CS Capstone Technology Review

November 22, 2017

Molecules in 3D?! And in color!? That I can hold in my hand? No way!!!

Prepared for

Oregon State University

Victor Hsu

Prepared by

Group 63

The Polycules

Jackson Wells

Abstract

This document is intended to describe each team member's role in the project and to give an in-depth look at the technology to be used in the project. Investigation of different technologies preceding development is extremely important, doing so allows for informed decisions as to what tool or hardware will be the best fit for the project. Each piece of the project will be researched, and three potential choices for each piece will be considered. Of the three competing technologies, one will be selected for use in the project.

CONTENTS

1 Role

My individual role in the Polycules team pertains to the development of the project's ability to handle biological data files, general software development and managing communication between the computer and the 3D printer. With this, I will lay the ground work for the program to allow the other group members to familiarize themselves with the software, and to get an idea where their piece of development might be placed. Additionally, I will implement the feature that handles the input of biological data files. This feature sits at the core of the project, but is also non-functional without the other essential components. I will also be addressing communication between hardware devices, making sure the computer can correctly supply the 3D printer and multi-filament interface with the information that they need.

2 Programming Language

2.1 Overview

The selection of a programming language is a crucial decision. Programming languages exist in countless number, each having their own specific task or purpose. Some are designed to interact with other programs, others are designed for data processing or visualization. The programming language chosen to be used will dictate how we solve the problem of producing polychromatic objects and what intermediary software tools will be put to use.

2.2 Criteria

Due to the large variance between programming languages, we must consider our group's background knowledge and preferences so that an appropriate language is selected unanimously. That being said, preference and background knowledge will not be the deciding factor for language selection. A mid to high-level scripting language will be ideal for project development. Scripts are intended to handle user input and send the input to other programs for processing, which is essentially the task our project aims to accomplish. High-level languages enable us to execute powerful commands without having to get into the hairy details of an algorithm, which can slow development exponentially.

2.3 Potential Choices

2.3.1 Python

Python is a widely known and respected high-level programming language. The documentation for this language is endless, just like the packages available for use. Created 26 years ago, Python still has a major place in the world of programming and is used in a plethora of disciplines[1]. A key point of Python's popularity is its readability; file syntax forces indentation of particular sections of code, making scripts easier to read and comprehend. Availability of packages makes python a strong choice, allowing users to accomplish dense tasks in dismal amounts of code.

2.3.2 Perl

Perl is a conglomeration of programming languages, borrowing essential features from multiple sources. Like Python, Perl is a high-level scripting language, capable of pipeline management and file manipulation. Perl differs from Python in its text handling abilities, focusing more on modification than unification. This language has been around for 29 years, mainly being used in bioinformatic applications and system/network management. The documentation and number of packages available for Perl are present in much lower levels than Python, as it has not reached the same number of disciplines.

2.3.3 Bash

Bash is a free command language, basic in function and familiar to any individual that has made use of a command-line interface at some point. Running on the majority of versions of Unix, Bash is the default shell for GNU based systems and a few other operating systems [2]. This language can run files containing Bash commands, called scripts. Due to its nature, Bash is able to run any program that can be compiled or interpreted on a command-line, which is feature of almost all programs or scripts. Development of Bash is ongoing, and its current version is 4.4.

2.4 Discussion

The three choices investigated all provide varying perks to their usage. Bash is very basic and will likely work with any piece of software that we select to execute, but scripts in this language can grow advanced quick and complications may be encountered. Perl employs high power text parsing and quality system call support, while maintaining simple syntax. One major drawback to Perl, in our case, is the low number of relevant modules for 3D printing applications or pipelines. Python succeeds where Perl fails, with multiple existing packages intended for use in 3D printing.

2.5 Conclusion

Python has proven itself as the best choice of programming language to use in the project. This language will enable us to use multiple free and open-source tools uniquely supported by Python. All of our group's members have experience in Python scripting, another reason for selection. Additionally, if problems are encountered in use of Python, Bash will likely be selected for use.

3 Conversion Software

3.1 Overview

The software tools to be listed do not function solely as tools for file conversion, but this feature and the general function of each software choice will be investigated. Essentially, the software will be used to convert biological structure data files (CIF or PDB) into printable three dimensional structure files (STL or WRL). During the process of conversion, the provided structure might have to be visualized and have certain settings tweaked, particularly color or orientation.

3.2 Criteria

The conversion software to be selected will be the tool that most adequately performs our desired functions. An ideal conversion software will be open-source and free to use by the public. The software must be able to take PDB and CIF files as input, and output in STL WRL or OBJ formats. File output restrictions are less stringent due to the wide input support of software tools further down in the work-flow. Visualization of the biological data might be necessary at this step in the work-flow, to verify the integrity of the input file, so a visualization feature will be required. It is not currently known if programs further on in the project work-flow will be used to color structures originating from biological data, so support of this feature may be necessary.

3.3 Potential Choices

3.3.1 Pymol

The Python Molecular Viewer (PVM or PyMol) was developed on top of existing and well-known Python packages [3]. This means that PyMol will be fully supported by our selected programming language. PyMol is a diverse tool, supporting many advanced features that may be useful to users of our work-flow. A feature for molecule coloring scheme is included, as well as features for structure representation, and command-line support. PVM is open-source but requires a license for use. This license is free to students and educators affiliated with accredited universities, but users of the work-flow may not associated with a university. PyMol supports CIF and PDB files as input and outputs in WRL [4].

3.3.2 Jmol

Jmol is an open-source molecule viewer, supported by Windows, OSX and Unix systems [5]. This piece of software exists in many forms for usage in multiple ways. Specifically we are interested in Jmol's standalone application, which can be run from a command-line. Jmol is Python based, like PyMol, making it a viable choice for use with our selected programming language. This software supports CIF, PDB and many other biological data formats for input, and outputs in WRL [6], [7]. The program's documentation also hints at the ability to color atoms in the rendered structure, but little information is given on this subject.

3.3.3 Chimera

University of California, San Francisco produced a robust software tool for biological structure visualization called Chimera. This piece of software is designed by biochemists for biochemists, including a large array of features outside of structure manipulation and rendering [8]. Among its many features, Chimera includes the functionality to color structures that are input to the software [9]. Chimera can take PDB and CIF files as input, and outputs in WRL and STL [10]. Command-line execution and access to this program may not be possible.

3.4 Discussion

Each piece of software investigated exhibited diverse functionality and could interface effectively with our required input file types. Jmol and PyMol both are Python based, making them easier to use with our selected programming language. All programs have support for coloration of rendered structures, although Chimera and PyMol seem to address this feature in a more involved manner. Chimera is the only software of the three that supports output of STL files, but STL files will likely not be used in the project's work-flow because they fail to store color information.

3.5 Conclusion

The Python Molecular Viewer will be used in the project work-flow. The issue of licensing is not being considered, as the work-flow is intended for use by educators and their students. PyMol was selected over Jmol due to the model coloration support included in PyMol. While Jmol likely has support for this feature, it is not well documented and is suspected to be rudimentary.

4 Hardware Communication

4.1 Overview

Communication between hardware devices is an important step in the project's work-flow. In this section, modes of communication between the computer running our project and the 3D printer will be investigated. These choices are limited to the inputs of the printer we have selected for use in the project, the Zmorph 2.0S. This printer can take in object files via USB connection, SD card or network connection. No method of communication is favored by the manufacturer, all of which are listed in the product manual [11].

4.2 Criteria

The selected medium of communication between the computer and the printer should effectively and accurately transport file information. Whichever method is chosen shall cause the user to experience no confusion in operation and usage, this assumes basic computer knowledge. Availability of the medium will also be considered, as some technologies may be obscure to some.

4.3 Potential Choices

4.3.1 SD Card

The SD card is a storage technology, used most commonly in mobile devices. The card itself is very small, around 1.5 X 0.75 inches, but can hold large amounts of memory. The printer involved in our work-flow has an SD card reader built into its interface, enabling direct file transfer [12]. Files to be printed must first be transferred to the card on the computer, this will require a computer with an SD card reader.

4.3.2 LAN

Local area networks (LAN) are a form of network consisting of multiple computers within a limited physical area. The Zmorph 2.0S supports the use of LAN for input of structure files. This enables users of the printer to remotely setup and initiate prints from anywhere with an internet connection, assuming the printer itself is connected to the internet[13]. The documentation on LAN operation is very short, and provides few key details for debugging of potential issues. Connection speeds go unmentioned by Zmorph, these become important when attempting to send large object files over a network.

4.3.3 USB

This technology is what SD card technology was built on. The Zmorph 2.0S can be interfaced with via direct USB connection to the computer in use. If selected, this choice would require a single one-time action to establish a connection, with connection only being lost if physically unplugged. The price of this technology is under $10, depending on cable length, and is readily available at most electronics stores.

4.4 Discussion

SD cards are very small, making them easy to lose. The process of using an SD card with our printer could potentially get tiresome after many prints, and some computers may not have SD card readers built in. Use of a LAN would lessen the amount of work required for usage, but poses security, connectivity and transfer speed issues. The project work-flow would have to manage and verify the connection is valid during run-time and implement troubleshooting methods when disconnections occur. Direct USB connection requires minor user interaction and maintains connection with the computer a majority of the time.

4.5 Conclusion

The Zmorph product developers have provided their customers with three possible communication technologies. Of these technologies, USB connection will be used in the project for communication between the computer and 3d printer. This choice will provide our project with the best file transfer speeds and most manageable connection.

References

[1] "About python." [Online]. Available: https://www.python.org/about/

[2] C. Ramey, "Bash," May 2017. [Online]. Available: https://tiswww.case.edu/php/chet/bash/bashtop.html

[3] "Overview," Jul 2017. [Online]. Available: http://mgltools.scripps.edu/packages/pmv

[4] "File formats," Dec 2016. [Online]. Available: https://pymol.org/dokuwiki/doku.php?id=format

[5] "Jmol: an open-source browser-based html5 viewer and stand-alone java viewer for chemical structures in 3d." [Online]. Available: http://jmol.sourceforge.net/

[6] "Jmol input file types." [Online]. Available: https://sourceforge.net/p/jmol/code/HEAD/tree/trunk/Jmol-datafiles/

[7] "File formats and 3d objects," Oct 2016. [Online]. Available: http://wiki.jmol.org/index.php/File_formats/3D_Objects

[8] "Chimera." [Online]. Available: http://www.cgl.ucsf.edu/chimera/

[9] "Coloring in chimera," Jun 2011. [Online]. Available: http://www.cgl.ucsf.edu/chimera/current/docs/UsersGuide/coloring.html

[10] "Exporting a scene," Nov 2016. [Online]. Available: https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/export.html

[11] *Zmorph Personal Fabricator User Manual*, Zmorph.

[12] M. Leniak, "Where is the micro sd card?" Sep 2017. [Online]. Available: https://support.zmorph3d.com/hc/en-us/articles/207313455-Where-is-the-Micro-SD-Card-

[13] ——, "How to connect and manage your zmorph machine remotely," Sep 2017. [Online]. Available: https://support.zmorph3d.com/hc/en-us/articles/208882325-How-to-Connect-and-Manage-your-ZMorph-machine-Remotely