientations [Heckenlively 2019] 74
Figure 5.17: Eyelid CAD Model 76
Figure 5.18: Servo Mount CAD Model 76
Figure 5.19: Printed eyelids - layer heights 0.06 to 0.6 mm from left to right 77
Figure 5.20: Printed servo mounts - layer heights 0.06 to 0.6 mm..........................77
Figure 5.21: Comparison of diagonals with layer heights 0.2 and 0.3 mm 78
Figure 5.22: Tensile and Yield Strength by Infill Density
(Concentric Pattern) [Pandzic 2019] 80
Figure 5.23: Rectilinear infill pattern compared to concentric [Engineerdog 2015] 81
Figure 5.24: Orientation for a rod with tension applied on the length 82
Figure 5.25: Shell thickness example [Jesimon 2014] 83
Figure 5.26: Extrusion thickness [exaCNC Kitchen 2019] 83
Figure 5.27: Comparison of Orientations for Flexural Stress 85
Figure 5.28: Grid input method..93
Figure 5.29: Ranking input method 94
Figure 5.30: Question tree arrangement...95
Figure 5.31: Example .stl file 98
Figure 5.32: Example orientation of the model 99
Figure 5.33: Maximum surface area 100
Figure 5.34: Minimum surface area 101
Figure 5.35: Low Overhang Score (Meaning Lots of Overhang) 102
Figure 5.36: High Overhang Score (Meaning Little Overhang) 103
Figure 5.37: Example of low slenderness score
(Meaning high amounts of slenderness) 104
Figure 5.38: Example of High Slenderness Score
(Meaning low amounts of slenderness) 104
Figure 5.39: R1 and R2 example...105
Figure 7.1: 3D Printing experience of test users 127
Figure 7.2: Very experienced users' speed evaluation 128
Figure 7.3: Moderately experienced users' speed evaluation 129
Figure 7.4: Not at all experienced users' speed evaluation 129
Figure 7.5: Very experienced users' orientation evaluation 130
Figure 7.6: Moderately experienced users' orientation evaluation..........................131
Figure 7.7: Not at all experienced users' orientation evaluation............................131
Figure 7.8: Very experienced users' difficulty evaluation.....................................133
Figure 7.9: Moderately experienced difficulty evaluation.....................................133
Figure 7.10: Not at all experienced difficulty evaluation......................................134
Figure 7.11: Very experienced users' most disliked aspects................................136

Figure 7.12: Moderately experienced users' most disliked aspects……………………….136
Figure 7.13: Not at all experienced users' most disliked aspects…………………………137
Figure 7.14: Moderately experienced users' unclear questions ………………………...138
Figure 7.15: Not at all experienced users' unclear questions…....……………………...139
Figure 7.16: User satisfaction………………………………………………………………140

List of Tables

Table 5.1: Cause and Effect of Settings on Physical Properties of 3D Print................71
Table 5.2: Layer Height Profile Names and Corresponding Heights..........................76
Table 5.3: Pro and Con score ranges 109
Table 5.4: Models used in testing..111
Table 5.5: Orientation tests 114
Table 5.5: Orientation tests (cont.)..115
Table 5.5: Orientation tests (cont.)..116

1. Introduction

3D printing can be overwhelming to those new to the process. Currently, if a user wants to print a 3D model from an .stl file, which contains a digital representation of spatial coordinates, they need to have substantial knowledge of multiple complex printer settings. Among these settings are the model's orientation, layer height and infill density. The object's final appearance and physical properties could be significantly different depending on the exact parameters of these settings.

These problems are compounded by the current selection of software that is supposed to assist with 3D printing. These programs, such as Cura (Ultimaker Cura, 2021), MatterControl (Matter Hackers Inc, 2021), and Slic3r (Slic3r, n.d.), convert an .stl file to G-Code, which is a file type that can be interpreted by 3D printers and contains machine instructions. However, creation of G-Code requires specification of the previously-mentioned complex printer settings, as well as the printed part's orientation. Currently available slicing programs overwhelm users with an abundance of options to configure these settings. Novice users are unlikely to know how these settings and the orientation affect the quality of their printed part, leading to possible low-quality prints.

The goal of this project is to create an application that simplifies the G-Code generation process for users by analyzing the user's .stl file and automatically determining the best settings and orientation for printing the part. This program, titled G-Code and Printing Automation (GaPA), aims to be easy to understand for both 3D printing novices and experienced users alike. The application would serve as an introduction to other slicing software. It will offer an intuitive user interface and automate the setting and orientation selection process and give users information about the settings until they are comfortable enough choosing them on their own. This report concerns the creation of this application, and how it automatically determines printer settings and orientation from an .stl file. It then discusses testing and final evaluation of the program.

Throughout this report, we discuss relevant background information for the project in the Literature Review chapter. We then establish the motivation behind the project and the requirements of the final application. In the Methodology chapter, we outline the process through which GaPA was developed. The Design chapter goes into detail about the various decisions behind the project and implementation. We then explain the testing process of the software and the analysis of the results that followed. Finally, the report concludes with a reflection on the goals of the project and future work that could be done to improve GaPA.

2. Literature Review

The aim of GaPA is to assist with and simplify the 3D printing process by automating certain steps and providing an intuitive interface while doing so. To better understand this program's goals, this chapter provides information on 3D printing, slicing, and user interface design. It also contains an analysis of commonly used 3D printing software. The features and UI in current slicing tools will be analyzed and applied to GaPA's design.

2.1. 3D Printing and Slicing

3D printing is currently the most accessible and popular form of additive manufacturing available. Characterized by layering material to build up the finished piece, additive manufacturing is in contrast to subtractive machining or molding. 3D printing technology was first developed in the late 1980s for the purposes of rapid prototyping. Since then, the technology has become more widely available, becoming commercially available in 2009, and is now used in many different sectors, from students to hobbyists to industrial corporations with the growth of the industry illustrated in Figure 2.1 (CBRE et al., 2017).

3D printing provides many advantages over traditional manufacturing. Because every part is built up by layers using the same tool, there is no need for unique molds or specialized cutting tools. This also provides the benefit of greater complexity, especially on the interior of parts. There are a variety of different processes that are considered 3D printing that use different materials and heating methods, but the most common one is Fused Deposition Modelling, or FDM (3D Printing Industry, 2017).

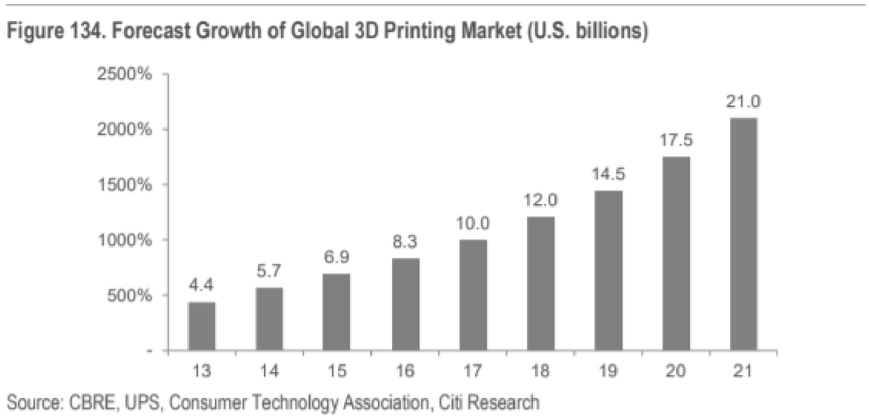


Figure 2.1: 3D Printing market growth (CBRE et al., 2017)

FDM is the most commonly used 3D printing method and will be the focus of our project. . In this process, the material is heated up to just past its glass transition temperature and extruded through a nozzle. The nozzle is moved in a specific pattern to create the individual layers on the printing bed, building them up one by one. FDM allows for greater scalability as the limiting factor of the process is the bed size. As the technology of 3D printing progresses, FDM presents the easiest path to larger scale printing. Another major advantage of FDM is the variety of materials that can be used; anything that can be heated to a temperature at which it can be extruded is usable. There are, however, some downsides to this process. The part quality is ultimately dependent on the nozzle of the printer, as the resolution it can print at is limited by the size of extrusion (Grames, 2020). Additionally, because parts are printed layer-by-layer, there is an inherent weakness between the layers because of the limited connection along the Z-Axis. As a result, parts fabricated with FDM are stronger on their XY-Plane (M. Anand, personal communication, October 6, 2020).

In order to print an object, an FDM printer requires instructions in the form of G-code. G-code gives instructions to the printer through movement patterns, dictating where the nozzle moves and when it extrudes material. An example of G-code is shown in Figure 2.2.

```
G00 Z5.000000
G00 X33.655106 Y11.817060

G01 Z-1.000000 F100.0(Penetrate)
G01 X247.951560 Y11.817060 Z-1.000000 F400.000000
G01 X247.951560 Y30.935930 Z-1.000000
G01 X106.963450 Y30.935930 Z-1.000000
G03 X106.587404 Y32.243414 Z-1.000000 I-7.576860 J-1.471361
G03 X105.974610 Y33.458880 Z-1.000000 I-6.445333 J-2.487300
G03 X104.697090 Y35.083261 Z-1.000000 I-7.601246 J-4.663564
G03 X103.141830 Y36.435630 Z-1.000000 I-10.087550 J-10.030472
G03 X102.969400 Y38.107779 Z-1.000000 I-20.252028 J-1.243405
G03 X102.369430 Y39.685740 Z-1.000000 I-3.842423 J-0.557919
G03 X100.419761 Y41.664361 Z-1.000000 I-6.181245 J-4.140917
G02 X98.333794 Y43.482560 Z-1.000000 I7.045018 J10.188229
G02 X95.783544 Y47.017541 Z-1.000000 I9.647185 J9.647199
G02 X94.101654 Y51.024620 Z-1.000000 I28.957871 J14.510988
G03 X92.872672 Y54.561719 Z-1.000000 I-340.631289 J-116.371936
```

Figure 2.2: G-Code example

Each line of the code dictates the movement of the nozzle with GXX (where XX is the process number) describing the type of movement (Linear, Curved Clockwise, Curved Counterclockwise, etc.). The X, Y, and Z values describe the end position, and the F value gives the feeding rate (Dejan, 2020). G-Code is created from a Stereolithography or .stl file. .stl files are graphical representations of a 3D model, composed of vertices and directions that combine together to create triangles. With the positioning given, these triangles align to create a mesh of the model which can be interpreted visually. .stl files can be generated through a CAD software and are given a resolution when created. Lower resolution .stl files have fewer triangles, consequently making the shape rougher, whereas a higher resolution will be smoother (Plewa, 2018).

An .stl file can also be exported in either binary format or ASCII format, with the binary format being a smaller file size, while the ASCII format is readable by a human user.

.stl and G-code file types are very different and do not have direct conversions. Instead, the .stl file must be "sliced", which is a term describing the program-assisted conversion of an .stl file into the G-code used to print the object, readable by the printer. This process is called "slicing" as it breaks down the model into layers that can be printed.

The process of slicing requires several settings selected by the user in order to generate the correct G-Code. The material used for printing will affect the temperature required for the nozzle as well as the thickness of the extrusion. Additionally, the chosen printer will also affect these settings as nozzle and bed sizes vary between different printer models.

Along with material and printer variations, there are a vast array of settings within slicers that determine the G-Code. Most of these have little effect on the resulting print but there are several that will have a direct effect (All3DP, 2016).

- **Orientation** is a key part of the resulting strength of the print, determining the relative position of the resulting print within the bed, both positional and rotational. While not a slicer setting, orientation is often performed by the user within the slicer program, moving the model within the simulated printer bed. As mentioned previously, a part fabricated using FDM will be stronger along the XY-Plane than along the Z-Axis. Proper orientation can also prevent holes and overhangs within the part by changing the orientation in such a way that they can be supported. This also affects the first layer of the print as the orientation can determine how much of the model is contacting the printer bed. A good orientation provides the strength in the direction that requires it, minimizes the need for extra support for holes and overhangs, and provides a solid base on the printer bed.

- **Layer Height** - This setting determines the height of each layer in the object as the material is extruded from the nozzle. The layer height will affect both the print time and the smoothness of the part. When the layer height is smaller, the resulting print will have a higher resolution and smoother surfaces. With a larger layer height the individual layers may be visible but this also has the advantage of significantly shortening the print time. A common default setting is 0.2 mm.

- **Infill Density** - To conserve the amount of material used, the interior of a print is often formed using a mesh of patterned geometry. The density of this mesh is determined by the “infill density” which can range from 0 to 100 percent, or entirely hollow to entirely filled respectively. The infill will affect the strength of the resulting print as well as printing time and the material use. When the structural integrity of the part is not a concern, the infill density can be set to between 10 and 25 percent. However, for a fully functioning part, the infill would be between 75 and 100 percent. Several infill meshes of different densities are shown in Figure 2.3 (Siber, 2018).

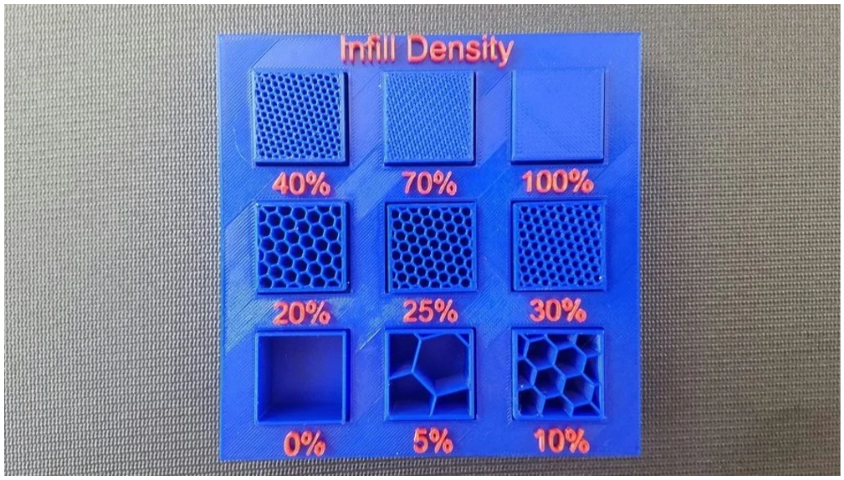


Figure 2.3: Infill density visual example (Siber, 2018)

- **Shell Thickness** - The shell is the solid exterior of the printed piece that is supported on the interior by the infill mesh. This setting adjusts the thickness of the solid material before the infill mesh begins. This greatly affects the strength of the object as the walls provide durability to the overall part. A common default setting is 0.8 mm.

- **Platform Adhesion** - When printing a model, the first layer must adhere strongly to the bed to prevent movement and warping of the model. Models with little surface area contacting the bed may require additional support for the first layer. There are three ways a slicer can improve adhesion to the bed in this way: a raft, a skirt, or a brim. These all offer starting support for the part, but have different advantages and disadvantages. Rafts are a lattice frame that the model is printed on top of rather than directly onto the bed. This helps prevent warping and is very useful for parts that have small footprints however it can leave roughness where the part breaks away from the raft. A skirt is an outline that surrounds the model but does not touch it. The primary function of this is to prepare the extruder and ensure the flow is consistent while also giving the user a chance to see if there are any issues with bed leveling. However, a skirt does not provide additional support. The last type is a brim. This is similar to the skirt but it touches the part. Brims can help prevent warping and give more support to thinner parts of the model that are on the bed. Brims require less filament than rafts and can be carefully snapped from the part after printing is complete (Simplify3D, 2020). Visual examples of these are shown in Figure 2.4.

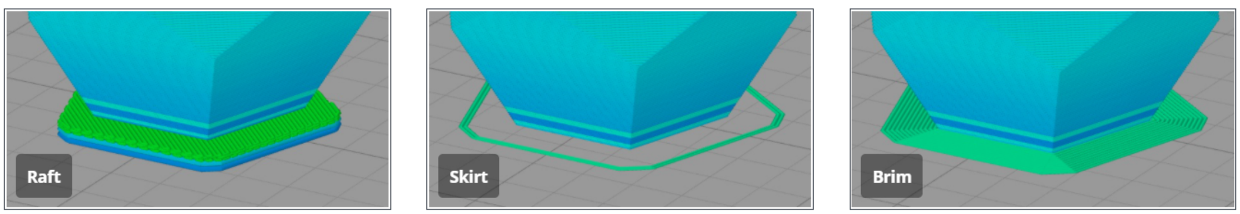


Figure 2.4: Platform adhesions (Simplify3D, 2020)

- **Supports** - If a model has overhangs then, as the model is printed, these overhangs will begin to sag under their own weight as they are built up layer by layer. Additionally steep angles away from the vertical may result in filament being printed without enough surface area from the previous layer to support the edge. To compensate for this, supports are added underneath these overhangs. Any extruding angle of more than 45 degrees from the vertical will not have enough support nor will bridges longer than about 5mm. Supports are most commonly constructed using an accordion shape, making them easier to remove and requiring less material (Chakravorty, 2020). Figure 2.5 displays a print with angles under 45 degrees that does not need supports as well as a piece with overhangs and accordion shaped supports. While supports are necessary for certain parts, they will increase the material cost as well as the duration for the printing process.

Figure 2.5: Supports Visualization (Chakravorty, 2020)

These settings will have a major effect on the resulting part, altering the resolution, material cost, and duration of the print. When all the settings and the orientation are determined by the user, the slicer can generate the G-Code using the .stl file and chosen settings, finally readying the model to be printed

2.2. Usability and UI Design

User interface design is a very important part of creating software. After all, program users interact directly with a user interface to accomplish their end goal in using the program. Because of this, user interfaces are designed with a target group of users in mind. According to Steve Krug, effective user interface design can be simplified to a single phrase: “Don’t make me think” (Krug 2014). This means that users should understand how to use a program simply by looking at it and experimenting with it, not by parsing through extensive documentation. In other words, a high-quality user interface should not require any explanation to the target audience.

There are several defined dimensions of interface usability which are used to determine the quality of user interface designs. According to Stone, the five most important criteria are effectiveness, efficiency, level of engagement, error tolerance, and learnability.

- **Effectiveness:** For a user interface to be considered effective, it must allow the user to complete their goals in a thorough and accurate manner. The effectiveness of an interface can be tested simply by seeing if a user can complete a task with it or not, making it the baseline requirement for a usable user interface.
- **Efficiency:** An efficient user interface is one that users can navigate quickly. The efficiency can be quantified by counting the number of actions, such as clicks or keystrokes, needed to complete a task, with fewer actions meaning higher efficiency.

- **Level of Engagement:** The level of engagement is the extent to which the style of the interface makes the program satisfying to use. This can be measured through user satisfaction surveys or qualitative interviews to gauge what users find appealing about the interface.

- **Error Tolerance:** Error tolerance helps users recover from errors or prevent them entirely. A user interface can feature popup windows or simple messages to fulfill this requirement.

- **Learnability:** A user interface should be easy to learn to encourage both novice and experienced users to keep using the program. This dimension can be judged by comparing how long it takes a new user to complete a set task versus how long it takes an experienced user to complete the same task (Stone 2005).

These are the baseline requirements that a user interface has to meet for it to be considered well designed.

2.3. UI Design Principles

The design of a user interface is made up of several elements, ranging from static ones, such as text and pictures, to ones that accept user input or interaction, such as text fields, buttons, and graphics. According to Williams (2015), there are four basic design principles that are necessary for a user interface design to be effective: contrast, repetition, alignment, and proximity.

- **Contrast:** If two items on a page do not serve a similar purpose, they should be visually distinct from each other. This helps the user navigate visually dense pages by distinguishing between associated actions.
- **Repetition:** Similarly, repetition in factors such as color, shape, and size, can be used to group similar actions.
- **Alignment:** Every element in a user interface should be connected with other elements by lining up the boundaries of interface objects.
- **Proximity:** This is another way of grouping elements with a related function together. If several elements are physically close to each other, the user will interpret this as a sign that they are related.

As previously mentioned, the usage of user interface elements is very important to consider while designing a program. These elements can be grouped into four major categories: input controls, navigational components, informational components, and containers. Input controls refer to anything that can accept user input, including buttons, text fields, checkboxes, dropdown lists, and radio buttons, to name a few. Navigational components allow a user to navigate within a program, including elements such as sliders, search bars, and pages. Informational components are mostly made up of static elements that tell the user how to use the program, consisting of message boxes, notifications, and tooltips. Lastly, containers separate page elements to reduce on-screen clutter and prevent the user from feeling overwhelmed (usability.gov 2014).

When designing an interface, universal accessibility must also be taken into consideration. Some users of a program might have a disability that makes it difficult for them to navigate an interface, such as colorblindness or poor eyesight. Therefore, interface design has to accommodate these users and meet all of the criteria needed for a usable interface at the same time. Many of the techniques that interface designers use to make a program more accessible also improve usability for all users. Some of these techniques include providing sufficient contrast between foreground and background elements, not using color alone to convey information to the user, and providing clear and consistent navigation options (WAI 2020).

2.4. Interface Evaluation Principles

Once an interface has been designed, it must be evaluated to determine its degree of useability. The traditional way that interfaces are evaluated is called heuristic evaluation. Instead of a random set of test users, a heuristic evaluation is conducted by usability experts who compare the interface in question against a set of well defined heuristics (usability.gov 2013). The most common set of heuristics used is the one defined by Nielsen and Molich (1990), consisting of 10 general principles for interaction design. Those principles are:

1. **Visibility of system status** - The system should always keep the user aware of what is happening in the program through feedback.
2. **Match between system and the real world** - The program should be in the native language of the target user and make use of terminology that would be familiar to them.

3. **User control and freedom** - supports the ability to undo and redo, allowing the user to make mistakes without serious penalty.

4. **Consistency and standards** - The system should remain consistent in the terminology it uses across the different interface elements.

5. **Error prevention** - eliminating possible error conditions so that the user can never make a mistake in the first place.

6. **Recognition rather than recall** - making as many options as possible visible throughout the program so that the user does not have to remember information from one part of the dialogue to another.

7. **Flexibility and efficiency of use** - The program allows experienced users to interact faster while still remaining accessible to the novice users.

8. **Aesthetic and minimalist design** - The design of the program should not contain irrelevant information.

9. **Help users recognize, diagnose, and recover from errors** - It should provide user friendly feedback to indicate the problem and possibly suggest a solution.

10. **Help and documentation** - Help is available to the user if they need it (Nielsen 1994).

Many of these heuristic principles are based on the five key requirements of a usable interface, which is why it is important to consider them throughout the entire development process.

2.5. Available Slicing Software

There are a number of downloadable slicing programs currently available, such as Slic3r, MatterControl, and Cura. Each program allows the user to input .stl files, edit slicer settings to control how the model will be printed, and generate the corresponding G-code (or, in the case of SolidWorks, print directly to the 3D printer). The MQP team chose to examine each of these publicly-available, advisor-recommended softwares as reference for how the UI of slicing softwares is usually designed, and evaluate each interface according to UI design principles to determine which aspects of each contribute towards usability and which aspects can be improved in a new automated slicing program.

Slic3r's main screen consists of a 3D representation of the print bed on the left side with a distinctive blue background and a menu of options on the right side, as shown in Figure 2.6.

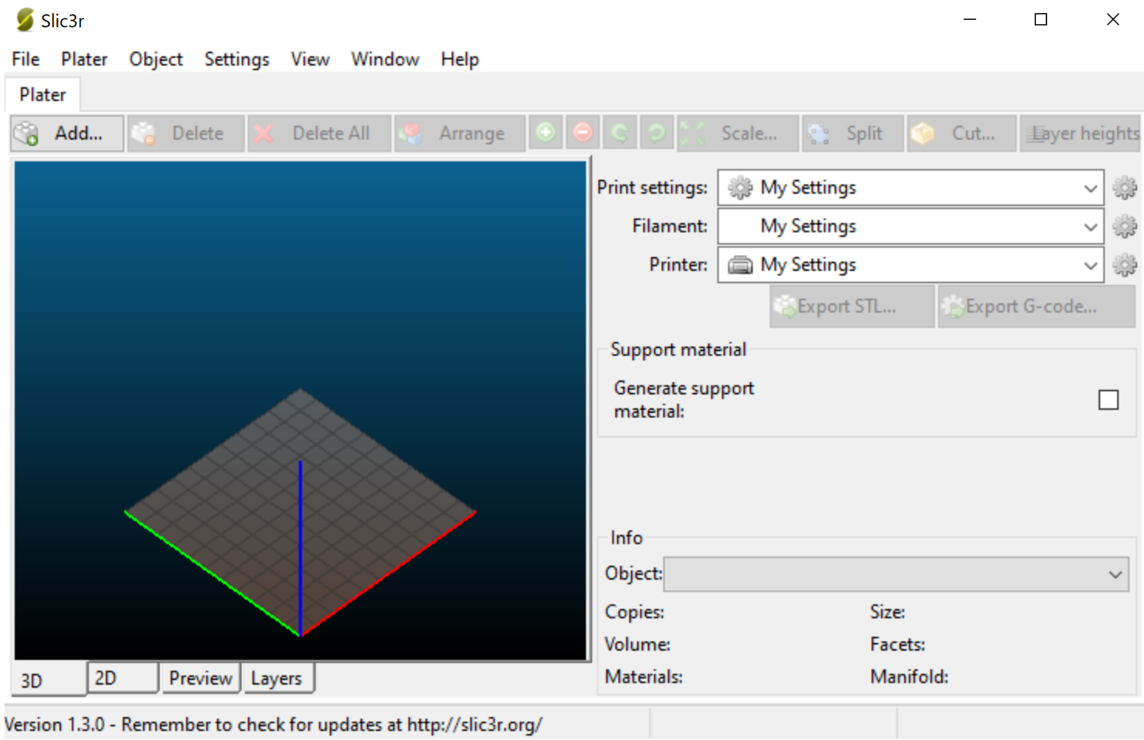


Figure 2.6: Slic3r main screen

Upon launching the application, all of the buttons except for the *Add…* button are grayed out, signifying the user that a model must be added using the button before any editing options are available. Using the *Add…* button prompts the user to select an .stl file, and once one is loaded, more buttons become available, as shown in Figure 2.7.

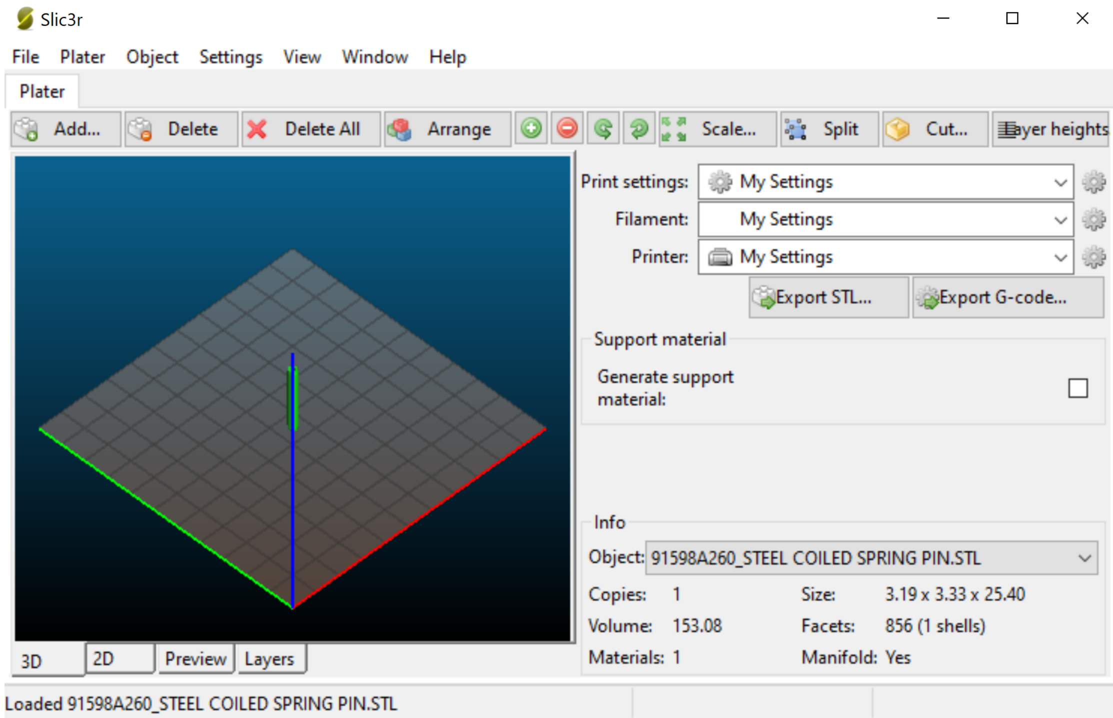


Figure 2.7: Slic3r main screen with an .stl file loaded

Preventing the user from accessing model editing and printer settings before a model is loaded in prevents the user from being overwhelmed with too much information at once. Menu buttons all have symbolic icons associated with them, making each option recognizable and memorable.

Options to physically edit the appearance of the printed model (such as adding new models, rotating the model's orientation, and scaling the model) are all contained within the easily-accessible top bar. To prevent impossible printing jobs, Slic3r prevents the user from moving a model off of the print bed. However, it does not prevent users from scaling a model to be too large for the bed.

In addition, options to edit the slicer settings, such as infill density and support type, are not easily visible. Users must navigate to the Settings menu and pick from

Print, Filament, or Printer Settings to access the respective settings tab (shown in Figure 2.8), where each option must be manually selected and edited. The high number of options in each settings tab might be overwhelming to new users.

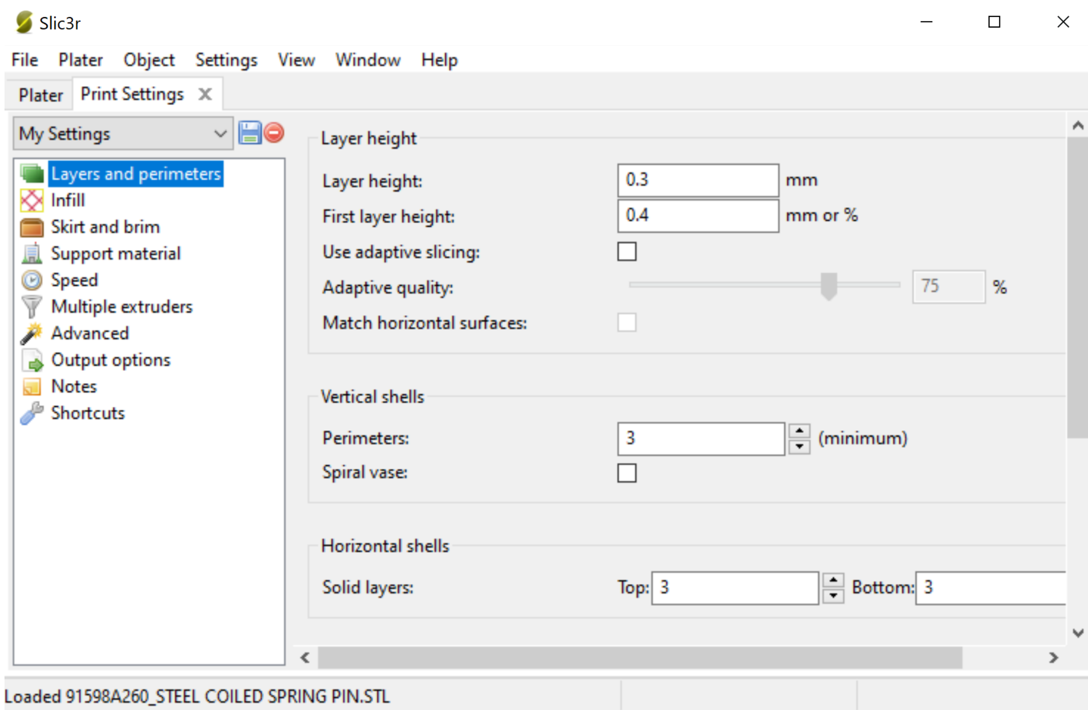


Figure 2.8: Slic3r's print settings window

Slic3r does contain a Configuration Wizard, located in the Help menu and shown in Figure 2.9, intended to assist new users with easily editing slicer settings. The wizard simply prompts the user for various details about their desired configuration, and those settings will be automatically changed without the user having to navigate the settings menu. However, the wizard is limited in its utility. It only prompts the user on a scant few settings, and does not allow the user to change settings like layer height or shell thickness from inside the wizard itself.

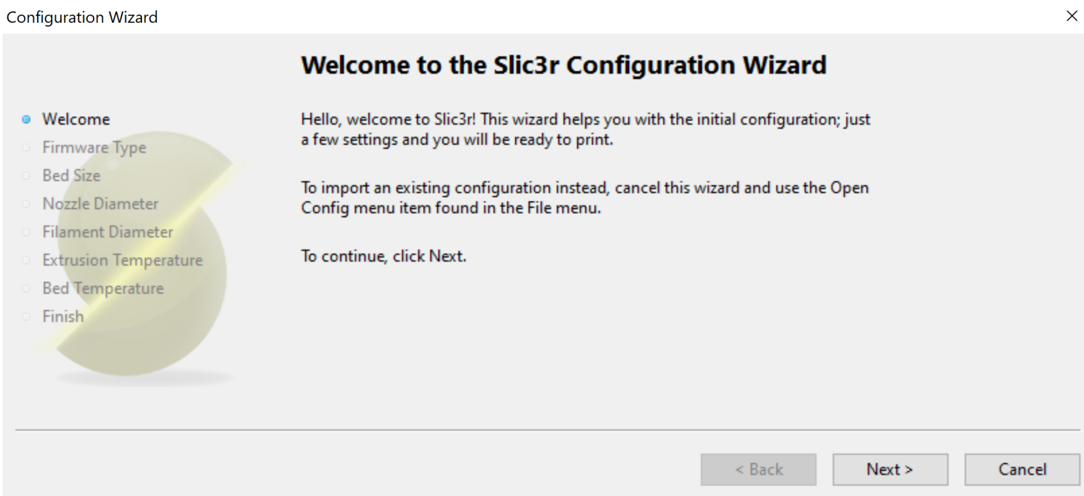


Figure 2.9: Slic3r Configuration Wizard

MatterControl's main screen, shown in Figure 2.10, is almost entirely composed of a 3D grid, which the user can intuitively move using the scroll wheel. Like Slic3r, it has a top bar with orientation editing options that are grayed out until the user loads a CAD file using the Open File button.

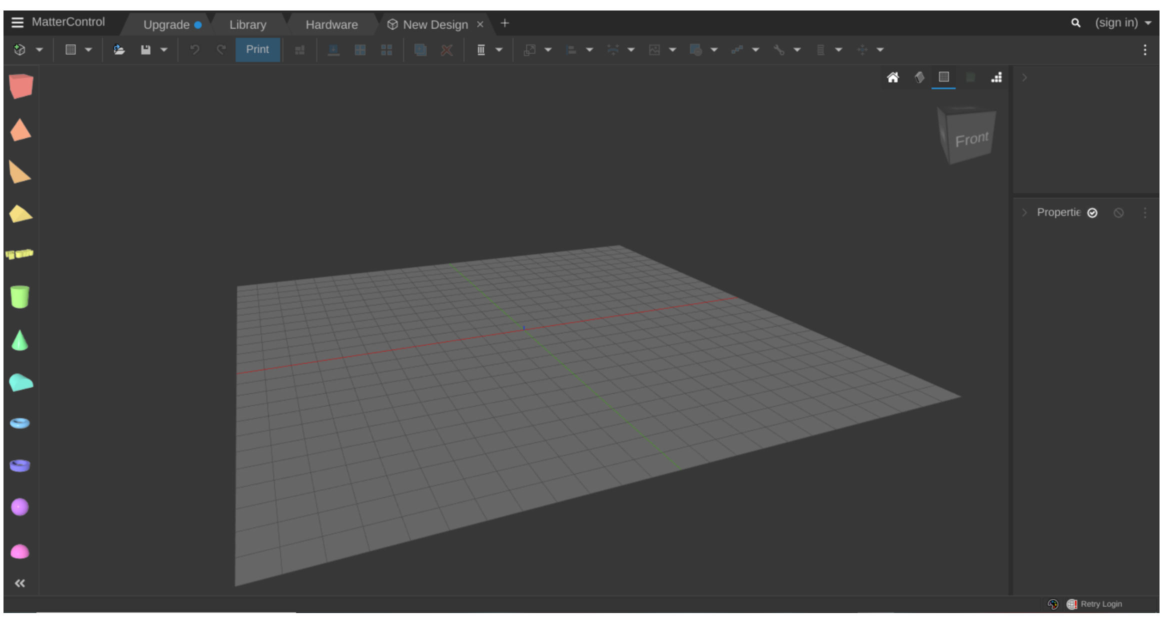


Figure 2.10: MatterControl main screen

Like Slic3r, MatterControl does not prevent the user from creating a model too large for the print bed. However, it also does not prevent the user from moving a model off of the print bed, or even warn the user when this has been done.

In order to access slicer settings, the user must load in a model and press the notably contrasted “Print” button, upon which they are given a large selection of 3D printers. Choosing one leads to a second screen with a simulated print bed of the chosen printer. On this screen, a “Slicer Settings” menu can be accessed on the right side of the screen, in which a variety of options are presented in a single streamlined menu, shown in Figure 2.11. Although there are still a high number of settings that can be configured, the low number of tabs and ease to switch between them makes it less overwhelming to use than Slic3r.

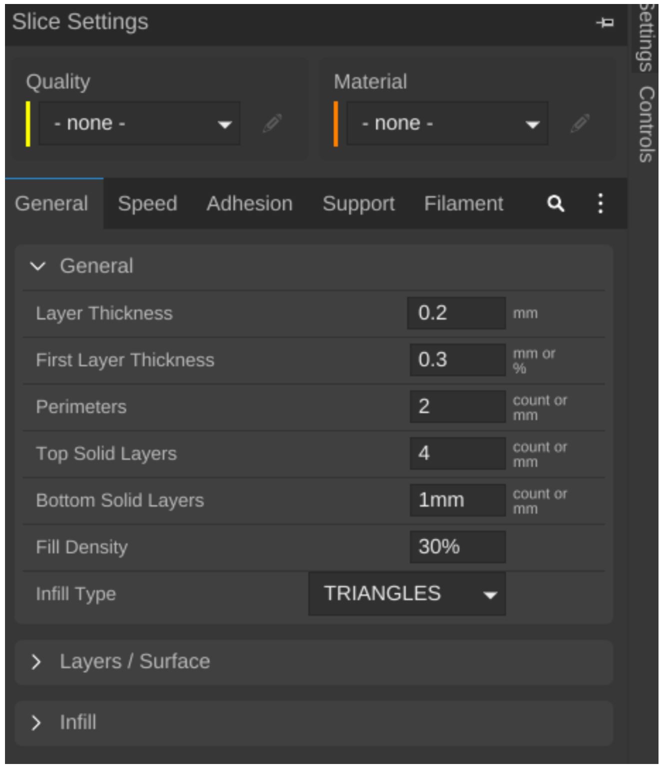


Figure 2.11: MatterControl slicer settings

However, the settings menu itself is still difficult to access, requiring the user to click "Print" and choose a printer before the option is even available. In addition, the option to generate G-code is further hidden, requiring the user to press a "slice" button on the second screen and then selecting the "Export" button from the "Print" drop-down menu. This stands in contrast to Slic3r, whose option to generate G-code is easily visible from the main screen as soon as an .stl file is loaded in.

Cura contains an extremely simple interface, consisting almost entirely of a simulated 3D printer bed shown in Figure 2.12. Like Slic3r and MatterControl, all options are grayed out until the user imports an .stl file. Options to edit the orientation of the model, including scaling and rotation, are easily visible on a left panel, while print settings are listed on the right of the top bar. Cura also will automatically return floating parts to the printer bed, and highlights in gray areas which fall outside the range of the bed. This provides robust error prevention that Slic3r and MatterControl lack.

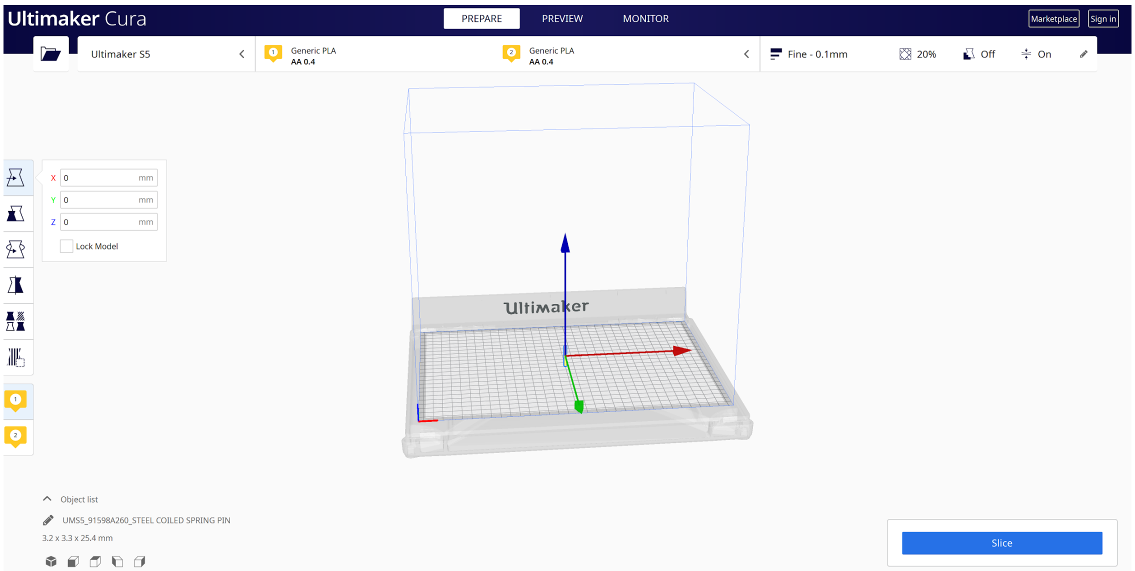


Figure 2.12: Cura main screen

Clicking on the right side of the bar — indicated clickable through the ubiquitous pencil icon — opens the print settings menu, which displays a small number of commonly-changed settings, such as infill percent and supports. This helps make the menu more accessible for novice users- however, if a user wants to edit more than just those small number of settings, clicking on a “Custom” button shows a much larger list of settings that a user can easily change from the same menu, shown in Figure 2.13. Although there are a large number of tabs, each of these tabs seldom contains more than 3 sub-options, making the menu highly categorized and easy to navigate. All of the options being instantly accessible from the main screen contributes greatly to the program’s efficiency.

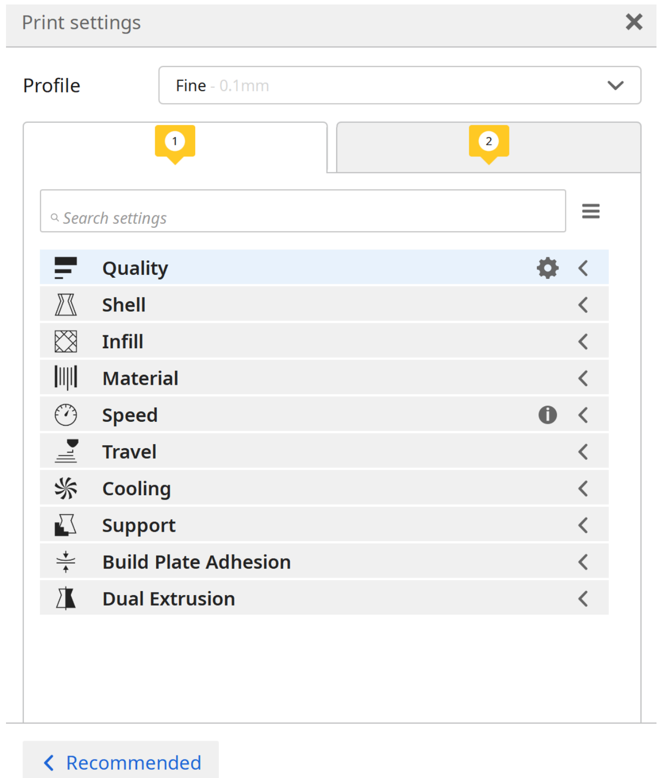


Figure 2.13: Cura print settings

Lastly, the “Slice” button, located in the bottom left and emphasized by its contrasting blue color, will generate the G-code for the model. After pressing the “Slice” button, Cura shows a progress bar in its place that, upon completion, becomes a “Save to File” button that allows the user to save the G-code to any place on their computer. This implementation of G-code generation is easy to locate due to its position on the bottom left, and easy to use due to its automated process.

2.6. Testing and Evaluation

Testing is an important part of software engineering to ensure the software runs as expected. There are two parts of testing, UI testing and software testing. UI testing is the act of making sure that the user interface is easy to understand and use. Software testing is the act of making sure all of the software that was written acts as intended.

According to Krug, there are six parts in a proper UI test: the welcome, the questions, the homepage tour, the task, probing, and wrapping up (Krug, 2014).

The welcome section involves explaining how the test works to the participant. The questions section of the test is used to learn more about the participant and any aspects of their character that might influence how a tester could interpret the data collected. For example, it is usually important to ask the participant how experienced they consider themselves with computers.

The homepage tour step has you open up the homepage (or default screen of the software) for the participant and gauge their initial reaction. The homepage is the most important part of a UI system, so understanding how users react to it is vital.

The tasks section is the longest part of a test and will be where testers receive most of their data. This section consists of asking the participant to complete a variety of tasks. It is important to ask the participant to think out loud, as this can inform a substantial part of why a user is struggling. It is important to ensure that the tester gives as little information to the participant as possible; otherwise, the data could become biased.

The penultimate step is probing. The tester asks the participant questions about the product, as well as any specific events that happened during the tasks section that were unexpected.

Wrapping up is when the tester thanks the participant for their help and shows them the door. During all of these steps, it is important to have as many people watching the test as possible, as the data collected is going to be very subjective.

When addressing problems that come up during UI testing, Krug recommends a four-step process. The first step is to meet with everyone observing the testing to make a collective list of all the serious problems they observed. This can be done by writing on a whiteboard or simply going around the room and asking everyone. After that is done, the two next steps are to choose the ten most serious problems and rate the severity of those problems from 1 to 10, with 10 being the most severe. Finally, Krug recommends creating an ordered list of what problems the developers wish to fix and what resources those fixes will require. Krug emphasizes that often all developers need to do is make small tweaks rather than complete overhauls (Krug, 2014).

Software testing is another process that is important to making a quality product. According to James A. Whittaker in his publication “What is software testing? And why is it so hard?” there are several problems that users could run into if developers do not test sufficiently (Whittaker, 2000). These could be caused by the user running untested code, the user performing actions in an order the testers did not test, the user entering inputs that were not tested, or the user running the software on an environment that was never tested.

Whittaker recommends breaking up software testing into four steps: modeling the software's environment, selecting test scenarios, running and evaluating test scenarios, and measuring testing progress. Modeling the software's environment needs to simulate interactions between software and its environment. In order to do this, testers need to identify all interfaces that might interact with the software. These include human interfaces, software interfaces (API calls), file system interfaces, and communication interfaces (device drivers). Testers should also identify what should happen during edge cases, such as deleting files currently being used by the software (Whittaker, 2000).

Selecting test scenarios can be a difficult process because there are infinite numbers of possible test scenarios that could be created, and each one costs the testers time and money. In order to simplify the process, testers should be selecting sets of tests that run every line of code, run both options of branching statements, cause all data to be created and used, and expose each of the programmed error catches. Failure to complete even one of these criteria could lead to software that runs with unintended consequences (Whittaker, 2000).

Running and evaluating tests is another important step, but can also be the most complex. This is because knowing what a test should return might be too complicated for a person to figure out on their own. If this is the case, the problem can be solved by making a self-testing program. An example of this would be to program something in two different ways and see if they create the same result. After the developer fixes all the failed tests, it is important to run all tests again. This is because, while trying to fix one thing, developers will often unintentionally break other things, or simply fail to fix what they intended (Whittaker, 2000).

Evaluating the progress you have made in software testing is important for keeping everyone in your team informed. The subjective nature of such an evaluation can make it difficult to measure progress, but Whittaker proposes the following questions to keep in mind when trying to judge how close a developer is to completing testing:

- “Have I tested for common programming errors?”
- “Have I exercised all of the source code?”
- “Have I forced all the internal data to be initialized and used?”
- “Have I found all seeded errors?” (seeded errors are deliberate errors put into the code, this is done because the act of exploiting them could lead to finding unintended errors)
- “Have I thought through the ways in which the software can fail and selected tests that show it doesn’t?”
- “Have I applied all the inputs?”
- “Have I completely explored the state space of the software?” (The state space is the name given to a combination of all possible configurations of a system (Math Insight, 2020))
- “Have I run all the scenarios that I expect a user to execute?

After asking themselves these questions, a tester should be better equipped to make a progress report (Whittaker, 2000).

3. Motivations, Requirements, and Goals

The primary goal of this project is to create a program which will simplify the 3D printing process by automating the printer settings, orientation selection, and slicing processes. Its creation was motivated by the lack of visual clarity and overabundance of customizable options present in other slicer applications, as described in section 2.5. The requirements for this program are:

- Input .stl files and correctly display and interpret their contents
- Detect flat surfaces on 3D objects
- Automatically determine printer settings for a print based on the given .stl file
- Slice an object for printing in automatically-generated orientations
- Allow the user to print G-Code for each generated orientation
- Show the users pros and cons about whichever orientation they select (including but not limited to estimated printing time and material cost)
- Correctly generate the proper G-Code for each part

Originally, we planned that a requirement for this program would be to connect to 3D printers and be able to print G-Code directly from the application window. As described in Chapter 4, we eventually decided this was outside the scope of the project.

Our goals, differing from our requirements, are lower-priority items that our group would implement if we have the required time necessary, but are not essential to GaPA's completion. If possible to meet, our goals for this program are:

- Have it be able to interpret SolidWorks CAD files
- Have experienced users be able to customize individual print settings specifically,

regardless of the program’s recommendations

- Ensure that prints using the G-Code succeed >99% of the time
- Include a help section for new users
- Generate all orientation options in under 2 minutes for .stl files with under 300 polygons

4. Methodology

This chapter details GAPA's research and creation process, told in chronological order by terms and supplemented with the reasoning for why each step was completed at that time.

4.1. A Term Methodology

At the project's inception, several members of the group were unfamiliar with basic 3D printing techniques and terminology. Also unfamiliar were common slicing softwares and their user task flow. As a result, the first two weeks of project work were centered around research into the process of 3D printing, divided among each team member and then concluded with presentations on findings.

In the first week, our group divided research on 3D printing and terminology into categories, including how to 3D print from start to finish, CAD files, G-Code, and slicing. We then presented our findings in presentation format to the rest of the group. This was done in order to establish a group baseline of knowledge on which we could build our understanding of the project, as well as provide background for further research. We also drafted a problem statement to guide us as we worked, summarizing each of our ideas on what we thought the purpose of our project was. After one round of revision from our advisors, our statement was as follows:

> Our project is a program to automate the G-Code generation process for an STL file. The program will feature an easy-to-use streamlined process to enter details about the printer, such as nozzle size, bed size, and temperature. The program will then read a user-inputted STL file and automatically generate multiple orientation options for printing that file with the given settings. The

> user can select an option, and the program will automatically slice that orientation and allow them to easily print G-Code for it. The program will also show users pertinent detail about whichever orientation they select, including but not limited to estimated printing time and material cost.

This statement was later broken down into bullet points summarizing requirements and goals for the application, which can be found in Section 3: Motivations, Requirements, and Goals.

Afterwards, through an advisor-assigned activity, each group member individually researched one of the existing slicing softwares of Slic3r, Cura, SolidWorks, and MatterControl. Each group member was asked to record various observations about their assigned slicing software, including where buttons to perform basic functions (such as importing STL files) were placed on the screen, how the slicer responded when the user attempted to move the model of an STL file to a position off the printer bed, and how many other screens were accessible from the main view. These observations were then compiled into a spreadsheet, the first eight rows of which are pictured in Figure 4.1.

Questions/Application	Slic3r	Cura	SolidWorks	MatterControl
What OS did you use?	Windows	Windows	Windows	Windows
How did you download the software?	The slic3r webpage has an easily visible "Downloads" tab and a large "Windows" icon.	Googled cura download and clicked the first page	Already had Solidworks downloaded through WPI, the slicer functionality is built in	The MatterControl webpage had a Download button on their homepage
How did you install the software?	The download tab was a .rar file. I simply extracted the contents to a new folder.	The thing I downloaded was an exe installer		There was a setup.exe file that I ran to install it. It allowed me to choose the install location on my computer.
How do you import an STL file into the software?	There is a large green "add" button in the top left corner of the screen. It is the only thing highlighted upon starting the program. Clicking it takes you to a file explorer window, from which you can add a file.	At the top left there is a button of a folder which takes you to a file explorer. There you can inport an stl file	The model is opened through "File" "Open" and will open the file in the main display. Clicking "File" "Print3D" will open the slicing software with the model already opened.	On the top bar, there is a small "Open File" button. Clicking this button brings you to a File Explorer window, where you can select which file you want to open.
How does the user see the imported part?	A large display of the part occupies the left of the main screen. A tab below it indicates settings for 3D view (where the user can click and drag the part around the bed), 2D view (which shows a top-down view of the object on the bed), preview (where the user can click and drag to rotate the bed and see the part from all angles), and layer view (where the user can drag a sliding scale that shows a corss-section of the part at the layer on the scale).	There is a large display of a printer bed in the middle of the UI. The user can rotate and pan arround them as they wish.	The main part of the display shows the model in a 3D space. Being the Solidworks workspace, the controls for manipulation are the same. The user can click and drag to rotate or middle click to pan. There are also several view options that allow for different viewing angles.	There is a large display pane in the center of the screen that shows a 3D representaion of your model.
What features are accessible on the main screen?	Clicking on a part in the display window highlights it and turns it green (although if there's only one part, it's highlighted by default). A top bar above the display that stretches to the right contains buttons that edit the model in the display, including buttons to add more parts, delete parts, duplicate, rotate, scale, cut, and edit the layer height of a highlighted part, and "arrange" parts by returning them to their default position. On the right, the user can also select print settings, filament, and printer settings from any presets they've previously saved. Right below these are the options to export a new STL file or print the gcode. Lastly, there is a final checkbox to generate support material.	There are several options for how you can manipulate the STL file. There are options for panning, rotating, scaling, mirroring, and mesh settings. At the top you can select your printer type and all the things that come along with it. In the bottom right you can select slice to generate gcode. Once you click slice, there is an option for save as file. There is also a drop down menu for printer settings. These settings include Quality, Shell, Infill, Material, Speed, Travel, Cooling, Support, Build Plate Adhesion, and Dual Extrusion	Given that Solidworks is not primarily used as a slicer, this will be considered with the Print3D function already open. With this open, the user still has the ability to manipulate the object in space but lose the ability to edit the part (as the Print3D function is currently running). The side bar includes the printer settings which includes: Face Selection for Base, Printer Bed Size, Model Orientation (Translation and Rotation on the bed plane), Scale, Job Quality, Infill Percentage, Support and Raft Option, Save File, Supported Faces, Layer Height, Thickness/Gap Analysis.	The user can use left mouse button to select the part, right click and drag to rotate the camera, middle mouse scroll to zoom, and middle mouse click and drag to pan the camera. There are settings in the top right corner of the display pane to change the way the model is represented (shaded, outlines, polygons, materials, and overhang). You can also toggle the print bed on and off from here. On the top bar, there are buttons to: Group/Ungroup parts, Duplicate, Remove, Toggle Wipe Tower, Generate Supports, Arrange All Parts, Lay Flat, Align Multiple Parts, Combine, Subtract, Intersect, Subtract and Replace, and Fit to Bounds. There is also a blue Print button that allows the user to select a printer and then slice the model.
How are these features organized?	Buttons towards left side of the bar (add, delete, delete all, arrange) deal with multiple parts, buttons towards the middle (duplicate, rotate) edit an individual part, and buttons on the right end (scale, split, cut, layer height) edit a single part in more depth, bringing up their own menus. All options that do not edit the model in the display (printer settings, generating support material checkbox) are on the right panel.	These buttons are orginized fairly well. There is a toolbar on the left for manipulating the model. At the top there are overall settings like select model and printer. To the right of those printers are the printer settings drop down. Each printer setting catagory also has its own drop down.	All of these features are on the sidebar menu. There are two tabs with all the options following Save File in the second tab (which is notably not all too noticeable). At the top of the workspace are the options for manipulating the object.	These features are located on a bar directly above the 3D model viewer pane.

Figure 4.1: Slicer Analysis Spreadsheet

This analysis of pre-existing slicers early on into development of the application was done to both allow group members to see the functionalities of available slicing programs, as well as the UI design of those programs and how those functionalities were presented to the user. It also provided background on what slicers would do to handle invalid model orientations.

Up until this point, each group member had been doing their own independent research into best practices in UI design and evaluation. To centralize and streamline this research, an outline for a Literature Review chapter was subsequently developed. Research was then refocused around this outline, and all references were compiled into a single reference document. For the rest of the week, the group's work was focused on analysis of print settings such as infill density, support type, and layer height, to determine which settings were most essential for our program to integrate.

For further information about FDM, interviews were performed with professionals in the field. The first interview was performed with Professor Erica Stults at the WPI Rapid Prototyping Lab and the second with Mitra Anand at the WPI Makerspace. These interviews gathered basic information about 3D Printing and pointed our team toward resources and similar programs such as 3DPrinterOS.

While the outline for the Literature Review chapter was being filled in with more specific data from our research, the group decided to switch their main focus from research to design. At this point, we considered it appropriate to form a basic foundation for the program's functionality and design. Although the majority of the first term of work was focused on research and synthesis, before beginning the two-week process of writing the Literature Review itself, we wanted to know what kind of program we would be developing in the following term.

The group took the following week to make a number of important design decisions about how the program would function. Among these decisions were an initial draft of the User Flow diagram (see Figure 5.1) as well as a decision matrix (shown in Figure 4.2) to determine whether the program should be a downloadable application for Windows or a web app, settling on the former. Full rationale for this decision can be found in Section 5.2.

Issues	Points	WebBased	WebBased Totals	Windows	Windows Totals
Ease of access	8	0.9	7.2	0.4	3.2
Keep a server running	3	0	0	0.8	2.4
Ease of update	5	1	5	0.2	1
Systems it can run on	3	1	3	0.33	0.99
Development experience	5	0.75	3.75	0.6	3
Speed	9	0.5	4.5	0.9	8.1
Need constant connection	1	0	0	0.4	0.4
Testing time	5	0.8	4	0.45	2.25
Ease of other applications to hook onto	10	0.9	9	0.85	8.5
How long into the future it lasts	10	0.1	1	0.9	9
Totals	59		37.45		38.84

Figure 4.2: Platform Decision Matrix

During this early design process, the team also made edits to the User and Task flow analysis to properly reflect the design elements of the program we had solidified. Also factoring in to our decisions was the recent discovery of the 3D Printer OS paid software at the Makerspace, whose purposes and streamlined interface seemed to serve a similar purpose as our application. Because of this, we decided an important part of our application would be that it further simplified the slicing process, and would also remain free to use.

Nearing the end of the term, the group started work on writing the Literature Review chapter itself in its entirety. The outline, now fully fleshed-out, was separated into four distinct sections. Each group member was assigned to work on one section, and did more directed research into their selected subject as necessary to form a robust analysis. Just as before, all sources were documented in the references page. Near the end of the last week of the term, the group finished this chapter of the report, and put together a schedule for B Term based on their progress. This plan was summarized into a Gantt chart, shown in Figure 4.3.

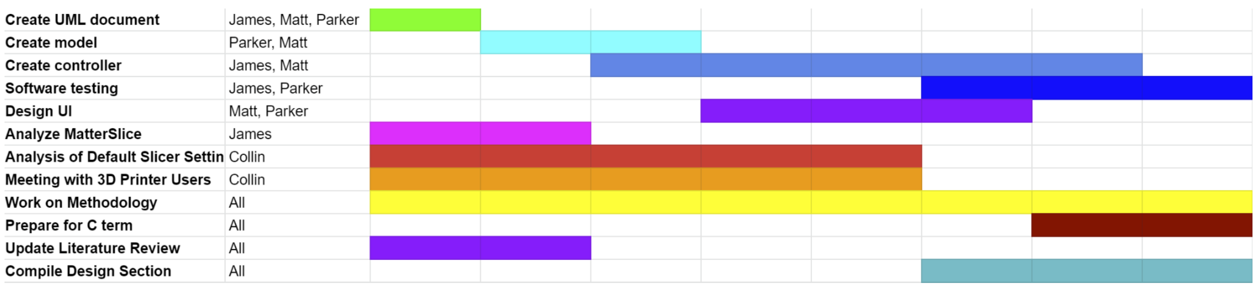


Figure 4.3: Gantt Chart for B Term work schedule

Creating such a chart was, in addition to being an advisor recommendation, a measure to keep our development more focused. Each of the listed tasks was a part of the design process, which was now fully informed by the research in our Literature Review.

4.2. B Term Methodology

During B term, our team planned on finalizing the program's design through analysis of our data from the Literature Review and continuous communication with our advisors. Thus, we determined that our main deliverable for this term would be the Design section, explaining the process and rationale behind each design decision. Originally, we planned to create UML diagrams, model, and controller classes in order, and then designing the UI over the course of three weeks. We eventually decided to move up the UI design process and backend development in order to facilitate iterative design through feedback from our advisors.

After the Literature Review was submitted and edited based on advisor feedback, work started on the implementation of specific third-party libraries, such as MatterSlice, Octoprint, and the SolidWorks API. While these tools had already been looked into during the previous term, work now had to be done on how they would factor into our program, and what methods we would use to implement them.

Also done during the first week of B term were UML diagrams that laid the foundation of the program's code. The group created an initial class diagram (shown in Figure 4.4) to determine what classes needed to be made and what kinds of objects and methods those classes held, and how they would interact with each other. We also created a system block diagram (shown in Figure 4.5) to demonstrate what systems (including the user, OS, and third-party libraries) classes would interact with each other. The group would then use these diagrams to develop model files in C# during the second week. These model files were classes which featured no functionality, but were created with all of the elements and connectivity specified in the class diagram.

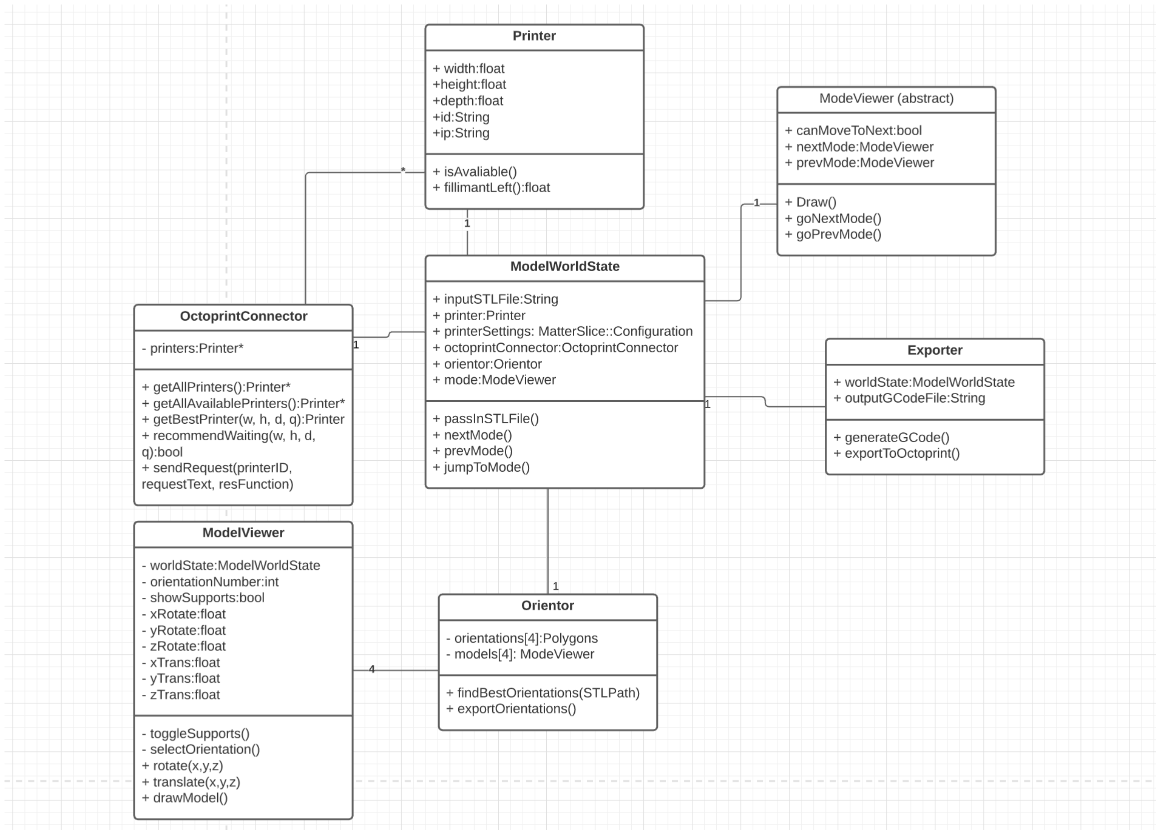


Figure 4.4: UML class diagram

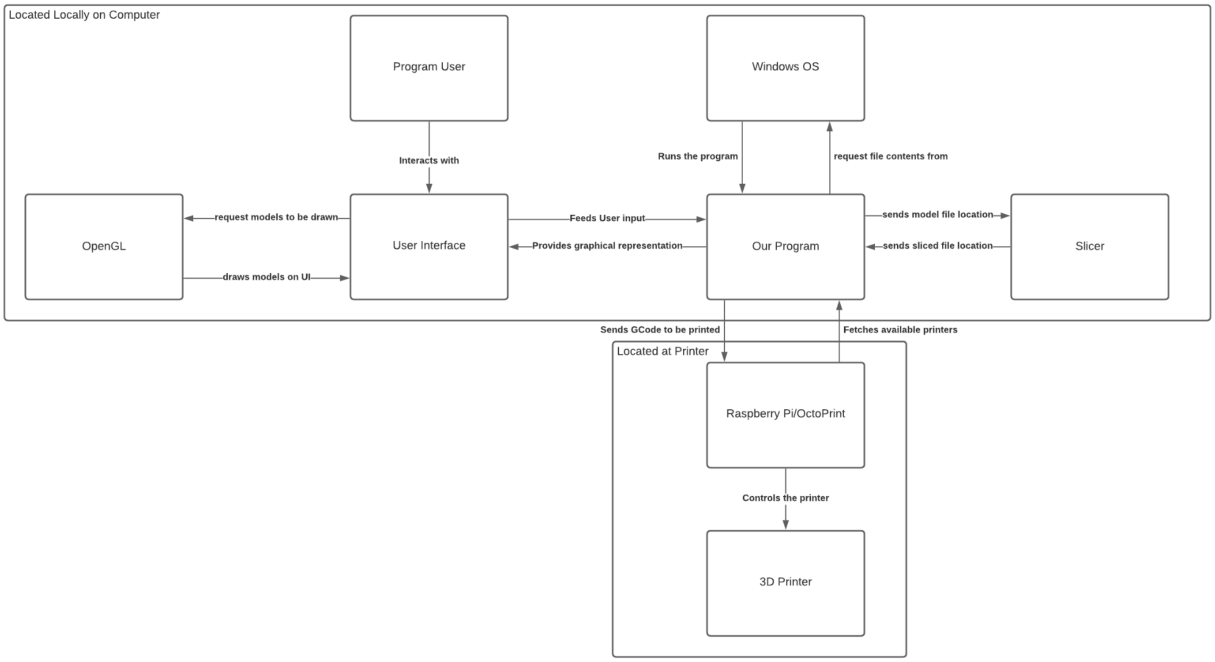


Figure 4.5: System Block Diagram

Early interviews were performed with potential users. These interviews were intended to ascertain some of the user expectations for what our program can do as well as ensure that there is a niche our program could fill.These interviews helped influence some of the functionality of the application.

Having finished the first set of model classes, the group also worked on aspects of the design which could be developed in parallel. In particular, the group worked on the backend code for generating G-Code. Initially, we planned to use the SolidWorks API to generate G-Code, but that proved to not be possible, so we pivoted to MatterSlice. We also began working on the orientation selection algorithm, completing a backend function that found the top 4 orientations of an STL file with the most surface area on the printer bed— a primitive way to rank orientations, but still a start.

In response to advisor requests, we also began drafting low-fidelity drafts of the UI (see Figures 5.5-5.9), starting earlier than originally planned in the Gantt chart. In developing these prototypes, we used our prior research on printer settings that affected part performance, as well as our reviews of other slicing softwares and what mechanisms they used to set their own settings (sliders, drop-down menus, etc). All of this design work was done at this point in order to facilitate getting iterative feedback from advisors, which was not possible for the nonfunctional model classes.

Throughout this period of initial weeks, the group was also meeting with 3D printer users to determine what kind of parts they made and which kinds of settings they believed to be valuable. These user profiles factored into the UI design process as we chose which options would be available to users.

The next few weeks were dedicated to iterative design and testing. The G-Code generation function was finished, and several rounds of low-fidelity UI prototypes went through advisor feedback and discussion. The effects of infill, layer height, and orientation on the compressive and tensile strength of a printed part were also tested, and a cause and effect table mapping the settings of a printed part to part's eventual physical properties was created. This table would serve as a basis for the ranking algorithm and the questionnaire, as important physical properties of a print would be asked about and the corresponding settings would be prioritized by the algorithm. A refined version of this chart can be seen in Table 5.1 of the Design chapter.

After several rounds of UI drafting and evaluation resulting in advisor and unanimous group approval, work began on recreating the most recent low-fidelity prototypes in C#. Included in the initial round of design creation was the functionality to resize the window, basic navigation to change from one mode to another, and a functional 3D model viewer panel. Backend development also progressed, and the orientation selection algorithm was edited to incorporate more settings than just the surface area with the printer bed.

At this point, the first iteration of the “Determine Settings” screen’s user survey began to take form, starting as a document with a list of if/then questions whose answers would determine the purpose of the printed object, and thus which of its settings would be a priority according to prior research. The full list can be found as Appendix A. Some examples include “If tensile strength is desired, then the infill density should be between 60 and 70%”, “If flexural strength is desired, then the model should be on-edge with the direction of the force acting on either the x or y axis”, and “If the material is ABS, then the nozzle temperature should be 235 degrees Celsius”.

As the end of B term approached, the team once more created an outline for the deliverable chapter of the report. This time, each group member was assigned a part of the report from the outset, as each team member had a distinctly different role in contributing to the overall design of the program. Over the short duration of a single week, this Design chapter was completed, edited, and submitted for advisor feedback, and a plan for C term was created. Just as for B term, this plan was finalized in the form of a Gantt chart, shown in Figure 4.7.

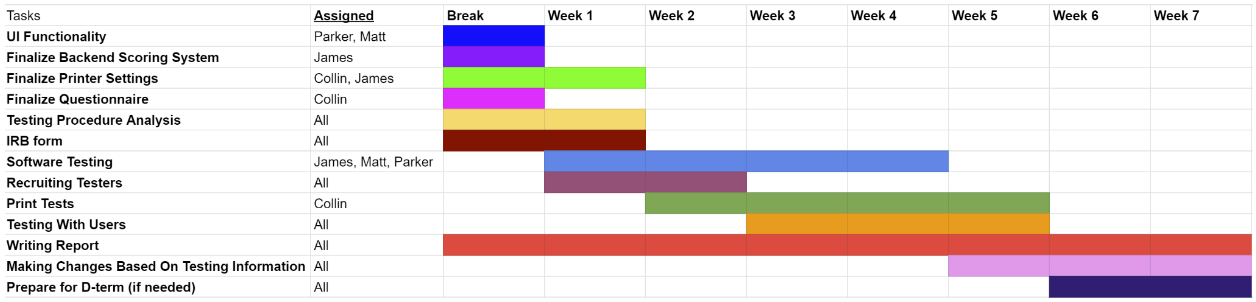


Figure 4.6: Gantt Chart for C Term work schedule

The intended schedule for C term was to spend the time over the extended winter break adding functionality to the UI, including navigation to other screens, working buttons, and finalizing the “Determine Settings” screen questionnaire, as well as submitting the IRB form required for testing. At the end of C term, we planned to be almost entirely done with the design of the program itself, and as such, our deliverables would be a completed Design document and a complete Methodology chapter.

4.3. C Term Methodology

Editing the Design chapter based on advisor feedback started early and was around half-complete by the time C term started. The creation process for IRB-form-related materials took longer than expected, however, and although the group completed a feedback survey for participants and outlined the testing procedure over the break, requirements for submission were not fulfilled until later. Various backend bug fixes also had to be made as well, involving the progress saving system and the merging with frontend functionality.

It was at this point that weekly communication with advisors began over UI specifics. Although the program was built on several iterations of low-fidelity prototypes, several aspects of the interface still needed to be tweaked for user accessibility. Such edits included changing the colors used, font sizes, and button placements to be consistent across modes, and, as such, a style guide was created to ensure a uniform application of visual design choices.

In addition, before the group submitted the IRB form for testing with other students, we ran a number of tests with ourselves. The results concluded that more work on the scoring algorithm for orientations was needed, and as such, the group met with Makerspace technician Mitra Anand to discuss the scoring system. This discussion and subsequent testing led to a discovery that the algorithm also had no protection in place for printing excessively tall pieces, and so it continued to be tweaked.

The IRB form was completed and submitted three weeks into the term. While waiting for it to be processed and approved, the group continued with adjustment to the orientation selection algorithm as well as iteratively changing the UI based on presentations with advisors and subsequent feedback. Such change included the addition and functionality of a printer settings window, a help popup, and a loading screen for parts that took a long time to generate orientations of. Since the IRB form remained unapproved for several weeks, most of the latter half of the term was spent implementing and presenting these changes rather than testing with users. Programming these changes also proved to be a difficult task for the team, and major changes coupled with several small tweaks to button shape and placement would take team members upwards of a week at a time. Bugs were also commonly discovered at this point, but none took more than a week of work to fix.

Near the end of C term, the IRB form was accepted, and, confident that the program's UI had undergone enough weeks of revision to be valuable to test with real users, individual team members began running the first tests using the procedures described in the form. These tests were scheduled and conducted whenever possible with personal acquaintances first, as the team decided the UI needed more refinement before a campus-wide request for testers was distributed. The team received feedback from the survey form at the end of the tests, and now could update the UI (including the "Determine Settings" screen's questionnaire) based on both that feedback and recommendation from the advisors.

As the end of the term approached, the team briefly shifted their attention to the report, and quickly finished edits to the Design chapter and completed the Methodology chapter. In the last days of C term, we also finished a plan for D term work, shown once again as a Gantt chart in Figure 4.8.

Figure 4.7: Gantt Chart for D term work schedule

The group planned for D term to primarily focus on testing, testing analysis, and completion of the required submission materials such as the videos, poster, and presentation. The report was also to be slowly revised and added to as the term progressed.

4.4. D Term Methodology

In order to properly prepare for testing, which was scheduled to start just over 2 weeks into D term on April 10th, multiple changes to the UI based on advisor feedback needed to be made. Changes included rephrasing of questionnaire answers, formatting of the help screens, and rewording of error messages. Also necessary for testing was completion of a help website, hosted at gapahelp.wordpress.com, designed to explain to first-time users how GaPA's functionality worked. This website was drafted, revised, and completed within the first two weeks, and can currently be viewed through any internet browser. Lastly, other materials were created to aid in the testing process, such as explanatory handouts for how to run the program, links to .zip files containing GaPA, and a selection of sample .stl files. These materials were completed and distributed by the date of testing, along with a specialized version of the program designed specifically for testers that featured the newly-designed logo.

While testing was in progress, the group turned their attention to the overhaul of the printer selection screen, the script for the videos, and the poster. In the case of the selection screen, the design went through a comprehensive critique and was almost completely redone. The screen was inaccessible in the version of GaPA used for testing, so as not to generate feedback about an out-of-date interface. The script for the CS and ME submission videos were outlined and refined over the third and fourth weeks. Work on editing the report also progressed incrementally, with the drafting of the Broader Impacts chapter and continued editing on the Design chapter. When the deadline of the 20th for the CS poster and video approached, the group shifted almost entirely to recording the video and editing the poster as quickly as possible. Both items were completed and submitted on time. The ME video was due much later, and was completed shortly after testing for the program finished.

During the fifth week, the full paper was compiled, combining every previous submitted chapter into one unified document. By the sixth week, a full first draft had been completed, including chapters on testing results and program evaluations written after a thorough analysis of the most recent studies.

5. Design

This chapter outlines GaPA's design, including its user interface, backend algorithms, and user flow, as well as the development process that resulted in the final product.

5.1. User Flow Diagram

Our team developed a user flow diagram, shown in Figure 5.1, that describes the process of how a user should use this program. Our general idea for the process of the program was that the user would give the program an .stl file and print preferences, and then the program would automatically determine optimal orientations and generate the G-Code for those orientations. Figure 5.1 realizes this information as a linear process for the user to take.

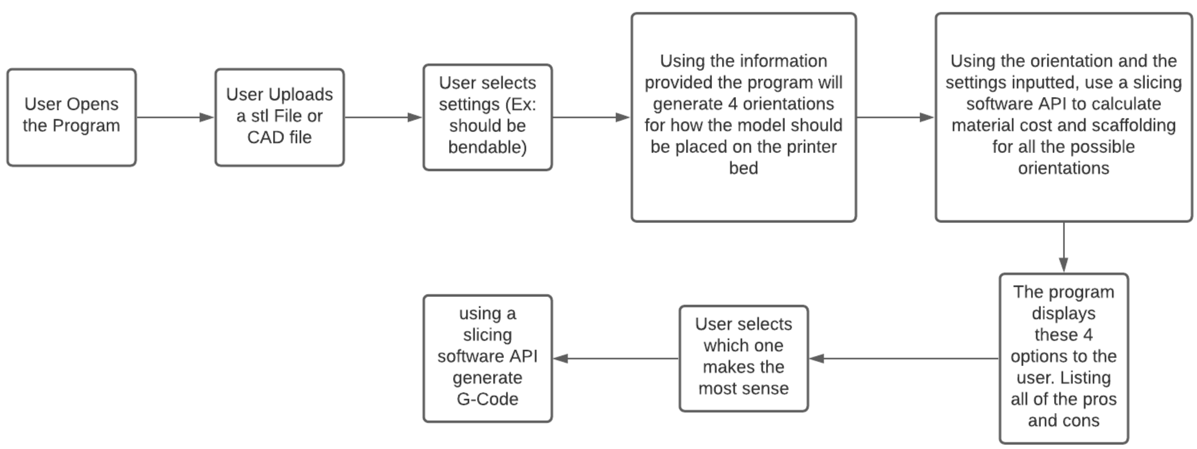


Figure 5.1: First draft of User Flow Diagram

This lays out our initial goals for the application. First, the user opens the program and inputs an .stl file for printing. The third box explains that the user would select their settings. But, in reality, the user would simply input their preferences for the use of the part, and the program would select the specific settings. The program would then use the .stl and the preferences provided to automate the orientation process, decide the top 4 orientations, slice the model, and display the orientations along with details about the print to the user. These details would include a list of pros and cons. For example, if an orientation had a low amount of overhang but a long print time, those would be listed as a pro and con respectively. This would provide the user with some flexibility in which orientation they choose. Finally, the program would generate the G-Code for the selected orientation. While our general design remains similar to what was described above, the processes would become more defined in the next iteration, shown in Figure 5.2.

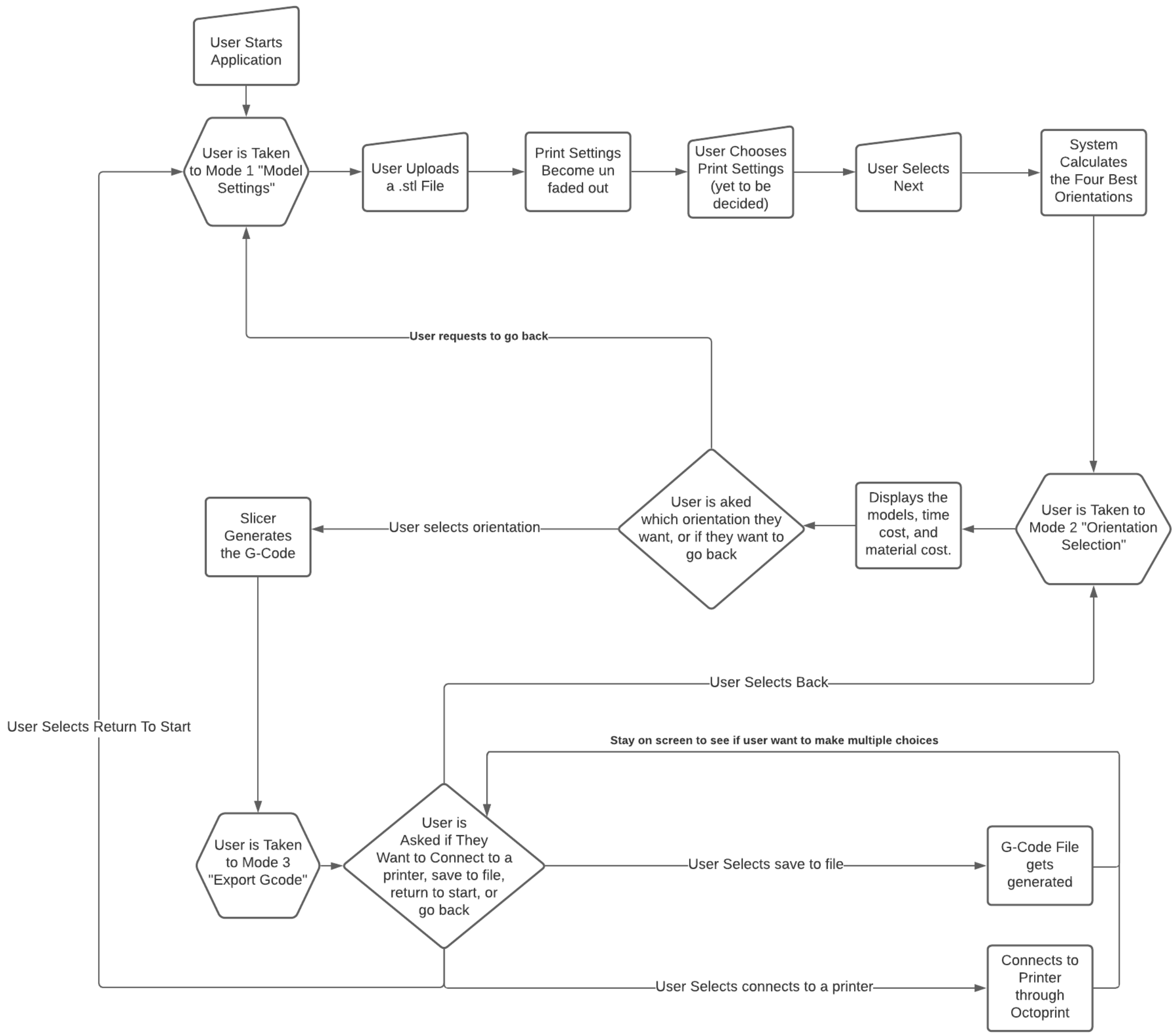


Figure 5.2: Second draft of User Flow Diagram

This user flow diagram expands on the general user experience defined in Figure 5.1 by defining and specifying certain buttons that the user needs to press in order to work their way through the program. It also introduces three distinct modes for the application, based on the three actions the user takes when proceeding through the program. Firstly, they give input in the form of print settings and an .stl file, which forms mode 1. Secondly, they select an orientation based on the given time cost and material cost for each option, forming mode 2. Finally, they can generate G-Code or print through Octoprint, forming mode 4. The user can return from each mode to the one prior, or return from the third mode to the first.

5.2. Program Technical Details

When deciding what platform our program would run on, we were primarily deciding between a web-based application and a Windows application. We ended up deciding to make a Windows application. There were several reasons for this decision, primarily running speed and not needing to maintain a server.

This program is a WPF (Windows Presentation Foundation) application running on the recently released .NET 5 platform. Our team decided to use WPF over UWP (Universal Windows Platform) because WPF, while being slightly less performant, offers greater customizability of interface components using third-party XAML, which stands for Extensible Application Markup Language, toolkits, such as Helix Toolkit which we use to display 3D models, and multi-platform support, due to deployment and distribution of the app not being limited to the Microsoft Store (Hunter 2020). Due to this decision, our team created the user interface layout for this program using XAML and the interaction code was written in C#. The IDE we used was Visual Studio 2019, which has a built in feature called XAML Designer. This provided our team with a visual, drag-and-drop toolkit to design the user interface of our program while automatically generating XAML code in the background. In order to display the .stl files as models in the interface, our group used Helix Toolkit, an open-source collection of functionalities that expand on the internal WPF 3D graphics technology (Helix Toolkit Documentation, 2015).

5.3. Evolution of the Interface Design and Rationale

At the beginning of the project, our team wanted to make a program that would act as a novice-friendly alternative to Cura, a popular slicing application for 3D printing. Users can only use Cura effectively if they know exactly how to orient the 3D model and what settings to apply to a print for it to have the desired quality. So, our program was designed with 3D printing novices in mind, not assuming that they know what the aforementioned settings mean.

Our original concept revolved around generating four unique orientations for an uploaded model based primarily on the flat surfaces present on the model. Figure 5.3 shows an example of four different orientations generated for the same model.

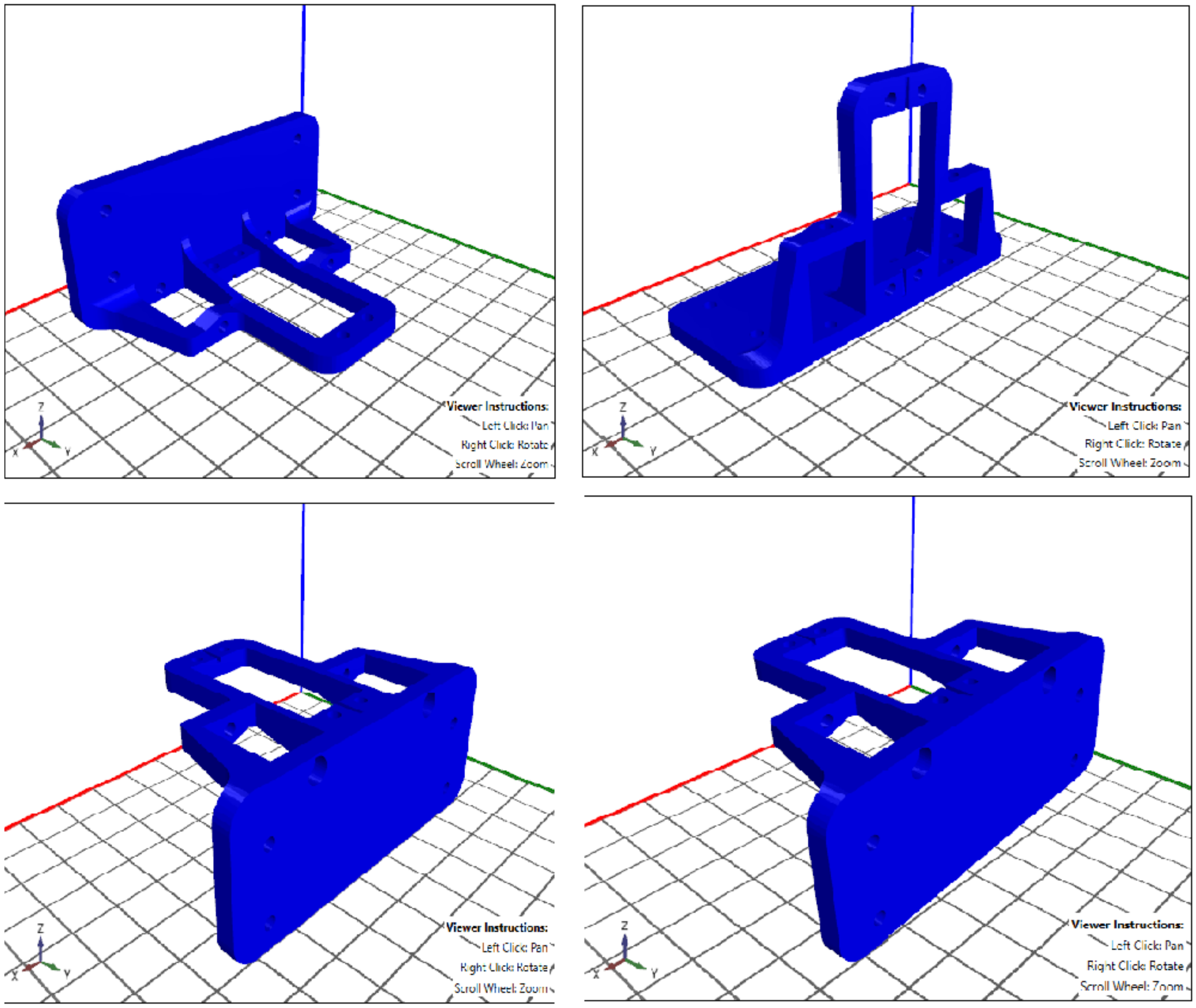


Figure 5.3: Example of four orientations generated for a model by our system

Our group decided to generate four orientations because we decided that three orientations provided too little flexibility and five orientations would overwhelm the user. Each orientation would be displayed with a 3D model and some statistics about the 3D print itself.

When we conceptualized the program, we split the 3D printing process into three steps: importing a model, configuring the print settings for it, and printing it. Our team decided to divide the program into three separate “modes,” or screens, based on these steps. Figures 5.4, 5.5, and 5.6 are early low fidelity prototypes of the three modes.

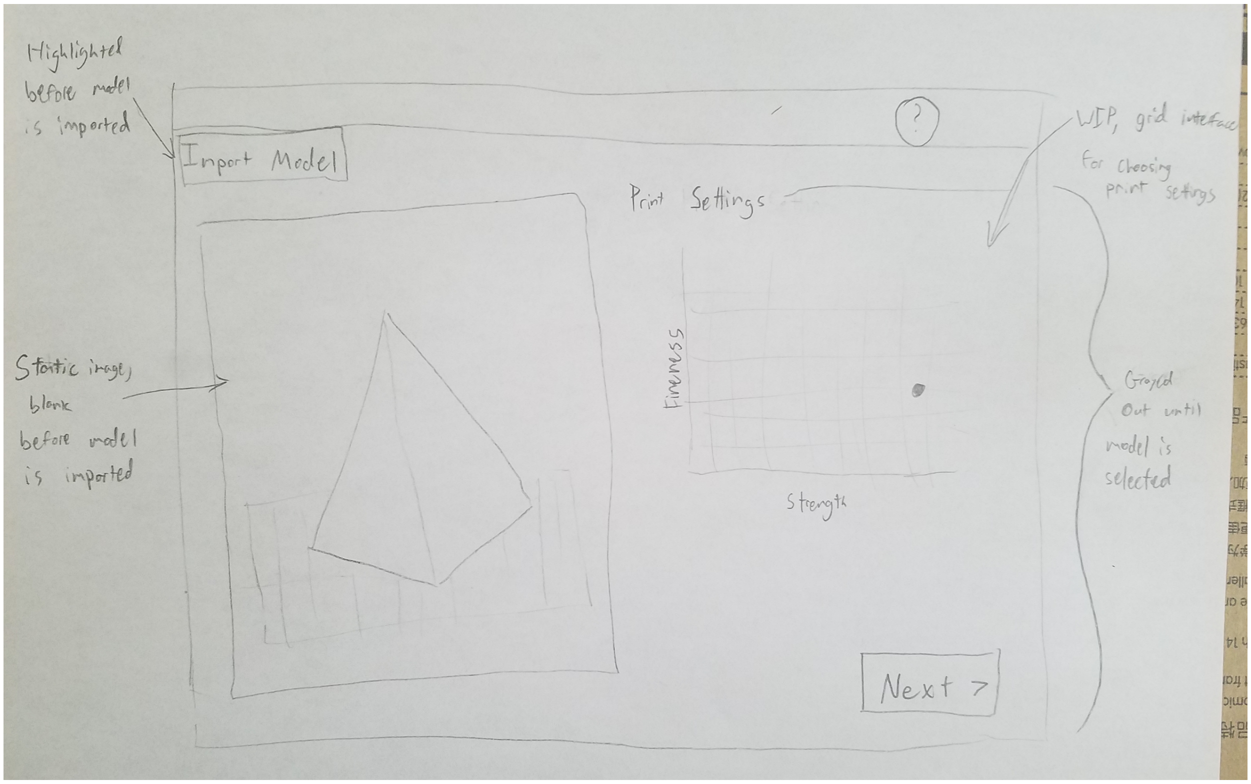


Figure 5.4: Early low fidelity prototype of Mode 1

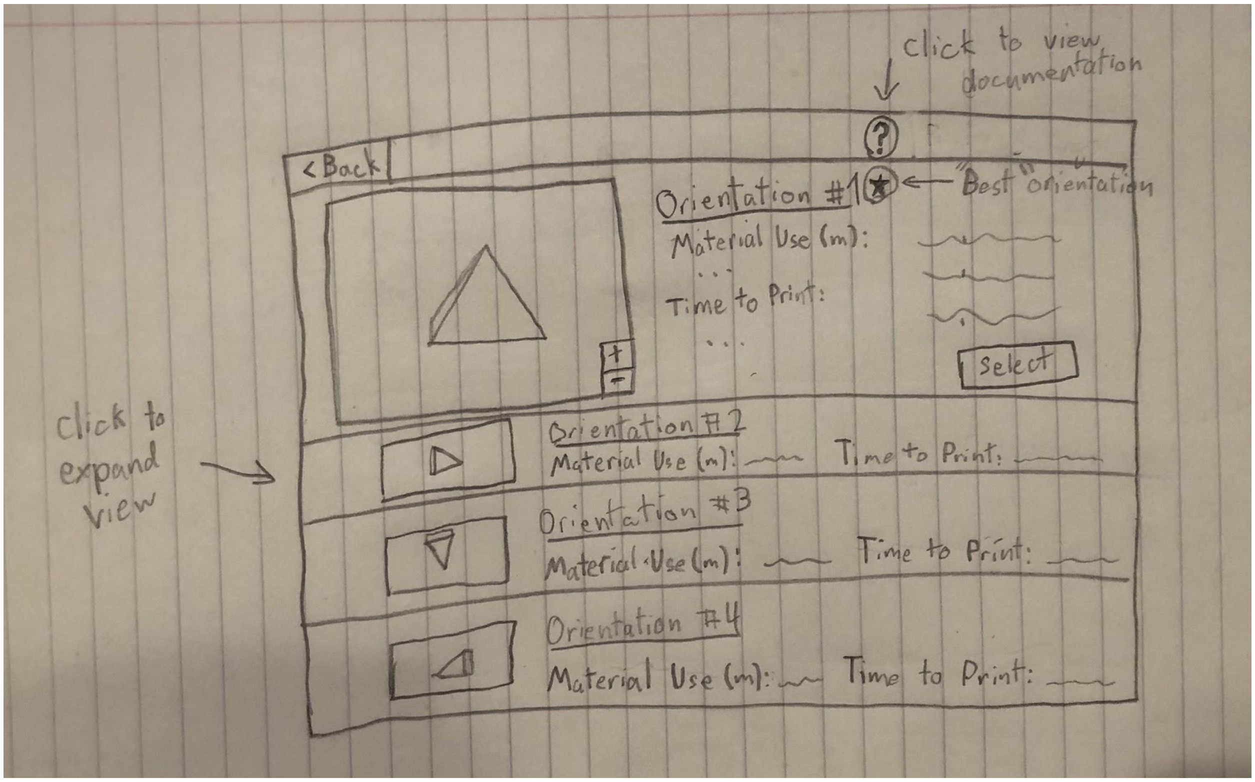


Figure 5.5: Early low fidelity prototype of Mode 2

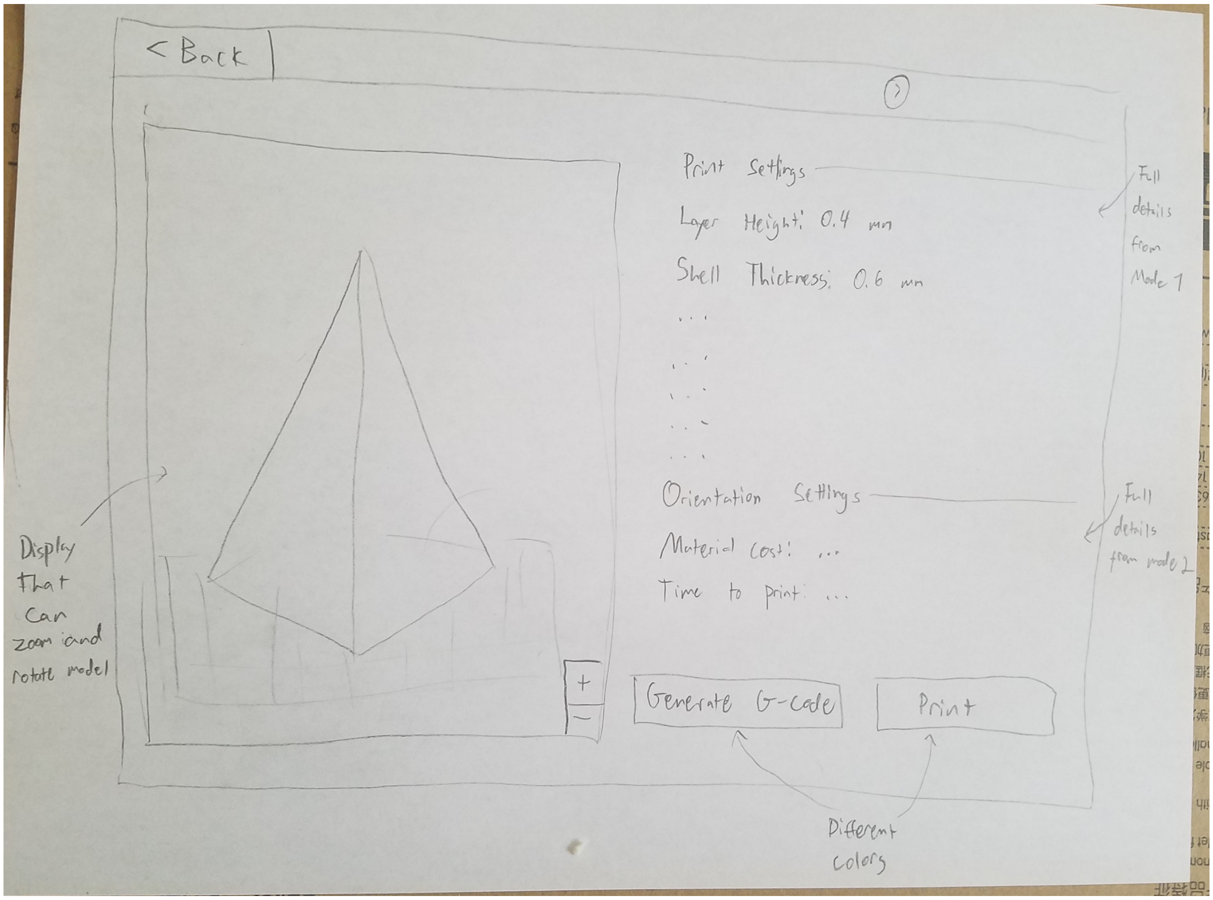


Figure 5.6: Early low fidelity prototype of Mode 3

Figure 5.4 shows a low-fidelity prototype of Mode 1, where the user would be able to import an .stl file, view it, and select settings relating to the print quality of the model. The “Import File” buttons in the top left corner of the screen would be highlighted in a bright color before a model is imported. This would serve as a visual indicator for the beginning of the program’s usage, directing the user’s attention towards the button when they first opened the program. After importing the model, it would be graphically displayed in a static image box so that the user could view roughly how it would look oriented on a printer bed. On the right half of the screen, the user would be able to set general specifications for the print quality on a grid with two axes, one representing print strength and the other representing fineness. The user would be able to drag a point on this grid to set a ratio between these two attributes for their print. This grid would be greyed out prior to the import of a model so as not to confuse the user upon opening the program. Finally, the “Next” Button in the bottom right corner would take the user to the next mode.

In our original concept for Mode 2, as shown in Figure 5.5, the user would be able to view the four orientations and detailed information that were generated for the model. The first of the four orientations, which would be ranked as the most optimal based on the user’s preferences, would be expanded by default, showing detailed information such as material use, time to print, and a short justification for the ranking. The expanded orientation would also have a 3D viewer displaying the model, which the user could use to rotate, pan, and zoom the model. The other three orientations would be collapsed, showing only basic statistics. The user would be able expand any of the other three orientations by clicking on it, which would result in the currently expanded orientation being collapsed. When the user finds an orientation that they like, they could click the “Select” button to proceed to Mode 3.

Our early design of Mode 3, as shown in Figure 5.6, would serve as a summary screen of the print and orientation settings and allow the user to either save the G-Code for the selected orientation or print it. The left half of the screen would be filled by a 3D model viewer, similar to the previous mode, where the user could view the selected orientation. The right half of the screen would be dedicated to the print and orientation information from the previous modes. The two buttons at the bottom, “Generate G-Code” and “Print” would serve as the end of the application process.

In the early design of Mode 1, our application had a grid (explained in further detail in section 5.15 and shown in Figure 5.28) where users could select a print quality ratio between the fineness and strength of the printed model. We eventually determined that figuring out this ratio would be hard to initially grasp for users, especially 3D printing novices. After brainstorming, we designed a much more simple, easy-to-understand questionnaire system. Our program would pose a series of simply worded, multiple choice questions to users about the print quality that they desired. The questions would be worded in a way that both printing experts and novices would understand. After answering the questions, the program would slice the imported model, generate four solutions, and provide a ranking of the solutions based on how well they fit the criteria set forth by the answers to the previous questions. So, our group refined the three modes to focus on the questionnaire concept: one for importing the model, one for the questions, and the final one for looking through the generated solutions and selecting one.

For our next set of low-fidelity prototypes, in addition to implementing the questionnaire concept, we also reorganized all three modes to implement the conclusions that our group came to as described above. Shown below in Figures 5.7 and 5.8 are the low-fidelity prototypes for Modes 2 and 3 on which we based our final interface designs:

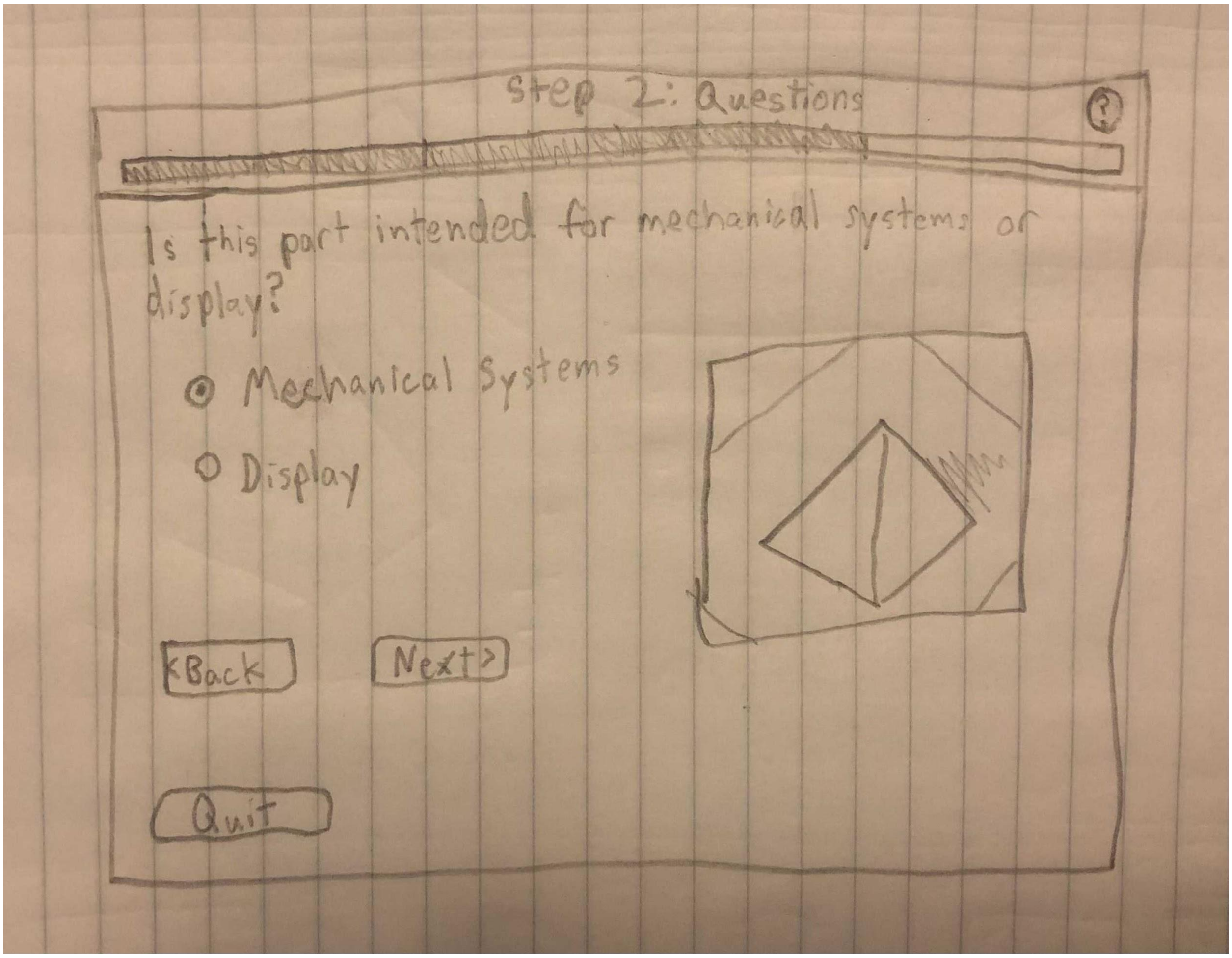


Figure 5.7: Final low fidelity prototype of Mode 2

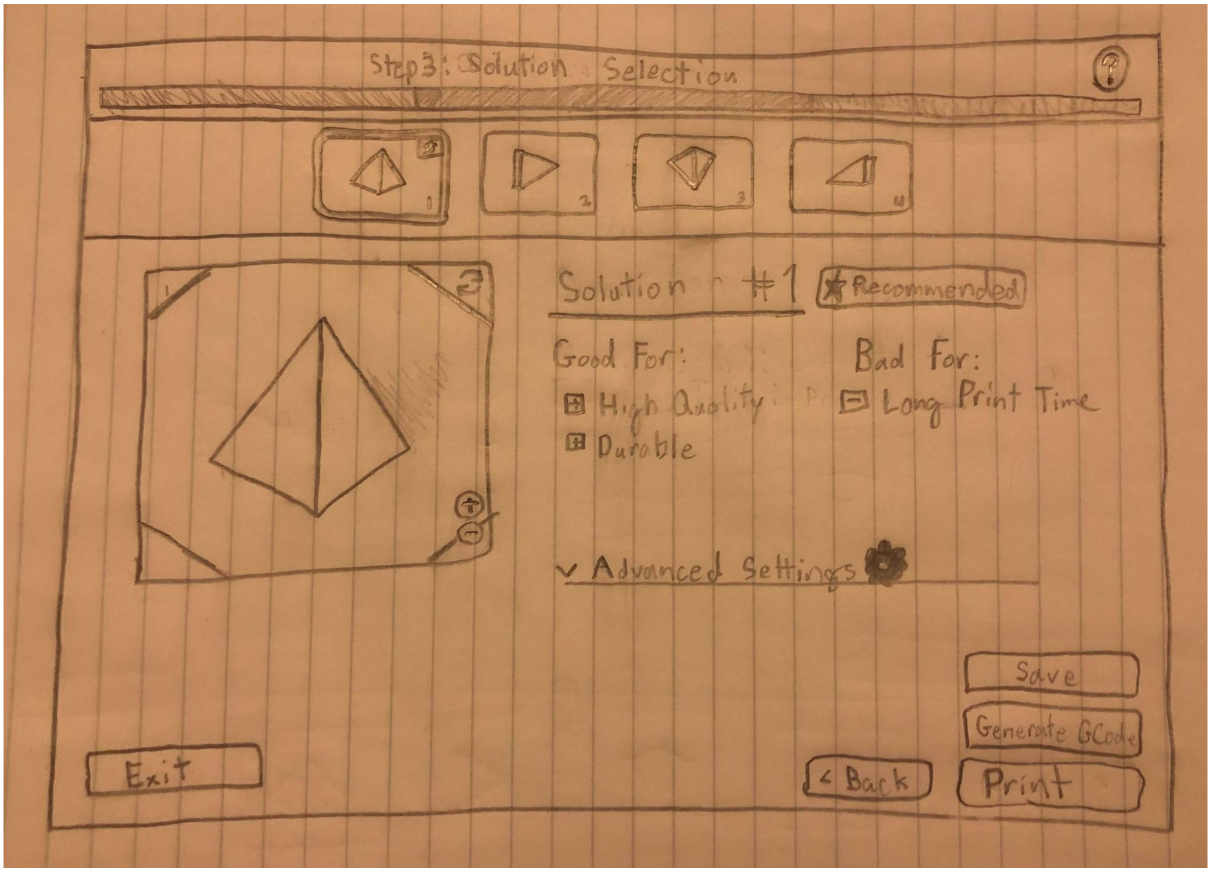


Figure 5.8: Final low fidelity prototype of Mode 3

Our redesign of Mode 2, shown in Figure 5.7, features the questionnaire as the main focus of the screen with an accompanying 3D viewer to view the imported model. Each question would be multiple choice, with the user selecting their desired answer from a list and clicking the “Next” button to move on to the next question. The user could also press the “Back” button to return to the previous question if they wanted to change their answer. When all of the questions have been answered, clicking the “Next” button would take the user to Mode 3.

Our final concept of Mode 3, as shown in Figure 5.8, features a more expanded view of the generated orientations, now known as “solutions,” than the previous low-fidelity prototype. All four solutions would be ranked at the top of the window, allowing the user to click on the one that they wished to view. The detailed information about the selected solution now took up the majority of the screen, so that it would be the focal point for the application users. It also listed the pros and cons of the solution based on the user's answers to the previous questions. All of the print statistics were now in a dropdown panel labelled "Advanced Settings." The buttons at the bottom of the screen stayed mostly the same except for one new addition: "Save." This new button would allow the user to save the current set of solutions as a file on their computer, which they could load at a later time to access the set again.

5.4. Mode 1: Import File

The first mode of the application (referred to as "Step 1" from within the application) allows the user to select an .stl format file that describes the object that they wish to 3D print, then view a simplified version of it.

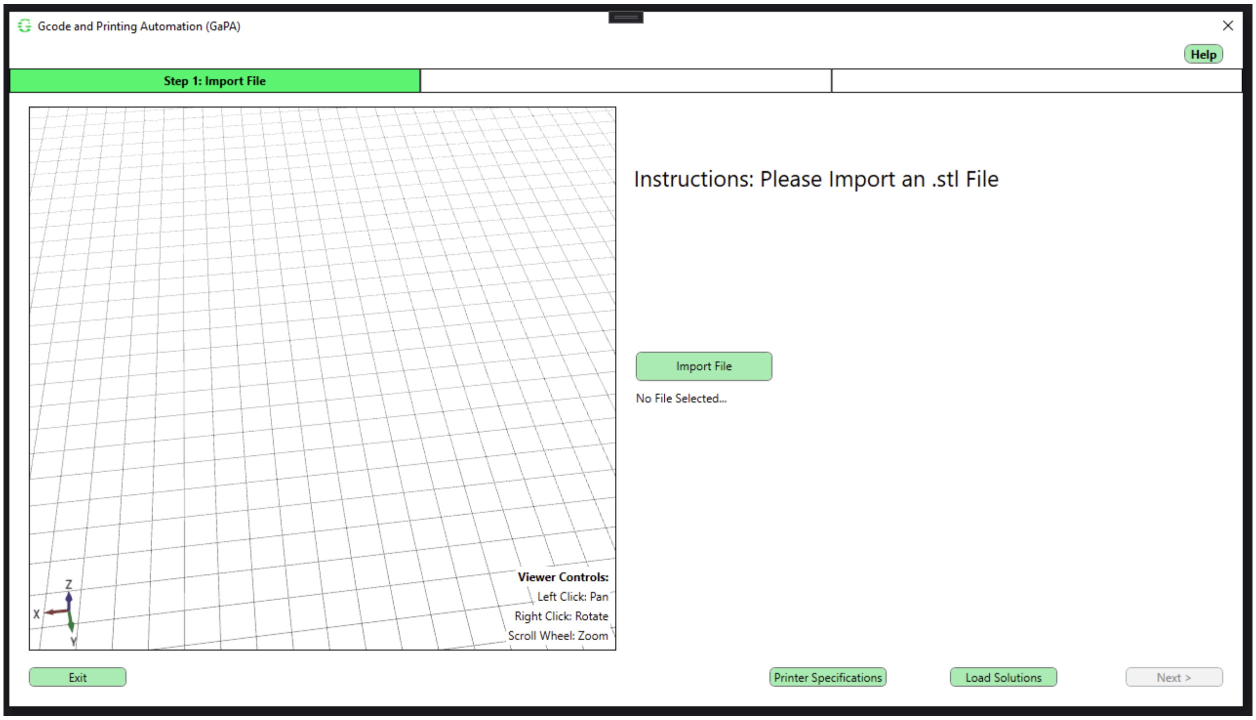


Figure 5.9: Mode 1, before importing a model

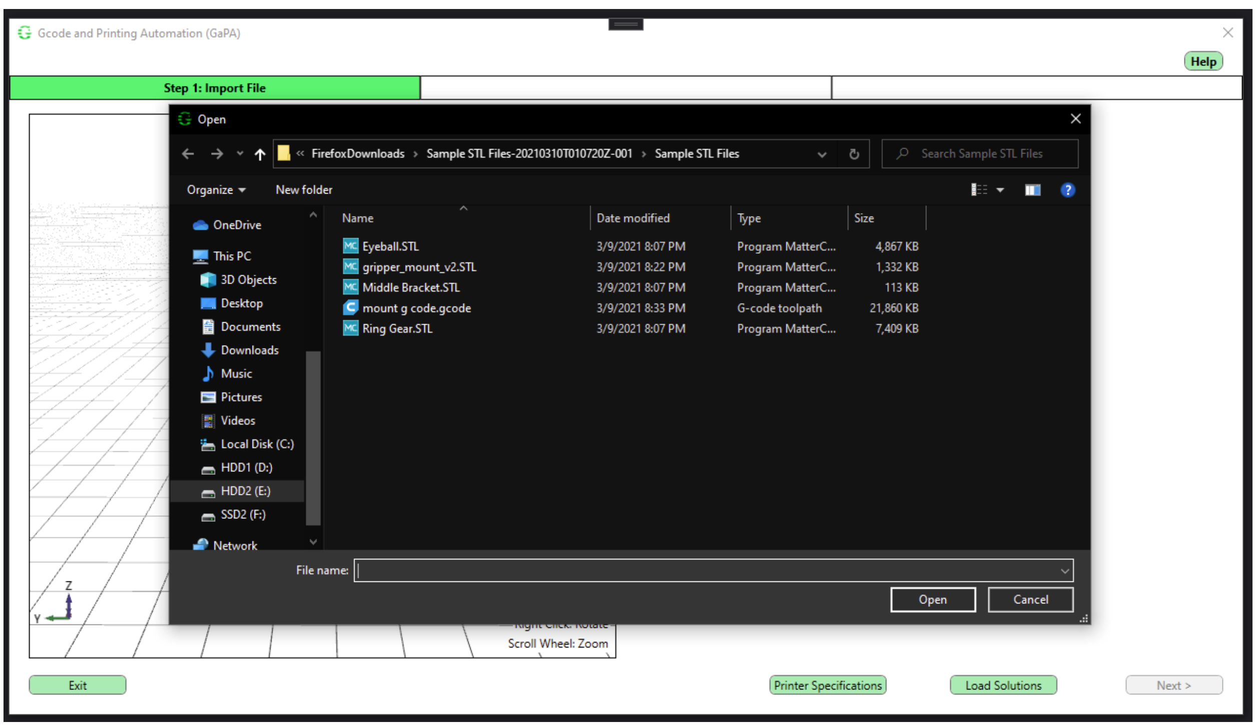


Figure 5.10: Mode 1, with File Explorer window open to import a model

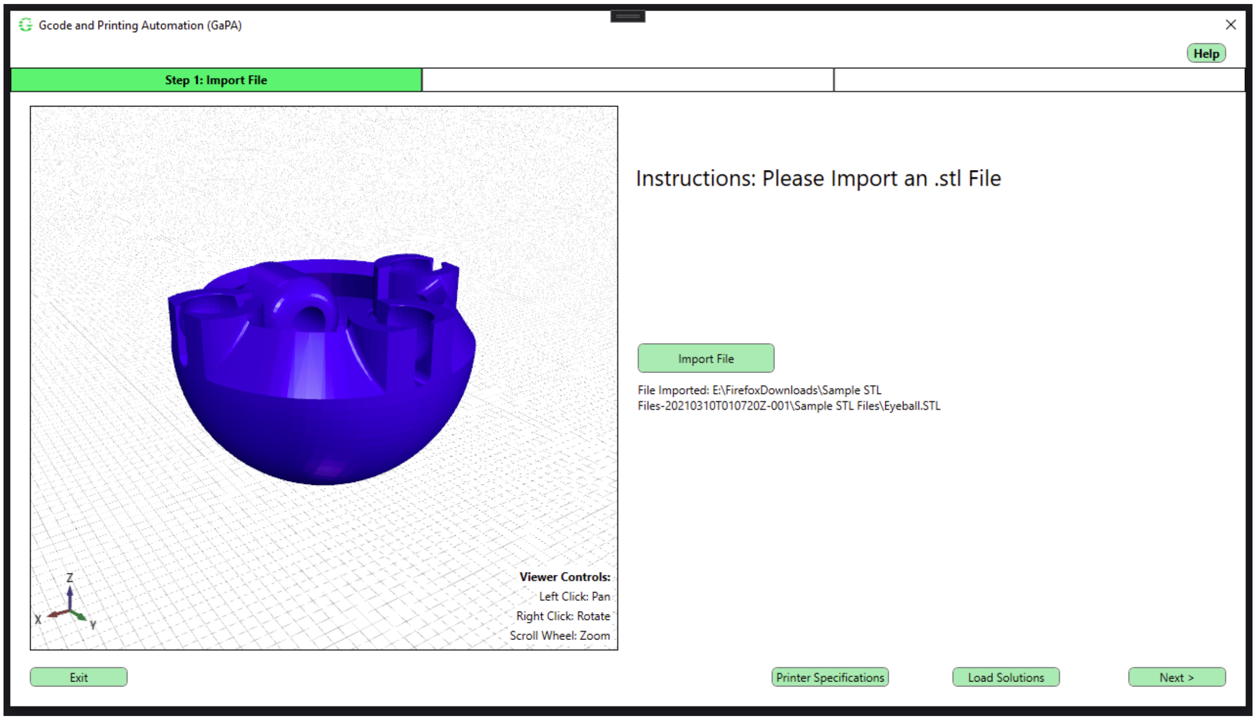


Figure 5.11: Mode 1, with model imported

The top of this window features the name of the mode, a progress bar, and a help button. The progress bar is segmented into three parts to convey to the user how much of the program's process they have completed. The help button, when clicked, displays contextual information pertaining to the interface and the function of each element. In order to help users further, a large label with instructions for the current mode is displayed on every screen. The majority of the interface consists of a 3D model viewer, an "Import File" button, and a text label to display the name of the uploaded file. The 3D model viewer, located in the left half of the window, allows the user to pan around, rotate, and zoom in on the imported 3D model. The window will, before a model is imported, look similar to Figure 5.9. When the user clicks the Import button, a File Explorer window opens as shown in Figure 5.10, allowing them to select an .stl file from their storage to use in the program. After this, in Figure 5.11, the selected file displays as a 3D model in the viewer on the left side of the window, and the text label changes to show the name of the uploaded file.

The four buttons at the bottom of the screen allow the user to navigate to different modes within the program. From left to right, these buttons are: “Exit,” “Configure Printers,” “Load Save State,” and “Next.” The “Exit” button allows the user to close the program. The purpose of this screen was to show the user their model after some initial processing, more specifically orienting it on a grid. This would give the user an idea of what the model would look like after being printed. The “Configure Printers” button, when clicked, navigates the user to a different mode in the program that allows them to view and change settings and properties for printers that are already in the system and even specify information for new printers. The “Load Saved State” button opens a File Explorer window that lets the user import a .gapa type file in order to restore a specific configuration of the third mode. Finally, the “Next” button allows them to go to the next step.

5.5. Mode 2: Determine Settings

Mode 2 poses a series of questions to the user about what qualities they want the 3D printed object to have in order to automatically determine the settings for the print. These questions would be worded in a way that users can answer with minimal thought.

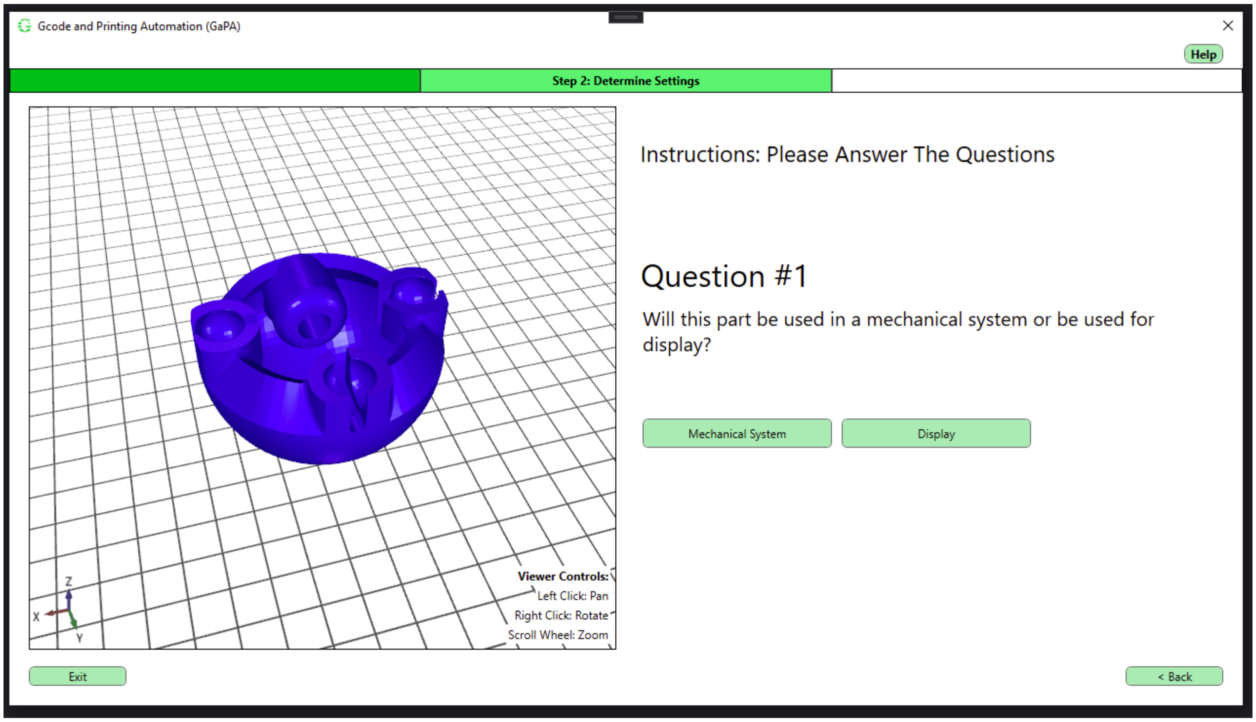


Figure 5.12: Mode 2, with an example multiple choice question

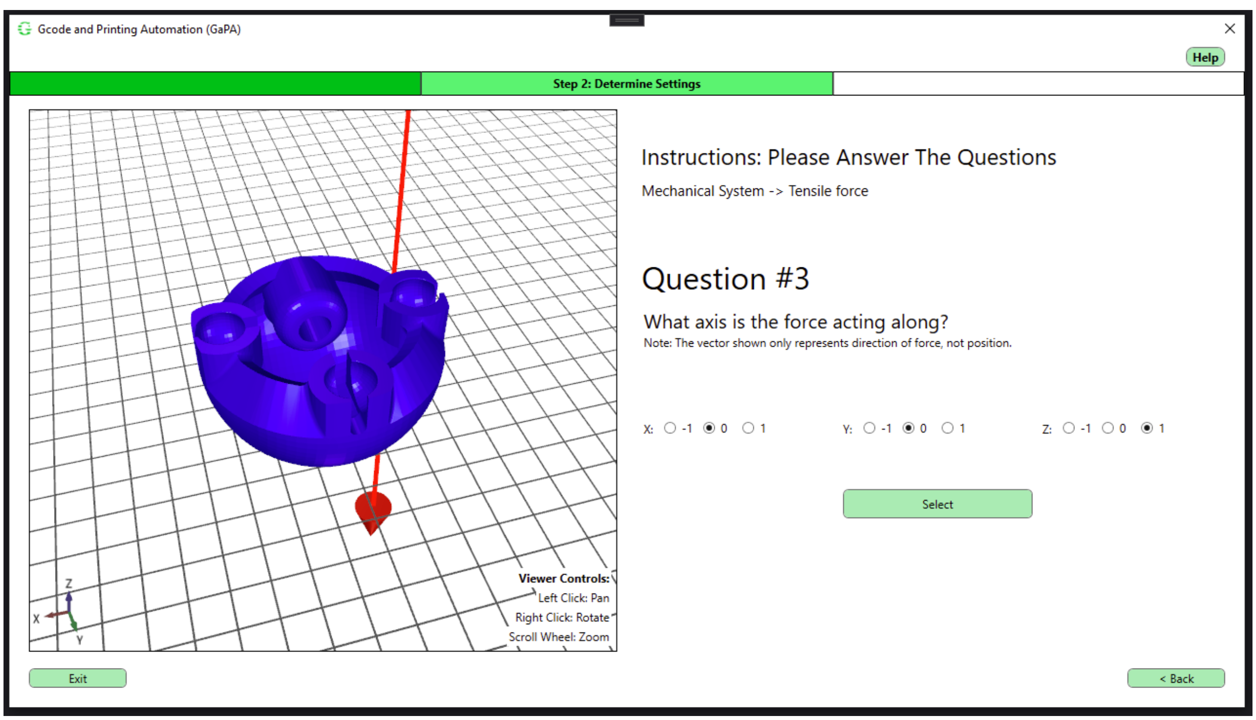


Figure 5.13: Mode 2, with example force question

The top of the window remains the same as the previous mode, except with the label reading “Step 2: Determine Settings.” The left side of the window shows the same 3D model as Mode 1. As seen in Figure 5.12, the right side of the screen shows the current question being asked and the answer choices associated with it. There are some special questions relating to forces exerted on the printed piece. For these questions, our program provides the user with a special series of inputs, as shown in Figure 5.13, to declare either positive, negative, or zero force on the x, y, or z axes. We initially made the user input the forces into text boxes but, after discussion and testing, we decided on radio selection buttons instead so as not to confuse the user. Once all of the questions are complete, the user is taken to the next mode automatically, which is the reason why there is no “Next” button present. The “Back” button takes the user back a question or, if they are on the first question, to mode 1.

5.6. Mode 3: Solution Selection

Mode 3 presents four different orientation and setting configurations, known as Solutions, to the user which have been automatically generated based on the answers to the previous questions.

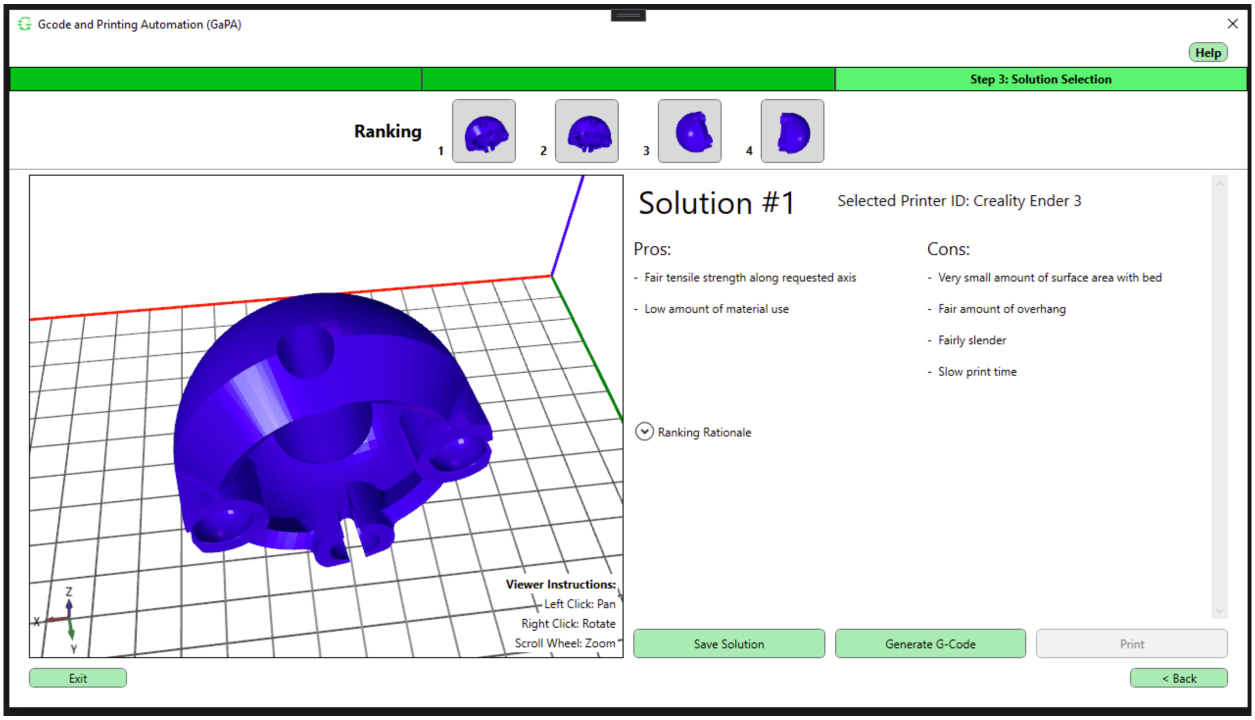


Figure 5.14: Mode 3 Interface Design

In addition to the usual help button and progress bar, the top of the screen features four buttons that will allow the user to navigate between the different solutions numbered according to their ranking. When a solution is selected, the middle section of the screen will populate with the 3D model of the solution and justification for the solution's ranking based on the answers to the previous questionnaire. The justification consists of a brief explanation of its ranking and lists of a few pros and cons of the solution according to the user's preferences. Below the justification, there is a "Ranking Rationale" dropdown to show the user all of the settings that were automatically set based on their responses in the previous mode. At the bottom of the window, there are three buttons: "Save" allows the user to save the current solution as a file, "Generate

GCode” lets them generate the GCode for the solution, and “Print” sends the solution to OctoPrint to be printed (not yet implemented).

In order to successfully automate the slicing process, our program provides settings for the slicer based on the user’s requirements. Our program will change the settings that have a significant effect on the final print while maintaining the defaults for settings that do not. In order to give the user control over the print, the user answers a small number of questions that prioritize certain features of the print, and the program will generate the settings based upon these selections to deliver the desired result.

5.7. Printer Specification

As our group designed the application, we realized that a new screen where admin users could add new printers to the system and edit the details of existing printers was necessary. From the printers that were specified on this screen, the system would select the one that best fit the user’s preferences before generating solutions. The design of this screen can be seen in Figure 5.15.

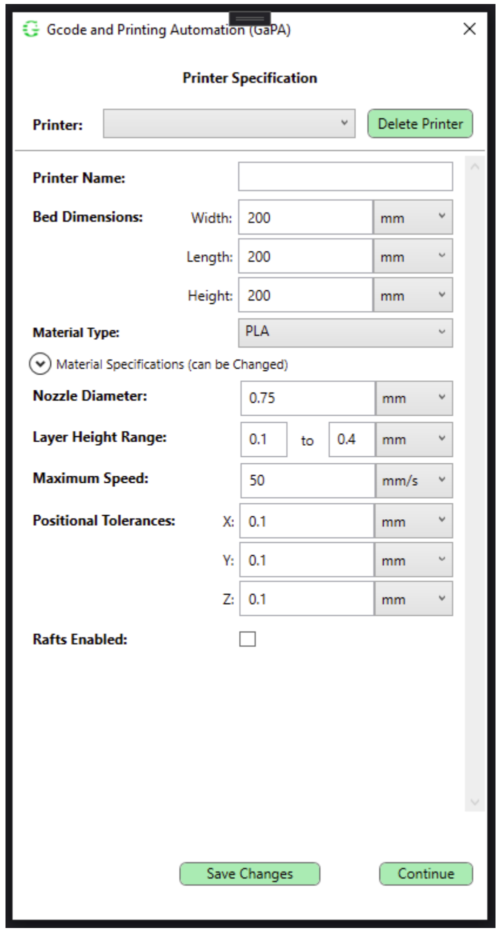


Figure 5.15: Printer Specification Screen

The top of the screen features a dropdown menu where users can select which printer to view or edit the specifications for, as well as a "Delete Printer" button if the user wants to delete the current printer. If the user wishes to add a new printer to the system, they can select the "Add New Printer" option from the dropdown, and the fields below it would populate with default values. The rest of this screen consists of textboxes where users can fill in the information about the selected printer. These fields include the name of the printer, bed dimensions, filament material type, nozzle diameter, the range of the layer heights, maximum nozzle speed, positional tolerances, and whether rafts can be printed or not. Under the material type selection is a submenu where the user can input specifications about the selected filament, including recommended nozzle and bed temperatures as well as the filament diameter. At the bottom of the screen are two buttons, “Save Changes” as well as the “Continue.” The Save Changes button saves the current text fields on the screen as a Printer object for use in the program, checking to see if every field contains valid inputs. The Continue button navigates the program back to the Import File screen.

5.8. Similarities and Differences to Currently Available Slicers

Our program is very different from traditional slicers. Despite this, several aspects of our program are inspired by the research we conducted into other slicers. For example, the design of our 3D viewer on the left of all three main modes is heavily inspired by the 3D viewers present in Mattercontrol and Cura. From Cura we also got the idea to grey out the Next button on the first mode until an .stl file is uploaded from Slic3r, and the placement of the “Generate G-Code” button in the bottom left of Mode 3. Additionally most of our default slicer settings are based on Mattercontrol’s default slicer settings.

However, the most inspiration that we got from the other slicers is where our program could be streamlined. The large number of options available in other slicers could overwhelm a new user. Our program does not allow users to select slicer settings. Instead our program will select the settings for the user based on the questions they answer and the printer that gets selected. We also tried to take the steps required to print in a traditional slicer and guide users through that in a linear sequence of events (i.e. import an .stl file -> figure out part usage -> orientate the part -> input slicer settings -> generate g-code). This streamlines the process in a way that other slicers do not.

5.9. User Requirements

The effects that these settings change are the physical properties of the 3D print and include external appearance, tensile strength, compressive strength, flexural strength, and dimensional tolerance. What the user wants the print will determine which of these are prioritized and as such these will be referred to as user requirements.

- **External Appearance:** The external appearance is characterized by the smoothness of the print. A print with improved external appearance will have smoother curves and finer details.
- **Tensile Strength:** Tensile strength is the component's ability to withstand stress from pulling or stretching. This is important for prints used as a component of a mechanical system that applies tension to said component.
- **Compressive Strength:** Similar to tensile strength, compressive strength is the print's ability to withstand force pushing into the print. This setting is important for prints that will be under compression in a mechanical system.
- **Flexural Strength:** The last strength component, flexural strength, is the component's ability to withstand bending forces. This is important for mechanical system components that have a bending force applied to them.

- **Dimensional Tolerance:** Not to be confused with external appearance, dimensional tolerance is how closely the resulting print matches the dimensions set by the original CAD model. This will be important for prints whose dimensions must be as precise as possible.

5.10. Cause and Effect of Settings on the User's Desires

Research was performed on how each user requirement is affected by slicer settings. Table 5.1 shows the properties of the 3D print and all of the settings that significantly affect each of these properties. The interactions will be described as well.

Table 5.1: Cause and Effect of Settings on Physical Properties of 3D Print

Effect (Properties of 3D Print)	**Cause (Slicer Settings)**
External Appearance	Layer Height Orientation Material
Tensile Strength	Infill Density Infill Pattern Orientation Shell Thickness Extrusion Width Material

Table 5.1: Cause and Effect of Settings on Physical Properties of 3D Print (Cont.)

Compressive Strength	Layer Height Orientation Shell Thickness Extrusion Width Material
Flexural Strength	Layer Height Orientation Material
Dimensional Tolerance	Material Printer Resolution Layer Height Extrusion Width

As shown in Table 5.1, the key settings that affect the results of the print are Layer Height, Extrusion Width, Infill Density, Infill Pattern, Shell Thickness, Orientation, Material, and Printer Resolution. For the purposes of clarity throughout this section, a brief description of each of these settings is provided.

- **Layer Height:** Layer height is the thickness of each layer when the model is sliced. This is commonly measured in millimeters and commonly varies between 0.06 mm and 0.6 mm. Layer height informs many other settings as well, such as the number of top and bottom layers (these values are equivalent to the shell thickness divided by the layer height).

- **Extrusion Width:** The extrusion width is the width of each line that the printer lays. This is set by multiplying the nozzle width by a predetermined ratio to achieve the desired width. Similarly to Layer Height, the Extrusion Width informs the number of outer perimeters to create the shell.
- **Infill Density:** To conserve both material and print time, the interior of 3D prints is normally formed by a mesh called the infill. Infill density is a percentage measurement of how much material is used out of the possible interior material. Solid infill is more common when print strength is of utmost importance, but a 90% infill has a tensile strength reduction of only about 15% with a print time reduction of 30% [Pandzic 2019]. This means that the loss of strength from the density is balanced by the greater cost reduction.
- **Infill Pattern:** The geometry of the infill is determined by the infill pattern. There are a wide variety of infill patterns varying from a simple grid pattern to more complex cubic patterns. The geometry of each of these patterns can change how forces are dispersed through the model giving certain patterns better strength under particular scenarios.
- **Shell Thickness:** The shell of a print is the solid material that makes up the outside of the print, essentially forming a hollow shell that is then supported with the infill. Shell Thickness is the measurement in millimeters from the exterior of the part to the beginning of the infill mesh. Shell thickness is not a distinct setting within MatterSlice [Landry 2015], but is instead composed of three settings: number of top layers, number of bottom layers, and number of perimeter threads.

The shell thickness can be found by multiplying the number of layers by the layer height for the top and bottom of the shell, while the sides of the shell can be found by multiplying the number of perimeters by the extrusion width.

- **Orientation:** Orientation is how the part is set on the bed relative to the origin. This is both the translational location of the part in the printer bed as well as the rotation of the part. There are three primary orientations considering the part's dimensions as a rectangular prism: standing, on-edge and flat. Figure 5.16 is a visualization of these three orientations.

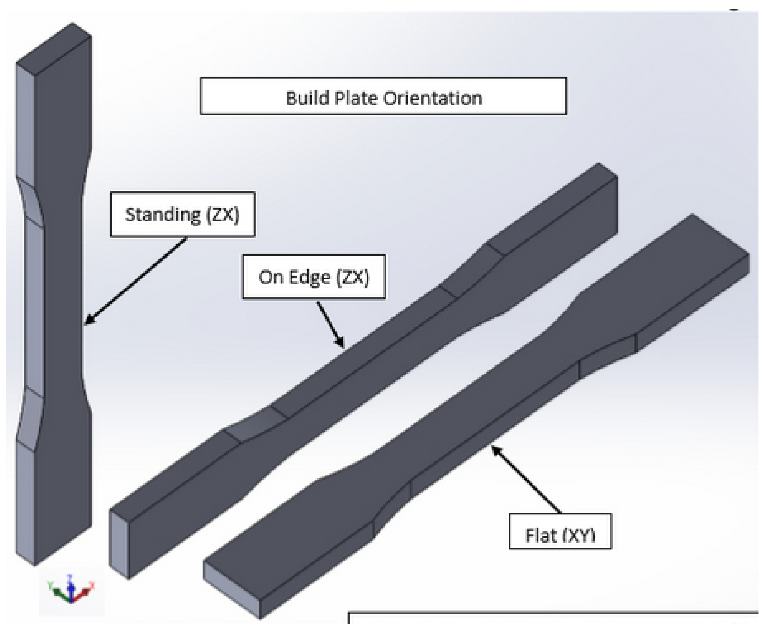


Figure 5.16: Standing, On-Edge, and Flat Orientations [Heckenlively 2019]

- **Material:** The chosen material has a major effect on the properties of a print. Specific properties of materials and how they affect the print will be examined later in this chapter.

- **Printer Resolution:** Each printer has a resolution rating that estimates the size of detail that can be printed, usually ranging between 50 and 150 microns.

The next section will outline the interactions between the settings and the user requirements as they are displayed in the table.

External Appearance

- **Layer Height:** Layer height has the greatest effect on external appearance. To visualize the difference in prints, two sets of pieces were printed: an eyelid piece with more complex geometry and a servo mount with simpler geometry. The eyelid CAD model is shown in Figure 5.17 and has a height of 22 mm, depth of 18.5 mm, and width of 28 mm. This model has several small details including the ridges along the front. This provides visual clarity for the comparison of layer heights. The Servo Mount Model in Figure 5.18 is primarily straight lines which is easier for a 3D printer. The servo mount has a height of 92 mm, width of 62 mm, and a depth of 36 mm.

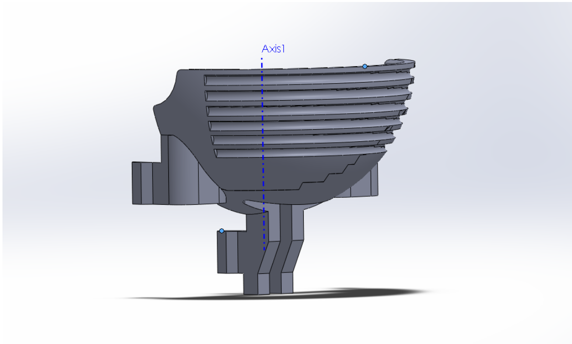


Figure 5.17: Eyelid CAD Model

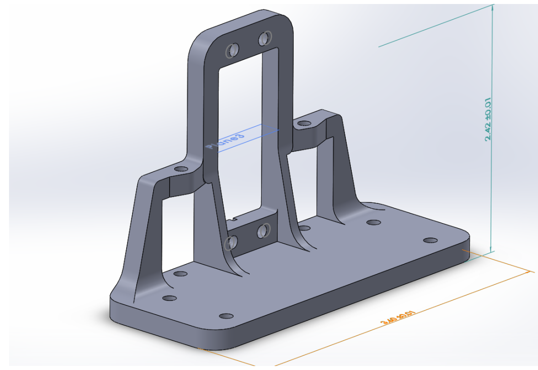


Figure 5.18: Servo Mount CAD Model

Both pieces were printed with layer heights varying between 0.06 mm to 0.6 mm with the corresponding layer heights in Table 5.2.

Table 5.2: Layer Height Profile Names and Corresponding Heights

Layer Height Name	Layer Height (mm)
Extra Fine	0.06
Fine	0.15
Normal	0.2
Draft	0.2
Extra Fast	0.3
Coarse	0.4
Extra Coarse	0.6

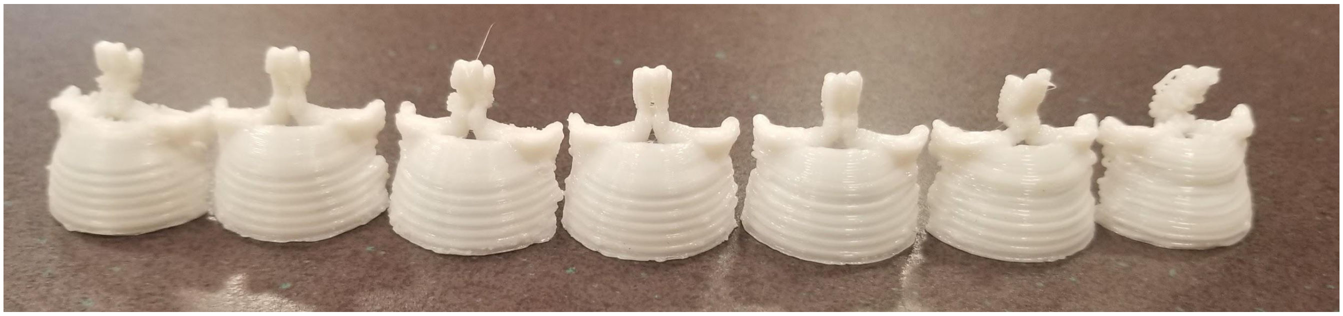


Figure 5.19: Printed eyelids - layer heights 0.06 to 0.6 mm from left to right

The printed eyelids displayed in Figure 5.19 are ordered left to right from smallest layer height (0.06 mm) to the largest layer height (0.6 mm). The slits on the front of the eyelid provide excellent visualization of how the layer height affects the final print. Beyond the fourth model, layer height 0.3 mm, the detail fades due to the increased layer height. Layer heights that exceed 0.3 mm lose both the slits and the smaller details on the top of the piece. Looking left toward the smaller layer heights, the exterior is overall improved with smaller layer height.

Figure 5.20: Printed servo mounts - layer heights 0.06 to 0.6 mm

Similarly in Figure 5.20, the servo mount models are ordered from smallest layer height to largest layer height. Given the simple geometry and straight edges, there is less variation between the finer layer heights. There is not a significant visual difference in surface finish between layer heights in the 0.06 to 0.3 range. Going towards thicker layer heights, diagonals lose their smoothness in the Z-Direction. Figure 5.21 shows a comparison between diagonals on the draft with a 0. 2 mm layer height and ExtraFast prints with a 0.3 mm layer height.

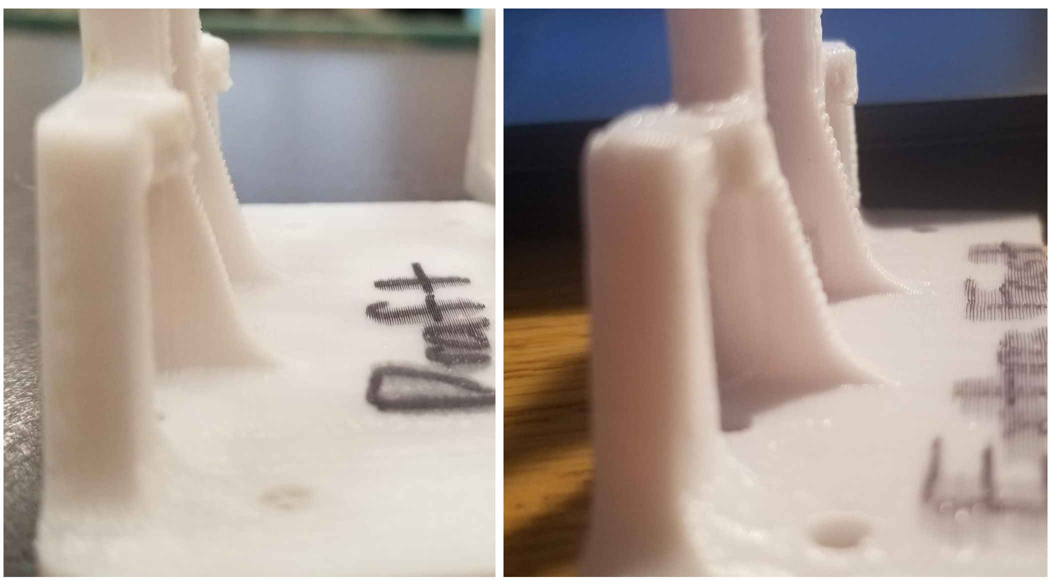


Figure 5.21: Comparison of diagonals with layer heights 0.2 and 0.3 mm

The Draft diagonal has a smoother face while Extra Fast is ridged. The diagonals get further ridged and the surfaces rougher as the layer height continues to increase. However, even the largest layer height (Extra Coarse - 0.6 mm) gives a print with the right shape. While these parts are not visually appealing, they are applicable for prototyping and have the advantage of improved printing times. Layer height correlates linearly with print time as the print time is mostly dependent on the number of layers. Since smaller layer heights require more layers to build the object, the print time will increase. A layer height of 0.1 mm will take twice as long as a layer height of 0.2 mm and three times as long as a layer height of 0.3 mm. Smaller layer heights give clear advantages for smoother appearances and, additionally, the smaller layer height allows for finer details that could not be achieved by a thicker layer height.

- **Orientation:** Orientation has little effect upon the external appearance of the print, with the exception of specific cases. There can be some shortening in the z-axis direction, primarily due to the calibration of the printer. With the difficulty that comes with calibrating the z-axis of certain models of printers, the first layer can have some variance in its height if the nozzle is closer to the bed than intended. This has no effect on later layers, as the z-axis movement of the nozzle is unaffected beyond this layer. This issue can also be alleviated with a raft, making the first layer height unimportant for the model itself.

  The orientation of the model also affects where the raft and supports are attached on the model. Supports and rafts help avoid warping and potential failure, but surfaces on the model where a support or a raft are attached will have a rougher finish than the rest of the model's exterior. The generation of supports If the user is concerned about the finish of a particular face, this face should not be oriented down toward the bed, as this would likely cause it to require supports. The user should be informed of this if they have chosen to prioritize external appearance.

- **Material:** The material will have an effect on the external appearance due to warping. Materials such as ABS and Nylon are more likely to warp and negatively affect the external appearance [von Ubel 2020]. Additionally, the tolerance of the filament has an effect on the quality of the extrusion. A small tolerance, about +/- 0.05 mm, will avoid irregularities in the filament to prevent backflow within the nozzle and ensure a consistent flow rate [Bouthillier 2016].

Tensile Strength

- **Infill Density and Pattern:** As the infill density increases, the tensile strength of the print also increases, as illustrated in Figure 5.22 where the ultimate tensile strength (RM) and tensile yield strength (R02) were measured at different infill densities [Pandzic 2019].

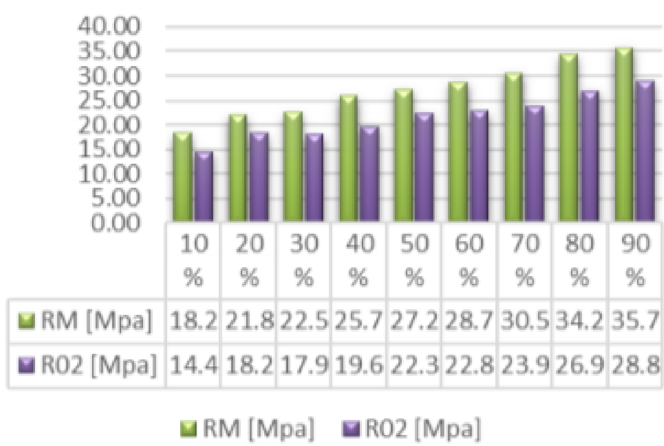


Figure 5.22: Tensile and Yield Strength by Infill Density (Concentric Pattern) [Pandzic 2019]

This trend is fairly consistent across all infill patterns with slight variations where some patterns are stronger at certain densities. For example, the strength of the ZigZag pattern peaks at 60% infill density. However, even at this peak, the strength provided by the ZigZag pattern is less than that of the Concentric pattern. The Concentric pattern has the highest tensile yield strength of the patterns when the infill is 20% or higher [Pandzic 2019]. Figure 5.23 compares the Concentric Infill Pattern with the Rectilinear Pattern visually.

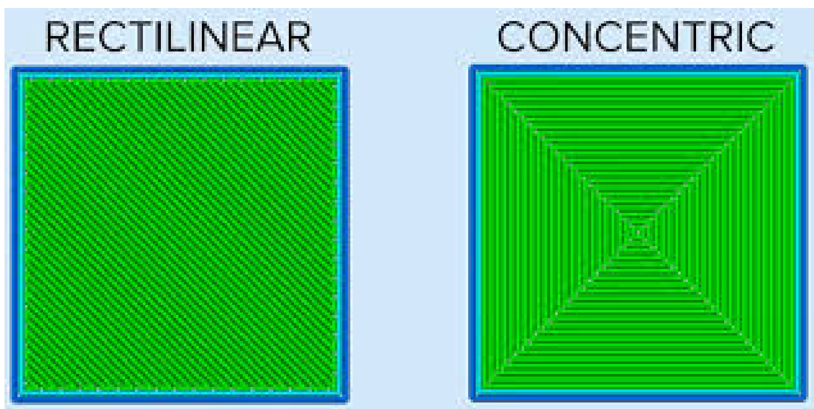


Figure 5.23: Rectilinear infill pattern compared to concentric [Engineerdog 2015]

Since tensile strength increases as the infill density increases, the infill will always be above 20%, so the infill pattern for prints that prioritize tensile strength should use the Concentric pattern. Additionally, greater tensile strength is achieved when the infill pattern is printed at a 45 degree angle to the applied tension, as this helps disperse the force into the stronger shell of the print [Engineerdog 2015]. While a print would have maximum tensile strength at 100% infill density, this is neither cost nor time efficient. Prints that prioritize tensile strength are often printed at 70% infill to balance the cost saving from material use and print time with the tensile strength of the print.

- **Orientation:** In order to provide the maximum tensile strength, the print should be oriented in such a way that the applied tension should be along the xy-plane. Figure 5.24 shows a rod that may have tension applied along the length. Since the bonds between layers in FDM are weaker than within the layer, this orientation will ensure that the tension is applied in the strongest plane. Additionally, the model should be oriented on-edge or flat [Chacóna 2017].

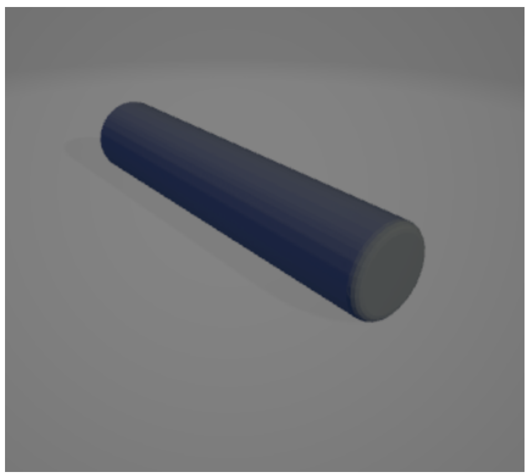


Figure 5.24: Orientation for a rod with tension applied on the length

- **Shell Thickness and Extrusion Width:** A thicker shell provides greater strength for all types of forces. Figure 5.25 compares examples of varied shell thickness and extrusion width. Additionally, a shell with a greater extrusion width provides more strength (though print quality suffers when the extrusion exceeds 160% of the nozzle width), even when the thickness remains the same [CNC Kitchen 2019]. Figure 5.26 shows differences in the extrusion thickness while still keeping the same shell size. However, the thickness of this shell can be limited by the geometry of the model. If there are walls too thin for the selected extrusion width, these will have poor tolerances. As such, this is a limitation on the shell thickness and extrusion width.

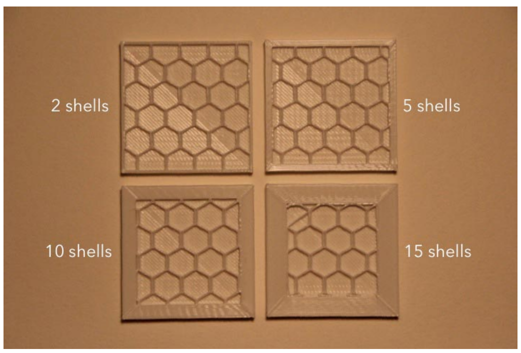


Figure 5.25: Shell thickness example [Jesimon 2014]

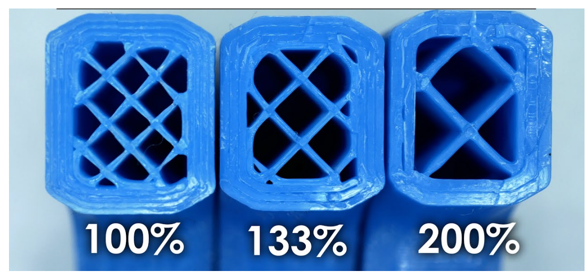


Figure 5.26: Extrusion thickness [exaCNC Kitchen 2019]

- **Material:** Due to its properties, Polyethylene terephthalate glycol (PETG) has stronger bonds between layers, making it ideal for tensile forces. This is followed by Polylactic acid (PLA) and then Acrylonitrile Butadiene Styrene (ABS) [3D Pros].

Compressive Strength

- **Layer Height:** In compressive models, layer height is a less significant factor than orientation or infill [Zaman 2018]. An average layer height of 0.2 mm is recommended as such.

- **Orientation:** Similar to tensile strength, the print should be oriented in such a way that the applied compression should be along the xy-plane. This orientation will ensure that the compressive is applied along the strongest plane. Again, the model should be printed on-edge or flat [Chacóna 2017]. This means that the orientation will work in the same way as tensile strength, as was shown in Figure 5.25 previously.
- **Shell Thickness and Extrusion Width:** The interaction between these settings and compressive strength are the same for that of tensile strength. A thicker shell provides greater strength for all types of forces. Additionally, a shell with a greater extrusion width provides more strength even when the thickness remains the same: although print quality suffers when the extrusion exceeds 160% of the nozzle width [CNC Kitchen 2019]. However, the thickness of this shell can be limited by the geometry of the model. If there are walls too thin for the extrusion width selected, these will have poor tolerances. As such, this is a limitation on the shell thickness and extrusion width.
- **Material:** Similarly to tensile force, PETG is ideal for compressive forces due to its stronger internal bonds. This is again followed by Polylactic acid (PLA) and then Acrylonitrile Butadiene Styrene (ABS) [3D Pros, n.d.].

Flexural Strength

- **Layer Height:** Smaller layer heights provide greater flexural strength and have the most significant effect of any printer settings. This relationship is not linear as thicker heights lose strength disproportionately. Changing the layer thickness from 0.1 to 0.3 caused the flexural strength to drop by more than three times [Luzanin 2014].

- **Orientation:** Similar to the other force types, the print will have the greatest flexural strength when the force is applied along the xy-plane. Unlike the other forces, however, flexural strength is noticeably increased with an on-edge orientation over a flat orientation. The graphs in Figure 5.27 show the flexural stress capacity compared to layer thickness. As can be seen in the middle graph (the On-Edge orientation) the strength is consistently higher. As such, on-edge orientation is ideal for flexural strength [Chacóna 2017].

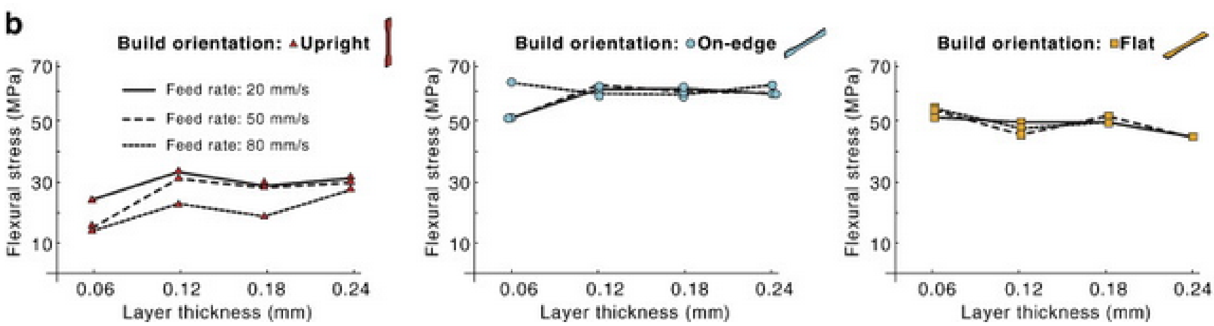


Figure 5.27: Comparison of Orientations for Flexural Stress

- **Shell Thickness and Extrusion Width:** Again, the shell thickness and extrusion width interact with strength in the same way as previously described for tensile strength. A thicker shell with fewer perimeter lines improves the strength of the print though this is limited by the geometry of the model.

- **Material:** In flexural strength tests, PETG and ABS performed about the same. PLA, however, was significantly weaker due to its greater rigidity causing earlier fractures [3D Pros, n.d.].

Fine Details

- **Printer Resolution:** Printer resolution determines how fine details can be made. A lower resolution value means that the printer can make finer detailed prints. This is measured in microns or hundredths of mm: this tends to be between 50 and 100 microns.

- **Positional Tolerance:** Positional tolerance is the tolerance that the nozzle has for movement accuracy. These should be very small measurements, in the range of 0.004 to 0.020 mm.

- **Material:** While the materials should theoretically all result in the same dimensions, in practice, warping and other material properties cause variance in dimensions. Due to its greater rigidity, PLA has the best dimensional tolerances. This is followed by PETG, Nylon, and ABS [3D Pros, n.d.].

- **Layer Height:** Layer height is important when the model has fine details, as a thicker layer height may not be able to make these details. A smaller layer height makes it easier for the printer to capture the details of the model.

- **Extrusion Width:** Similarly to layer height, extrusion width can limit the detail that the printer can make. Additionally, the extrusion width should not be more than 150% of the nozzle width, as any more than that leads to irregularities in the print [CNC Kitchen 2019].

5.11. Other Cases

While these user expectations cover many cases for printing models, there are some other common cases that do not fall under the previously covered user expectations.

- **Gears:** Due to their functionality and application of force at each tooth rather than a single point, gears do not fall under the categories of user requirements above. As such, gears have specific settings for printing. Recommendations for printing gears were given by Professor Radhakrishnan from experience. A 100% infill is recommended for gears, as this gives the maximum strength. Additionally, due to the generally thin size of gears, there is a smaller cost loss for 100% infill. Orienting the gear at an angle to the bed both improves its strength by placing the weaker bonds between layers on diagonals as well as improving hole tolerances that are often skewed by the first layer [Radhakrishnan, P; WPI ME; Personal Communication, November 2020]. Nylon provides the best overall strength for gears, followed by PLA, ABS, and PETG. Printers with superior resolutions will deliver better results [3D Pros, n.d.].
- **Flexible Prints:** Some prints are meant to be flexible: for example, the wheel of a remote controlled car. A flexibility requirement means that the model should be printed with the material TPU, which decides many of the settings due to its inherent difficulty to print. In order to maintain flexibility, the infill density should be 5-10% with a gyroid pattern, as this provides support to the model while maintaining its flexible nature [AMFG, 2018]. Due to the material properties of TPU, the feed rate should be about 15-20 mm/s. Additionally, keeping the layer

height range between 0.1 mm and 0.2 mm will help ensure bonding due to the thinner layer height [Simplify3D Flexible, n.d.]. Unlike many other materials, TPU should have retraction disabled.

5.12. Materials

The most common materials used for 3D printing are PLA, ABS, PETG, TPU, and Nylon [von Ubel, 2020]. In addition to these materials, the WPI Foisie Makerspace also has Polycarbonate and Polypropylene available for use. The properties of each material was researched to learn their strengths and weaknesses. Additionally, the material determines the temperature of both the nozzle and the bed. While our program will prioritize manufacturer-specific temperatures, the following list provides a baseline temperature for each material. Each material includes a brief description, the advantages of the material, the disadvantages, the baseline temperature, the print speed, and any additional considerations for the material.

PLA [von Ubel 2020] [All3DP 2020]

- A biodegradable plastic derived from cornstarch
- Easy to print with
- Less durable and more susceptible to heat than ABS but more rigid
- Printed between 180 and 230 ^{o}C with a bed temperature at 30-50 ^{o}C (though not necessary)
- Print speed of 30-40 mm/s

ABS [Bhavnagarwaia 2019] [Kondo 2019]

- A durable plastic similar to that used in Legos
- Provides durability at a low cost and weight
- Considerable shrinkage/warping
- Printed between 220 to 250 degrees Celsius with heated bed at 110 degrees C
- Print speed of 30-40 mm/s
- First layer should be printed at 150% height and 70% speed

PETG [von Ubel 2020] [All3DP 2020]

- Glycol combined with plastic
- Can withstand higher temperatures and prints with faster speed
- Poor surface finish
- Printed between 220 to 250 degrees Celsius and heated bed at 60 degrees C
- Print speed of 40-50 mm/s

TPU [AMFG 2018] [Simplify3D Flexible]

- Combines properties of plastic and rubber
- Useful for any flexible plastic piece
- Less dimensionally accurate
- Printed at 225 to 250 degrees C and the bed at 50 degrees C
- Print speed of 20-30 mm/s
- Retraction disabled

Nylon [von Ubel 2020] [All3DP 2020]

- Provides both strength and flexibility
- High overall material strength with good flexibility
- Worse exterior finishes than PLA and ABS and considerable shrinking
- Printed at 225 to 250 degrees C and the bed at 85 degrees
- Print speed of 40-50 mm/s
- Use a brim or raft

Polycarbonate [Simplify3D Polycarbonate]

- A very strong material with little flexibility
- Resistant to heat and physical impact and is transparent
- Prone to warping and requires high printing temperature
- Prints at 290 to 310 degrees C and bed at 100 degrees C
- Print speed of 30-40 mm/s
- Print first layer with 150% height, 120% width, and 20% speed

Polypropylene [Simplify3D Polypropylene]

- Semi rigid and lightweight plastic
- Ideal for low strength applications
- Heavy warping
- Prints at 235 degrees Celsius and bed at 90 degrees
- Print speed of 50-60 mm/s
- Requires raft and cooling fan. 10 mm/s for first layers.

5.13. Printer Information

As part of the slicing process, GaPA selects a printer to use. In order to do so, the program needs access to certain information about the printer. As such, we have decided to include specifications for each available printer. The necessary specifications are as follows.

Model Name

- Bed Dimensions
  - Bed Width (x-axis)
  - Bed Length (y-axis)
  - Bed Height (z-axis)
- Available Materials
  - Material Type
    - Manufacturer
    - Recommended Nozzle Temperature
    - Recommended Bed Temperature
    - Diameter of Filament
    - Filament Tolerance
- Nozzle Diameter
- Nozzle Temperature Range
- Bed Temperature Range
- Resolution
- Maximum Speed
- Positional Tolerance
- Raft Enabled?
- Drive Type (Bowden or Direct)
- Metal Extruder?

The model name allows the program to inform the user which printer is being used. The bed dimensions ensure that a printer is selected that can fit the model within the printing space. Each material available to the printer should be listed to ensure the printer has the desired material. The material information also includes the temperatures the nozzle and bed should be at when printing. The nozzle diameter is important for the extrusion width as this setting is a multiplier to the original diameter. The temperature ranges of both the bed and nozzle should be included primarily to ensure that the material temperatures are within this range. Setting this will have no direct effect on GaPA's results. The resolution specifies what layer height range the printer is able to make, while the speed provides a baseline for the travel speed that can be varied depending on the material. Tolerances inform the user how much the part will vary from the initial model as a result of the printer. While printers often include this in their specifications, the tolerance should also be determined experimentally by measuring printed parts and comparing them to the intended measurements. This should be done across a variety of settings however in order to isolate the tolerance caused by the printer [Formlabs, n.d.]. Enabling a raft will result in parts printed by that printer to include rafts.

5.14. IF/THEN Statements

Using the information gathered through research of best 3D Printing Practices, as well as the properties of both materials and printers, a list of If/Then statements was created. This list includes all of the user desires as the "If" statement to categorize them and minimize the amount of information required. The "Then" statements refer to how a setting is changed or a material is selected given the user requirement. In addition to these If/Then statements, there is a second set of If/Then statements that rely on information obtained from the first set, such as the material. This list is used to create relationships between requirements and settings so that our program has a method of determining the settings. The full list of statements can be found in the Appendix A as "List of If/Then Statements for Settings."

5.15. Survey Questions

In order to apply these If/Then statements, the user requirements need to be known first. To accomplish this, a tree of questions was developed that obtain the necessary information in as few questions as possible. The questions that are used in the application, as well as the possible answers and effects, are listed in Appendix B. This method of gathering information was selected to minimize the user input. By asking what the user expects of the use and characteristics, the system does not rely on the user's knowledge of the settings or the properties of the print.

Other input methods that were considered include a grid with strength and exterior finish as the axes shown in Figure 5.28. Users would select a point on the grid, with the point's position on the Y-axis representing their desired mechanical strength for the print, and the point's position on the X axis representing their desired finish quality for the print.

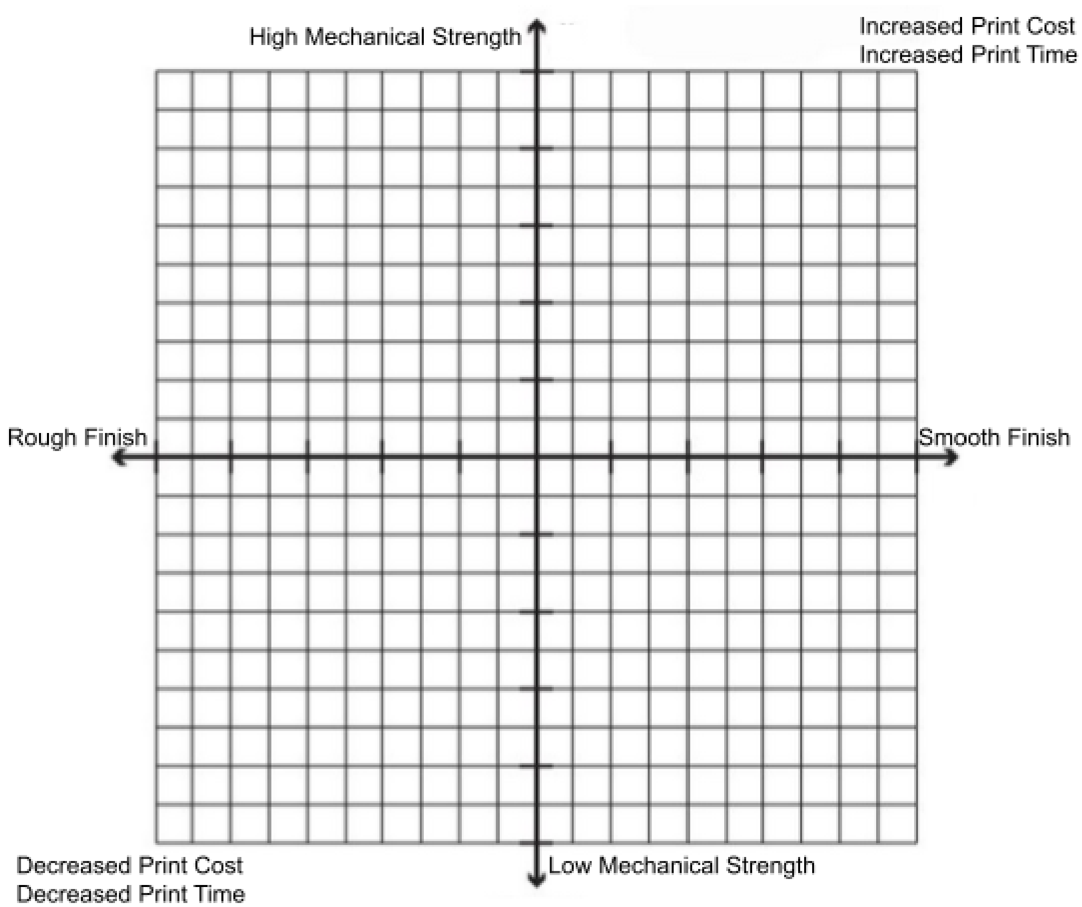


Figure 5.28: Grid input method

However, this design lacks the complexity necessary to differentiate between traits such as tensile strength vs. compressive strength. Additionally, the user would have to choose a point on the plot with little reference to how the position within different quadrants would affect the outcome.

Another drafted design was to have the user prioritize their requirements from a list of user requirements. The user would be able to move the items and indicate their preference from the list shown in Figure 5.29.

- ☐ Dimensional Tolerance
- ☐ Exterior Finish
- ☐ General Mechanical Strength
- ☐ Tensile Strength
- ☐ Compressive Strength

Figure 5.29: Ranking input method

While this improves on the previous design by adding the complexity necessary to differentiate between the user requirements, this still does not account for cases such as the gears or flexible prints. Additionally, it would still require the user to make decisions about properties they might not understand.

Survey questions were selected as they provide the complexity needed to determine all the settings. The questions also limit the user's decision making as they provide concrete answers from which the program can determine the user requirements, rather than the user having to pick blindly. The questions in Appendix B were selected to include an answer to all the If/Then statements that were previously discussed. Further, the number of questions was minimized so as to not dissuade potential users. Figure 5.30 shows how the questions are arranged and how different answers can lead to different questions.

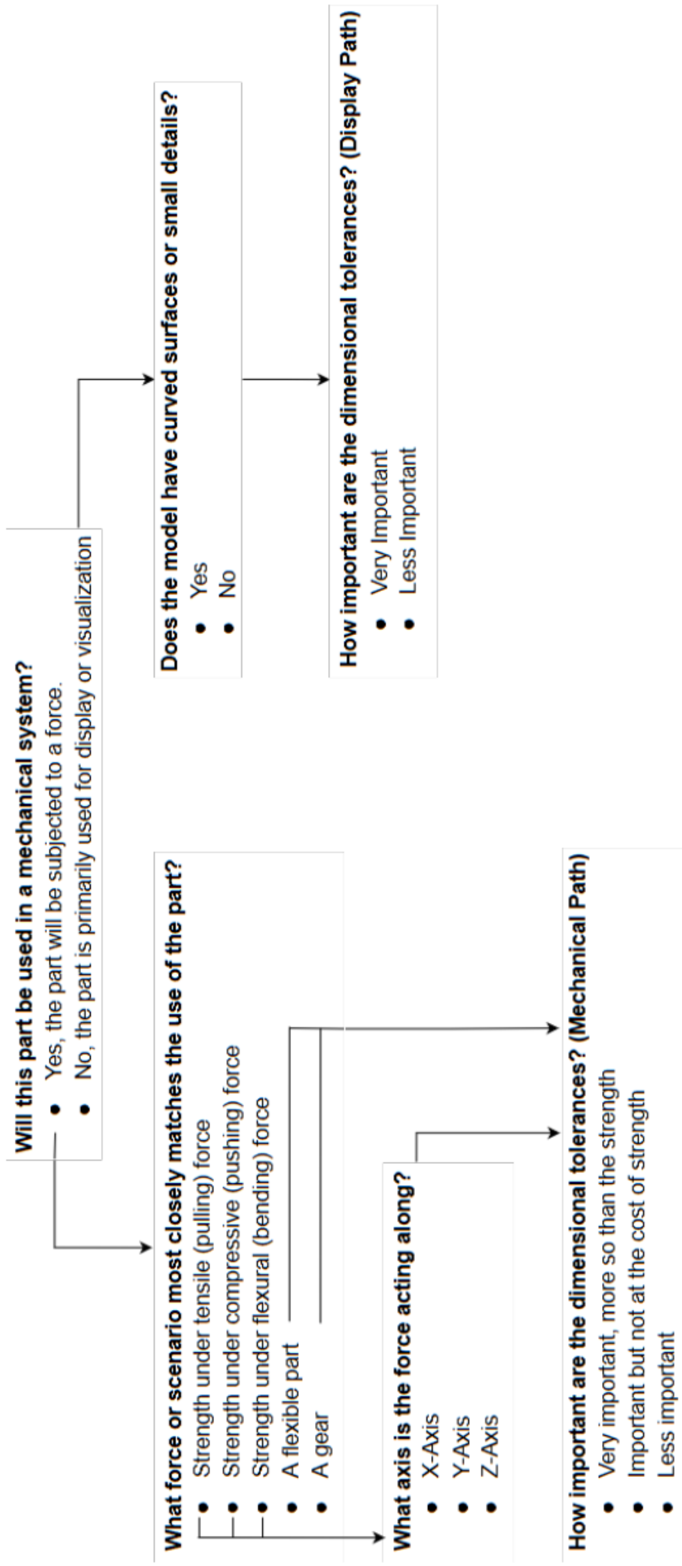


Figure 5.30: Question tree arrangement

In order to generate the model's orientation for printing, the program goes through three procedures: generating a list of all valid orientations, scoring each orientation, and outputting the .stl and G-Code files.

5.16. Generating a List of all Valid Orientations

In order to generate a list of all valid orientations, the program parses through the .stl file provided by the user. Afterwards, it iterates through every face on the model and determines what plane it lies on by determining the values in a point-normal form equation of the plane ($ax + by + cz + d = 0$). Pseudocode for this process is shown below.

```
A face is made up of points p1 p2 and p3

v1 = p2 - p1
v2 = p3 - p1
cross = cross product of v1 and v2
normalize c
If cross.x does not equal 0
{
        Times cross by -1 if cross.x < 0
}
Else if cross.y does not equal 0
{
        Times cross by -1 if cross.y < 0
}
Else if cross.z does not equal 0
{
        Times cross by -1 if cross.z < 0
}
plane.a = cross.x
plane.b = cross.y
plane.c = cross.z
plane.d = -(a*p1.x + b*p1.y + c*p1.z)
return plane
```

This ensures that if two faces lie on the same plane, they will have the same a, b, c, and d values.

The program then adds all these planes to a list while removing any duplicates. Once all the planes are determined, the program needs to create a list of all possible orientations that it can evaluate. To do this, the program creates duplicates of the information parsed from the .stl file for each plane, this information is stored in a list of objects that store all the information of the stl file. It then rotates all the points to make it so each model's corresponding plane will be equal to the x-y plane. Meaning that when printing the model, its corresponding plane will lie flush with the printer bed. The program will then go through each of the points in each orientation, if any of them have a z value less than 0 then the model is not valid and removed from the list. This is because if any points have a z value less than 0, the model would start below the printing bed which is impossible.

5.17. Scoring Each Orientation

Now that the program has a list of valid orientations, it scores them. Orientations are scored on six criteria: surface area contact with the printer bed, overhang, slenderness, tensile strength (if applicable), compressive strength (if applicable), and flexural strength (if applicable). Each score is on a scale from 0 to 10. If any of the equations below lead to a score greater than 10, the score will be set to 10. An example of the scoring process on the example model shown in Figure 5.31 is provided below.

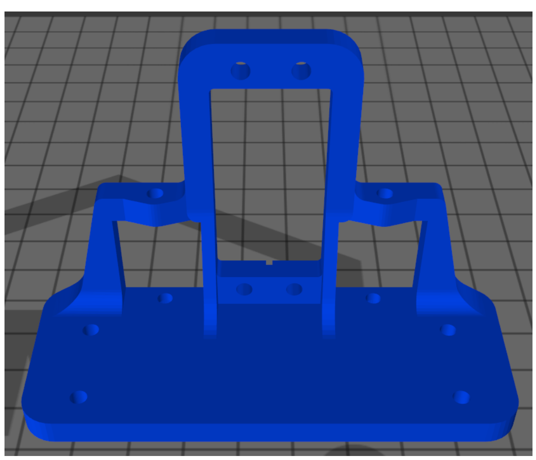


Figure 5.31: Example .stl file

During the survey questions, the user states that they want to prioritize tensile strength along the Z-axis (the axis directed out of the bed).

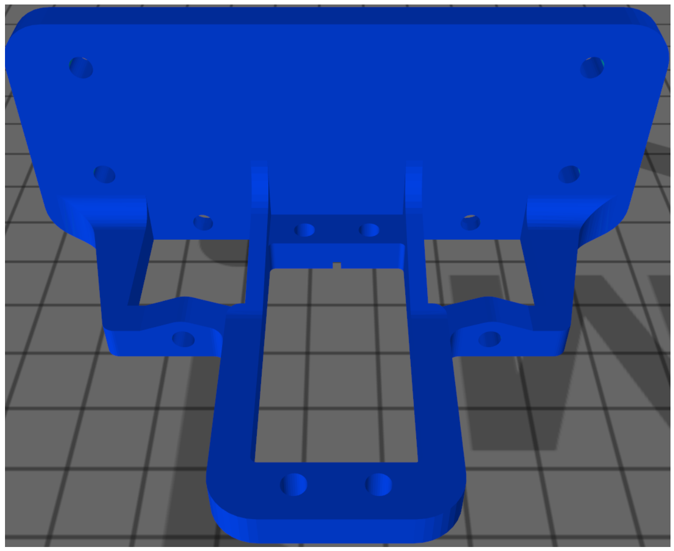


Figure 5.32: Example orientation of the model

The program will generate an orientation similar to the one provided in Figure 5.32, and will then proceed to score it. This orientation has more surface area in contact with the bed compared to most other orientations, so it will score fairly high in terms of surface area in contact with the printer bed. This score is calculated by dividing the surface area making contact with the printer bed that this orientation has by the largest amount of surface area making contact with the printer bed out of all the orientations, as shown in equation (1).

$$Surface\ Area\ Score = 10 * (ThisSurfaceArea/MaxSurfaceAre \quad (1)$$

Figures 5.33 and 5.34 show examples of the orientation with the largest amount of surface area making contact with the printer bed and the orientation with the smallest amount of surface area making contact with the bed, respectively. Figure 5.33 scores 10 for surface area score. Figure 5.34 scores 0.0173 for surface area.

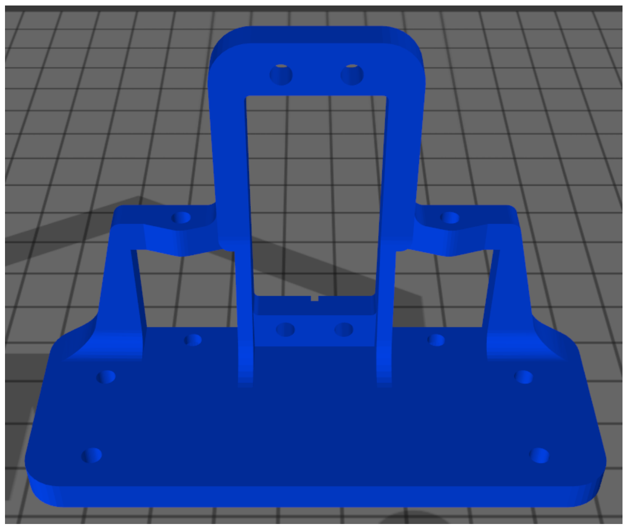


Figure 5.33: Maximum surface area

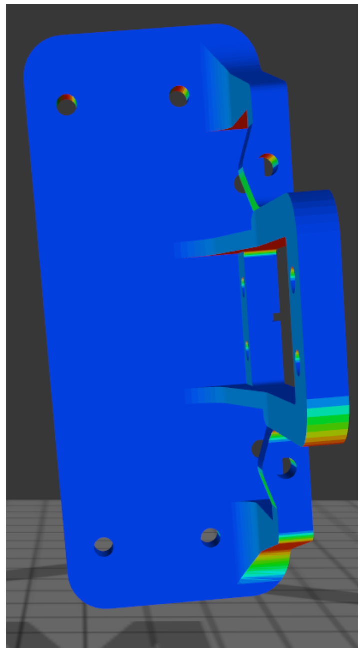


Figure 5.34: Minimum surface area

The next value is overhang. The overhang scoring process is to take every face on the model and find out it's area and normal vector. Then using that information, calculate the percentage of surface area that has a normal vector within 45 degrees of the vector [0, 0, -1]. The greater the overhang, the greater this percentage will be. Invert this percentage by subtracting 1 from it, then this percentage is raised to the 5th power to spread out the values more. For example, if before applying the 5th power you have scores 0.99 and 0.97. Their scores after applying the 5th power will be 0.95 and 0.85 which is a lot more distinct from each other which is good because the overhang score is usually very high.

The full equation is shown in (2). Shown below in Figures 5.35 and 5.36 are two examples with high and low overhang scores. It is important to note that a high score for overhang does not mean that there is a lot of overhang. Instead, it implies that it is going to score high overall due to the small amounts of overhang. High scores contribute positively in the overall orientation score where low scores contribute negatively to the overall orientation score. Figure 5.35 has an overhang score of 4.028 (Total overhang area = 2151.849. Total Area = 12939.762), and Figure 5.36 has an overhang score of 9.286 (Total overhang area = 190.217. Total Area = 12939.762).

$$Overhang\ Score\ = 10 * (1 - (TotalOverhangArea/TotalArea)) \tag{2}$$

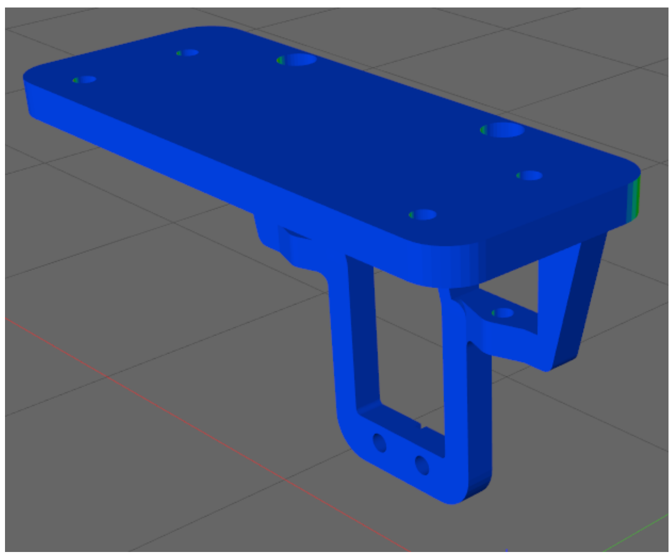


Figure 5.35: Low Overhang Score (Meaning Lots of Overhang)

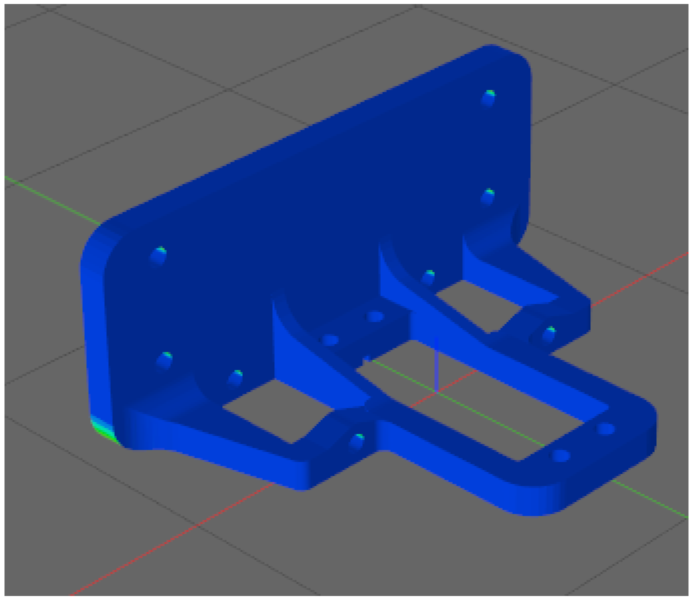


Figure 5.36: High Overhang Score (Meaning Little Overhang)

The next score to be evaluated is the slenderness score. The slenderness score is calculated by dividing the model into 8 pieces by its height. Then the program finds the width and depth of each of these pieces by inscribing them in a right rectangular prism (3D rectangle). Afterwards the program calculates the average width/depth of the prism and divides that average by the height, as shown in (3). This is done in order to prevent models from being printed tall and 'skinny'. Below is an example of how slenderness would be scored. This one uses a different model in order to better show what we are looking for. Similar to overhang, a high slenderness score does not imply there is a lot of slenderness. Instead it means that there is a small amount of slenderness so the overall score will be higher. The model in Figure 5.37 has a slenderness score of 0.152 (Average width = 0.494. Height = 9.741). Figure 5.38 has a slenderness score of 8.643 (Average width = 5.702. Height = 1.975).

$$Slenderness\ Score\ = 3 * (AverageWidth/modelHeight) \quad (3)$$

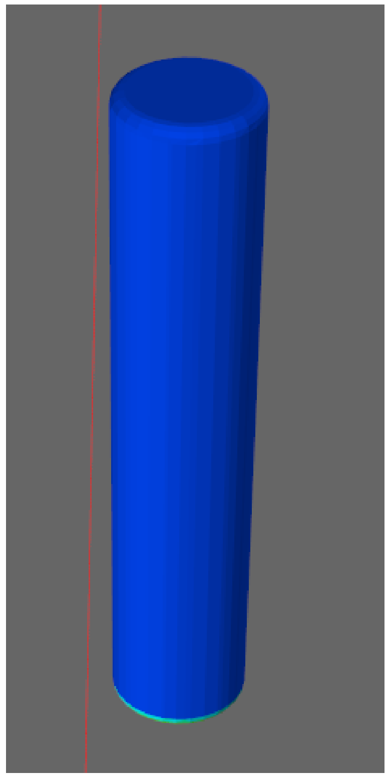


Figure 5.37: Example of low slenderness score (Meaning high amounts of slenderness)

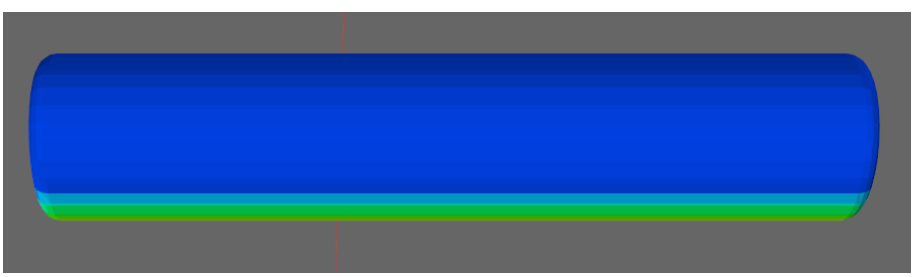


Figure 5.38: Example of High Slenderness Score (Meaning low amounts of slenderness)

Because the user indicated that tensile strength was a priority, the program will score tensile strength next. According to research, tensile strength is maximized when a model fulfills two conditions. The first is when the axis of force is parallel to the bed, and the second is when the model is lying on an edge. This is shown in (4), where *n* is the dot product of the applied direction of tensile force to the Z-axis (meaning the more parallel the force is to the printer bed the lower n is), R1 is the longer dimension of a rectangle circumscribed around the model that is parallel to the bed, and R2 is the shorter dimension of the same rectangle. Figure 38 is an example of R1 and R2.

$$Tensile\ Score = 5 * (1 - n) * R1/R2 \qquad (4)$$

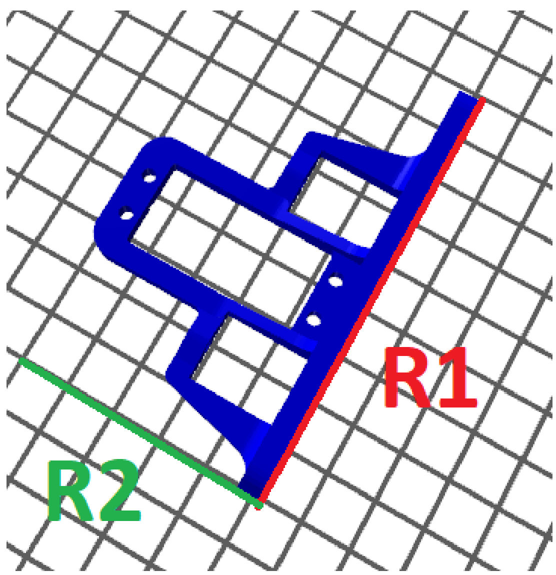


Figure 5.39: R1 and R2 example

Next to be scored is compressive strength. In this example, the user determined not to have this setting as a priority, so it will not be scored. However, if it needs to be scored, compressive strength is maximized by having the axis of force as parallel to the bed as possible (just as for tensile strength). This calculation is shown in (5); where *n* is the same as it was in tensile force. Meaning the more parallel it is to the printer bed the lower it is.

$$Compressive\ Score\ =\ 10 * (1 - n) \tag{5}$$

The last force score to be scored is the flexural score. Just as for the compressive score, the user determined that this is not a priority so it will not be scored. However, if it needs to be scored, it is scored identically to the tensile score, as shown in (6); where n is the same as it was for tensile force. Meaning the more parallel the force vector is to the printer bed the lower *n* is. R1 and R2 are also the same as in the tensile score

$$Flexural\ Score\ =\ 5 * (1 - n)\ *\ R1/R2 \tag{6}$$

To get the final score, the program multiplies each score by a weight and then adds them all together, shown in (7). These weights were determined through trial and error until all test models produced what we considered was the ideal orientation. Our goal for the weights was to make the force score just as important as all the other scores combined. This is because the force score is optional and if a user is prioritizing a force it probably means that is the most important aspect of the piece for them. More details on how these weights were determined can be found in section 5.19.

$$\textit{Total Score} = \textit{Surface Area Score} + 2*\textit{Overhang Score} + 3*\textit{Slenderness Score} + 5.5*\textit{Force Score (if applicable)} \quad (7)$$

Below is pseudocode that describes the entire orientation selection process:

```
Create list of planes
Create map of planes to surface area
For(every face f on the model)
{
	Determine what plane face f belongs to.
	Determine surface area of face f
	if(plane exists in list)
	{
		Add surface area to the map
	}
	Else
	{
		Add plane to list
		Create a map from current plane to surface area value
	}
}
List of Models;
For(every plane p in list of planes)
{
	Create a new model identical to the first model;
	Orientate this model so plane p is equal to the x-y plane
	if(no points on model are below z=0)
	{
		Score the model using Equation #7;
		Place the model in the list of models ordered by its score;
	}
}
```

5.18. Selecting a Printer and Applying Settings

The printer used for G-Code generation is selected from one of the printers that GaPA has saved. Specifications about the printers are used to ensure that the printer is compatible with the user's needs.

Selection is based on three parameters: the bed size, the material, and the tolerance. Data about the size of the model being printed is received from the .stl file, so GaPA ensures that the bed size is able to fit the part with any given orientation. The material is determined through the questionnaire section, using the user's determined preferences to select the most optimal material. Then, a printer with the material is selected. Finally, the tolerances of the printer are taken into account through the questionnaire, using the user's answer to the question about the importance of the dimensional tolerances.

The answer to the question about dimensional tolerances also affects the priority that these parameters have. Normally, bed size is given the highest priority when determining the most optimal printer. However, if the user answers that strength is a greater priority than tolerance, the material selection is taken into account first. The opposite holds true for prioritizing the tolerance. When the printer is selected, the specifications of the printer are entered into the slicer to generate G-Code appropriate for that printer.

5.19. Exporting .stl and G-Code files

Once the best 4 solutions have been determined (i.e. the highest scores), the program will save these orientations to .stl files. These .stl files are not meant to be accessed by the user and are only meant to be used by the program internally. Then the program will use Matterslice to generate the G-Code files. The program has Matterslice's open source code directly in its code base and uses that to generate G-Code. The G-Code generation is affected by the answers to the questions in Mode 2 (see Appendix B) as well as by the printer that was selected by the program (see section 5.18). Finally, the program will send the signal to the UI so that it can load the solution selection screen, and populate it with images of the chosen orientations. On this screen, the user will find the pros and cons for each orientation. These pros and cons are generated by taking the scores that were generated and splitting them up into 5 levels: very good, good, neutral, bad, and very bad. Each level has a different range, shown in Table 5.3. These values were calculated by using the test models found in Table 5.4, trying to match the pros and cons of each orientation with what we would expect.

Table 5.3: Pro and Con score ranges

Level	Very Good	Good	Neutral	Bad	Very Bad
Surface Area	8-10	6-8	5-6	2-5	0-2
Overhang	9.5-10	9-9.5	7-9	5-7	0-5
Slenderness	9-10	7-9	4-7	2-4	0-2
Force	9-10	7-9	6-7	5-6	0-5

5.20. Scoring Evaluation

Our scoring system has gone through quite a few changes over development, mostly changing the weights, changing how overhang is calculated, and the addition of the slenderness score. The slenderness score was suggested during interviews with experts and subsequently added. In our original system, overhang was calculated by inscribing the base of the model in a rectangle and dividing that area by the area of a rectangle surrounding the entire model. We determined this method of calculation was fundamentally flawed due to it being more of an estimate that was vulnerable to edge cases with a lot of curves.

Our weighting for each of the scores also went through some changes during development. Originally we had surface area, overhang, and slenderness all weighted as one and the force score weighted as three. This was because we thought that if a user was going to specify a force, it should be considered important enough to be the primary deciding factor for orientation. Over time, we discovered that in order to consistently get the orientations that we needed, we needed to weight slenderness and overhang higher than surface area. In Table 5.4, we go over the models that were used during evaluation and what we thought their ideal orientations should be going into testing. In Table 5.5, we go over the final scores and results we got for each of the models.

Table 5.4: Models used in testing

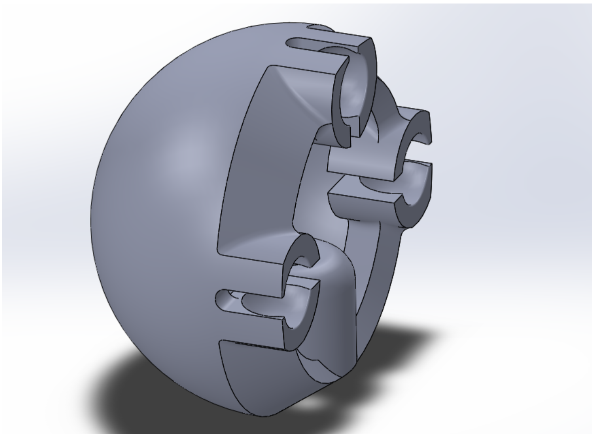
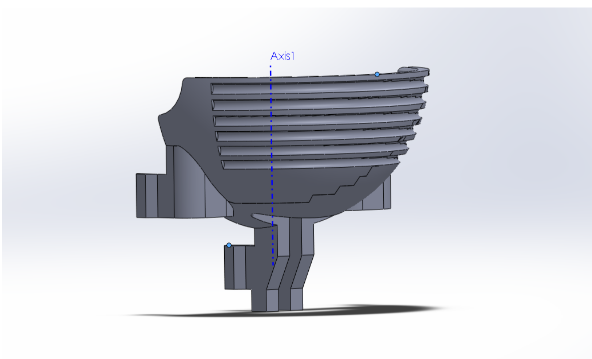
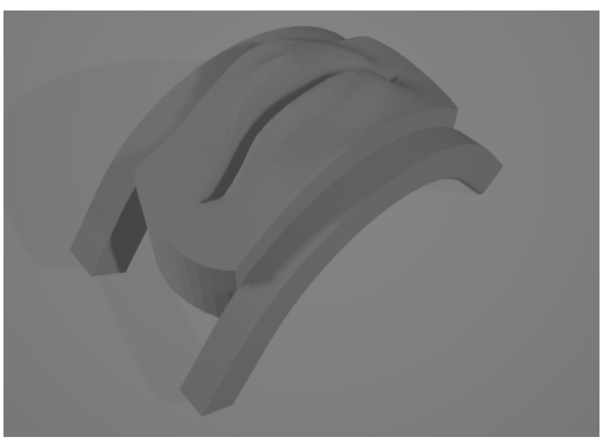
	This part provides a curved surface as well as minimal surface area that can contact the bed. Additionally, it's hollow interior will require supports so the program will be tasked with minimizing the supports.
	A smaller piece with fine details. The smaller parts of this model, especially the ridges will provide an easy visualization of the resolution of the printer. The part also has little surface area that can contact the bed at any one time and regardless of the orientation, supports will be required.
	Provides a unique surface with many polygons. The gap between the lips seen in Figure 3 will also provide visual representation for tolerances. With the upper and lower brace pieces, there will also be interior support generated for the model.

Table 5.4: Models used in testing (cont.)

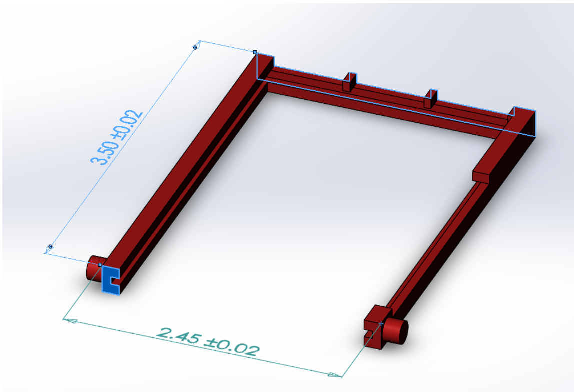
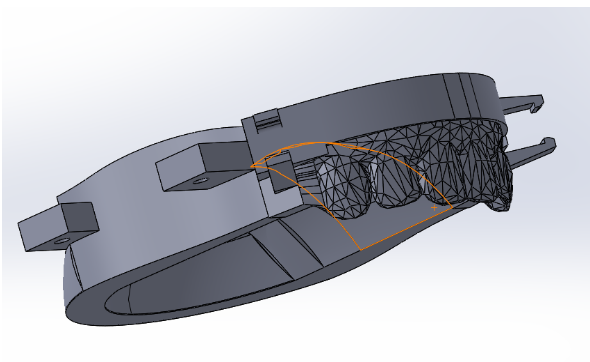
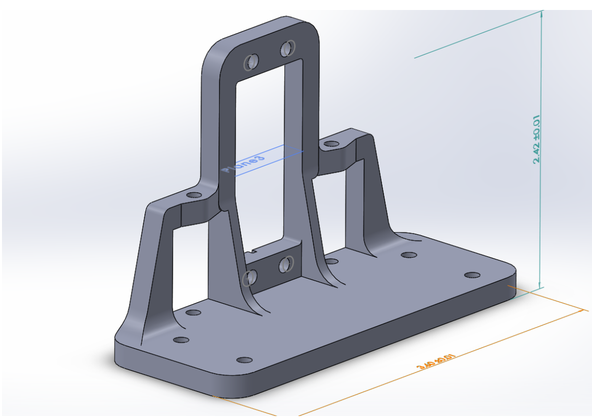
	A simpler piece for orientation, this part provides many sharp angles and straight measurements that make it useful for tolerance measurements.
	The thin protrusions around the outside edge of this model as seen in Figure 5 below, as well as the teeth themselves, complicate the orientation of this model. Furthermore it has many curved surfaces on both the interior and exterior that will require supports and will be a useful visual demonstration of support angle due to the gradual curve.
	This part will require internal supports when printed and has more straightforward geometry than the rest of the pieces. The slight curves in the surfaces also aid in visualization of the part's resolution as a result of layer height.

Table 5.4: Models used in testing (cont.)

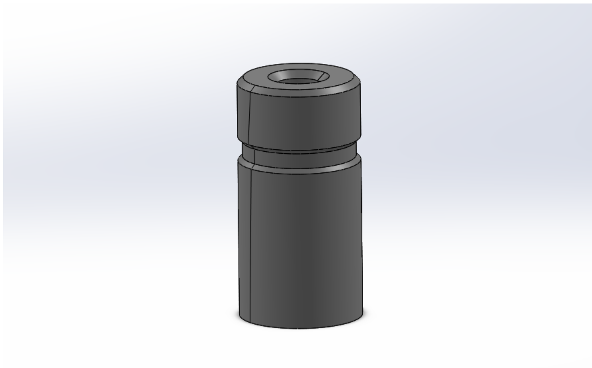
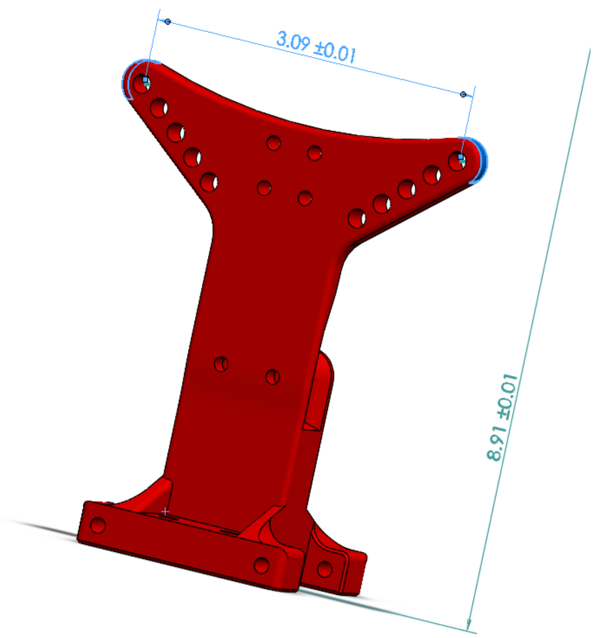
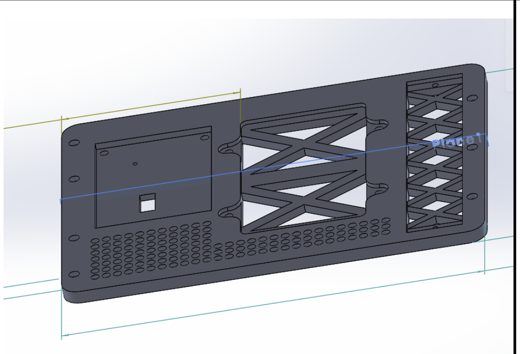
	The smallest test piece, with a 4 mm diameter and an 8 mm height. The hollow interior forces the program to contend with interior overhang and due to the cylindrical shape it should be on it's side when printed. However, since the ends are flattened and have more surface area, this model can be used to ensure the scoring system balances the surface area scoring and the overhang scoring against the slenderness/height ratio.
	A tall and thin piece that will require supports no matter how it is oriented. This piece can be used to test how our program handles pieces with a high height to slenderness ratio that would likely collapse during printing if upright.
	A large model with the highest surface area and polygon count of the test cases. While the orientation is fairly simple due to its basic shape, the many holes create far more complexity and polygons that the program must check. This results in the longest runtime of any of the test cases despite the seeming simplicity of orientation

Table 5.4: Models used in testing (cont.)

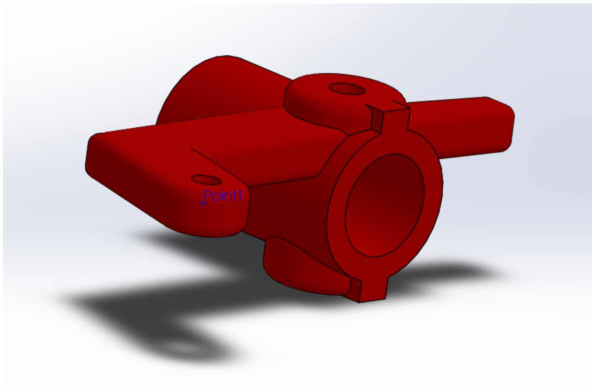
	This piece has no straightforward orientation due to the extrusions on all sides of a cylinder. No matter which way the part is oriented there will need to be supports and there will be little bed contact. Additionally, because of the offset of the extrusion from the center of the cylinder as seen in Figure 10, the program could save support material by orienting with the extrusion closer to the bed.

Table 5.5 includes orientation tests through GaPA of the selected parts described previously. The table includes the name of the part as well as the answers given to each question, with the numbers indicating which option was chosen for each subsequent question. The second column is an image of the first ranked orientation from GaPA and then subsequent columns are the pros and cons for that orientation as well as the scoring breakdown. Following the table, there is a brief explanation of why this orientation meets the requirements of the part.

Table 5.5: Orientation tests

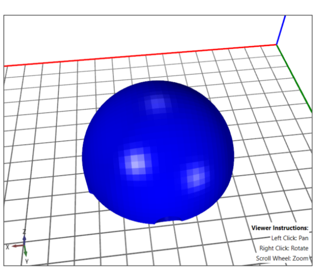
Part and Answers	First Ranked Orientation	Pros and Cons	Scoring Breakdown
Eyeball 2, 1, 1		Pros: - High amount of surface area with the printer bed - Quick print time Cons: - Fair amount of overhang - Fairly slender - High amount of material use	Print Time: 30.05 mins Material Cost: 1245 mm Surface Area Score (out of 10): 9.771 Overhang Score (out of 10): 5.254 Slenderness Score (out of 10): 3.675 Force Score (out of 10): N/A

Table 5.5: Orientation tests (cont.)

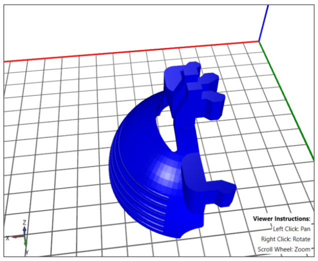
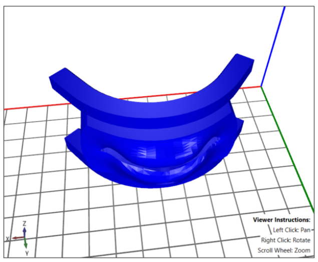
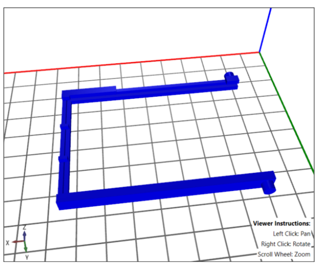
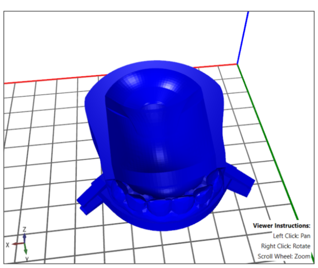
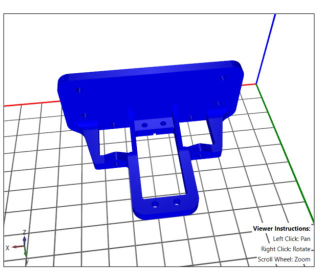
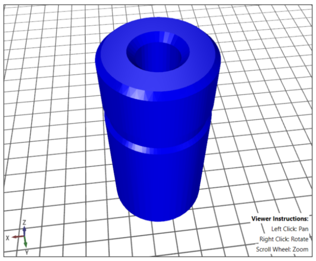
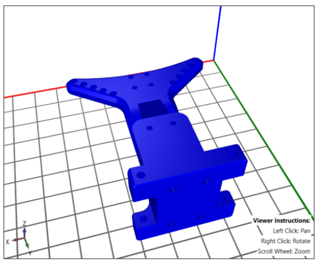
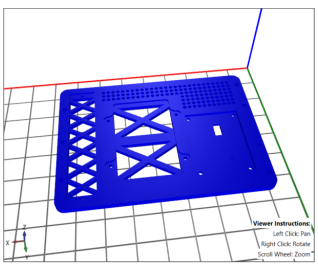
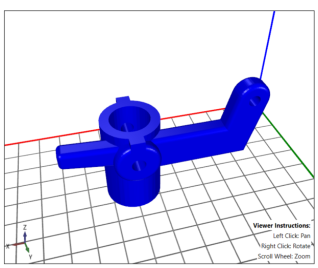
Eyelid 2, 1, 1		Pros: - Low amount of material use - Quick print time Cons: - Low amount of surface area with bed compared with other orientations - Fair amount of overhang - Very slender	Print Time: 18.533 mins Material Cost: 822 mm Surface Area Score (out of 10): 4.621 Overhang Score (out of 10): 5.334 Slenderness Score (out of 10): 1.984 Force Score (out of 10): N/A
Lips 2, 2, 2		Pros: - Low amount of material use - Quick print time Cons: - Low amount of surface area with bed compared with other orientations - Fair amount of overhang - Fairly slender	Print Time: 78.1 mins Material Cost: 9458 mm Surface Area Score (out of 10): 5.017 Overhang Score (out of 10): 5.848 Slenderness Score (out of 10): 3.777 Force Score (out of 10): N/A
Middle Bracket 1, 1, (0,0,1), 1		Pros: - High amount of surface area with the printer bed - Low amount of overhang - Not slender - Low amount of material use - Quick print time Cons: - Low tensile strength along requested axis	Print Time: 37.85 mins Material Cost: 2813 mm Surface Area Score (out of 10): 10 Overhang Score (out of 10): 9.391 Slenderness Score (out of 10): 10 Force Score (out of 10): 6.364
Mouth and Teeth 2, 1, 2		Pros: - Not slender - Low amount of material use - Quick print time Cons: - Very small amount of surface area with bed - High amount of overhang	Print Time: 273.35 mins Material Cost: 14691 mm Surface Area Score (out of 10): 0.898 Overhang Score (out of 10): 2.084 Slenderness Score (out of 10): 9.671 Force Score (out of 10): N/A
Servo Mount 1, 2, (0,1,0), 2		Pros: - Low amount of overhang - High compressive strength along requested axis - Low amount of material use - Quick print time Cons: - Low amount of surface area with bed compared with other orientations - Fairly slender	Print Time: 150.217 mins Material Cost: 9400 mm Surface Area Score (out of 10): 4.811 Overhang Score (out of 10): 9.286 Slenderness Score (out of 10): 3.837 Force Score (out of 10): 10

Table 5.5: Orientation tests (cont.)

Servo Shaft 2, 1, 1		Pros: - High amount of surface area with the printer bed - Little to no overhang - Low amount of material use Cons: - Very slender	Print Time: 1.483 mins Material Cost: 102 mm Surface Area Score (out of 10): 10 Overhang Score (out of 10): 9.69 Slenderness Score (out of 10): 0.445 Force Score (out of 10): N/A
Suspension Riser 2, 1, 2		Pros: Cons: - Very small amount of surface area with bed - High amount of overhang - Fairly slender - High amount of material use - Slow print time	Print Time: 133.267 mins Material Cost: 18932 mm Surface Area Score (out of 10): 1.846 Overhang Score (out of 10): 4.722 Slenderness Score (out of 10): 3.514 Force Score (out of 10): N/A
Top Plate 2, 1, 1		Pros: - High amount of surface area with the printer bed - Little to no overhang - Not slender - Low amount of material use - Quick print time Cons:	Print Time: 1100.7 mins Material Cost: 34166 mm Surface Area Score (out of 10): 10 Overhang Score (out of 10): 10 Slenderness Score (out of 10): 10 Force Score (out of 10): N/A
Wheel Coupler 2, 1, 1		Pros: - Low amount of material use - Quick print time Cons: - Low amount of surface area with bed compared with other orientations - Very slender	Print Time: 54.017 mins Material Cost: 2225 mm Surface Area Score (out of 10): 3.64 Overhang Score (out of 10): 7.537 Slenderness Score (out of 10): 1.396 Force Score (out of 10): N/A

- **Eyeball:** Any orientation that this .stl has will require supports. This orientation limits the support to be on the underside of the print, providing a smooth exterior.
- **Eyelid:** With little surface area available for the printer bed, this orientation sits on the bed with a wide edge. This orientation also provides a smooth exterior on the exterior part. The downside of this orientation is that the clips are being

printed vertically which could negatively affect their strength. However, since this procedure did not consider forces, this is unsurprising.

- **Lips:** This orientation maximizes the surface area in contact with the bed. Additionally, support is only required on the interior of the part.
- **Middle Bracket:** This orientation keeps the tensile force direction along the xy-plane meeting the force standard. In addition, the orientation maximizes the surface area on the bed.
- **Mouth and Teeth:** This orientation maximizes the surface area on the bed through the flat backside of the mount. This additionally minimizes supports as they are not needed for the roof of the mouth.
- **Servo Mount:** This orientation places the force vector in the xy-plane to satisfy that requirement. While this orientation has less surface area on the bed, this is counteracted by limiting the support that is needed on the interior of the part since this would be needed in the gaps of the part.
- **Servo Shaft:** The orientation maximizes the surface area on the bed while also preventing the need for support. The downside of this orientation is the very low slenderness score. However, due to the low height of the part, the slenderness will not hurt the print.
- **Suspension Riser:** The orientation given is a bit unique as there are no listed pros. The slenderness of this part adds further difficulty for orienting this piece. Maximizing the surface area and limiting the supports would result in a very tall and thin orientation. As such this orientation balances those parameters to generate this orientation.

- **Top Plate:** This orientation has the maximum score for all of the parameters. The orientation has lots of contact with the bed, has no overhang, and has a very low slenderness ratio.
- **Wheel Coupler:** This orientation minimizes the support needed for the print. There is less contact with the bed and it is more slender though. Notably, orientation 2 changes the balance, prioritizing low slenderness and increasing the surface area on the bed.

6. Testing Procedure

6.1. Software Testing

In addition to testing GaPA with students from Mechanical Engineering courses at WPI, some software testing was done before then to filter out bugs. Due to time constraints, we were not able to do as much software testing as we would have hoped. With this stated, we have tested our software while keeping in mind the questions Whittaker proposes to evaluate our progress (Whittaker, 2000).

- "Have I tested for common programming errors?"
    - Our testing for common errors led to several bug discoveries we would have missed otherwise, and were able to fix. This includes importing multiple .stl files leading to a crash, importing invalid .stl files, and invalid answers being stored if a user fills out a questionnaire and then goes back and changes their answers.
- "Have I exercised all of the source code?"
    - While the IDE we are using does not let us check code coverage in test cases, we were careful to test code as we developed it. We understand that this is not a full substitute for full code coverage.
- "Have I forced all the internal data to be initialized and used?"
    - Yes. All variables for each class are properly initialized during the creation of that class. Most classes are also initialized on program startup.

- “Have I found all seeded errors?” (seeded errors are deliberate errors put into the code, this is done because the act of exploiting them could lead to finding unintended errors)
    - Yes. Seeded errors will appear if a user inputs an .stl file with improper formatting. This has led to no other errors being discovered.
- “Have I thought through the ways in which the software can fail and selected tests that show it doesn’t?”
    - Yes, but not to the fullest extent. Most of the tests in this area involve trying to confuse the program by giving it invalid inputs or moving back and forth between modes to test the program’s capability to handle those transitions.
- “Have I applied all the inputs?”
    - We have not tried all the inputs that the program’s internal functions could theoretically allow, but we have tried all the inputs that a user would have access to through the UI.
- “Have I completely explored the state space of the software?” (The state space is the name given to a combination of all possible configurations of a system (Math Insight, 2020))
    - No. The only computers the program is known to run on are the computers of team members and study participants. This is not a

complete view of the state space which should include all windows computers.

- “Have I run all the scenarios that I expect a user to execute?
    - Yes. Several tests have run through the program, replicating every activity we expect a user to do. When testing with users performing expected tasks, our study reported very few crashes. Those few crashes that were reported have been fixed.

Our software testing overall is somewhat lacking and is a definite area of improvement for future developers to work on. The key areas that could be improved are including more test cases, measuring code coverage, and testing more seeded errors. Despite this, we believe user testing, edge case testing, testing common programming errors, and applying all inputs were accomplished sufficiently. Therefore, despite some shortcomings in testing, we are confident that the program works as intended, and whatever bugs remain (if they exist at all) will be obscure and hard to find.

6.2. User and UI Testing

In order to test GaPA and its UI properly, we decided to set up a study that would gauge how users react to our program and find out what needs to be changed. This study consisted of users downloading GaPA, supplying their own .stl files to test with (if they did not have an .stl file, three sample files were provided). They then accessed a Google Forms webpage that explained how to use the program and asked the user questions about their skills and experience.

Using the completed G-Code and the participants' answers to the survey questions, we checked the result of the program's orientation against manual orientation of the same object. Furthermore, follow-up questions helped us determine where the user interface of our program falls short and where it succeeds, allowing us to make informed adjustments.

Our team sent out a handout to individual WPI classes that involve Mechanical Engineering students designing CAD models that could be used for printing. These classes align with the target user base of this program. These classes were ME4320, Advanced Engineering Design, and ME2300, Intro to Engineering Design. The handouts included a download link to our program and the previously-mentioned Google Form that asked them questions about their experience.

The following sections explain the contents of the form in detail.

6.3. Introduction and Consent Form

The form begins by giving the user the consent information for the study, and providing a brief description of what the test includes. After consenting (by marking a checkbox), the participant was given information about the .stl files and given preliminary questions. The consent form is based on standard IRB Consent Information and is included as Appendix C.

6.4. Background Information

Users were first asked the following question:

Do you own STL files that you would like to test with or can you convert a Solidworks component to one (File > Save As >STL File)?

- Yes
- No

If the user selected "No", they were taken to a link where they can download sample .stl files. The participant was then asked preliminary questions to determine their experience with 3D printing. This was also used to better contextualize later questions when the participant was asked to compare our program with standard slicing programs.

What slicing programs have you used for 3D printing?

- I have never used a slicing program for 3D printing
- Cura
- MatterControl
- Slic3r
- Other _________

How experienced are you with slicing an .stl file for 3D printing?

- Not at all experienced
- Moderately experienced
- Very experienced

Asking for the participant's experience level helped us determine whether users of different experience levels react differently to our program. The question about specific slicer programs allowed us to see which programs our participants will be comparing to ours.

6.5. Testing Sequence

In this section, the user was given instructions on how to complete the test. They were first asked to upload the .stl file they were going to use so that our team had access to it (for analysis purposes). Afterwards, instructions for running the .exe file were given.

The user was then asked to import their .stl file into the program, and to complete the slicing process. When the part was sliced, the user was asked to save and upload the G-Code that was generated.

Completing this task brought the user through the major parts of our program and gave them the chance to interact with it under circumstances similar to real usage. Additionally, the questions that followed allowed us to see how well users interacted with the program, and provide information that allowed us to determine where users may have encountered problems in the program.

6.6. Follow-Up Questions

Since we did not expect all the participants to be able to tell if the resulting G-Code met their expectations, the follow-up questions focused on the usability of the program and user interface:

Was the slicing process using this program faster or slower than you expected?

- Faster than expected
- About the same as expected
- Slower than expected

Please indicate the duration (in seconds or minutes) it took to complete each step: 'Answering Questions' about the file being 3D Printed and the software 'Generating G-Code'

Do you think that the first orientation the program presented would be optimal for your part?

If you think another orientation would have been optimal, please describe the orientation you would have preferred and why.

What did you find the most difficult in using this program?

What aspect of the program did you find easiest to use?

What did you like about this program?

What did you dislike about this program?

Were the questions that the program asked you clear/easy to understand? If not, which questions were unclear and why?

Did the program include enough information to help you understand the slicing process?

Were there any errors that interfered with the use of the program in a way not covered by the previous questions?

Most of these questions were open-response and were intended to get more information about the user's experience with the program. The questions asking where the participant faced difficulty was used to evaluate the effectiveness of the program's user interface. Parts that participants find difficult can be improved, potentially using stylistic choices from sections that participants found easy to use. The like and dislike questions were intended to track engagement with the interface while the final two questions were specifically targeted at the language used in our survey questions and the help functionality of our program.

6.7. Concluding the Test

This final section included questions for the participant regarding their opinion of the program. In comparison with the previous follow up questions, these questions were more focused on possible future improvements for the program, getting overall opinions of the program as well as future recommendations. The questions included to this end are as follows:

> Please rate your satisfaction with the program on a scale from 1 to 10 with 1 being the lowest while 10 is the highest.
> Why did you select this number?
>
> How does this program compare to other slicing tools you may have used?
>
> Do you have any recommendations for features that could be added to or improved about the program?

These questions will be useful for future improvements to the program, identifying benefits over other programs as well as features we may have overlooked. We conclude by thanking the participant for their time and participation in the study.

7. Testing Results and Analysis

In order to get a better idea of the strengths and shortcomings of our program we needed to analyze the results of the study to see what people liked and disliked. Given all of the survey questions, we isolated the results with the most pertinent responses and created graphs to better understand the shortcomings of our program.

7.1. User Experience

The first major question that provides relevant results is how users would describe their level of experience with 3D printing. Figure 7.1 describes the distribution of how users would rate their experience with slicing software.

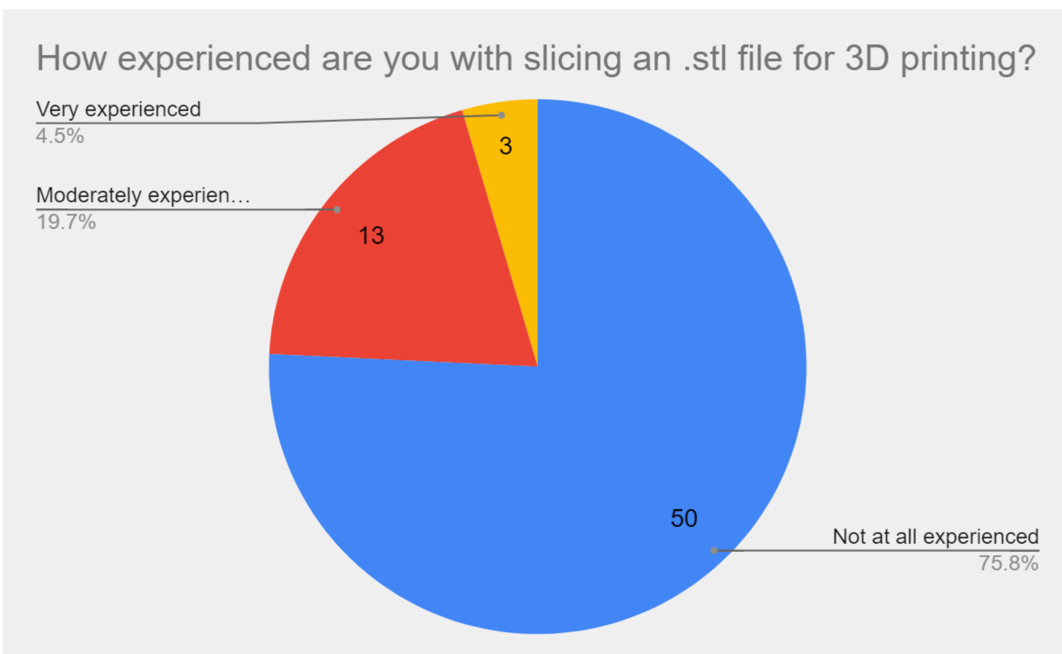


Figure 7.1: 3D Printing experience of test users

Because this graph seems to indicate that the vast majority of ME students claim to be not at all experienced with 3D printing, this further reinforces the idea that our program must be as easy to use as possible.

7.2. Program Speed Expectations

For the next couple of questions, we split each graph up by level of experience that we obtained from the question above. This following question is intended to gauge how quickly our program runs vs. user expectations. Figures 7.2, 7.3, and 7.4 show how users with different experience levels answered this question.

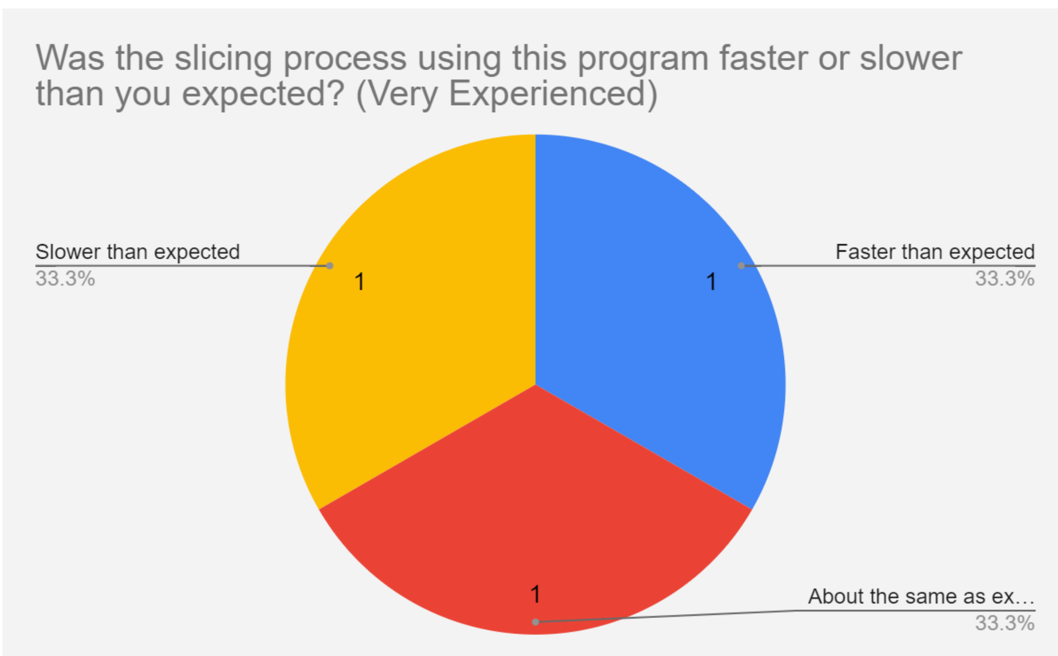


Figure 7.2: Very experienced users' speed evaluation

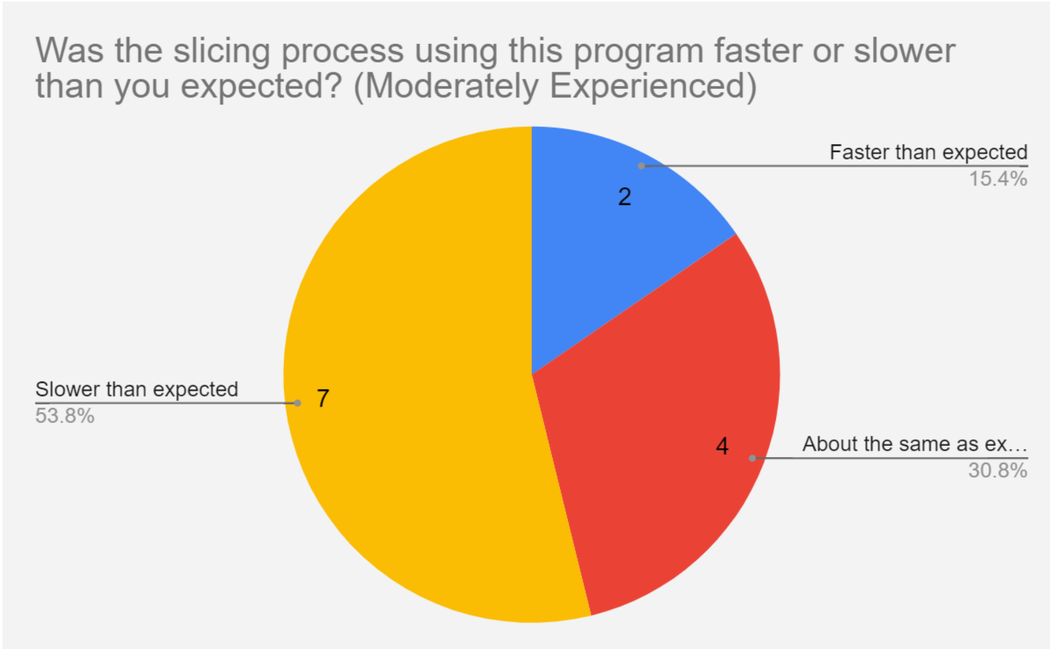


Figure 7.3: Moderately experienced users' speed evaluation

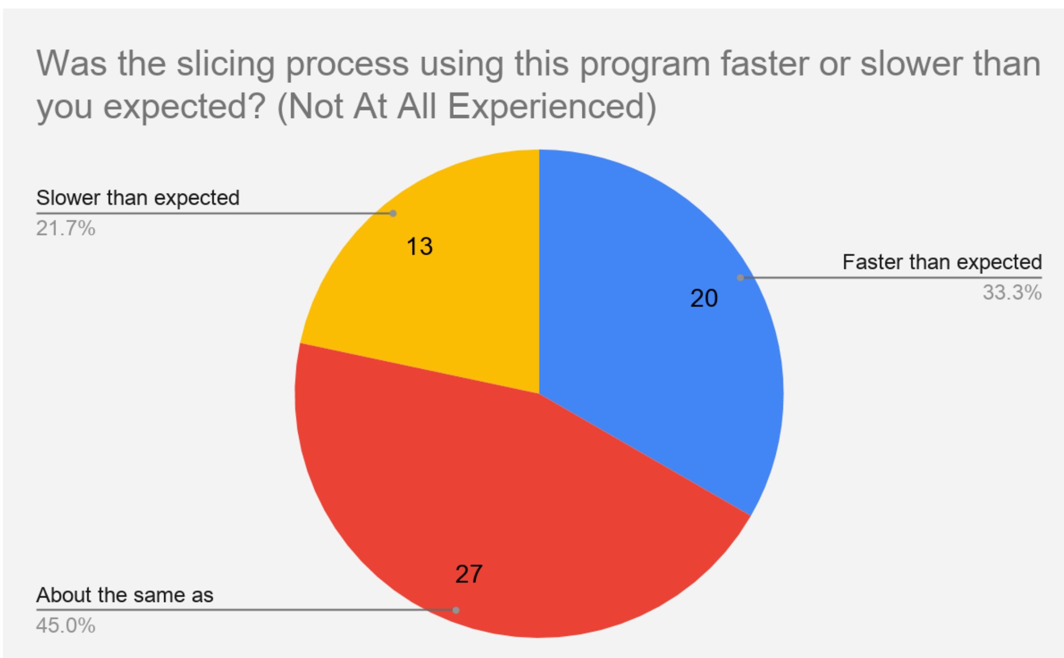


Figure 7.4: Not at all experienced users' speed evaluation

These results suggest that most people were satisfied with the speed, but the fact that there are a large number of users that answered ‘slower than expected’ does suggest more research could be done into optimization. Every experience level had at least 21% of people saying that the program ran slower than expected. This number is also much higher in the moderate experience group where 53% of users responded that way.

7.3. Orientation Optimization

Orientation is a vital part of our program, so the way users react to the generated orientations is important to understand, as well as how users react to the software as a whole. Figures 7.5, 7.6, and 7.7 show how users with different experience levels answered this question.

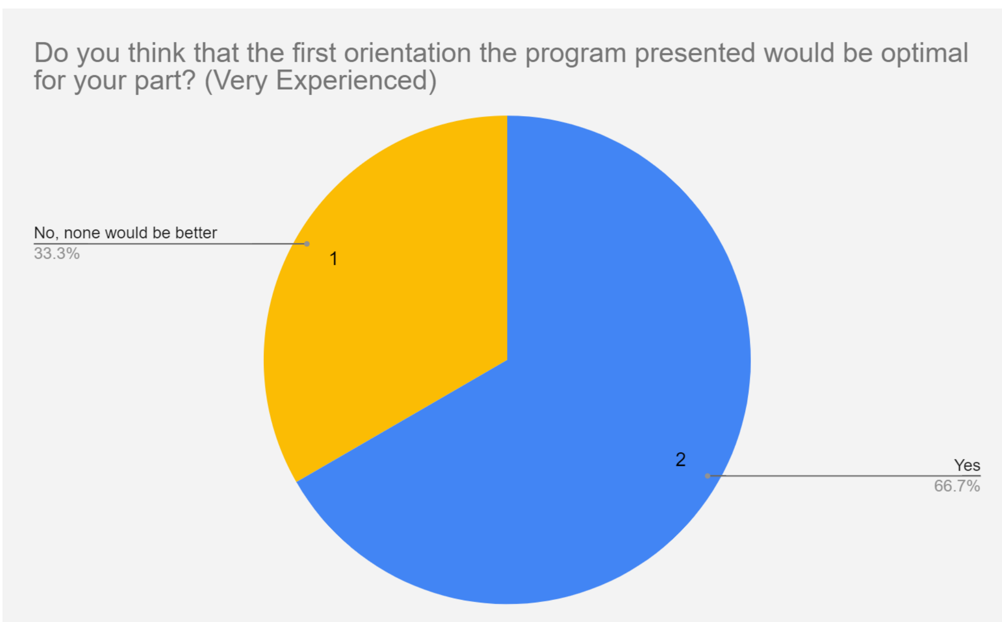


Figure 7.5: Very experienced users’ orientation evaluation

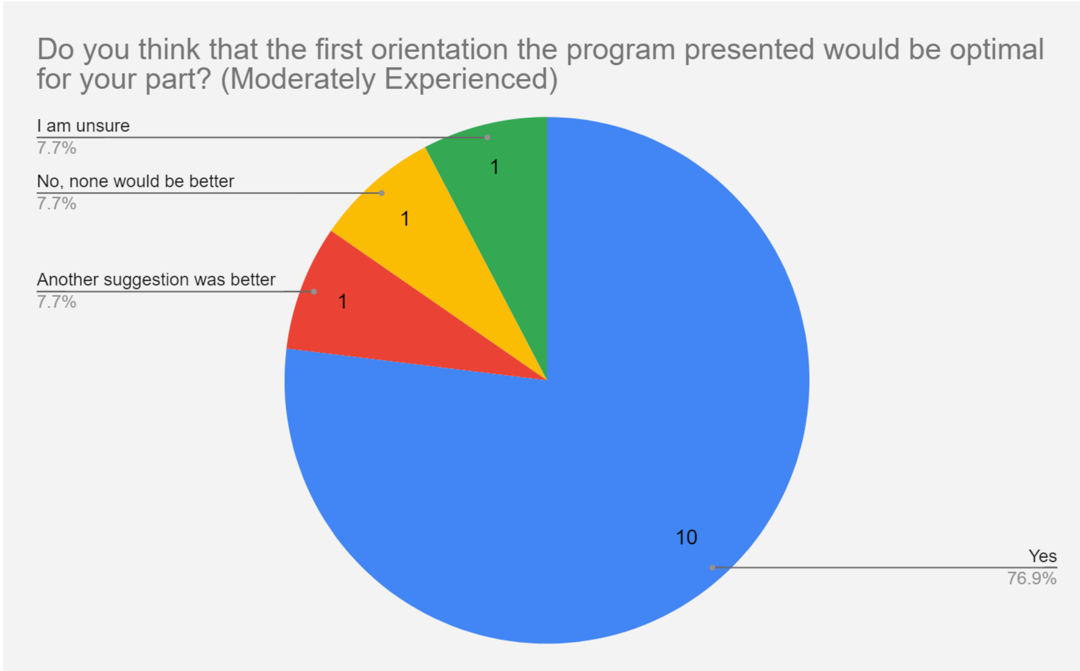


Figure 7.6: Moderately experienced users' orientation evaluation

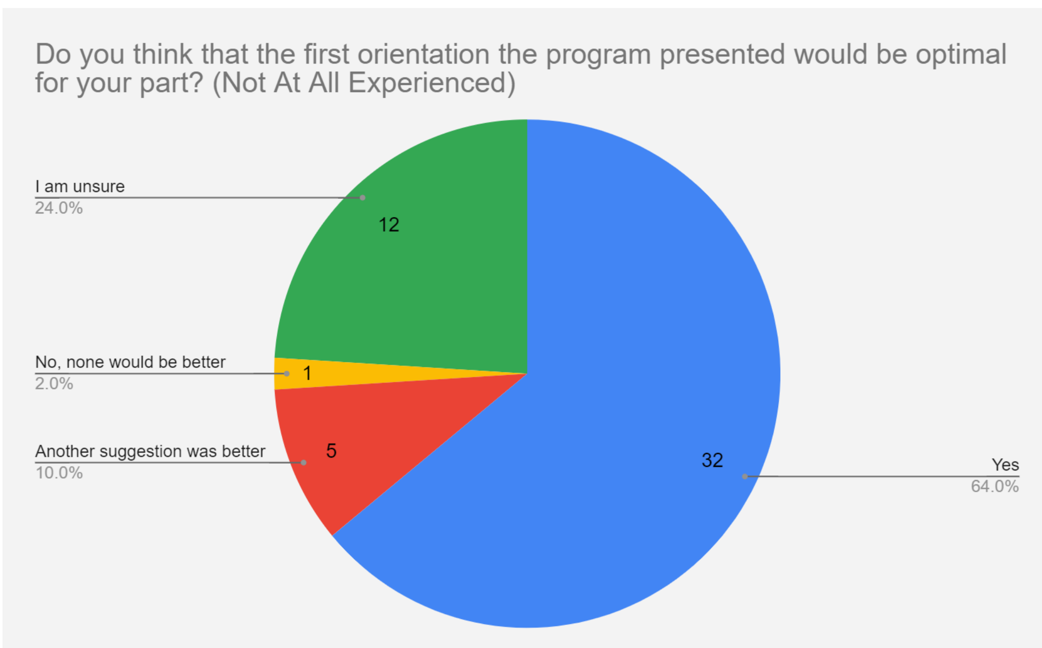


Figure 7.7: Not at all experienced users' orientation evaluation

These results seem to imply that a vast majority of the time, an optimal orientation was selected. Each experience level had at least 64% of its users report an ideal orientation.

7.4. Areas of Highest Difficulty

The questions described in section 7.4, 7.5 and 7.6 are all free response. When collecting data on free response questions we split the answers into common response categories. The complete list of free response answers can be found in Appendix D. The categories for this question are: program crashes, confusing UI, confusing questions, loading time, 3D viewer controls, and other. We also removed responses that said “nothing” or something similar. Additionally, we did not include responses that were not relevant to the program’s development, such as “It took too long to download.” Figures 7.8, 7.9, and 7.10 show how users with different experience levels answered this question.

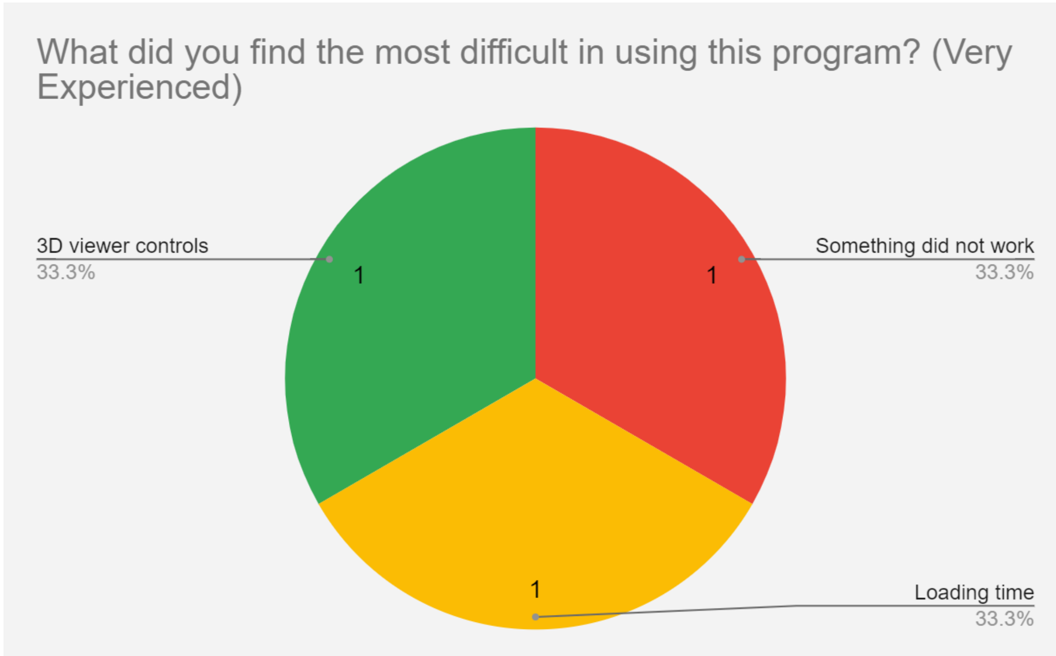


Figure 7.8: Very experienced users' difficulty evaluation

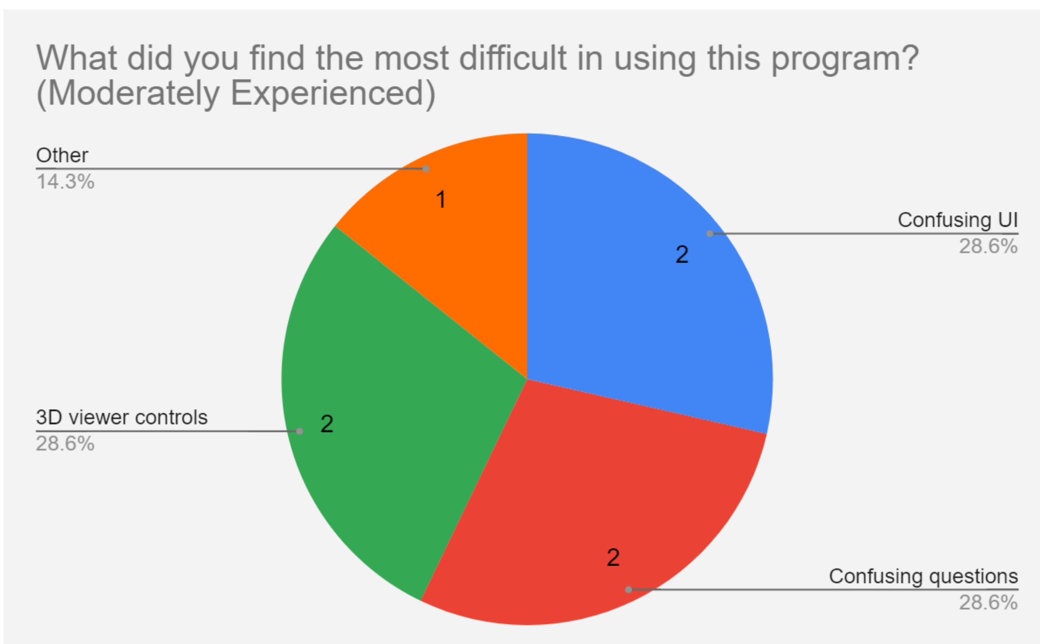


Figure 7.9: Moderately experienced difficulty evaluation

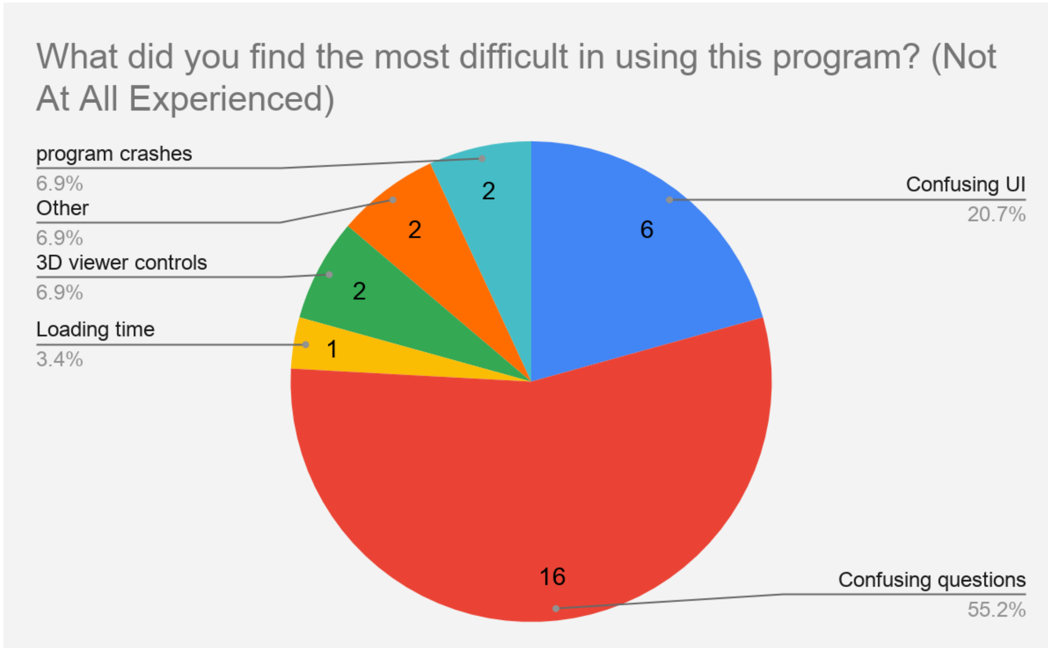


Figure 7.10: Not at all experienced difficulty evaluation

These results seem to suggest that the biggest difficulty an average user would encounter when using our program is the questionnaire. The questionnaire is the list of questions our program asks the user in mode 2. This problem is compounded by the fact that the less experience a user has (and becomes closer to our target user base) the more prevalent this confusion becomes. Based on the wording of the responses it would probably be helpful to better explain terms and explain how the result will affect the print in a simple way. We provide more information on what specific terms users found confusing in section 7.6.

7.5. Most Disliked Aspects

In this question, we asked users what they disliked most about our program. Figures 7.11, 7.12, and 7.13 show how users with different experience levels answered this question.

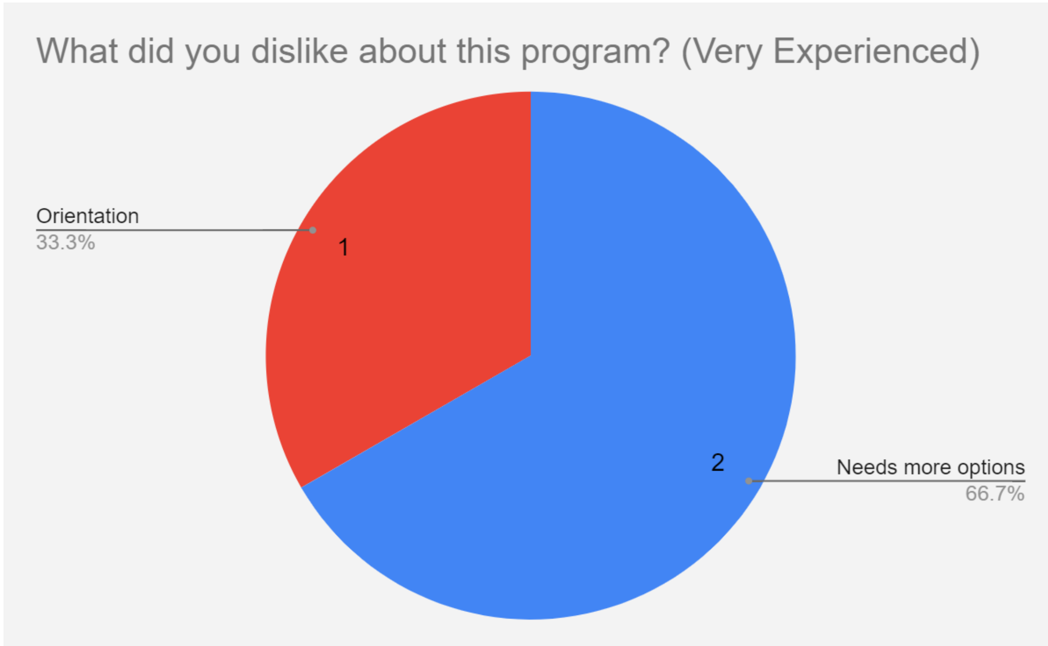


Figure 7.11: Very experienced users' most disliked aspects

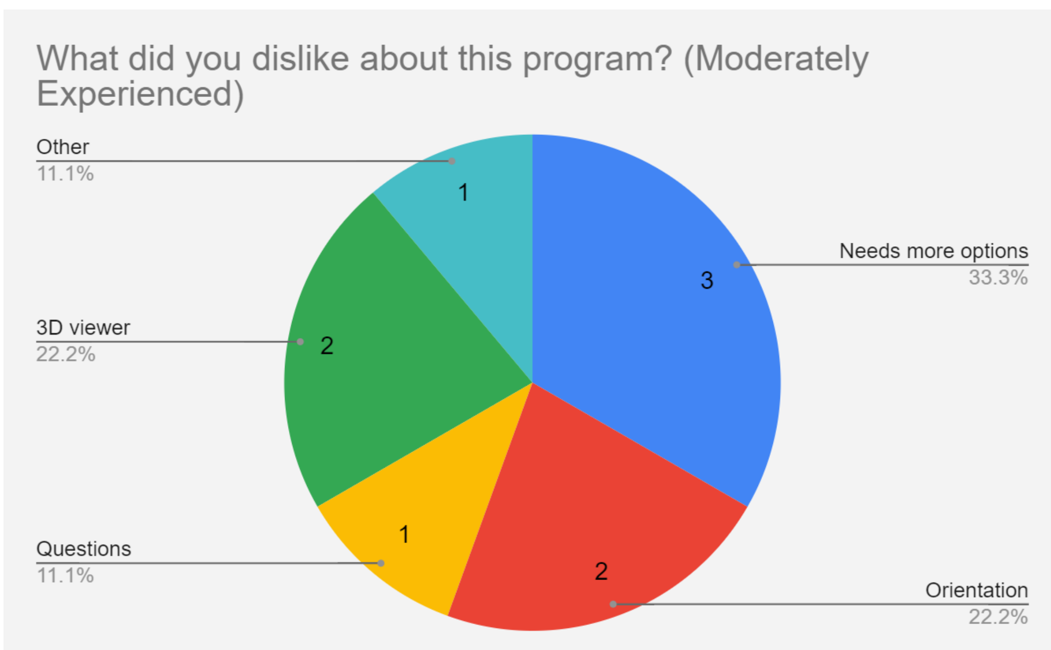


Figure 7.12: Moderately experienced users' most disliked aspects

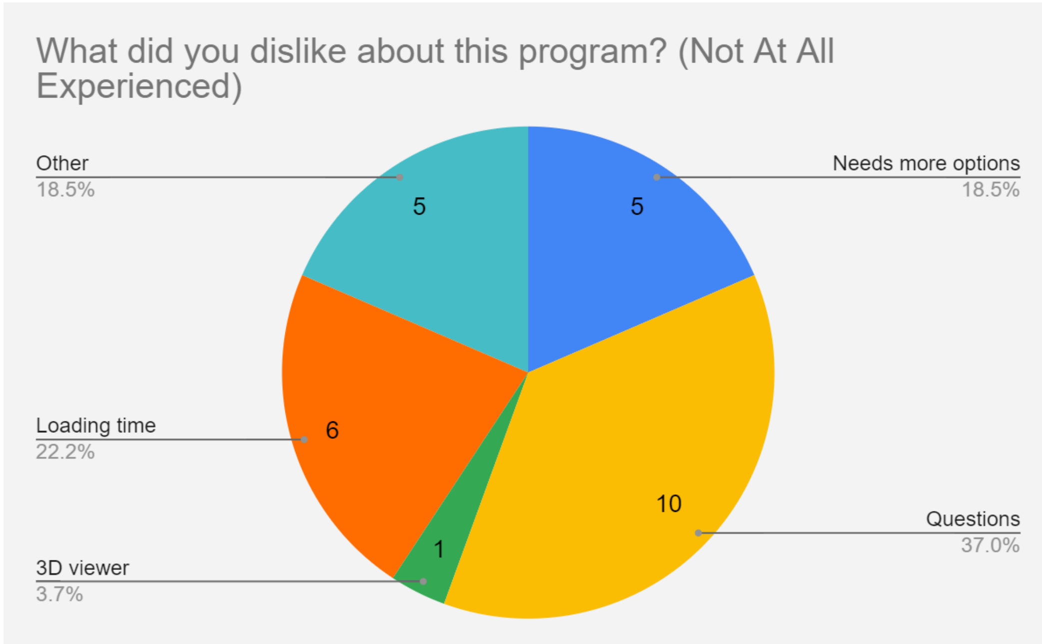


Figure 7.13: Not at all experienced users' most disliked aspects

While most of these results seem to reinforce the results found when we asked what they found most difficult, there were also an interesting request for more options. Most of these options seem like they would overcomplicate the program, such as to edit slicer settings manually. That being said there were also some suggestions that are seriously worth looking into such as being able to drag and drop ..stl files into the program.

7.6. Question Clarity

If a user found a question unclear in the program, we also asked users to explain why. This information is important because it will allow us to know how to improve the questions, as they were the highest cause of complaints. Due to the small number of highly experienced users and the specific nature of this question in the survey, we did not have any usable information from highly experienced users regarding the clarity of questions. Figures 7.14 and 7.15 show how users with different levels of experience answered this question.

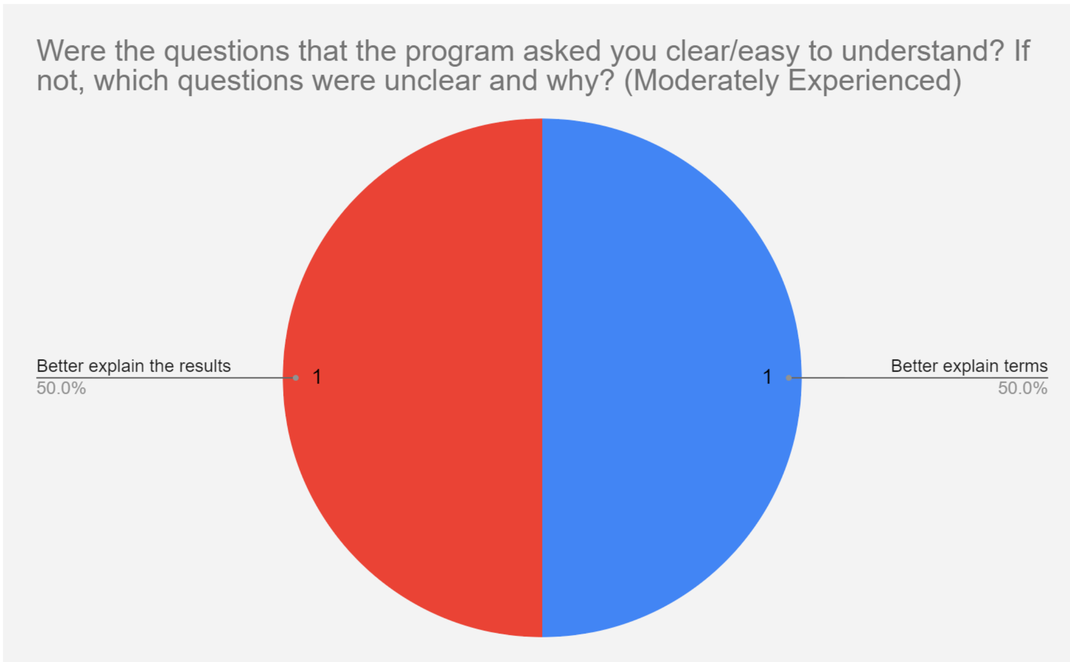


Figure 7.14: Moderately experienced users' unclear questions

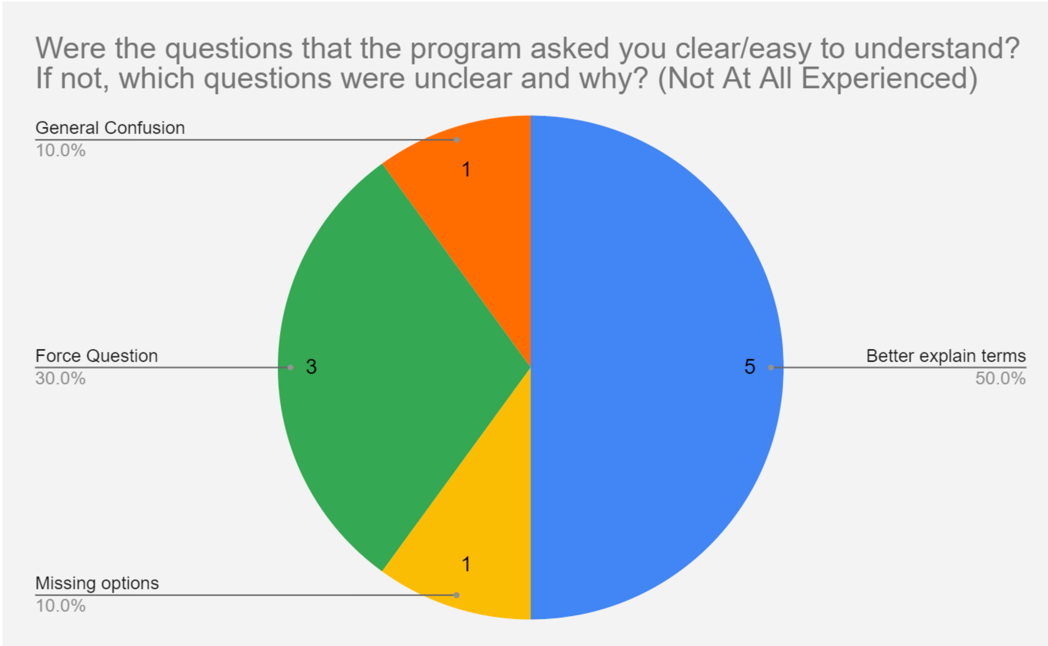


Figure 7.15: Not at all experienced users' unclear questions

Based on these results, it seems that we need to have better explanations of terms that appear in the questions. The terms that users seemed to be most confused by were “dimensional tolerances” and “flexural force”. It also seems that the question about the direction in which a force is applied confused some people and should be made clearer.

7.7. Satisfaction With Program

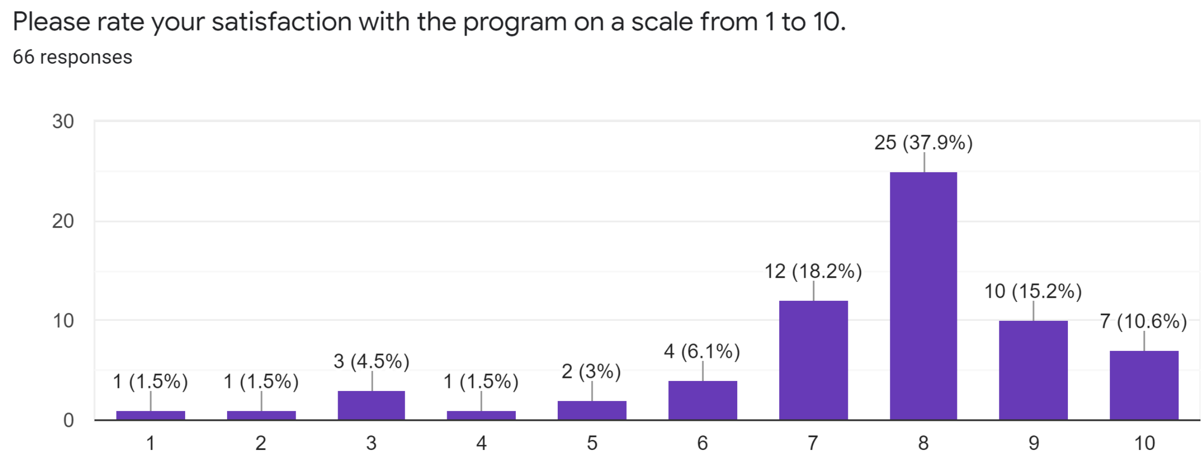


Figure 7.16: User satisfaction

Based on the information in Figure 7.16 we can say that the majority of people are satisfied with our program. More than 80% of users rate their satisfaction as a 7 or above, and every user was required to answer this question to complete this survey.

7.8. Program Evaluation

According to the results of our study, it seems that most users are satisfied with our program. However, there are key areas that could use some improvement. The part of our program that falls shortest appears to be the questionnaire. Users seem to be generally confused about the terms used in the questions and what the impact of each answer would be once selected. There were also a notable number of complaints about the “force” question, “What axis is the force acting along”, and how to interact with it.

Most users seemed to be happy with the loading time, but it could still be improved with further optimization.

7.9. Feedback from Professionals

Our team met with FDM printing professionals to discuss GaPA and get feedback from them over the course of the project. The individuals we met with were Mitra Anand, an Advanced Technology and Prototyping Specialist at WPI's Makerspace, and Professor Erica Stults, the head of WPI's Rapid Prototyping Lab. Both offered positive feedback on the application, as well as suggestions for possible improvement, some of which was implemented over the course of the term.

Our team met with them when we had an early working version of GaPA. In this meeting, we demonstrated GaPA's orientation algorithm using the Servo Mount model and the Eyelid model (visualized in Table 5.4). These were selected because the Servo Mount is an easy piece to visualize the qualities that GaPA considers (overhang, surface area, and forces), while the Eyelid serves as contrast, being a much more ambiguous piece in terms of orientation. This provided an opportunity to demonstrate how GaPA deals with parts that may have less surface area as well as gather the opinions of experts for a more complex orientation. In this meeting, we also described how the orientation algorithm works and what aspects are taken into account.

Both Professor Stults and Anand agreed with the first ranking orientation that we showed in both cases. They also verified aspects of our scoring algorithm. This included both the way that forces are oriented along the xy-plane, and the calculations for surface area on the bed. One issue with the scoring system that was discovered in these meetings was the system for measuring overhang. As described in section 5.20, the initial method did not take into account support that would be required within the model as it only found overhang above the printer bed. Following this, we were able to implement another method that we discussed with Anand, the one found in the final version of the application. Another issue that both experts noted was the potential for tall and thin models being printed upright, a common cause for print failure. Professor Stults suggested the implementation of a slenderness ratio score into the application as well. This would compare the height and width of the model and include this score as one of the weights. A Slenderness Ratio score was added into the application later, through the methods outlined in section 5.20.

Near the end of the project, members of our team met with Mitra Anand and Professor Erica Stults to follow up on the progress of the application and gather their feedback. During these presentations, we reviewed what the program did as well as the updated features since the previous meetings, including the suggested Slenderness Ratio. GaPA was demonstrated using the Servo Mount model as an example.

The majority of the positive feedback was focused on the presentation of the results. Both professionals we interviewed praised the use of the pros and cons list to provide the user with more information about the orientation. This method also establishes to novice users that there is no single ideal orientation for any given part. Professor Stults praised this for teaching purposes, as it encourages the user to critically analyze the generated orientations rather than just picking the first option.

Both agreed that the first orientation GaPA generated was the most optimal, based on their experience and the use case we outlined. However, one of the major criticisms from Professor Stults, which will be addressed further in the Future Work section, was the impracticality of the lower ranking orientations. She felt that some of these lower ranked orientations were not viable orientations for printing at all. Her suggestion was to find a way to create a threshold for viable orientations and limit the number of orientations to less than four if there are not enough viable orientations.

Mitra Anand gave positive feedback on the flow and interface of the program. He thought that the overall design of the interface was very helpful for new users, guiding them through the process with reasonably few clicks. He appreciated the minimal interaction between the program's user and the slicing tool.

Both Professor Stults and Mitra Anand noted the practicality of this program for teaching people how to get a part ready for 3D printing. They noted the ease of navigation and the way the results are presented. Anand suggested that GaPA could possibly be used as part of basic training for the Makerspace as a more interactive tool.

8. Broader Impact on Mechanical Engineering

When a product is being developed, the greater impact the product will have must be considered. No product exists in isolation and it will have a social, environmental, and economic impact. This chapter considers the broader impacts that GaPA could have.

GaPA is designed to enhance manufacturing and Mechanical Engineering as a whole by making 3D Printing more accessible. The target audience of GaPA is Mechanical Engineering students, specifically those with little experience in 3D Printing. GaPA provides easier access to a popular form of manufacturing, helping students further both their own projects as well as their knowledge of 3D Printing.

8.1. ASME Code of Ethics

Engineers have a responsibility to produce objects that satisfy the requirements of the customer and function well. By automating some of the printing process, GaPA eases the process and can help ensure a positive result. The application supports intelligent design and, by helping novice users, GaPA helps prevent problems that arise from a lack of knowledge of 3D Printing.

8.2. Societal and Global Impact

The use of 3D printing for manufacturing has grown exponentially in the past decade. GaPA has the potential to increase the number of people who can utilize 3D printing by making the process more accessible. Students of all disciplines can utilize the software to create effective prototypes without having to process a lot of background knowledge related to 3D Printing. At the same time, the software also provides valuable information related to the process. We hope that this will lower the barrier of entry to using 3D Printing at WPI, and around the world, helping to produce engineers with a stronger foundation of knowledge in this growing field.

8.3. Environmental Impact

As a software application, GaPA has little direct environmental impact. However, the function that it provides has the potential to improve the environmental impact of 3D printing. By coding knowledge for users, GaPA limits the failure rate of 3D Printed parts. This could reduce material use for 3D printing as material is wasted whenever a print fails. Beyond material reduction this would also save energy as the printer would not have to be run as many times.

8.4. Codes and Standards

Since GaPA is a software application, the traditional mechanical standards are not applicable.

8.5. Economic Factors

By fostering more knowledge and interest in 3D printing, we hope that students will in turn take this knowledge to improve the field beyond WPI. Both 3D printing and additive manufacturing in general have the potential to grow much further than they already have. As an alternative to most common methods of manufacturing, 3D printing can fill more niches and open more manufacturing opportunities than it already has. We hope that GaPA can play a role in furthering this field through improved accessibility, bringing more perspectives into the industry that can develop and grow the market.

9. Team Members' Applied and Developed Skills

An important factor for success in a project is knowledge of each team member's strengths so that they can be effectively applied. At the beginning of the project, each team member listed some of their relevant skills for the project. These are listed below:

Collin:
- Mechanical engineering knowledge and skills
- CAD experience
- Basic computer science theory for team communication
- Fairly strong technical writing skills
- Creating and giving presentations
- Art and color theory

James:
- How to use MVC based frameworks
- Analysis and creation of algorithms
- Experience with node js based web development
- Fundamental understanding of databases
- Fundamental understanding of displaying 3D graphics
- Experience developing using a variety of OOP languages

Matt:
- Web app development using C# (ASP.NET)
- Experienced with database querying and integration
- Background in Java, Python, C#
- Technical writing experience
- Experience debugging programs

Parker:
- Effective writing skills
- Editing the writing of others for cohesion
- Organizing disparate points into structured paragraphs
- Extensive background in Java
- Fundamental understanding of databases
- Effective note taking

Over the course of the project, we ensured that work we were doing aligned with our strengths while still developing other skills. Near the end of the project, our team revisited what we had recorded as our strengths and considered what skills we applied most to this project. We also considered what skills we had developed through working on this project. Both of these are listed below.

Collin:

Applied Skills

- Mechanical background for reading and extracting info from studies
- Debugging and communication from a consumer point of view
- Technical writing and presentation skills

Learned Skills

- How to use a 3D Printer
- Application Testing
- Using if/then statements for design

Matt:

Applied Skills

- Technical writing
- Debugging
- C# app development

Learned Skills

- XAML for user interfaces
- WPF framework

James:

Applied Skills

- Algorithm development
- Background in several OOP languages

Learned Skills

- XAML based UI
- WPF framework
- Converting a large codebase into a single executable file

Parker:

Applied Skills

- Technical writing
- Debugging

Learned Skills

- C# app development
- XAML
- UI design and critique
- How to use a 3D printer
- Writing for clarity

These skills have helped our team work together effectively by working with our strengths as well as allowing room for team members to grow. An important factor in our success was collaboration between team members, both applying their skills to the project and helping other team members develop their own skills. It is through this effective teamwork that we were able to develop GaPA.

10. Conclusion

Over the course of the 2020 - 2021 school year, our team has created an application that automates the 3D printing and G-Code generation process for users, called G-Code and Printing Automation (GaPA). The main objective was for the program to be easy to understand by both novice and expert 3D printers alike and not require advanced knowledge about the 3D printing process. To that end, our project team created a WPF application that allowed users to upload an .stl file, answer questions about their desired print quality, and select one out of four generated solutions to print.

At the beginning of the project, our group made a set of testable requirements, as previously recorded in Section 3, that our program would be able to fulfill upon completion. Over the course of program development, we implemented functionality that fulfilled all of these requirements. The front-end allowed users to import and view .stl files, save G-Code for generated solutions, and view the pros and cons of each generated solution. The back-end detected flat surfaces on imported models, determined the optimal settings for the print, generated solutions based on them, and correctly generated G-Code for the solutions. Our team had some other, longer-term requirements of the program. Some of these were met, such as providing documentation for new users through a help website, but others fell out of the scope of the project, such as the ability to interpret Solidworks CAD files.

Our program also met the 10 general principles for interaction design. The state of the system was always visible on the screen by means of the progress bar at the top of every mode. The application matches between the system and the real world by using easy to understand terminology which remains consistent throughout the application process. The program will prevent most user errors with helpful accompanying dialogue, but for those it cannot, the user is able to navigate between modes without losing progress. The minimal design of GaPA enables users to navigate it efficiently. The Help button and website serve to fulfill the last design principle regarding help and documentation. Therefore, GaPA fulfills all ten of the interface design principles.

As previously mentioned, there are five dimensions of interface usability that are used to qualify user interfaces: effectiveness, efficiency, level of engagement, error tolerance, and learnability. After completing this project, GaPA's user interface has characteristics of all five of these dimensions. Our interface is effective as it guides the user through the 3D printing process in three basic steps, which are clearly explained. It is also efficient, with a minimal number of clicks needed to generate G-Code for the file. From our user evaluation, we determined that the majority of users were very satisfied by the design of the program, fulfilling the level of engagement dimension. Our program is error tolerant, displaying pop-up messages if the user is about to make a mistake. Finally, the interface is very straightforward and easy to learn for all users. So, GaPA meets all five of the dimensions of interface usability.

This program will serve as a convenient way for WPI students to quickly print 3D models with little effort. It will also serve as a gateway for students to learn more about the 3D printing process. This program has the potential to be used by students from other universities as well.

10.1. Future Work

One of the initial goals for this project was to allow users to print their selected solution directly from the program. While our team was not able to implement this in our program, we researched ways to do this. One of these ways was through an open source application called OctoPrint. OctoPrint is a web-based application that allows users to queue prints to different printers and view information for prints in progress. At the beginning of the project, our team planned to send prints from the GaPA application to a microcontroller running OctoPrint which would allow users to print the G-Code they generated with the click of the “Print” button on Mode 3. However, as time passed, this functionality fell out of the scope of the project because our team decided to prioritize other functionality of the program. An OctoPrint connection is a feature that could be implemented by another team in the future.

Additionally, our team fell short on software testing. As discussed in our section on software testing (section 6.1), a future team could improve test cases, measuring code coverage, and testing more seeded errors.

Another goal of our team that could be accomplished by a future team would be the improvement of the user interface. Through our user study, we received feedback from students on aspects of the user interface of GaPA that they found unintuitive or difficult to use. Since our team received this feedback so close to the end of the project, we did not have time to make changes to the interface. So, a future team could improve aspects of the user interface based on the feedback from our user study.

Further suggestions for improvement were provided from our discussions about the application with Makerspace Specialist Mitra Anand and Professor Erica Stults. These were suggestions for improved functionality as well as improved interface.

One of the major critiques of the application was the inability for it to consider small details, such as walls being too thin for the selected layer thickness. A recommendation was to consider the smallest distance between walls and compare this to the layer thickness, warning the user if the wall is too thin. This is similar in functionality to the Wall Thickness Analysis in Solidworks. Another suggestion involved force selection, recommending the ability to place point forces for a more robust orientation. While this would improve the accuracy of GaPA with further development, implementation of this would have to contend with the simplicity of the user interface. A final functionality suggestion was the limitation of possible orientations when they are impractical. As it stands, GaPA always suggests four possible orientations but, especially when adding a force constraint, the lower ranked orientations might have no benefit over the higher ranked ones. Possibly including a score threshold and limiting the number of orientations to viable ones would limit the decisions the user has to make as well as provide only orientations that the user can learn from.

One recommendation for changes to the interface is to have the supports visible in the orientation recommendations. This would help the user to visualize where the supports will be and which faces of the print will be affected. Our team had considered this, however the implementation proved to be difficult and was not a high priority. A major obstacle to implementation was the added processing time of creating a visualization. Another recommendation was regarding the pros and cons list. Professor

Stults recommended adding color to the text to emphasize the severity of the pros and cons. GaPA already differentiates through language (‘very little surface’ area versus ‘little surface area’) but color would help further indicate this to the user. Along with this recommendation was the suggestion of help dialogue about the pros and cons so the user can get a little more information.

11. References

- 3D Printing Industry (2017). The Free Beginner's Guide. Available: https://3dprintingindustry.com/3d-printing-basics-free-beginners-guide. [Accessed November 29 2020]
- 3D Pros. "Optimizing 3D Printed Parts for Strength," *3d-pros.com*. Available: https://www.3d-pros.com/optimizing-strength-of-3d-printed-parts. [Accessed November 29 2020]
- All3DP (2016). 3D Slicer Settings for Beginners: 8 Things You Need to Know. Available: https://all3dp.com/3d-slicer-settings-beginners-8-things-need-know/ [Accessed November 28 2020]
- All3DP (6 February 2020). "2020 3D Printer Filament Buyer's Guide," *All3DP*. Available: https://all3dp.com/1/3d-printer-filament-types-3d-printing-3d-filament/. [Accessed November 28 2020]
- AMFG (23 July 2018). "TPU 3D Printing: A Guide to 3D Printing Flexible Parts," *AMFG Autonomous Manufacturing*. Available: https://amfg.ai/2018/07/23/tpu-3d-printing-guide/. [Accessed November 28 2020]
- Bouthillier, J. (10 February 2016). "The Importance of High Quality 3D Printer Filament," *Boots Industries.* Available: http://bootsindustries.com/the-importance-of-high-quality-3d-printer-filament/ [Accessed 30 November 2020]
- Bhavnagarwaia H. (25 November 2019). "The Perfect ABS Print & Bed Temperature," *All3DP*. Available: https://all3dp.com/2/abs-print-bed-temperature-all-you-need-to-know/. [Accessed November 30 2020]
- Chacóna, J. M.; Caminerob, M. A.; García-Plazab, E.; Núñez, P. J. (15 June 2017). "Additive manufacturing of PLA structures using fused deposition modelling: Effect of process parameters on mechanical properties and their optimal selection," *Materials and Design*, Vol 124, pp. 143-157. Available: https://www.sciencedirect.com/science/article/abs/pii/S0264127517303143 [Accessed November 2020]
- CitiGPS. (2020, February 24). *Technology at Work v3.0- Citi Global Perspectives and Solutions*. Available: https://www.citivelocity.com/citigps/technology-work-v3-0/. [Accessed April 24 2020]

- Chakravorty, D. (2020). 3D Printing Support Structures - The Ultimate Guide. All3DP. Available: https://all3dp.com/1/3d-printing-support-structures/ [Accessed April 25 2021]
- CNC Kitchen (October 19, 2019). “Extrusion Width - The Magic Parameter for Strong 3D Prints?,” *Youtube*. Available: https://www.youtube.com/watch?v=9YaJ0wSKKHA&feature=youtu.be [Accessed November 21 2020]
- Dejan. (2020) G-code Explained | List of Most Important G-code Commands. How to Mechatronics. Available: https://howtomechatronics.com/tutorials/g-code-explained-list-of-most-important-g-code-commands/ [Accessed April 25 2021]
- Engineerdog.com (September 2, 2015). “Mechanical Testing 3D Printed Parts: Results and Recommendations,” *engineerdog.com*. Available: https://engineerdog.com/2015/09/02/mechanical-testing-3d-printed-parts-results-and-recommendations/ [Accessed November 21 2020]
- Fabian. (2018). How to Choose the Best File Resolution for 3D Printing. I.materialise. Available: https://i.materialise.com/blog/en/how-to-choose-the-perfect-file-resolution-when-turning-your-3d-model-into-a-3d-print/ [Accessed April 25 2021]
- Formlabs (n.d.). “Understanding Accuracy, Precision, and Tolerance in 3D Printing,” *Formlabs*. Available: https://formlabs.com/blog/understanding-accuracy-precision-tolerance-in-3d-printing/. Accessed [Accessed November 30 2020]
- Grames, Emmett. (2020). FDM 3D Printing - Simply Explained. All3DP. Available: https://all3dp.com/2/fused-deposition-modeling-fdm-3d-printing-simply-explained/ [Accessed April 25 2021]
- Heckenlively, T. (26 September 2019). “Effects of Build Orientation and Raster Angle on the Tensile Behavior of 3D Printed Parts,” *Teton Simulation*. Available: https://www.tetonsim.com/blog/post/effects-of-build-orientation-and-raster-angle-on-the-tensile-behavior-of-3d-printed-parts. [Accessed November 30 2020]
- Helix Toolkit's Documentation. Helix Toolkit 2015.1 documentation. Available: http://docs.helix-toolkit.org/en/latest/ [Accessed April 29, 2021]
- Houser, Franklin (2016). The Ultimate Guide to 3D Printing - 101 Questions Answered. All3DP. Available: https://all3dp.com/3d-printing-3d-printer-guide-101-questions/ [Accessed April 25 2021]

- U.S. General Services Administration (2013, October 09). Heuristic Evaluations and Expert Reviews Available: https://www.usability.gov/how-to-and-tools/methods/heuristic-evaluation.html [Accessed October 13 2020]
- Hunter, S. (2020, May 21). Introducing .net multi-platform app ui. Available: https://devblogs.microsoft.com/dotnet/introducing-net-multi-platform-app-ui/ [Accessed March 19, 2021]
- Jesimon0801, (2014, November 2014,) “Tech Talk Thursday: Intro to 3D Printing,” *Enabling the Future*. Available: https://enablingthefuture.org/2014/11/13/tech-talk-thursday-intro-to-3d-printing/. [Accessed March 15 2020]
- Kondo H. (12 May 2019). “How to Prevent ABS from Warping on a Heated Bed,” *All3DP*. Available: https://all3dp.com/2/how-to-prevent-abs-from-warping-on-a-heated-bed/. [Accessed November 30 2020]
- Krug, S. (2014). *Don't make me think! A Common Sense Approach to Web & Mobile Usability.* New Riders. Available: http://www.scottsdevelopers.com/dont-make-me-think-revisited.pdf [Accessed April 25 2021]
- Landry, T. (1 June 2015). “Slice Settings Explained - Part 2,” *MatterHackers*. Available: https://www.matterhackers.com/articles/slice-settings-explained-part-2. [Accessed November 28 2020]
- Luzanin, O.; Plancak, M.; Movrin, D.; (January 2014). “Effect of layer thickness, deposition angle, and infill on maximum flexural force in FDM-built specimens,” Journal for Technology of Plasticity, Volume 39, Number 1.
- Math Insight. (n.d.). Available: https://mathinsight.org/definition/state_space [Accessed October 26 2020]
- Matter Hackers Inc. (2021). “MatterControl - 3D Printing Software,” *MatterHackers*. Available: https://www.matterhackers.com/store/l/mattercontrol/sk/MKZGTDW6. [Accessed 5 May 2021].
- Nielsen, J. (2020, November 15). 10 Usability Heuristics for User Interface Design. Nielsen Norman Group. Available: https://www.nngroup.com/articles/ten-usability-heuristics/. [Accessed November 28 2020]

- Pandzic, A; Hodzic, D & Milovanovic, A (2019). Effect of Infill Type and Density on Tensile Properties of PLA Material for FDM Process, Proceedings of the 30th DAAAM International Symposium, pp.0545-0554, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734- 22-8, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/30th.daaam.proceedings.074
- Plewa, Kat. (2018). 3 Easy Steps for Best STL Files Resolution. Sculpteo Available: https://www.sculpteo.com/blog/2018/07/17/3-easy-steps-for-best-stl-files-resolution/ [Accessed April 25 2021]
- Siber, Blu. (2018). 3D Printing Infill: The Basics - Simply Explained. All3DP. Available: https://all3dp.com/2/infill-3d-printing-what-it-means-and-how-to-use-it/ [Accessed April 25 2021]
- Simplify3D (2020). Rafts, Skirts, and Brims! Simplify3D. Available: https://www.simplify3d.com/support/articles/rafts-skirts-and-brims/ [Accessed April 25 2021]
- Simplify3D (2020). "Materials Guide: Flexible," *Simplify3D*. Available: https://www.simplify3d.com/support/materials-guide/flexible/. [Accessed November 29 2020]
- Simplify3D (2020). "Materials Guide: Polycarbonate," *Simplify3D*. Available: https://www.simplify3d.com/support/materials-guide/polycarbonate/. [Accessed November 30 2020]
- Simplify3D (2020). "Materials Guide: Polypropylene," *Simplify3D*. Available: https://www.simplify3d.com/support/materials-guide/polypropylene/. [Accessed November 30 2020]
- Slic3r (n.d). "Slic3r: Open source 3D printing toolbox," *slic3r.org*. Available: https://slic3r.org/download/. [Accessed 5 May 2021]
- U.S. General Services Administration (2014, May 21). User Interface Design Basics. Available: https://www.usability.gov/what-and-why/user-interface-design.html [Accessed October 12 2020]
- Ultimaker Cura. (2021). "Ultimaker Cura 4.9 Download," *Ultimaker*. Available: https://ultimaker.com/software/ultimaker-cura [Accessed 28 November 2020]
- von Ubel, M. (4 September 2020). "2020 3D Printing Materials Guide," *All3DP*. Available: . https://all3dp.com/1/3d-printing-materials-guide-3d-printer-material/. [Accessed November 28 2020]

- Web Accessibility Initiative (WAI) (n.d.). *Designing for Web Accessibility – Tips for Getting Started*. Available: https://www.w3.org/WAI/tips/designing/ [Accessed April 25 2021]
- Whittaker, J. (2000). What is software testing? And why is it so hard? *IEEE Software*, 17(1), 70-79. doi:10.1109/52.819971
- Williams, R. (2015). *The non-designer's design book : design and typographic principles for the visual novice* (Fourth edition.). Peachpit Press.
- Zaman, U; Boesch, E; Siadat, A; Rivette, M; Baqai, A. 2018. "Impact of fused deposition modeling (FDM) process parameters on strength of built parts using Taguchi's design of experiments," *The International Journal of Advanced Manufacturing Technology*, 16 November 2018, vol. 101, pp. 1215-1226.

12. Appendices

Appendix A: List of If/Then Statements for Settings

General Strength

If strength is desired, then the shell thickness should be wider

If strength is desired, then the extrusion width multiplier should be greater with fewer perimeters

Flexural Strength

If flexural strength is desired, then the layer height should be smaller

If flexural strength is desired, then the infill should be 20% at 30 degrees from the applied force

If flexural strength is desired, then the model should be on-edge with the direction of the force acting on either the x or y axes

If flexural strength is desired, then the material preferences are as such: ABS > PETG > PLA

Tensile Strength

If tensile strength is desired, then the infill density should be between 60 and 70%

If tensile strength is desired, then the infill pattern should be at a 45 degree angle to the force and the pattern should be Concentric

If tensile strength is desired, then the model should be on-edge or flat orientation with the force acting along either the x or y axes

If tensile strength is desired and the model is on-edge or flat, the layer height should be less than 0.1 mm (with negligible difference in anything higher)

If tensile strength is desired, then the material preferences(?) are: PETG > PLA > ABS

Compressive Strength

If compressive strength is desired, then the layer height should be smaller

If compressive strength is desired, then the infill density should be about 70%

If compressive strength is desired, then the orientation should be on-edge or flat with the force acting along the x or y axes

If compressive strength is desired, then the infill pattern should be diamond

Dimensional Tolerances

If accuracy is desired, then layer heights should be smaller

If accuracy is desired, then the selected printer should have greater resolution

If accuracy is desired, then the extrusion width should be < 150%

If accuracy is desired, then the material preferences are as such: PLA > PTEG > Nylon > ABS

Gear

If the part is a gear, then the infill should be 100%

If the part is a gear, then it should be printed at an angle to the bed

If the part is a gear, then the material priority is Nylon > PLA > ABS > PETG

If the part is a gear, then the printer should have a low resolution value for improved dimensions

Flexibility

If the part is flexible, then the material should be TPU

If the part is flexible, then the infill should be 10%

If the part is flexible, then the layer height should be between 0.1 and 0.2 mm

Other

If the part is in a harsher environment, then PETG is used as a material

If the part should be transparent, then material priority is PC > PETG

<u>Second Level if/then Statements</u> -These rely on information determined in previous if/then statements, such as material

Material Effects

If the material is PLA, then the nozzle temperature should be 205 degrees Celsius

If the material is PLA, then the bed temperature should be 30 degrees Celsius

If the material is PLA, then the feed rate should be average

If the material is PLA, then the first layer height should be 90% of the layer height

If the material is ABS, then the nozzle temperature should be 235 degrees Celsius

If the material is ABS, then the bed temperature should be 110 degrees Celsius

If the material is ABS, then the feed rate should be average

If the material is ABS, then the first layer height should be 150% of the normal layer height

If the material is ABS, then the first layer speed should be 70% of the print speed

If the material is ABS, then the part should be printed with a raft

If the material is PETG, then the nozzle temperature should be 235 degrees Celsius

If the material is PETG, then the bed temperature should be 60 degrees Celsius

If the material is PETG, then the feed rate should be higher

If the material is TPU, then the nozzle temperature should be 235 degrees Celsius

If the material is TPU, then the bed temperature should be 50 degrees Celsius

If the material is TPU, then the feed rate should be lower

If the material is TPU, the extrusion multiplier should be increased slightly

If the material is TPU, then the retraction should be enabled

If the material is TPU, then the retraction distance should be 5 mm

If the material is TPU, then the retraction speed should be 25 m/s

If the material is TPU, then the print and wall acceleration should be 300

If the material is TPU, then there should be a skirt for the part

If the material is Nylon, then the nozzle temperature should be 240 degrees Celsius

If the material is Nylon, then the bed temperature should be 85 degrees Celsius

If the material is Nylon, then the feed rate should be average

If the material is Nylon, then the first layer height should be 150% of the layer height

If the material is Nylon, then the part should be printed with a raft

If the material is PC, then the nozzle temperature should be 300 degrees Celsius

If the material is PC, then the bed temperature should be 100 degrees Celsius

If the material is PC, then retraction should be enabled

If the material is PC, then the first layer height should be 150% the layer height

If the material is PC, then the first layer width should be 120% the extrusion width

If the material is PC, then the first layer speed should be 20% the normal speed

If the material is PC, then the movement speed should be half the normal speed

If the material is PC, then the fan should be off

If the material is Polypropylene, then the nozzle temperature should be 235 degrees Celsius

If the material is PP, then the bed temperature should be 95 degrees Celsius

If the material is PP, then the cooling fan should be on

If the material is PP, then the part should be printed with a raft

Printer Selection

If the model is too big for a printer bed, then any printer with a smaller bed will be eliminated as an option.

If the material is known, then the printer must be able to print that material.

If there is not a printer that meets the first two criteria, the model cannot be printed and the user will be informed that there is no viable printer. It will do so by replacing the text on Mode 3 where the printer name would normally be displayed with “No valid printers”. A user will still be able to generate the G-Code, but it will be generic and won’t work properly on all printers. The user can solve this problem by returning to Mode 1 and adding a viable printer to the system.

If there are multiple printers that meet the first two criteria, then the printer will be selected by dimensional precision based on the user’s requirements for precision.

If there is only one printer that meets the first two criteria, then that printer should be selected.

Appendix B: List of Survey Questions

Format:

Question

- Possible Answer
  - Settings applied
- Possible Answer
  - Settings applied

Reasoning and explanation

<u>Survey Questions:</u>

Will this part be used in a mechanical system?

- Yes, the part will be subjected to a force.
  - Shell thickness: 1.2 mm
- No, the part is primarily used for display or visualization
  - Shell thickness: 0.8 mm
  - Infill Density: 20%
  - Infill Pattern: Grid
  - Material: PLA

This question determines if the part will require general strength which provides information to select a shell thickness. Since there is less variation in display parts than mechanical parts, selecting display will also determine the infill and material.

What force or scenario most closely matches the use of the part?

- Strength under tensile (pulling) force
  - Infill Density: 70%
  - Orientation: On-edge
  - Layer Height: 0.1 mm
  - Infill Pattern: Concentric
  - Material Ranking: PETG > PLA > ABS
- Strength under compressive (pushing) force

    - Infill Density: 70%
    - Orientation: On-edge or Flat (Maximize Surface Area)
    - Layer Height: 0.1 mm
    - Infill Pattern: Concentric
    - Material Ranking:
- Strength under flexural (bending) force
    - Infill Density: 20%
    - Orientation: On-Edge
    - Layer Height: 0.1 mm
    - Infill Pattern: Grid
    - Material Ranking: ABS > PETG > PLA
- A flexible part
    - Infill Density: 10%
    - Orientation: Maximize Surface Area
    - Layer Height: 0.1 mm
    - Infill Pattern: Grid
    - Material: TPU
- A gear
    - Infill Density: 100%
    - Orientation: At a 20 Degree angle to the bed
    - Layer Height: 0.2 mm
    - Infill Pattern: N/A
    - Material Ranking: Nylon > PLA

This question determines the basic function of the print: applied forces or other special cases. This decision also provides enough information to select the infill density, general orientation, layer height, infill pattern, and material rankings.

What axis is the force acting along?

- X-Axis
- Y-Axis
- Z-Axis

Provides information for the orientation program that allows it to finalize the orientation for the cases of tensile, compressive, and flexural strength.

How important are the dimensional tolerances? (Mechanical Path)

- Very important, more so than the strength
    - Selects printers with the best tolerance and then considers materials based on the ranking
- Important but not at the cost of strength
    - Selects printer with the best tolerance that has the best material available
- Less important
    - Selects the next available printer with the best material based on ranking

Provides selection criteria and ordering for printer and material selection.

Does the model have curved surfaces or small details?

- Yes
    - Layer Height: 0.1 mm
- No
    - Layer Height: 0.3 mm

Provides information for the layer height to provide better surfaces if necessary in the case of curved surfaces or small details.

How important are the dimensional tolerances? (Display Path)

- Very Important
    - Selects printer with best average tolerance
- Less Important
    - Selects next available printer

Provides information for selecting a printer based on the positional tolerance of the printer.

Appendix C: Consent Form

Informed Consent Agreement for Participation in a Research Study

Contact Information: pbcoady@wpi.edu mjgulbin@wpi.edu jjscherick@wpi.edu catouchette@wpi.edu
Title of Research Study: Software Testing of G-Code and Printing Automation Program
Sponsors: Professor David Brown, Professor Pradeep Radhakrishnan

Thank you for offering to participate in this study. We are conducting research on the effectiveness of an automated slicing software used for additive FDM Printing. The study is being conducted by the WPI MQP group of Parker Coady (CS), Matthew Gulbin (CS), James Scherick (CS), and Collin Touchette (ME). The project is advised by Professor David Brown (CS) and Professor Pradeep Radhakrishnan (ME).

Purpose of the study:
This study is designed to test the user experience with a software application developed by our team so that we may improve it. The application is meant to provide novice 3D printing users a tool that simplifies one of the steps of the process.

Procedures to be followed:
Over the course of this study you will be asked to use the program to slice an .stl model, answering questions with regards to its use. After using this software, you will then be asked some questions to gather your thoughts about the experience with the program. Participation in this study should take no more than 30 minutes.

Risks to study participants:
There are no known risks associated with participation in this study.

Benefits to research participants and others:
There is no direct benefit for participation in this study.

Record keeping and confidentiality:
Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators, the sponsor or it's designee and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data that identify you by name. Any publication or presentation of the data will not identify you.

Compensation in the event of injury:
This research involves minimal risk of injury or harm. You do not give up any of your legal rights by signing this statement.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact:
You may obtain a more detailed explanation of its goals by emailing the MQP group (pbcoady@wpi.edu, mjgulbin@wpi.edu, jjscherick@wpi.edu, and catouchette@wpi.edu). Furthermore, you may contact the IRB Manager (Ruth McKeogh, Tel. 508 831-6699, Email: irb@wpi.edu) or the Human Protection Administrator (Gabriel Johnson, Tel. 508-831-4989, Email: gjohnson@wpi.edu).

Your participation in this research is voluntary. Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit.

By signing below, you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

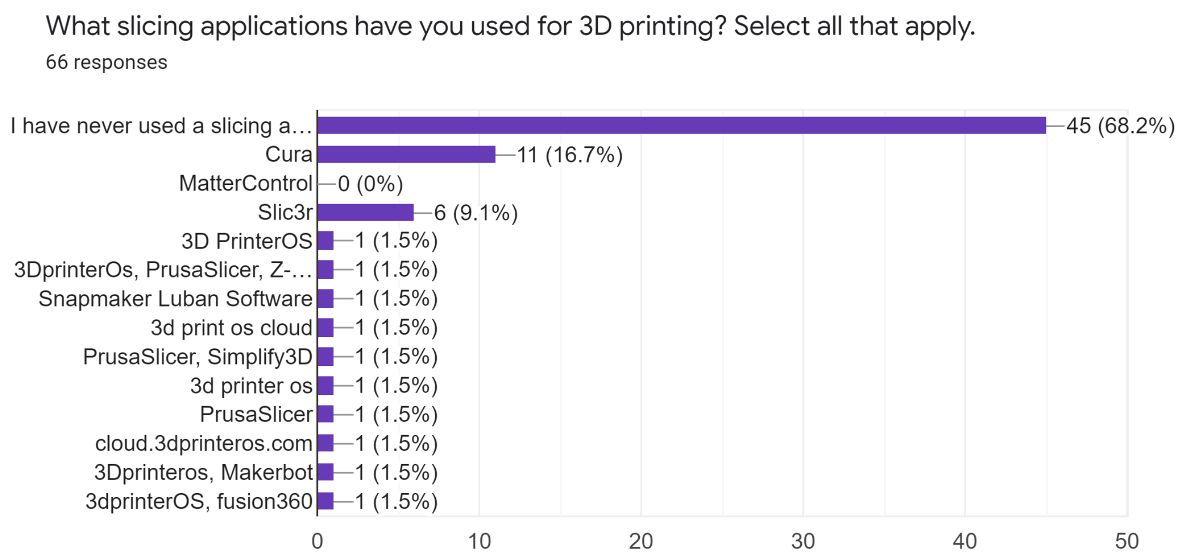
____________________________ Date: ____________________
Study Participant Signature

____
Study Participant Name (Please print)

______________________________________ Date: ____________________
Signature of Person who explained this study

Appendix D: List of Survey Free Responses

What slicing applications have you used for 3D printing?



Please indicate the duration (in seconds or minutes) it took to complete each step: 'Answering Questions' about the file being 3D Printed and the software 'Generating G-Code'

Very Experienced:

- 12 seconds
- Answering questions took a couple seconds (~5 seconds), G-Code generation took 12 minutes (720 seconds)
- 10-15 seconds

Moderately Experienced:

- 6 minutes
- Answering Questions: ~2min, loading file: ~10min, generating g-code: ~30sec, saving solution ~2min
- 1min
- Maybe 10 seconds
- about 10 seconds or so.
- 1 minute for the display model
- None of the steps took much time except for actually generating the G-code (2 minutes)

- questions: 1)import STL file=30 seconds 2)Mechanical system or display= 10 seconds 3)specific type of force= 1min 4) axis force is acting upon= 1min 5) dimensional tolerances importance= 10seconds
- 4 minutes
- 5 minutes
- seconds for the question, then about 2 minutes to load
- 30s
- 10 min

Not At All Experienced:

- 0000002 minutes
- 2 minutes
- 1 minute
- Around 20 min
- 3min
- 35 sec for answering question and 2 sec for generating G-Code
- 3 minutes
- 20 seconds
- Answering questions took about 2 minutes, generating G-code took about 1 minute
- 3:59
- Approx. 6 min
- 45sec-2min
- 9
- Answering questions: <1 minute, Generating 5 minutes
- Answering Qs - 1 minute. Generating Code -
- 7 minutes
- 3 mins for the questions asked about the file, 32 mins for the generating, and a second to generate and save the gcode
- Ansering Q's: 3 minutes, Generating G-Code: 4 minutes
- roughly a minute or a minute and a half
- 1 minute
- Answering questions took 1 minute, the file being 3D Printed took 3 minutes and the software generating g code took a 15 seconds. The print time states it is: 39.967 mins for the eyeball.
- 2 minutes
- Answering Questions=1 min and Generating G-Code= 2.5 minutes
- 2 minutes
- more than 5 minutes

- 5 minutes
- about .5 seconds for answering questions each, about 1 minute for the solution generator, and less than 2 seconds to generate the G-Code
- About three minutes to download and two to answer questions.
- 7 minutes
- 3 minutes
- It only took like 30 seconds to answer the questions and it took the program about 3 minutes for to generate the code
- Answering Questions: 1 minutes, Generating G-code: 3 minutes,
- Few seconds for answering questions and 3 mins for generating g-code
- 3 minutes
- 1 min and 30 seconds
- ~1 min
- 30 seconds
- 4 minutes
- Question 1: 4 sec; Question 2: 11 sec; Question 3: 32 sec; Question 4; 25 sec;
- 5 minutes and 3 minutes
- "Answering Questions" - about 15 seconds "Generating G-Code" - About 1 minute
- about 5 minutes
- 1 minute
- 10 min
- 2 minutes
- Answering Questions: 30 s ; Generating G-code: 100+ min
- a few minute
- 8
- 5 minutes
- Answering the questions took me about 1 minute, and generating the g-code took about 4 minutes.
- 15 mins for questions, 20 for G-code

If you think another orientation would have been optimal, please describe the orientation you would have preferred and why.

Very Experienced:

- I've always printed gears flat, printing at 45 degrees as suggested is a huge waste of time and filament (unnecessary support material). What I would've liked to see is a gear generated flat, with 10-20% infill the body and 100% infill inside the teeth and around the bore.

Moderately Experienced:

- I used the sample gear provided in the STL download, and I would have imagined that printing it with the flat gear face towards the bed of the printer would have been best, but this was not generated by the program, which said that "gears are printed at a 45 degree angle from the bed for strength." Three of the four solutions were like this and the other was standing on the gear teeth. If this is the best way of printing, then I guess I'd go with that, but it was not my initial understanding because of sagging that might happen with the overhanging 45 degree angle.
- I like the idea of the other potential orientations given but the fist solution was good
- No the first one was optimal
- The orientations which would have been optimal is not totally clear. I do not know a whole lot about 3D printing, much more about CNC milling. Generally, the solutions were very tall from the program, and that does make sense for decreasing print time so less distance is traveled. My second run on the program gave me flat laid down part.
- n/a
- I would have preferred the gear to print flat on the print surface
- yes, better tooth for gear

Not At All Experienced:

- N/A
- no
- My cylindrical part was laying down on the gridded plane, when it should have been upright with the circular face of the cylinder normal to the ground.
- The part was cylindrical and by default it had it on it's side instead of the flat end
- I will put it on the yz plane because that way the print can have better support
- n/a
- I would have preferred for the it to print with the bottom of the part, on the bottom of the print bed, instead it printed on one of the sides
- free orientation with degree entered, and also some other shape function orientation(instead of line) would be really professional.

What did you find the most difficult in using this program?

Very Experienced:

- The 3d viewer is not very good, when swapping suggested orientations I had it zoom out like crazy and was not able to get it back to where I wanted it.

  No way to delete a part after importing without restarting the program?
- It took to long to use (12 minutes) to generate G-Code
- laggy zooming

Moderately Experienced:

- Not comprehending how the support material will be included.
- I didn't find anything particularly difficult in this program, but some things took a long time to load, like the save solution option, which made the program not respond for a minute or so. The load time was relatively long, but I feel like this is to be expected.

Also, the Estimated Loading Time said 48 minutes, but this was way over what it was (about 10).

- The part color being a dark blue makes it hard to see overhangs and shadows
- The most difficult one would be for someone with no 3D printing experience or background on that
- The rotation was hard to use
- None of it was difficult
- Nothing, for my STL file, it was very simple and intuitive
- Most difficult in using the program was answering the force and axis questions. The purpose of your software has excellent pros and cons at the end for solutions, but perhaps showing some pros and cons for earlier steps would be helpful for less experienced users.
- Using mouse controls to orient the part was very slow
- I did not know how to answer the questions that were asked. If this was my piece then I might have know how to answer the questions.
- Nothing was difficult, would've liked to have more options
- it's pretty good
- too long

Not At All Experienced:

- Nothing
- instructions
- I didn't understand some of the terms used. It might be helpful to provide a description or definition.
- my lack of knowledge limits my ease in proceeding through this program
- Figuring out what forces went where
- none, it was very easy
- No
- Understanding the different reasons for selecting slicing parameters, having never used software like this before.
- Getting to the instructions, but that still very easy
- Selecting the direction of force with the X Y and Z axis adjusters was a bit difficult at first but once I realized how it moved the force it was simple
- I didn't find anything to be difficult about using this program.
- Not knowing which button to press next while using the application and what each button's function was
- Doing the g code
- choosing a solution
- Getting the force axis in the correct direction
- nothing
- If a file couldn't open the program crashed
- I do not have any background in splicing or 3D printing, so I needed to read the help section first to understand how to answer each of the questions.
- Getting the right axis for the direction of the force took a little while to figure out
- Not knowing the correct answers to the questions.
- I just have little experience in splicing so I was unsure of what to expect and if everything came out correctly. Based on my little experience, the software was very easy to use.

- Nothing was particularly difficult
- It was hard to decipher which directions were negative (-1) and which were positive (+1)
- If you didn't know the dimensions of the part off the bat you had to look them up to answer the questions.
- crashes so much and can only work for the provided part
- Properly answering some of the questions, given my inexperience with 3D printing
- I think you have to understand the question so for users that do not, that could be the difficulty
- I did not find much difficulty.
- I was a bit unsure whether there were any more steps at the last slide of the program
- It was slower than I expected
- Nothing really. The user interface was really easy to understand and the question were presented clearly.
- Describing if tolerances would be greater or less than strength (I wasn't super familiar with the original part)
- Everything seemed intuititve
- I do not know what I'm doing, especially with parts that aren't mine
- Their was nothing really difficult
- Downloading the program file took a while, but I'm not sure if it was just my computer/wifi
- Overall I found it very easy.
- I have no experience with slicing so I'm not exactly sure how the program works
- Moving the part around in the left window, make it the same as moving them around in Solidworks
- I thought it was pretty easy to use, however I am not very familiar with the subject of 3D printing. The most difficult part was trying to figure out what the program was talking about.
- Nothing, it was a very simple and straightforward program
- Nothing really, the program worked very smoothly.
- Needing to have an STL file
- I have not used this program yet so just getting used to it.
- understanding the terms in those answers
- n/a, it's all very intuitive
- Getting access to it because I am off campus
- The only difficult part was moving the piece around in the solution view was a little bit laggy but answering all the questions and generating the code was very easy to follow.
- when importing the STL file, it didn't let me go into the STL folder to get the individual files, also the orientation question view was confusing

What aspect of the program did you find easiest to use?

Very Experienced:

- The questionnaire was very easy!
- Questions were straightforward and simple
- simply setup, easy to use in general, simple layout

Moderately Experienced:

- The entire process was super simple. I am very pleased with it.
- I thought that the program was very straight-forward and gave me a linear process to follow. This made it easy to use.
- uploading, questions
- I really liked the use of axis orientation, that was really nice, also the rendering of the objective to give an idea of how it looks like and the placement of the axis
- The import process, pros/cons, and the different orientations was nice
- Checking the different solutions
- Program was very responsive, just took a while to generate the code
- I found the slier overall easier to use then 3D print OS at WPI, which was relieving, but it was not the most educational. It is unfriendly to users who have less background, but definitely gets the job done. Easiest to use was the interface with questions which have prepared responses, but these prepared responses and knowing what answers to select via a guideline would make it more user friendly.
- Importing the STL
- All of the program except its download was easy to use.
- nothing was difficult
- the question part
- yes

Not At All Experienced:

- no need to think
- Super simple UI, easy to understand.
- I think everything was very easy to use, I am mostly confused because I have never needed to use anything like this before
- Importing the files and all the work being done for me
- all of it was very simple
- N/A
- Selecting the different views to see how the part could be sliced.
- Selecting a part orientation at the end
- The questions were specific and clear
- Answering the questions was very easy because they were straightforward and quick to answer.
- The questions were straightforward and easy to understand
- Camera movement and angles
- uploading/modifying object
- interface/answering questions and importing STL file
- the options for parts and and types of forces that it could undergo
- The questions asked about your part are easy to answer
- I found the step by step process clean and easy to follow.
- Answering all the questions was easy
- Loading the STL file.
- The questions were self explanatory and to the point/direct therefore easy to answer/interpret.

- It was easy to upload files
- Answering the questions about the forces acting upon my object
- Uploading the file.
- the user interaction
- Importing the STL files
- the efficiency and time!
- The questions were simple and understandable.
- The UI response time was quick, which helped me to quickly go back and forth between steps incase I needed to change some of my answers
- The instructions were easy to follow
- Answering the questions was really straight forward and easy
- easy, straight forward questions!
- Every button is laid out well and easy to follow
- It was easy to follow the instructions on screen
- The program as a whole
- Everything was labeled clearly and easy to see
- I think the easiest part was answering the questions about the file that needed to be sliced.
- I found it easy to generate the G-code
- Simple and straightforward questions
- the interface was very straight forward
- Choosing the best g code for your program
- Strength selection
- The upload was smooth and instant.
- Questions are simple and straight forward, making it very hard to get confused
- The way it guided you through.
- put the files in and click several bottoms and then we got G code ready to print, that's sick!!! Love it.
- answering the questions
- Answering the questions that were asked at the beginning once you uploaded the fill.
- User friendly
- Uploading the file and answering the questions
- the G-code generation

What did you like about this program?

Very Experienced:

- The idea is very solid, but I'd definitely need to print some test parts to be convinced the material support cost and the increased print times are worth the strength gains.

  Love love the "force score," though I'd like a bit more transparency or clarity on how it is calculated/
- Looked nice and was easy to use
- very simple, readable, not much going on

Moderately Experienced:

- I love the pros and cons feature.
- I liked the questions part that allowed me to easily inform the program how I intended to use the part. I've used Cura before and I don't remember this kind of option.
- Very simple to use. The mechanical questions about what type of forces and where on the part the forces will be were cool to use and helpful. Wish I could add multiple at once though. Many mechanical parts incorporate more than just one force in one direction.
- I really like its simplicity and I also like the pros and cons, its something lots of people don't think about which gives a general understanding about the part.
- the orientation options were nice it doesnt have that in cura
- It's simple
- The ability to see why each orientation is ranked in its specific way
- I like everything about the program, I can tell it is a quick offline slicer application, which is nice compared to WPI 3D print OS cloud, which runs slowly as of the nature of its cloud. I like the almost immediate generation of g code upon clicking generate g code in on the final step.
- It has a very simple layout
- I like the simplicity of this program
- Fairly straightforward program
- The question part to show some info and slicing speed is pretty fast as well
- I didnt like the program

Not At All Experienced:

- everything
- Seems like it could be very helpful in determining the proper printing orientation. I also like how it provides pros and cons for each orientation and the ranking rationale.
- I didn't have to do much thinking to get it to work
- It gave me the best orientation to print and then gave me the times
- gives you options on how it's made
- It is simple and easy to use.
- I liked how the UI of the program was simple and easy to follow.
- Program was very simple.
- I think it's a great idea and could be super helpful for people (like myself) that aren't overly familiar with 3D printing
- I liked the ranking and pros and cons of each orientation. This was very helpful and informative.
- Great description of the purpose of the program, and made the whole process easier for a novice user
- Camera movement and angles much more free than other programs that i have used
- nice UI
- interface was easy and simple
- the step are easy to go through
- cad magic is cool and I like learning about new ways to use cad
- I liked the 3D visual that was included in the program and how clean and easy to operate the program looks.
- As someone who doesn't know much about 3D printing, having a program that can help me get the ideal properties for the application I'm using is really cool

- It was easy to download and install.
- I liked how easy and simple it was to go through the program.
- It gives you pros and cons at the end
- How easily solidworks files can be uploaded
- I appreciated having the 3D model to view.
- it look nice
- The overall simplicity in its execution
- I liked how every question was displayed and easy to understand and the software reacted appropriately
- I liked how it ranked all the solutions and listed pros and cons.
- It was quick and simple to use
- It looks very simple and straight forward
- I like the pros and cons offered for the different solutions
- presented multiple solutions!
- Simple and efficient
- It seems simple to use
- It was easy to use
- It was simple and easy to navigate
- I liked that it was quick and efficient, and that it was easy to follow. I needed very little instruction to be able to use it successfully.
- The program was easy and simple to use. There was also a help button to answer any questions.
- Gives great tips on what your part needs and does well
- I liked that the program worked relatively quickly and provided multiple solutions
- The simplicity of it
- Different solutions posted
- Everything was fast and efficient. I liked the details it gave about the print and the pros and cons list as well.
- It gives multiple options allowing for flexibility and a much higher chance for success.
- I liked how it was easy for beginners to use.
- I do like it, seriously.
- Great UI, easy to follow, very automated.
- I love how makes 3D printing much easier with the correct orientation
- Seems pretty intuitive
- The simplicity and ease of use and how each solution has a list of pros and cons
- it asked comprehensive questions

What did you dislike about this program?

Very Experienced:

- From my own experiences with 3d printing gears and brackets, laying the parts flat on the bed to as short and wide as possible is typically the best choice. Especially when printing tall and slender parts, a single skip on the stepper could ruin your entire print. Even if not recommended by the program, the "just lay it flat on the bed" orientation should always exist for the sake of the user. Again, I love the idea of the "force score" and if flat on bed with a 9/10 force score prints in 5 hours less than a 10/10 and uses a 1/4 of the material, I'd go with the flat on bed orientation.

No way to easily compare print time despite a (currently locked) field to select the printer.

- No way to see generated toolpaths or do layer by layer analysis of the toolpaths (I talk about this more in the "Did the program include enough information to help you understand the slicing process?" question.
- I felt like it didn't ask for enough information to create a successful print. For example, it never asked about printing material, so I had no way of knowing if it was printing at the right temperature or speed.
- - a bit laggy, the assessment of my part wasnt fully accurate, saying a lot of material use when only an inch thick

Moderately Experienced:

- One thing that stuck out to me was saving the file as a .GaPa file. I am only confused as to which format it is saved as and where else it can be opened.
- I thought that the 3D image on the left was hard to interact with. I liked the instructions, but it was very slow and when i paned/rotated/zoomed, there were several seconds between when I input a command and when the image updated, making it difficult to use.
- Too simple, doesn't give you options of how to print such as print speeds, infill, layer height, support settings etc. If I wanted to restart and use a different STL then I would have to close out of the whole program cause there is no option to delete or remove the first one. Also doesnt give u an option to move part around the print bed. Similarly it doesnt say the dimensions of the virtual print bed so you dont know where it might fit on yours. Also took longer than any other slicer i have used to generate the gcode.
- Maybe the fact its on a windows as a Mac user, but still not bad
- the rotation is a bit difficult
- I have no idea what the specifications of the filament temperatures are, the layer heights, wall thickness, or anything of the sort. It is too simple for me to have any confidence in the product that will be printed.
- I don't see a way to set the bed size which would help me have a better scale of everything
- I disliked the lack of immediate information for the early on questions, and I recommend making the programs little more educational if the purpose is to simplify slicing for 3D printing.
- It doesn't look like you can customize the slicer settings as well as with other softwares
- The time it took to run was around 3 minutes which I thought was long but I do not know.
- the printing orientation made no sense to me, it wouldve needed supports which I dont like dealing with
- na
- It was too slow and gave a bad orientation

Not At All Experienced:

- inability to add printers

- Would be nice to be provided the definition of some terms.
- nothing
- The forces part because it confused me at first
- nothing, very simple
- No
- Not much, it was simple and easy to use for a beginner.
- I wasn't able to change the size of the window, and my taskbar cutoff most of the "next" and "exit" buttons.
- Nothing honestly
- There wasn't anything that I disliked. If anything, the waiting time could be a little shorter, but even so, it was quicker than I expected it would be.
- Some of the button functions were unclear
- As a remote student there is some latency between action although this might not be on the programs fault maybe it could be minimized?
- kina slow, some confusing phrasing in questions
- Load speed
- some files take less than minutes but some show estimated time to be around 38 minutes
- the crashing issue
- I did not know how to answer some of the questions due to lack of experience/understanding, so I had to look of the help website which I was not able to find in the program. The help button was not very helpful for answering the questions
- The only problem I had was with selecting the axis
- The questions were unclear because I am not familiar with 3D printing or modeling.
- I did not strongly dislike anything in the program.
- There wasn't anything I didn't like about the program
- Not knowing which was the positive and negative direction for the XYZ axis
- Maybe offer different color options for the parts to be presented in.
- It run too slow crash often only work for the provided part
- The inability to resize the software's window
- Sometimes the back option glitches and you are stuck with zoomed in version of the part
- I disliked the very long download time of the ring gear download.
- The program was new to me, so I was unsure if the program had finished at the end
- It is pretty slow
- Nothing really
- three out of 4 of the solutions presented were essentially the same but different materials which made it difficult to initially see difference
- In my case (haven't ever done anything like this) it was hard to answer some questions.
- Not sure if this is because of remote desktop but kind of slow and the imaging can be improved
- It make my computer noisy
- Not that I disliked anything but colors are always nice to add :)
- I think there could be two different work paths, one "simple" one that you have here, but also an expanded one for more detailed designs that give more, perhaps less used options. This would probably be good for a future iteration.
- There were no aspects I really disliked. The program seemed easy to follow.
- It exits out of the program if you don't select a file the first time

- Not a part of the program, but I would have liked a small explanation as to what the .stl files provided did. That way I could be sure of the information I was entering for them.
- The wording on some questions may have been slightly confusing. Most likely due to my lack of knowledge of 3D slicing
- Not being able to reorient your STL in the beginning
- Nothing
- In the end options give the titles given that have information in them are cut off!!!!
- I did not like how I did not really realize what the purpose of the gcode was.
- Since I had a hard time understanding those answers, I kind of have no idea about what I am clicking.
- I don't think there was any drag and drop functionality for importing an STL file. Also, it took way too long to write the g-code
- N/A
- This probably would not be that helpful for low level 3d printing applications, it also had long loading time
- There was nothing I disliked about the way the program runs maybe just the aesthetics could be enhanced a bit.
- there was a question with multiple parts, it only lets you answer one then moves on.

Were the questions that the program asked you clear/easy to understand? If not, which questions were unclear and why?

Very Experienced:

- Questions were very easy.
- Yes, they were clear
- Not really, if i wanted to slice a part thats not for a mechanical purpose, wgy

Moderately Experienced:

- Very straight forward. Everything was explained in a super simple manor.
- I thought the questions were easy to understand, although I do have a somewhat mechanical background. Questions like the one about what forces will be applied to the object could be difficult for people without a mechanical background, but I understand that this program is intended for mechanical students so that's OK.
- Yes, they were clear
- It was good to understand
- Yes they were simple and easy to follow
- I think that the mechanical system/display question is a little vague, I'm unsure about which parameters are changed. Is the mechanical one printed with lower resolution? Does it affect infill?
- Yes
- Not all of the questions were clear, they were well worded and I understood them from my engineering background, but perhaps add more info on how those choices influence the final product at the end
- They were clear
- Simple to understand. Yet i did not know which choices to make

- yes, easy to understand
- yes
- clear

Not At All Experienced:

- Yes
- yes
- The questions were clear and easy to understand
- They were clear
- Let's say I select "Mechanical," but I don't want to add any force, there should be an option
- yes, they were very clear
- Yes they were seemingly clear enough for me to work it
- Yes. The questions are very clear but I don't understand why you need the answer to generate the G-Code.
- For the most part they were okay. The question asking about the curve size was hard, was the grid under the eyeball part at a certain scale?
- Yes they were.
- Yes, the questions seemed very straightforward and matched up with what was going on with the image on the left. For example, I changed the axis line and it moved on the graph on the left which was helpful for me to be able to visualize.
- Mostly, but more clarity on what "how important is tolerancing" means would be good
- The forces, what is flexurial force?
- I think the questions were, I simply did not have the background to understand what some of the question were asking, especially when asking what axis the force would be acting upon.
- Yes, everything was easy to understand
- I didn't understand the force question, the question about tolerances, or the question about orientations because I am not familiar with these concepts and did not know the correct answers.
- No questions were unclear.
- Yes they were clear.
- it is easy to understand
- All of the questions seemed clear to me
- Yes they were
- they were clear
- Yes, as long as there is understanding about this area.
- They were easy
- yes
- Yes, very clear
- Yes the questions were clear. However, I was unsure of how to answer for some of them since I was unfamiliar with the provided stl file.
- They were clear for someone with a little more knowledge on the topic than myself.
- They were great.
- clear

- Not sure which force my part causes, I think there are more than one force apply on my part, and we are suppose to click only one answer, so kind of have no clue.
- I only had trouble answering the question about if the part contained small or detailed parts since I downloaded one of the tutorial parts and was not sure the size of it.
- they were clear except for the orientation one because it was hard to move around and view

Did the program include enough information to help you understand the steps required to generate G-Code?

Very Experienced:

- No! Please let me see the generated tool paths! This is especially important for gears as I'd like to see the infill level on the teeth compared to the rest of the body.

  I'm also curious as to what kind of infill this slicer generates, is it squares, diamonds, hexes? I've found this can matter a lot when generating lower infill parts that undergo high stress.
- No, It didn't talk about printing material, which is an important element since it changes a lot of the printer settings.
- yes

Moderately Experienced:

- yes
- Yes, there was a sufficient amount of information regarding the print results.
- I thought it could have included a bit more. After the loading ended, there was just one pro listed and no cons, and the pro was the same for all 4 solutions. I was a bit confused as to why the gear was to be printed at a 45 degree angle, so more info could have been nice. In terms of the process of what happens behind the scenes to generate the g-code, there was no info I found, but this is probably not super necessary.
- Very very simple and basic information for beginners
- I feel maybe questions about layering and thickness might be good as well. Also an estimate on print time but I know that might vary depending on the printer
- Yes it did I was not confused about the process
- It helps me understand the rationale behind how the part is positioned
- No, I did not necessarily gain any more knowledge of slicing through this use of the program
- No, it had a lot of info and I previously understood slicing to an extent, but the program does not teach much about slicing to the user.
- This software would not help me learn how to slice independently
- Yes
- no

Not At All Experienced:

- Yes
- yes

- no
- possibly
- I do not know what G-code is so it didn't help me but did not not help me
- No
- I think so, the pros and cons list was a helpful summation of the options listed.
- Yes, it was easy to follow the steps to the G-code and understand what was going on.
- Not exactly, I was able to use the program after experimenting and it did not take long but after uploading a file the next steps were unclear
- Somewhat, it is still lofty but its starting to make sense, with more context and example would be better
- not really
- For the most part, I learned what things would be important to consider to generate G-Code, but I don't necessarily know the exact steps to list out.
- Yes, the process was fairly simple to complete.
- I wish it provided a little more description on understanding the steps required to generate the g-code.
- I believe so, but I am not familiar with G-Code
- Yes, I understand how this code is generated
- I think so, I just answered the questions with what I thought would be important for the eyeball stl file and the solution made sense for what I thought.
- yes!
- Yes.
- I still do not know what G-Code is but I could generate it again with other parts
- Yes it did/
- Yes, I was able to understand what variables were required to generate an acceptable code.
- There was a button option to generate the G-code so it was fairly simple.
- Yes I believe it does
- Yes
- Yeah kind of.
- Yes I learned more about G-code than I knew before by working through this tutorial.
- probably for someone with some experience, but I still have no idea what is going on lol

Were there any errors that interfered with the use of the program in a way not covered by the previous questions?

Very Experienced:
- No that I can think of!
- No Errors
- nope

Moderately Experienced:
- No
- no
- I didn't find any errors that I wasn't able to detail above.
- I did not face any

- no I used a simple part so it easily slice it
- There was no errors present that were unexpected, the lack of education to the user making selections was alright.
- n/a

Not At All Experienced:

- No
- no
- No
- No.
- There was an error with uploading the files, but that was only because I hadn't extracted the files yet. Once I did that, they were easy to import.
- No.
- Only function for the provided part, the STL files I made is not useable
- N/A
- nop
- No i did not run into any errors while using the program
- It exits out of the program if you don't select a file the first time
- G-code took way to long to compile, 100+ minutes.
- No there was not.
- no, other than what I said before it was great

Why did you select this number?

- pretty good tool
- Seems like it could be very helpful.
- The program is very sleek and simple i like it. The only thing that could be improved is the graphic interface simply for aesthetic purposes.
- I thought it was a straight-forward program that did it's job. I liked the questions it provided, too. Although, I did think there were some things that could potentially be improved, like the lag in updating the image on the left and that it became not responding a few times.
- it was easy to choose for this activity
- Great concept but major lack of specific print settings.
- Because while I may have done it wrong it did all the work for me and gave me everything I'd need to print the part
- generated the g code very easily
- I believe this is a very nice easy to use software, I really like the idea
- Because it works and simply to use but I am not sure if the G-Code can run correctly. It will be best if the program can simulate the G-Code, so the users can verify the program before putting it to the printer.
- Pretty easy to use, although I don't know what I am looking for in slicing software, it seemed fairly intuitive and good for a basic user.
- Program was simple enough to understand, and generated the best way for 3d printing
- It was quick, easy and selected an orientation I think was best

- The program was easy to use and helpful with visualizations and pros and cons of each orientation.
- Making the instructions more clear would greatly improve the application, when this element is changed my satisfaction with the program with increase
- It was fairly easy to operate as a person who did not know what g code was.
- Again, I didn't agree with the way gears were being generated: at the very least a "lay flat on bed" option should always be present.

  No way to see toolpath without using another slicing program.

  No way to delete an imported part without restarting the program.
- The slicing process took too long, and did not generate G-code that I could trust running on my computer. There was no way for me to verify printing temperature, print speed, nozzle size, or any other settings that might cause issues during printing if incorrectly set
- Seems like it would be helpful, but needs polishing
- I wasnt great, but I have nothing to compare it to.
- some stl files work fine but some take too long to process options
- I liked how the software works, but it took a while.
- It seems like a very clean and intuitive program, and I like the look of it. I am not completely satisfied because I lack some the background to understand all of the steps and questions throughout the process of using the program. Also, the program is kind of slow. When clicking between the different solutions and generating the G-code, I need to wait several moments for it to switch and load.
- The program was very quick and intuitive to use
- It was pretty good but the questions could be explained better.
- I selected 8 because it was an easy to use software with little to no wait time. But I wish it explained what it was doing behind the scenes a little more.
- It wasn't great, but it wasn't bad by any means
- It was overall a very good but the visual when trying to rotate was a little difficult
- The confusion regarding the axis direction.
- As my first time using a program such as this, it was a fine experience.
- it took way too long crash so often and most time do not work. I can not use my won file and have to use the provided file for it to work
- I liked this program's simplicity, and it makes generating G code much easier for someone like me who is inexperienced. Given that inexperience, I do not have much of a baseline for comparison with other software's.
- Very effective just the glitch that occurs when you click back from importing a part
- I had a good experience with it.
- The program was quick and easy to use, but there could be more words explaining the steps in greater detail
- I was mildly satisfied but I don't really understand the point in general
- I think the program did exactly what it said it was going to do and was really easy to use.
- very easy and gave multiple (ranked) options!
- I think I'm a more advanced user than this program is meant for, because I am comfortable placing parts in a slicer myself and I would prefer a slicer with more customizability.

- The program takes most of the thought away from the process, just takes some time
- It was simple and efficient. Not perfect, as there is always room for improvement
- It seems like it can be very useful for 3-D printing
- 9, great program, could teach user more about slicing in the process
- It was easy to use but makes my computer fan run fast
- This is a very basic software with almost no room to customize one's slicer settings.
- It was easy to use, and really simple. It has a lot of potential to add more intense functions
- This is a very cool program yet I did not have a purpose for it
- I felt that it was easy to use, and I feel like I would want to use it if I were actually 3d printing a part.
- I have never used slicing tools but this program was easy and simple to use. I was a little confused when I was asked the questions before generating the G-code.
- It gave me very useful information about the part that I can use for my project, but the wait time was slow, especially if I have to do this for several parts in a row, and it exits out when you don't select a file
- I thought it worked really well, but I also dont think I know enough about the subject material to rate it higher since there are probably things I looked over.
- Easy to use, but lacks features and print orientation doesnt make sense
- The program did as it was designed to do without any errors
- It did its job, but not as well as other slicer programs I have used.
- it's perfect
- I don't know much but the program ran smoothly.
- I like it a lot but it only takes in STL files. I don't like that but as someone who doesn't 3D print I don't know if that is just a standard file.
- it was too slow
- because it did not help me understand the gcode but everything was easier to use.
- Kind of like it, but hard time to understand those terms in the answer.
- It took too long. Probably could learn to use a slicer in that amount of time.
- I find this very helpful especially since I do not understand 3D printing super well.
- not as good as makerbot's software but still pretty good software
- It was easy to use but i do not think i will need to use it for my current 3d printing applications. i can see how it is useful though for heavy 3d printing users.
- The program was very fast and easy to use it could just use some visual enhancements
- it would be a 10 but they didn't explain well enough for someone with no background to understand.

How does this program compare to other slicing tools you may have used?

- I haven't used other slicing tools.
- N/A
- I have not used other slicing tools
- this is better than Cura for most applications
- I like how it provides multiple options and the pros and cons of each option. Other slicing tools I've used had nothing like this, so when I tried to print something in a orientation that was impossible to print, I had no idea.

- The pros and cons feature provided along a multiple solution list is extremely useful.
- I've only used Cura besides this. That had a lot more options, which is good, but it also made it more overwhelming and I never used most of the options. For someone new to 3D printing, this would be very good, but to someone looking for more control, it might be a bit limited. This seems to be the goal of the project, though.
- i have never used another slicing tool
- Very basic and simple.
- I have never used any other tools
- i havent used any others
- I do believe theres alot of potential for this software
- I don.t know
- I never used any other slicing tools before this one.
- Better than the others, there weren't any others
- I haven't used any others
- It is just as effective as some of the other slicing tools I've used, and it contains more information and in-depth analyses than others.
- I haven't used any other slicing tools in the past, therefor my knowledge is limited however it appears to be easier to understand as a novice user
- i havent used other slicing tools
- I can't say. I didn't enjoy it as much as I enjoy using Cura, but I really need physical parts to see whether an algorithmic (or AI, not sure how you did this) approach is better for part slicing that my manual techniques.
- While much simpler and more straightforward, I feel like it is missing the flexibility found in other programs.
- Haven't used anything else
- NA
- it is easier to use
- First time I've heard of slicing was from this software
- I have not used used other slicing tools
- I have not had any experience with slicing tools in the past
- I have not used other slicing tools.
- I haven't used other slicing tools
- I think that this has better functionailty but not enough aesthetic
- I have not used any other slicing tools.
- I have not used any other slicing tools so I can not say.
- slower and crash often, it is terrible
- I have not used another slicing tool before
- Have not sliced before!
- I haven't used any other
- I haven't used any others.
- There are no options or information on the specifics of the printing parameters.
- Much more intuitive, just slower
- First program I've used.
- I've never used any other slicing tools

- This is a good program to compare to the slicing tools used like 3dprinter OS at WPI. the program is faster than the cloud environment, as it is all offline, and I can see its value for personal owned printers and for cloud.
- I never use another one
- This is the lowest-end slicer I've used
- I've never used any other slicing tools so pretty well ha
- N/A have not used other programs
- It was a lot easier to use.
- I have never used any other slicing tools
- Have not used any slicing tools
- about as fast but not as developed
- No other tools for comparison
- Panning is slow color contrast is not that pleasing to look at, details get lost in the shadows
- I'm not sure
- I have not used any.
- It gives multiple answers which I like a lot and it just asks a couple of easy questions instead of forcing someone to know a bunch of information to fill in the proper formatting needed.
- worse
- n/a
- Haven't use anyone yet.
- Intuitive, simple.
- see above
- It was good. I like 3DprinterOS.
- I have not used any other slicing tools
- I have never used a slicing tool
- Do you have any recommendations for additions or improvements to the program?66 responses
- N/A
- No
- None
- none
- no
- nope
- Adding some definitions may help inexperienced users.
- I haven't have the chance to attempt printing straight from the program but that is my only concern at the moment.
- I think if the 3D image on the left is able to be interacted with, it should update faster. When I tried to zoom on it, it didn't update for a few seconds and then it zoomed out way too far.
- Mainly the blue color and highlights/shadows used on the part viewer make it difficult to actually see the part if there are a lot of angles.
- I don't think there are any I can give because this is my first time ever using a program like this
- Stated in the previous question

- It is great if it can directly transfer the G-code program to the printer.
- Adding a scale to the part viewing screens, and also the time predicted for completion was incorrect so maybe modify that estimation
- Ability to change window size, minimize or full screen
- I don't think either of the 3 orientations after the first would work at all as they have very little contact with the print bed, but I could be wrong.
- Just updating the instructions for application usage
- Ability to see toolpaths
-
- Always presenting a "lay flat on bed" orientation. I.E minimal material use, flat, and wide.
-
- Ability to delete imported parts.
-
- Ability to swap printers (I can see this is coming, but just want to leave it here).
-
- IDK if I really need this, but: Ability to select preferred infilling methods or density.
-
- Oh, one last thing that I forgot about on the previous page: The estimated loading time is very inaccurate on gear generation (Told me 48 minutes, took 2). I know how complicated estimating loading time is, so I'd personally get rid of it. OR have the estimated update as the loading bar fills.
- I would like to see a "view final settings" button for the solutions to view information like print temperature, print speed, bed settings, etc. Otherwise I can't verify that the code would run on my printer. I would also suggest adding in a "printing material" question. That way people could select the material they are printing with to get recommended settings
- no, I have never used a slicing G-code software
- option for cancelling a process
- faster slicing
- I don't have enough experience with these types of programs to really think of any suggestions
- Explain the questions better for those who are new to 3D printing.
- No.
- I have no further recommendations
- Definitely include a minimize window for the program, increase the rotation sensitivity, and increase aesthetics.
- Make the axis direction clearer.
- Make it more clear that the Ranking photos at the top of Step 3 are clickable.
- If the slicing can run faster, do not crash so much and can do files other than the provided files.
- Add the ability to resize the window, potentially the ability to generate multiple G codes at once to streamline the process.
- Look into glitch described- importing a file then hitting next, then going back to the import screen the part that is imported is zoomed in at one point
- nope good job

- Maybe an advanced tab that allows users to modify or at least view the print settings
- The ability to change the bed size
- Not really, just try and keep making it look more pleasant.
- The program could add more educational information as it progresses through the steps for completion. The slicing process is not clear to me from independent use of the program.
- not really
- More custom user settings
- Nope :)
- I think that having more options/questions would be beneficial if you had a more intricate part to slice.
- I am not familiar with any other programs so I do not have any recommendations
- Try to make it faster, have it so the window's size can change so I have have a screen for the program and directions
- Maybe improve the visualization. The remote server seemed to have a little problem displaying movement of the part.
- add more features
- Be able to manually reorient your STL
- improve ui?
- Make the program window able to be minimized, maximized, and to be resized within reason.
- make it faster
- make it more designed for people who have never used a slicing tools
- Give a little bit of explanation and examples about what each answer means. Like a pop gif.
- Maybe it was a bug on my end but it took way too long.
- no, pretty decent software
- Again, I really enjoy the ease of use and speed of the program, it could just be improved in terms of design.
- the orientation part was difficult