

Solablock2LaTeX

Ziel

Das Ziel dieses Projekts ist es, ein Programm zu schreiben, welches es ermöglicht, Sommerlagerdossiers nach den Vorgaben von J+S zu verfassen. Dabei sollen diese Dossiers jedes Mal gleich formatiert werden, unabhängig von Autor oder Betriebssystem. Dies wird über eine GUI ermöglicht, die den Text abfragt und nach LaTeX konvertiert. Das Programm hat dann seine Final-Version erreicht, wenn auch von Laien ohne LaTeX-Kenntnisse damit ein Lagerdossier erstellt werden kann. Dann wird es an die offiziellen Stellen von Jungwacht und Blauring übergeben.

Programmablauf

Das Programm funktioniert nach dem Start auf einem Prinzip von Fragen und Antworten. Das Programm stellt eine Frage, der Nutzer beantwortet sie, das Programm stellt die nächste Frage. Wenn alle Fragen beantwortet sind, kann der Benutzer in einem Fliesstext-Editor Einstieg, Hauptteil und Ausstieg eingeben. Danach kann der Text an einem beliebigen Ort gespeichert werden. Als Toolkit für die Benutzeroberfläche dient Shoes.

Vor- und Nachteile von Shoes

Vorteile:

- Einfache Programme lassen sich sehr schnell mit einer Benutzeroberfläche versehen
- Plattformübergreifend
- Opensource und Freeware
- Ruby
- Git Community, die weiterentwickelt

Nachteile:

- Komplexere Programme mit mehreren Fenstern sind nur sehr schwer zu realisieren.
 - Informationsfluss vom Kind zum Elternfenster funktioniert nicht gut.
- Dokumentation schwer zu verstehen, bis man die Grundlagen hat, Einsteigerhandbuch (<http://cloud.github.com/downloads/shoes/shoes/nks.pdf>) nicht wirklich lesbar
- Installation unter Linux kompliziert -> selbst kompilieren
- Installationen unter Mac und Windows können keine Programmpakete packen -> ebenfalls selbst kompilieren erforderlich
- Unter Mac OS X funktioniert Copy und Paste nicht

Verteilung

Die Verteilung des Programmes spielt im aktuellen Entwicklungsstand noch keine grosse Rolle. Interessierte Entwickler haben über Github Zugriff. Für Benutzer sind Programmpakete geplant, die sich per Doppelklick starten lassen. Allerdings sind dazu noch einige Punkte zu überarbeiten, wie die Verteilung von Ruby, Shoes und der Vorlage. Damit die Vorlage sicher ankommt, falls dennoch ein solches .shy Paket verteilt wird, ist sie momentan noch direkt als String im Code verankert. Dies ist allerdings eine Behelfslösung, bis die definitive Verteilung des Programms geklärt ist.

Copy und Paste

Zunächst war geplant, die Funktionen für Copy und Paste selbst zu implementieren. Per Rechtsklick funktionierten sie bereits auf Anhieb ohne irgendetwas programmieren zu müssen. Allerdings funktionierte die Tastenkombination unter Mac OS X nicht, daher wurde versucht, die Funktion zu ergänzen. Dies warf ein Problem auf, als im Editor zwei Textboxen sichtbar waren. Es war nicht möglich, herauszufinden, welche Textbox sich im Fokus befindet. Als sich dann noch herausstellte, dass es unter Linux auch ohne Tastenkombination funktioniert, führte das zum aktuellen Stand. Es ist nämlich im Editor nichts in der Art implementiert. Eine zukünftige Version wird sich diesem Problem annehmen. Es kann allerdings auch sein, dass es sich von selbst löst, falls das Problem in Shoes behoben wird.

Formatierungen im Fliesstext

Die geplanten Formatierungen im Fliesstext sind noch nicht eingebaut, da die Vorlage noch nicht zu 100% meinen Vorstellungen entspricht. Speziell das Setzen der Titel und Untertitel in der Tabelle braucht noch Zeit, da entsprechende LaTeX-Befehle noch kein schönes Layout erzeugen.

LaTeX-Sonderzeichen escapen

Sonderzeichen, welche vom Benutzer eingegeben werden, und von LaTeX interpretiert werden, müssen durch die entsprechenden LaTeX-Befehle ersetzt werden. Speziell das Ampersand-Zeichen (&) bereitet Probleme, da es in der Methode, welche die Zeichen ersetzt, eine Sonderbedeutung hat. Mit vorangestelltem Backslash wird es durch das ersetzte Zeichen ersetzt. Da nach dem Ersetzen des Ampersands noch eine unbekannte Zahl an Ersetzungen folgten, musste das Ampersand erst ganz am Schluss ersetzt werden, und konnte nicht an der eigentlich vorgesehenen Stelle im Code verarbeitet werden.

Fazit

Ich sehe das Programm, wie es jetzt ist, mittlerweile in einer Beta-Version, der aber noch einige Features fehlen. Die Frage ist jetzt, ob es in Shoes weitergeführt werden sollte, trotz der Nachteile dieses Toolkits oder ob ein anderes System verwendet werden sollte. Diese Entscheidung werde ich mir noch durch den Kopf gehen lassen. Sollte ich aber ein ähnlich einfaches, aber funktionaleres Toolkit finden, so werde ich das definitiv ausprobieren und den Code nötigenfalls umschreiben.

Generell sollte angemerkt werden, dass man das GUI-Toolkit kennen sollte und im Voraus einen Plan vom fertigen Programm haben sollte, bevor die erste Codezeile geschrieben ist. Dann wird die Entwicklung wesentlich einfacher. Mittlerweile kann ich Shoes einschätzen und weiss, in welchen Fällen es nützlich sein könnte, Programme mit diesem Toolkit zu schreiben.