**Table of Contents**

1. 1  Introduction ........................................................................................................................................ 4
2. 2  Background information ..................................................................................................................... 4
3. 3  Basic program operation .................................................................................................................... 5
4. 4  Requirements ..................................................................................................................................... 6
   1. 4.1  Business requirements .............................................................................................................6
   2. 4.2  User requirements ....................................................................................................................6
   3. 4.3  System Requirements...............................................................................................................6

**1 Introduction**

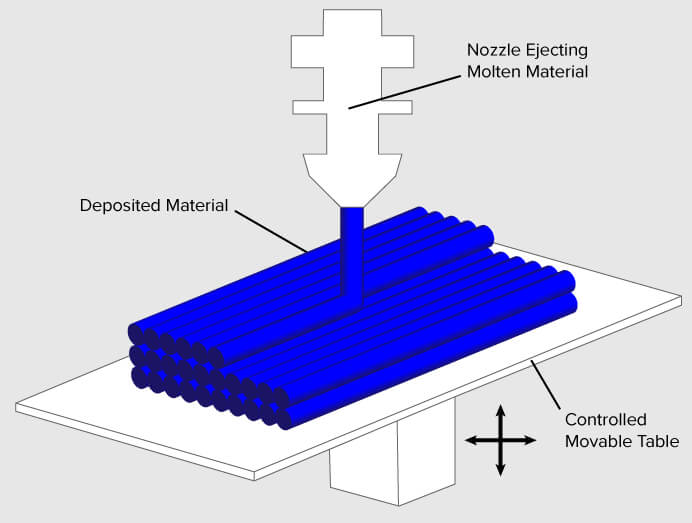
Jack started with a hobby of 3D printing with a single printer. Quickly friends and family started requesting various prints for which they paid, resulting in small profit. This allowed him to buy another printer, and another, and soon he was operating a small print farm. Offering his services online and via folders allowing him to grow the business. As the business grew, he also got different printers and modified some. Allowing him to print multi-color and print in special materials.

Quickly it got to a point where he felt printers were not running at the most optimal rate. While completing prints and handling printers still required manual activity, he figured he could optimize printing activity by writing a piece of software that allowed him to properly schedule the printers.

In creating this program and assignment certain liberties were taken to have an assignment that is both close to reality and still workable as a school assignment. Please keep this in mind when examining the working of this program.

**2 Background information**

FDM printers operate using large spools of filament which is fed through a tube to a hot end. The hot end will melt the filament and then using X, Y and Z coordinate instructions move along a set path. This results in an object being printed layer by layer.



A spool of a kilogram tends to have about 330 meters. A small print takes up about 2-3 meters of filament. Though this can vary from print to print.

PLA and PETG are the most common filaments. Where PETG requires the printer to run at higher temperatures. ABS requires a printer that is housed to prevent warping, ABS is resistant to higher temperatures. Though a printer can easily be placed in housing without much trouble. Similarly, printers can also be upgraded to carry more colors.

**3 Basic program operation**

Though some 3d printers can be networked, most are either USB connected or use micro-SD cards. In this case the number of 3d printers is too large to connect them to an actual computer. Since Jack focusses on selling them same small prints over and over, these can all be maintained on the same SD cards. Specialty prints are done on separate printers not in the system.

The program is designed to be simple in operation. Printers run on SD cards that contain all the possible prints he wants to do. A lot of orders are for existing things and only occasionally do new things come up. Whenever a new print shows up a new SD card is made with that the new print and swapped out (that SD card is then also updated), this process is not part of the software.

When the printer tells Jack what to print is assumes that print will happen. When a print is done, he tells the program a print is done. On completion the computer immediately tells him which print to start next on that printer, or if maintenance needs to be done. It will also tell him if he needs to swap out a filament spool for the next print.

Currently, if a print task fails, Jack still marks the printer as done and it will follow the usual process. He will then manually add the print to the queue again. This is not an ideal situation and something that needs to be fixed in the future.

For convenience all possible prints and printers are managed in JSON files. The reason for this is that while adding printers to the system is not that exciting and doesn’t happen a lot. The JSON contains the name of the print, GCODE filename, and printing time. This estimated printing time is generated by the slicing software (software that generates the Gcode file).

When adding a printing task to the system the print is chosen and the filament is chosen for that print. This will add it to the queue. The system should choose prints that are viable to do and ideally don’t require a filament change, unless there is no other choice.

In the future Jack would like to introduce an alternative method of scheduling where small remaining spools can be used up efficiently.

**4 Requirements**

To assist himself in writing the software he created the following requirements for himself. Since he is the only one using this system he has kept it simple and not too complicated. Favoring a program that can run on a single computer and runs a simple text interface so it can be run on a small computer in the shed dedicated to the print farm. Additionally, all requirements are must as the focus was on what was specifically needed.

**4.1 Business requirements**

|  |  |
| --- | --- |
| **Code Description** | |
| **B1** | Spend less time figuring which print to start next. |
| **B2** | Maintain an overview of all active printers. |
| **B3** | Maintain an overview of all print tasks. |
| **B4** | Get an overview of which spools need to be ordered. |
| **B5** | Have a central list of all available prints. |
| **B6** | Be able to get the most out of leftover printing spools |

**4.2 User requirements**

|  |  |  |
| --- | --- | --- |
| **Code Description Source** | | |
| **U1** | Add new printer to the system | B2 |
| **U2** | Add new print to the system | B2 |
| **U3** | Add a print task to the system | B3 |
| **U4** | Register a print completion | B1 |
| **U5** | Be informed on which printer to start a print task. | B1 |
| **U6** | Operation should be done using menu numbers for quick operation. | B1 |
| **U7** | Get information all printers and current spools and prints | B3 |
| **U8** | Get a list of all spools and their current length. | B4 |
| **U9** | Have a central list of all available prints. | B5 |
| **U10** | Be able to select a printing strategy related to spool usage. | B6 |

**4.3 System Requirements**

|  |  |  |
| --- | --- | --- |
| **Code Description Source** | | |
| **NF1** | Runs on one computer. |  |
| **NF2** | Does not need to interact with the printers directly. |  |
| **NF3** | Data is stored locally in simple files. |  |
| **NF4** | Interface is entirely text based. |  |

|  |  |  |
| --- | --- | --- |
| **Code Description Source** | | |
| **F1** | Display a menu of options with number input. | U6 |
| **F2** | List all printers with their current spools and jobs. | U7 |
| **F3** | List all available prints | U9 |
| **F4** | List all available Spools and their current length | U8 |
| **F5** | Add task to queue using numberpad. Selecting print, filament type and color. | U3, U6 |

|  |  |  |
| --- | --- | --- |
| **F6** | Start print queue which will try to assign a print to each printer. | U5 |
| **F7** | Register print completion which reduces spool length and selects a new print. | U4, U5 |
| **F8** | Register print failure which returns the print to the queue, reduces spool length, and selects a new print. | U5 |
| **F9\*** | Select different printing strategy. | U10 |

**5 Testing**

To support development of this program a *black box test suite* has been written. It is not the most elegant approach to testing but allows for testing the working of the program while leaving you free to change the code in any way you see fit.

These tests are not complete and great. There is a markdown file to explain matters.

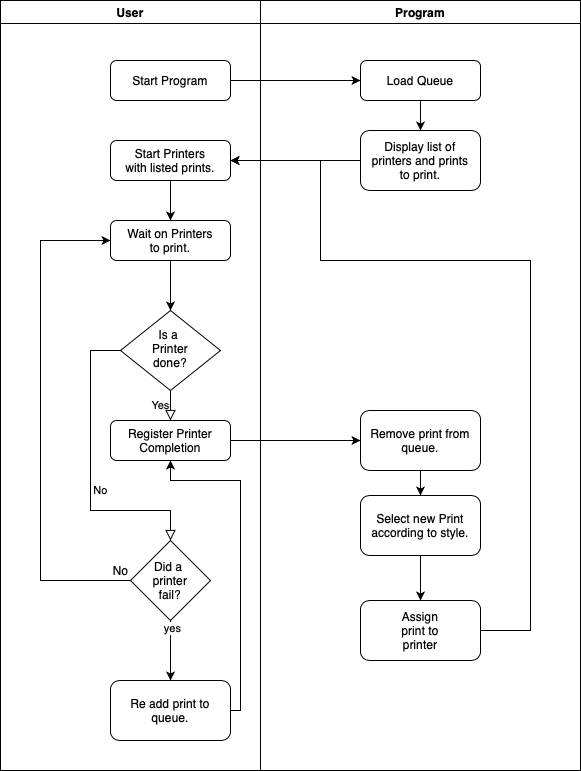
These tests will help you during development but it is up to you to prove that the program still works as it should. This might require writing new proper tests.

**6 Diagrams**

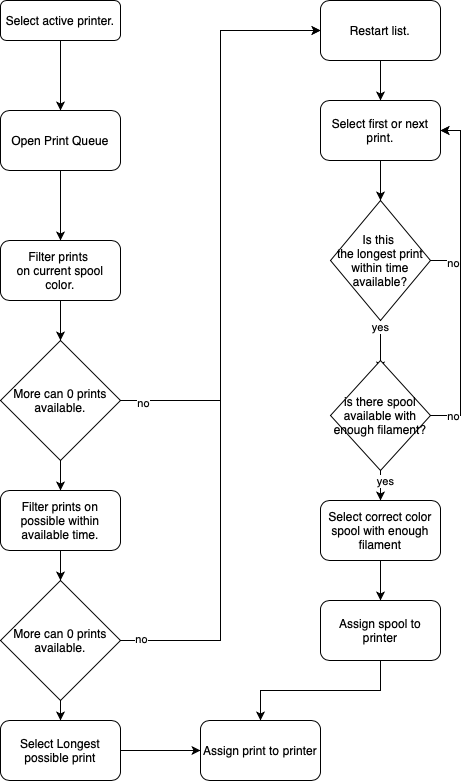
In figure 1 we see a general overview of how the program is used. Once the program is started and things are added to the queue, it waits for Jack to register a completion of a print.

For the possible strategies we can see the 2 main strategies in how a print is selected for a printer. Figure 2 has already been implemented while figure 3 has not yet been implemented.

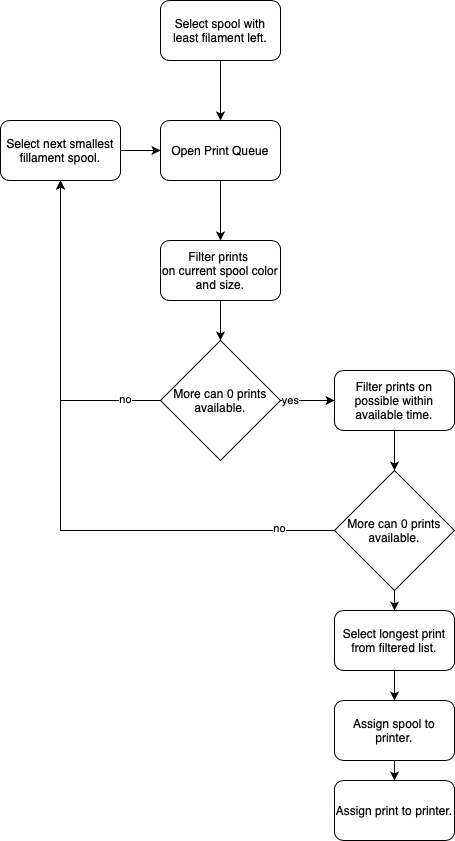
In both figures there is a step related to available time. This step has been dropped as it added unneeded complications to the creation of this program.



*Figure 1 General operating procedure*



*Figure 2 Less Spool Changes*



*Figure 3 Optimal Spool Usage*