

---

---

# Status Report: VandalForge - Software Team

Vulcan Team • 01.26.2018

---

# Overview

## Recent progress

- Gcode produced by Slic3r
- Modifying the Slic3r User Interface
- OctoPrint Modifications

## Future Plans

- Changing print-path algorithm
  - More UI modifications and upgrades
-

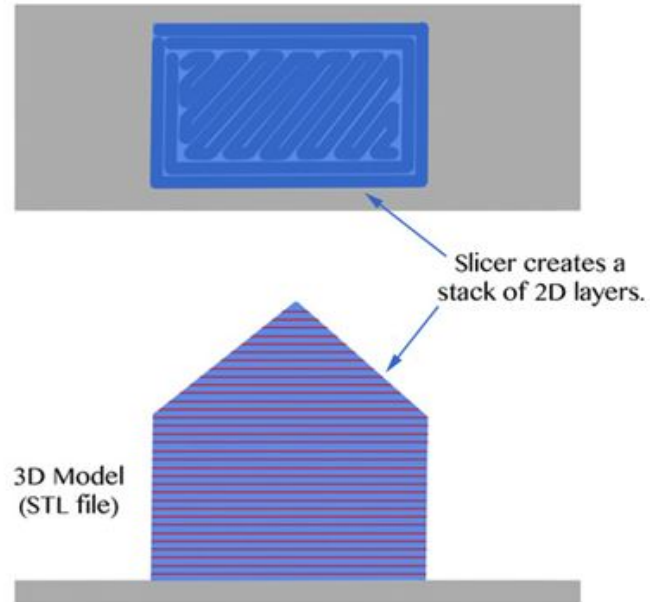
---

# Slic3r

---

---

# Slic3r: What it does



---

# Slic3r Progress - Gcode

## Original Code

```
G1 X123.138 Y80.654 F7800.000  
G1 E2.00000 F2400.00000  
G1 F1800  
G1 X124.247 Y79.545 E2.04868  
G1 X124.968 Y79.364 E2.07171  
G1 X124.968 Y78.824 E2.08845  
G1 Z0.650 F7800.000  
G1 E0.08845 F2400.00000  
G92 E0  
G1 X26.050 Y148.950 F7800.000
```

## Updated Code

```
G0 X123.138 Y80.654  
G1 X124.247 Y79.545  
G1 X124.968 Y79.364  
G1 X124.968 Y78.824  
G0 Z0.650  
G0 X26.050 Y148.950
```



---

# Slic3r Progress - Algorithm

## Problems Faced

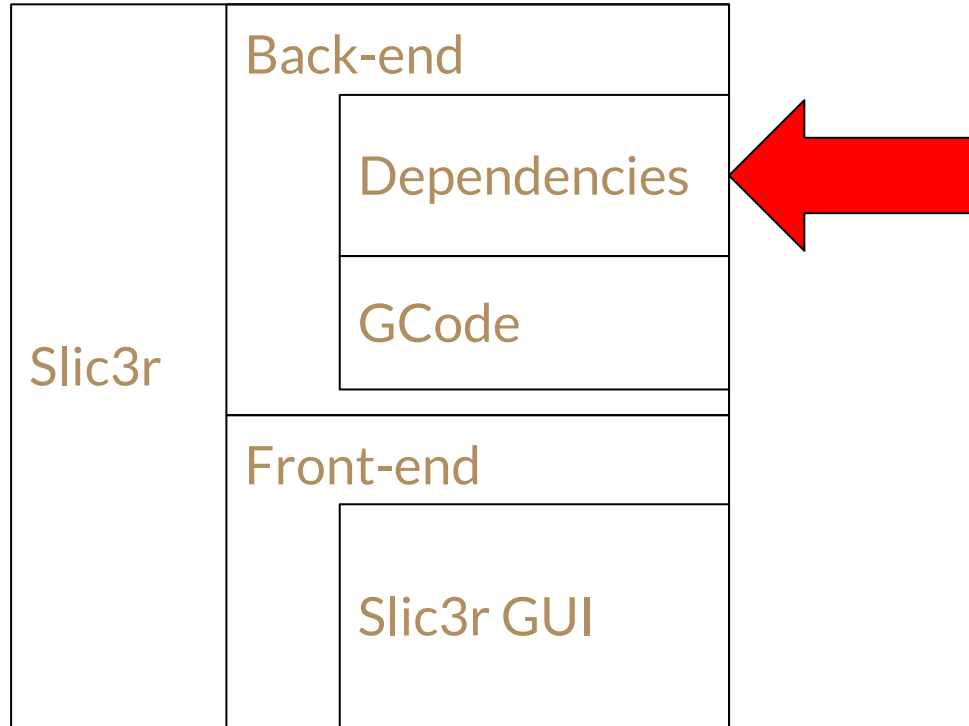
- Current algorithm designed for plastic printers uses too many G0 Commands

## Future Work

- Finding and implementing an algorithm that utilizes longest path.
-

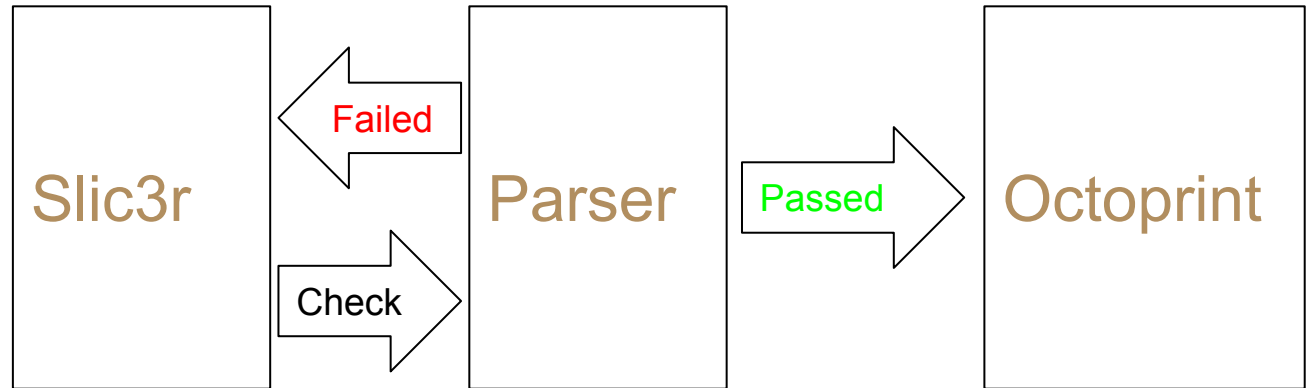
---

# Slic3r - Dependencies



---

# Slic3r - File Testing (Parser)





---

# Slic3r Progress - Creating new tutorial

## Problems Faced

- Current tutorial on how to install dependencies and run Slic3r is not optimal.
- Primarily a lack of clarity, but minor unprofessional aspects, such as typos, litter the current tutorial.

## Solution Proposed

- Fully understanding the install process, taking this knowledge to write a new, more accessible, and more verbose tutorial.
-

---

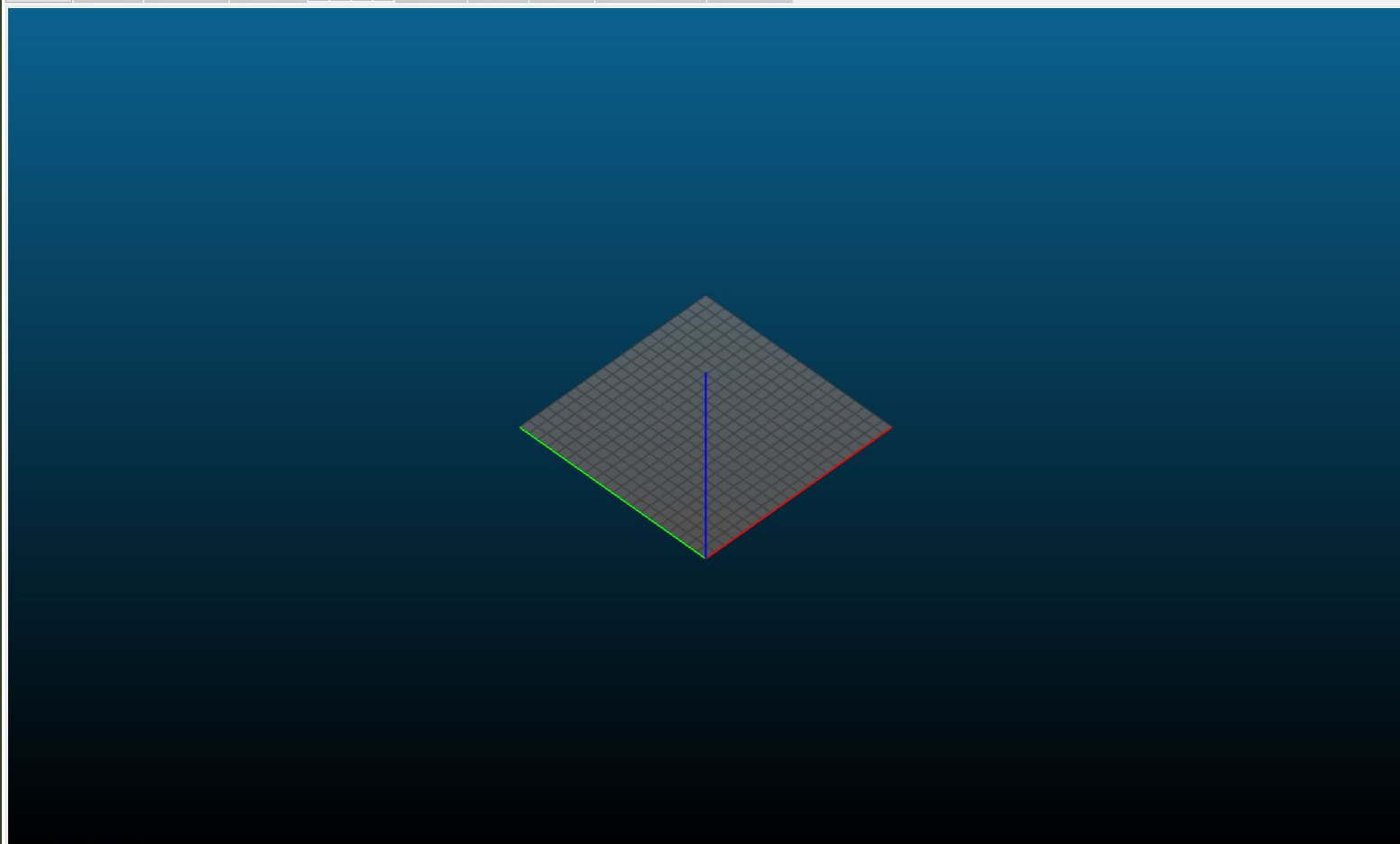
# Slic3r Progress - GUI

## Problems Faced

- The code for the GUI is written in PERL, an inconvenient language to work with.
- There is a lot of code to look at.

## Solutions Proposed

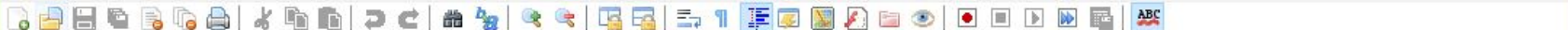
- Focus on goals needed to be completed and look at very small and specific portions of code.
-



Print settings: - default -  
Filament: - default -  
Printer: - default -  
Export STL... Export G-code...

Support material  
Generate support material:

Info  
Object:   
Copies:  Size:   
Volume:  Facets:   
Materials:  Manifold:



config.t x 2DBed.pm x OptionsGroup.pm x 2DToolpaths.pm x MainFrame.pm x

```
22     my ($class) = @_;  
23  
24     my $self = $class->SUPER::new(undef, -1, 'Slic3r', wxDefaultPosition, wxDefaultSize, wxDEFAULT_FRAME_STYLE);  
25     if ($^O eq 'MSWin32') {  
26         $self->SetIcon(Wx::Icon->new($Slic3r::var->("Slic3r.ico"), wxBITMAP_TYPE_ICO));  
27     } else {  
28         $self->SetIcon(Wx::Icon->new($Slic3r::var->("Slic3r_128px.png"), wxBITMAP_TYPE_PNG));  
29     }  
30  
31     $self->{loaded} = 0;  
32     $self->{preset_editor_tabs} = {}; # group => panel  
33  
34     # initialize tabpanel and menubar  
35     $self->_init_tabpanel;  
36     $self->_init_menubar;  
37  
38     # set default tooltip timer in msec  
39     # SetAutoPop supposedly accepts long integers but some bug doesn't allow for larger values  
40     # (SetAutoPop is not available on GTK.)  
41     eval { Wx::ToolTip::SetAutoPop(32767) };  
42  
43     # initialize status bar  
44     $self->{statusbar} = Slic3r::GUI::ProgressStatusBar->new($self, -1);  
45     $self->{statusbar}->SetStatusText("Version $Slic3r::VERSION - Remember to check for updates at http://slic3r.org/");  
46     $self->SetStatusBar($self->{statusbar});  
47  
48     $self->{loaded} = 1;  
49  
50     # initialize layout  
51     {  
52         my $sizer = Wx::BoxSizer->new(wxVERTICAL);  
53         $sizer->Add($self->{tabpanel}, 1, wxEXPAND);  
54         $sizer->SetSizeHints($self);  
55         $self->SetSizer($sizer);  
56         $self->Fit;
```

# Editing the GUI

The image shows a Perl IDE window titled "C:\Users\James\Downloads\VandalForge-master\VandalForge-master\Slic3r\lib\Slic3r\GUI\MainFrame.pm - Notepad++". The code defines a menu structure for a GUI application. The menu items and their corresponding code are as follows:

Menu Item	Code
Open STL/OBJ/AMF/3MF...	<code>wxTheApp-&gt;append_menu_item(\$fileMenu, "Open STL/OBJ/AMF/3MF...\tCtrl+O", 'Open a model', sub { \$self-&gt;{plater}-&gt;add if \$self-&gt;{plater}; }, undef, 'brick_add.png');</code>
Open 2.5D TIN mesh...	<code>wxTheApp-&gt;append_menu_item(\$fileMenu, "Open 2.5D TIN mesh...", 'Import a 2.5D TIN mesh', sub { \$self-&gt;{plater}-&gt;add_tin if \$self-&gt;{plater}; }, undef, 'map_add.png');</code>
Load Config...	<code>wxTheApp-&gt;append_menu_item(\$fileMenu, "&amp;Load Config...\tCtrl+L", 'Load exported configuration file', sub { \$self-&gt;load_config_file; }, undef, 'plugin_add.png');</code>
Export Config...	<code>wxTheApp-&gt;append_menu_item(\$fileMenu, "&amp;Export Config...\tCtrl+E", 'Export current configuration to file', sub { \$self-&gt;export_config; }, undef, 'plugin_go.png');</code>
Load Config Bundle...	<code>wxTheApp-&gt;append_menu_item(\$fileMenu, "&amp;Load Config Bundle...", 'Load presets from a bundle', sub { \$self-&gt;load_configbundle; }, undef, 'lorry_add.png');</code>
Export Config Bundle...	<code>wxTheApp-&gt;append_menu_item(\$fileMenu, "&amp;Export Config Bundle...", 'Export all presets to file', sub { \$self-&gt;export_configbundle; }, undef, 'lorry_go.png');</code>
Quick Slice...	<code>wxTheApp-&gt;append_menu_item(\$fileMenu, "Q&amp;uick Slice...\tCtrl+U", 'Slice file', sub { wxTheApp-&gt;CallAfter(sub { \$self-&gt;quick_slice; \$repeat-&gt;Enable(defined \$Slic3r::GUI::MainFrame::last_input_file); }); }, undef, 'lorry_add.png');</code>

The IDE also shows a menu structure on the left side of the screen, which is a visual representation of the code. The menu items and their corresponding code are as follows:

Menu Item	Code
Open STL/OBJ/AMF/3MF...	<code>Ctrl+O</code>
Open 2.5D TIN mesh...	<code>Ctrl+E</code>
Load Config...	<code>Ctrl+L</code>
Export Config...	<code>Ctrl+E</code>
Load Config Bundle...	<code>Ctrl+E</code>
Export Config Bundle...	<code>Ctrl+E</code>
Quick Slice...	<code>Ctrl+U</code>
Quick Slice and Save As...	<code>Ctrl+Alt+U</code>
Repeat Last Quick Slice	<code>Ctrl+Shift+U</code>
Slice to SVG...	<code>Ctrl+G</code>
Repair STL file...	<code>Ctrl+G</code>
Preferences...	<code>Ctrl+,</code>
Quit	<code>Ctrl+,</code>

The IDE status bar at the bottom shows: Perl source file, length: 28,568, lines: 680, Ln: 124, Col: 16, Sel: 0|0, Unix (LF), UTF-8, INS.



# Editing the GUI

The image shows a Notepad++ window editing a Perl script for a GUI application. The script defines a menu structure for a slicer application. A menu overlay is visible on the left side of the screen, listing various menu items and their keyboard shortcuts. Red arrows point from these menu items to the corresponding code lines in the script.

**Menu Overlay:**

- File
- Plater
- Object
- Settings
- View
- Window
- Quit
- Open STL/OBJ/AMF/3MF... (Ctrl+O)
- Quick Slice... (Ctrl+U)
- Quick Slice and Save As... (Ctrl+Alt+U)
- Repeat Last Quick Slice (Ctrl+Shift+U)
- Slice to SVG... (Ctrl+G)
- Repair STL file...
- Preferences... (Ctrl+,)

**Code Snippet:**

```
122 my ($self) = @_ ;
123
124 # File menu
125 my $fileMenu = Wx::Menu->new;
126 {
127     wxTheApp->append_menu_item($fileMenu, "Open STL/OBJ/AMF/3MF...\tCtrl+O", 'Open a model', sub {
128         $self->{plater}->add if $self->{plater};
129     }, undef, 'brick_add.png');
130     wxTheApp->append_menu_item($fileMenu, "Open 2.5D TIN mesh...", 'Import a 2.5D TIN mesh', sub {
131         $self->{plater}->add_tin if $self->{plater};
132     }, undef, 'map_add.png');
133     $fileMenu->AppendSeparator();
134     wxTheApp->append_menu_item($fileMenu, "&Load Config...\tCtrl+L", 'Load exported configuration file', sub {
135         $self->load_config file;
136     }, undef, 'plugin_add.png');
137     wxTheApp->append_menu_item($fileMenu, "&Export Config...\tCtrl+E", 'Export current configuration to file',
138     $self->export_config;
139     }, undef, 'plugin_go.png');
140     wxTheApp->append_menu_item($fileMenu, "&Load Config Bundle...", 'Load presets from a bundle', sub {
141         $self->load_configbundle;
142     }, undef, 'lorry_add.png');
143     wxTheApp->append_menu_item($fileMenu, "&Export Config Bundle...", 'Export all presets to file', sub {
144         $self->export_configbundle;
145     }, undef, 'lorry_go.png');
146     $fileMenu->AppendSeparator();
147     my $repeat;
148     wxTheApp->append_menu_item($fileMenu, "Q&uick Slice...\tCtrl+U", 'Slice file', sub {
149         wxTheApp->CallAfter(sub {
150             $self->quick_slice;
151             $repeat->Enable(defined $$Slic3r::GUI::MainFrame::last_input_file);
152         });
153     }, undef, 'cog_go.png');
154     wxTheApp->append_menu_item($fileMenu, "Quick Slice and Save &As...\tCtrl+Alt+U", 'Slice file and save as', s
155     wxTheApp->CallAfter(sub {
```

# Editing the GUI

The image shows the Slic3r application interface on the left and its source code in Notepad++ on the right. The Slic3r menu is open, showing options like 'Open STL/OBJ/AMF/3MF...', 'I changed the text here', 'Also here', 'Quick Slice...', 'Quick Slice and Save As...', 'Repeat Last Quick Slice', 'Slice to SVG...', 'Repair STL file...', 'Preferences...', and 'Quit'. The Notepad++ window displays the source code for the main frame, with red arrows pointing from the menu items to their corresponding code lines.

```
*C:\Users\James\Downloads\VandalForge-master\VandalForge-master\Slic3r\lib\Slic3r\GUI\MainFrame.pm - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
config1 x 2DBed.pm x OptionsGroup.pm x 2DToolpaths.pm x MainFrame.pm x
122     my ($self) = @_;
123
124     # File menu
125     my $fileMenu = Wx::Menu->new;
126     {
127         wxTheApp->append_menu_item($fileMenu, "Open STL/OBJ/AMF/3MF...\tCtrl+O", 'Open a model', sub {
128             $self->{plater}->add if $self->{plater};
129             }, undef, 'brick_add.png');
130         wxTheApp->append_menu_item($fileMenu, "I changed the text here", 'Open a model', sub {
131             $self->{plater}->add if $self->{plater};
132             }, undef, 'brick_add.png');
133         wxTheApp->append_menu_item($fileMenu, "Also here", 'Open a model', sub {
134             $self->{plater}->add if $self->{plater};
135             }, undef, 'brick_add.png');
136         #wxTheApp->append_menu_item($fileMenu, "Open 2.5D TIN mesh...", 'Import a 2.5D TIN mesh', sub {
137             # $self->{plater}->add_tin if $self->{plater};
138             #}, undef, 'map_add.png');
139         # $fileMenu->AppendSeparator();
140         #wxTheApp->append_menu_item($fileMenu, "&Load Config...\tCtrl+L", 'Load exported configuration file', sub {
141             # $self->load_config_file;
142             #}, undef, 'plugin_add.png');
143         #wxTheApp->append_menu_item($fileMenu, "&Export Config...\tCtrl+E", 'Export current configuration to file', sub {
144             # $self->export_config;
145             #}, undef, 'plugin_go.png');
146         #wxTheApp->append_menu_item($fileMenu, "&Load Config Bundle...", 'Load presets from a bundle', sub {
147             # $self->load_configbundle;
148             #}, undef, 'lorry_add.png');
149         #wxTheApp->append_menu_item($fileMenu, "&Export Config Bundle...", 'Export all presets to file', sub {
150             # $self->export_configbundle;
151             #}, undef, 'lorry_go.png');
152         $fileMenu->AppendSeparator();
153         my $repeat;
154         wxTheApp->append_menu_item($fileMenu, "Q&uick Slice...\tCtrl+U", 'Slice file', sub {
155             wxTheApp->CallAfter(sub {
```

Perl source file length: 28,928 lines: 686 Ln: 141 Col: 38 Sel: 0|0 Unix (LF) UTF-8

---

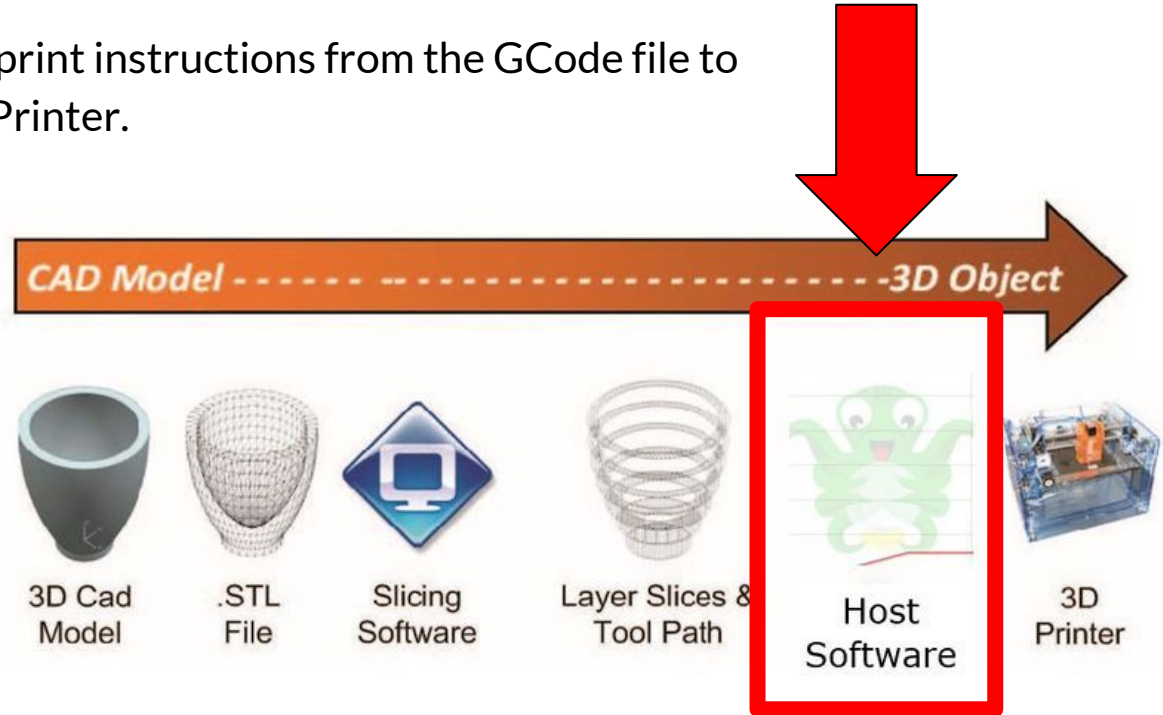
# OctoPrint

---



# OctoPrint: What it does

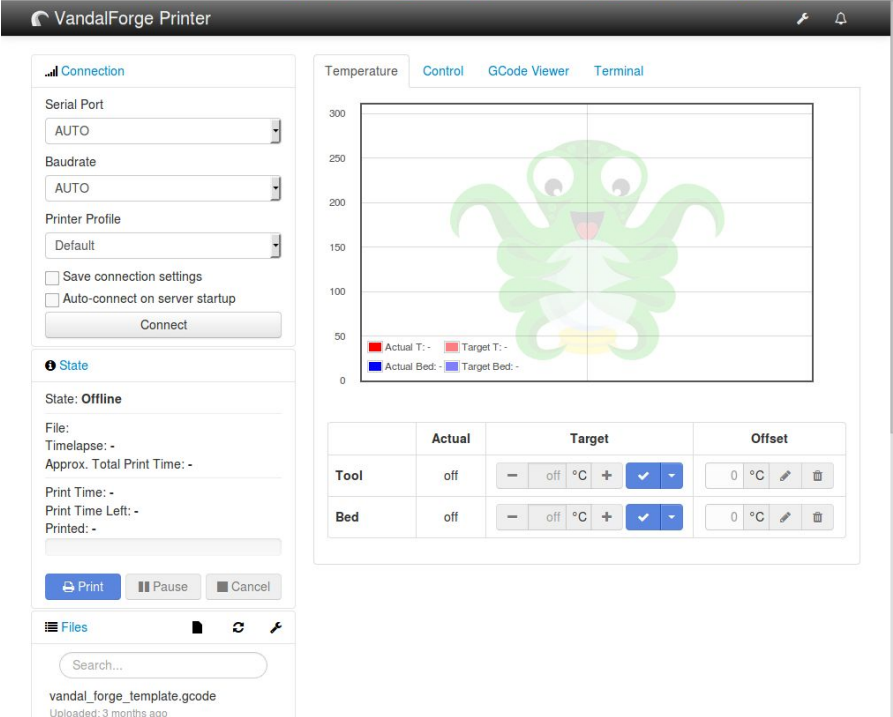
OctoPrint sends the print instructions from the GCode file to the Smoothieboard/Printer.



# OctoPrint: Modifications

## Features and Facts

- Originally Designed for Plastic Printers
- Open Source
- Easy to tailor for Metal Printing purposes



The screenshot displays the OctoPrint interface for a printer named "VandalForge Printer". The interface is divided into several sections:

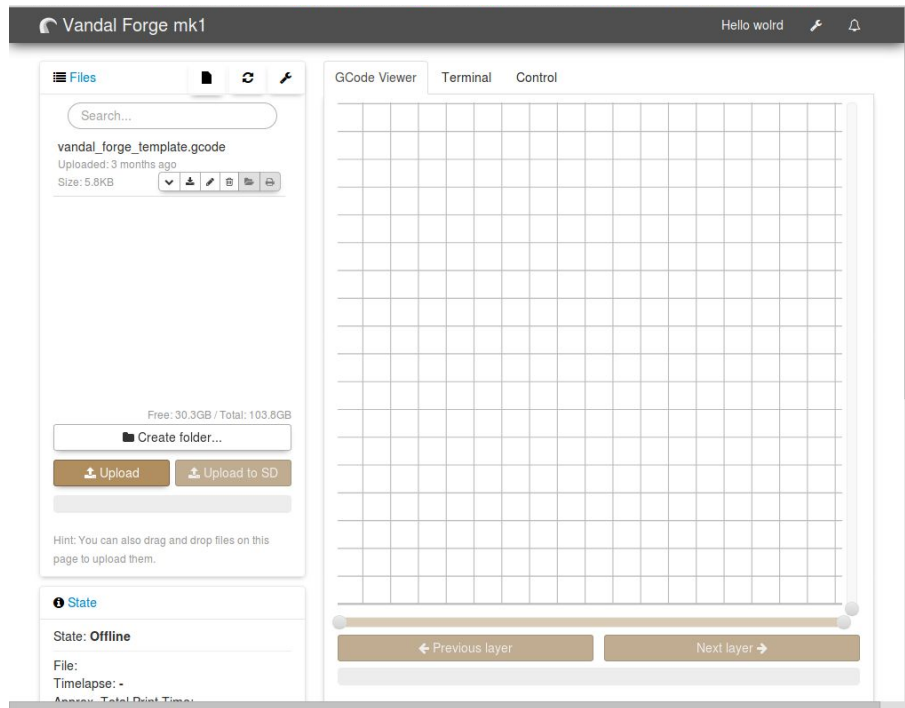
- Connection:** Includes dropdown menus for Serial Port (set to AUTO), Baudrate (set to AUTO), and Printer Profile (set to Default). There are checkboxes for "Save connection settings" and "Auto-connect on server startup", and a "Connect" button.
- State:** Shows the printer is "Offline". It includes fields for "File:", "Timelapse: -", "Approx. Total Print Time: -", "Print Time: -", "Print Time Left: -", and "Printed: -". At the bottom of this section are "Print", "Pause", and "Cancel" buttons.
- Files:** A search bar and a list of files, including "vandal\_forge\_template.gcode" which was "Uploaded: 3 months ago".
- Temperature Control:** A graph showing temperature over time. The y-axis ranges from 0 to 300. A green octopus character is overlaid on the graph. Below the graph is a table for temperature control:

	Actual	Target	Offset
Tool	off	off °C	0 °C
Bed	off	off °C	0 °C

# OctoPrint: Ways to Modify

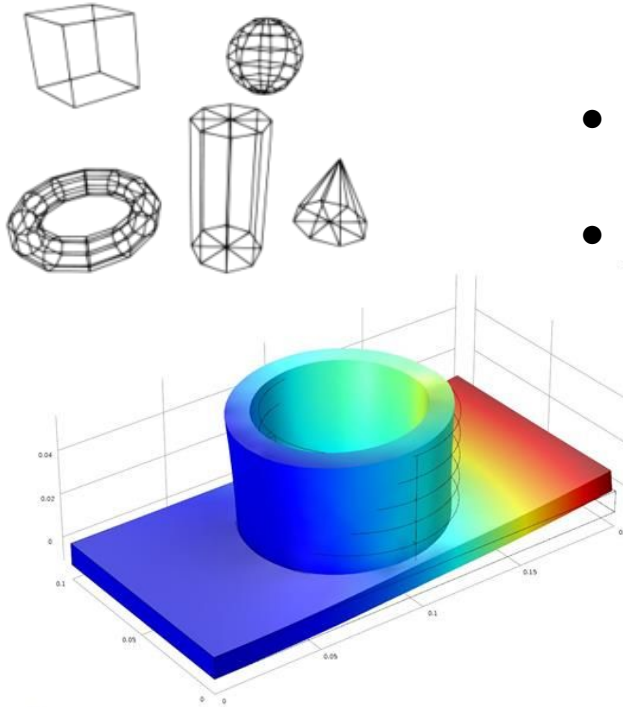
## Ways to Modify

- **Config.yaml**
  - Simple UI rearranging, new buttons tied to GCode commands or scripts
- **Premade plugins**
  - Easy install new features (GCode editor, themes, webcam)
- **Custom work**
  - More difficult, but customization possibilities are “limitless”



---

# OctoPrint: Future Plans



- **Repackage changes as a plugin**
  - Organize modifications into an easy to install plugin
- **Investigate Temperature wireframe model**
  - Create wireframe model of print in GCode viewer (create model in “dry-run” of print)
  - Add temperature data to wireframe model

---

# Next steps

## Add Mark II Gcode instructions

Add instructions that are compatible with Mark II printer

## Simplify Slic3r and OctoPrint UI's

Remove unnecessary UI elements from Slic3r, package up OctoPrint changes into an easy to install plugin

## Investigate modifying Print-Path Algorithm

Replace Slic3r's Print-Path algorithm with an algorithm that won't cause any problems from the Metal Printer (if possible).

---

---

# Contact Us



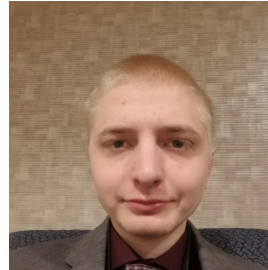
Michael Madsen  
Phone: 208-818-2061  
Email: [mads2998@vandals.uidaho.edu](mailto:mads2998@vandals.uidaho.edu)



James Young  
Phone: 208-881-6318  
Email: [youn5393@vandals.uidaho.edu](mailto:youn5393@vandals.uidaho.edu)



Tim Clemans  
Email: [clem8649@vandals.uidaho.edu](mailto:clem8649@vandals.uidaho.edu)



Jonathan Buch  
Phone: 208-789-8534  
Email: [buch4710@vandals.uidaho.edu](mailto:buch4710@vandals.uidaho.edu)

---