

Dokumentation Abschlussarbeit Informatik:

Notensatzprogramm (Musik)



Software und Dokumentation erstellt von März bis Juni 2023 am
Wilhelm-Ostwald-Gymnasium Leipzig

Inhaltsverzeichnis

1	Kurzreferat	4
2	Einleitung	5
2.1	Themenwahl	5
2.2	Projektziel	5
2.3	Motivation	5
2.4	Projektplanung	6
3	Vorbetrachtung	6
3.1	Partituraufbau	6
3.2	Musikschriftarten	7
4	Ergebnisse	9
4.1	Programmbeschreibung (Anleitung)	9
4.1.1	Einführung	9
4.1.2	Befüllen	10
4.1.3	Speichern / Öffnen	13
5	Methoden	13
5.1	Programmstruktur	13
5.2	Benutzeroberfläche	14
5.3	PyQt5 Grafiksystem	14
5.4	Dokumentaufbau	14
5.5	Algorithmen	14
5.5.1	Notationssystem	14
5.5.2	Positionierung	14
5.5.3	Speichereinheit	14
5.5.4	Editierungseinheit	14
5.6	Kompilierung	14
6	Diskussion	14

6.1 Diskussion des Ergebnisses	14
6.2 Erweiterungsmöglichkeiten	15
7 Quellen	15
7.1 Bibliotheken	15
7.2 Schriftarten	15
7.3 Textquellen und Links	15
7.4 Abbildungen	15
8 Selbständigkeitserklärung	15

1 Kurzreferat

Als Abschlussarbeit der Sekundarstufe 1 im Fach Informatik wurde von Noah Weiler der Klasse 10/1 ein Notensatzprogramm für das Fach Musik erstellt. Es ermöglicht Schülern und anderen Benutzern, ein leeres Notenblatt mit Musikzeichen zu befüllen, um somit musiktheoretische Inhalte zu üben und Musikstücke zu schreiben. Nach der Bearbeitung kann das erstellte Dokument auf verschiedene Wege gespeichert und vervielfältigt werden, was den Austausch zwischen Schülern und Lehrkraft ermöglicht. Bei der Erstellung wurde darauf Wert gelegt, dass das Programm sehr nutzerfreundlich ist, gleichzeitig aber nicht zu viel vorgegeben wird, um für die schulische Anwendung geeignet zu sein. Das Programm wurde mit der Programmiersprache „Python“ sowie dem GUI-Toolkit „PyQt5“ entwickelt.

2 Einleitung

2.1 Themenwahl

Das erstellte Notensatzprogramm ist eine Musiksoftware, die darauf ausgelegt ist, Notenblätter digital statt auf Papier zu befüllen, um ein gut lesbares Endresultat zu erhalten. Dabei liegt der Fokus auf dem Komponieren und nicht dem Erstellen von tatsächlicher Musik.

2.2 Projektziel

Die Zielstellung des Projekts beinhaltete neben dem Befüllen von Notenblättern mit Musikzeichen wie zum Beispiel Noten, Pausen, Artikulationszeichen und Dynamikzeichen auch das Verändern, also Löschen und Bewegen von Musikzeichen und insgesamt eine benutzerfreundliche und simple Gestaltung.

Die genauen Ziele mit Ordnung der Kriterien nach der Wichtigkeit sind in den Planungsunterlagen zu finden, welche sich ebenfalls, wie diese Dokumentation, im Dokumentationsordner des Projekts befinden.

2.3 Motivation

Für viele Menschen ist Musik ein wichtiger Bestandteil ihres Lebens. Dazu gehört auch, selber Musik zu produzieren und zu schreiben, weswegen Musik auch in der Schule unterrichtet wird, um die wichtigsten Grundlagen zu vermitteln. Dabei wird fast immer auf Stift und Papier zurückgegriffen. In Zeiten der Digitalisierung stehen dafür aber bereits Computerprogramme wie zum Beispiel „MuseScore“ zur Verfügung, die jedoch wegen einer aufwändig zu erlernenden Bedienung noch nicht im Schulalltag angekommen sind.

Um Schülern, Hobbykomponisten und anderen eine leicht zu bedienende Notationssoftware an die Hand zu geben, wurde eine solche eigens programmiert und dokumentiert und dann von Musikern getestet.

Da die Programmidee etliche Möglichkeiten des Hinzufügens von Funktionen bereitstellt, konnte an diesem Projekt optimal das Lösen ganz unterschiedlicher programmiertechnischer Probleme geübt werden. Daher wurde dieses komplexe Projekt einem vergleichbar vielleicht simpleren vorgezogen.

2.4 Projektplanung

Die allgemeine Projektplanung wurde bis zum 01.03.2023 getätigt, die Planungsunterlagen sind im angesprochenen Dokumentationsordner zu finden. Wie dort angemerkt, kann der gesamte Quellcode unter dem Link in den unten befindlichen Quellen eingesehen werden. ¹

Die Versionsverwaltung „Git“ mit der Oberfläche „GitHub“ wurde daher gewählt, weil es damit möglich ist, den Code in mehreren Teilschritten, sogenannten Commits, zu veröffentlichen und kostenlos online zu speichern und auf das gesamte Projekt über die Entwicklungsumgebung „Visual Studio Code“ zuzugreifen. Über das lokal installierte „Git“ konnte dann auch zu verschiedenen Commits zurückgewechselt werden, was die Fehlersuche beim Hinzufügen neuer Funktionen erleichtert.

Das Projekt war bis kurz vor Abgabedatum privat und nun öffentlich, um den Code sowie die kompilierte Version leicht zugänglich zu machen.

3 Vorbetrachtung

3.1 Partituraufbau

Eine Partitur, wie sie mit dem Programm erstellt werden soll, stellt ein mehrstimmiges Musikstück dar.

Um die Musiknotation programmtechnisch umsetzen zu können, muss im vornherein die genaue Objektstruktur genau geplant werden. Im Folgenden ist eine Abbildung zu sehen, die später mit dem selbst erstellten Programm abgebildet wurde und die Struktur einer Partitur, also eines Notensatzes, verdeutlichen soll.

Titel

Untertüberschrift

Komponist / Arrangeur

System

Tempoangabe

Notenzeile /
Stimme

Taktart

Takt

Notenschlüssel

Tonart

Musikzeichen

Dabei fällt auf, dass mehrere Objekte ineinandergreifen: So beinhaltet ein System mehrere Takte und mehrere Notenzeilen, wobei ein Takt mehrere Teilstücke von Notenzeilen enthält. Die Notenschlüssel, die für jede Notenzeile verschieden sind, sowie die Tonart werden in jedem neuen System neu angezeigt, wobei die Taktart nur im ersten Takt und bei Taktwechseln angezeigt wird.

Dies erschwert die programmiertechnische objektorientierte Umsetzung, da dabei genau darauf geachtet werden muss, welche Objekte welche anderen Objekte beinhalten. Dazu folgt später die Beschreibung der gefundenen Lösung.

3.2 Musikschriftarten

Beim Erstellen eines solchen Programms stellt sich schnell die Frage wie es möglich ist, die vielen Musikzeichen, die man benötigt um ein Stück zu komponieren, dem Nutzer grafisch zur Verfügung stellen kann. Die Verwendung von Bilddateien würde lange Ladezeiten, eine schlechte Skalierbarkeit und uneindeutige Positionen verursachen. Daher wird für solche Anwendungen üblicherweise eine Schriftart verwendet, also ein Satz von Schriftzeichen, in diesem Falle Sonderschriftzeichen. Die Schriftzeichen einer Schriftart werden über Unicodes definiert und

festgelegt. Ein Unicode besteht aus der Vorsilbe „U+“ und einer Hexadezimalzahl. Der Unicode für den Buchstaben „a“ ist zum Beispiel U+0061. Lässt man sich nun von verschiedenen Schriftarten das Schriftzeichen zurückgeben, welches bei diesem Unicode hinterlegt ist, bekommt man in allen Fällen ein „a“ zurück, welches sich aber im Aussehen unterscheiden kann.

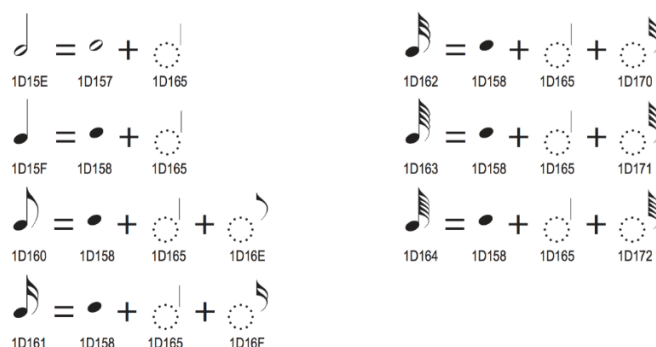
Das gleiche System wird bei Musikschriftarten verwendet. Dabei legt das „Standard Music Font Layout“, kurz SMuFL, fest, welchem Unicode welches Musikzeichen zugeordnet ist. Dieser offizielle Standard wurde von Daniel Spreadbury entworfen.

Wie auch bei Textschriftarten gibt es viele verschiedene, aus denen, abhängig von den Ansprüchen, gewählt werden muss. Die Musikschriftart „Bravura“, die ebenfalls von Daniel Spreadbury entwickelt und 2013 veröffentlicht wurde, gilt dabei als generelle Referenzschriftart für den SMuFL-Standard. ²

„Bravura“ ist frei unter der Lizenz „SIL OPEN FONT LICENSE Version 1.1 - 26 February 2007“ herunterladbar. ³

Mit dieser Schriftart wurde auch das blaue Icon des Programms erstellt.

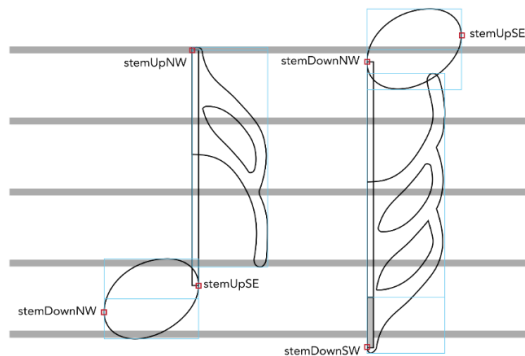
Die Zuordnung von Unicodes und Musikzeichen, sowie Anwendungen bestimmte Dinge implementieren können wurde online bereits dokumentiert. ⁴ So muss zum Beispiel eine Note aus verschiedenen Schriftzeichen zusammengesetzt werden.



Implementierung von verschiedenen Notenwerten ¹

Dabei stellen die meisten Musikschriftarten, wie auch „Bravura“, Metadaten in JSON-Dateien zur Verfügung, die genaue Abmessungen

und Koordinaten von wichtigen Verbindungspunkten der Schriftzeichen beinhalten, welche zur Implementierung essentiell sind.



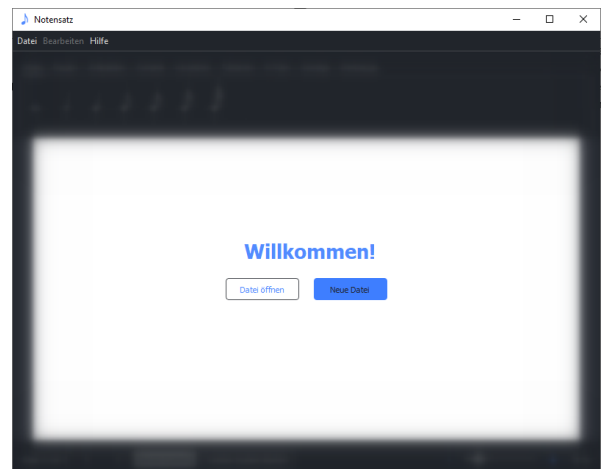
Ausdehnungsboxen (blau) und Verbindungspunkte (rot) von Noten²

4 Ergebnisse

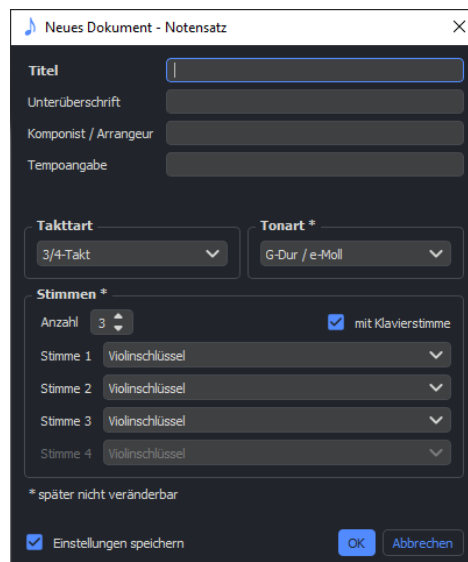
4.1 Programmbeschreibung (Anleitung)

4.1.1 Einführung

1. Im Startbildschirm lässt sich zwischen der Option „Neue Datei“, um ein neues Notensatzdokument zu erstellen, und der Option „Datei öffnen“, um ein gespeichertes Notensatzdokument wieder öffnen, wählen.



2. Bei einem Klick auf die Option „Neue Datei“ ist die sich öffnende Vorlage auszufüllen. Wenn die getätigten Einstellungen nicht verloren gehen sollen (auch nach Schließen des Programms), kann das Häkchen bei „Einstellungen speichern“ links unten gesetzt werden.



Die mit einem Sternchen gekennzeichneten Einstellungen können später nicht mehr geändert werden und sind im gesamten Dokument konstant.

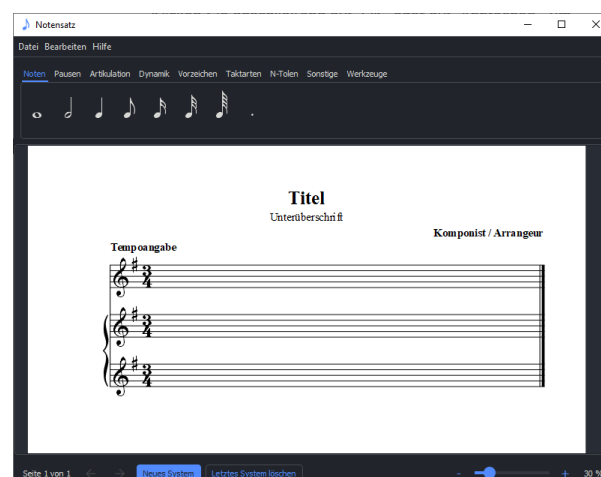
Mit dem Klick auf „OK“ wird ein neues Dokument erstellt, bei „Abbrechen“ gelangt man zurück zum Startbildschirm.

3. Nun kann das Notensatzdokument nach den jeweiligen Bedürfnissen befüllt, gespeichert und exportiert sowie bereits gespeicherte Notensatzdokumente geöffnet werden.

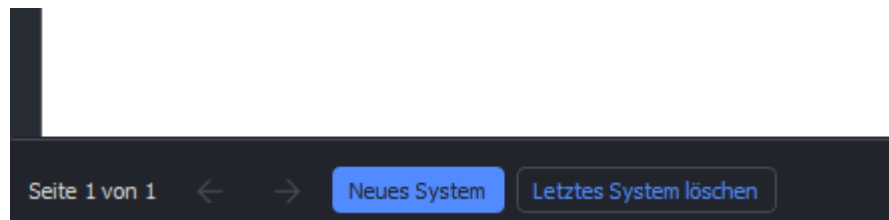
4.1.2 Befüllen

1. Nachdem das neue Dokument erstellt wurde, befinden man sich nun in der Hauptumgebung des Programms.

Unter der Menüleiste befindet sich der „Notensatzbaukasten“, darunter die Dokumentansicht, deren Seiten, sowie die Skalierung und die Notenliniensysteme sich über die ganz unten befindliche Statusleiste ändern lassen.

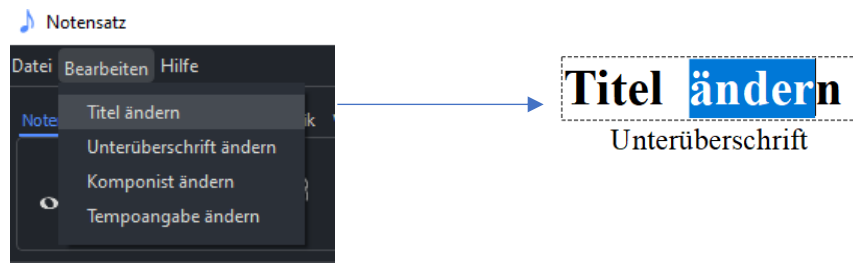


2. Über die zwei Buttons in der Statusleiste kann ein neues System hinzugefügt werden oder das aktuell letzte gelöscht werden.

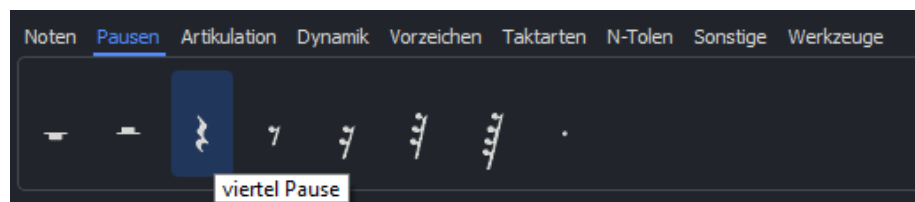


Das Programm erstellt dabei automatisch neue Seiten, zwischen diesen lässt sich über die zwei Pfeile neben der Seiteninformation wechseln.

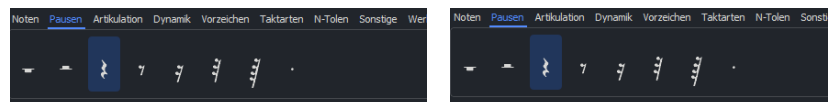
- Der Titel, der Untertitel, der Komponist und die Tempoangabe lassen sich mit einem direkten Klick auf das entsprechende Textfeld ändern. Generell, oder in Fällen in denen das Textfeld leer und somit schwer zu finden ist, kann auch über die Menüleiste in die Textfelder navigiert werden:



- Zum Eigentlichen Befüllen der Notenzeilen steht der „Notensatzbaukasten“ zur Verfügung. Dazu wählt man einfach den gewünschten Reiter (hier zum Beispiel „Pausen“) und klickt auf ein Symbol. Bei längerem Verweilen mit der Maus auf einem Symbol wird eine Erklärung angezeigt, um welches musikalische Symbol es sich handelt.



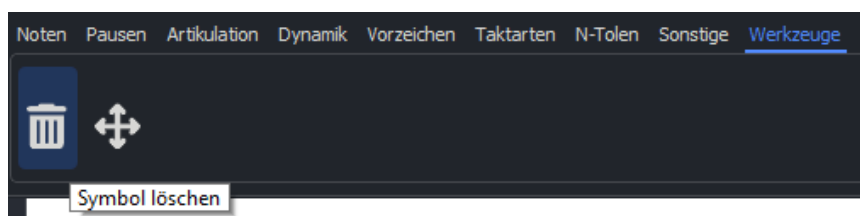
Das Programm schlägt nun automatisch freie Positionen für das ausgewählte Symbol vor, wenn man die Maus in die Dokumentansicht bewegt.



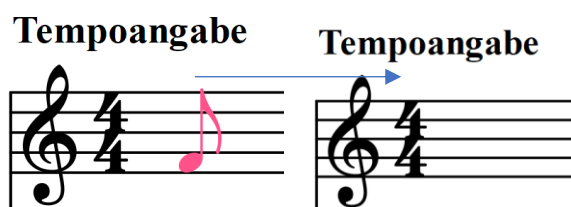
Mit einem Linksklick kann das Symbol an der gewählten Position platziert werden.

Löschen / Bewegen

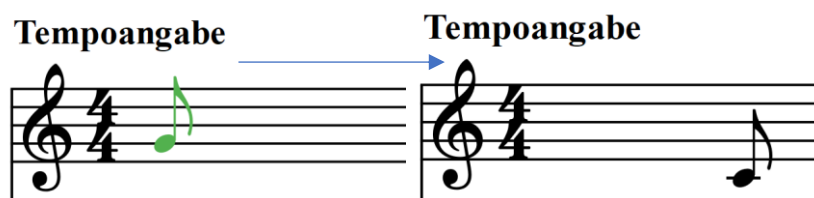
1. Zum Verändern des erstellten Notensatzes können die Optionen unter dem Reiter „Werkzeuge“ genutzt werden.



2. Wählt man die Option „Symbol löschen“ und fährt mit der Maus über das gewünschte Objekt, zum Beispiel eine Note, färbt sich dieses dann rosa und kann per Linksklick entfernt werden.

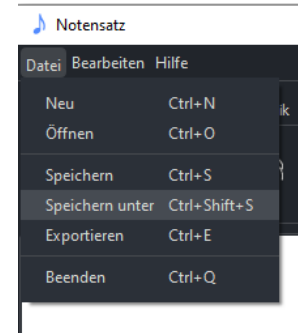


3. Bei der Option „Symbol bewegen“ färbt sich das gewählte Objekt grün. Dieses wird per Linksklick beweglich und kann dann per erneutem Linksklick auf der neuen Position platziert werden.



4.1.3 Speichern / Öffnen

1. Das erstellte Notensatzdokument kann über die Optionen „Speichern“, „Speichern unter“ und „Exportieren“ unter dem Reiter „Datei“ in der Menüleiste zur Vervielfältigung auf Ihrem Gerät gespeichert werden. Dazu wählt man im sich öffnenden Dialog einfach den



entsprechenden Ordner aus. Basierend auf dem Titel und dem Komponisten wird ein Dateiname mit der Dateiendung „nos“ oder „pdf“ erstellt, dieser kann entsprechend angepasst werden.

Eine Datei im Format „nos“ kann vom Programm wieder geöffnet werden. Soll das Dokument fertiggestellt und zum Beispiel gedruckt werden, kann die Option „Exportieren“ gewählt werden, die den Inhalt des Dokuments in eine PDF-Datei exportiert. Diese kann nicht mehr vom Programm eingelesen werden.

Ein Druckerdialog steht ebenfalls zur Verfügung.

Wie in der obigen Abbildung gezeigt, stehen für alle Optionen unter dem Reiter „Datei“ Tastenkombinationen zur Verfügung.

Eine Datei kann über die Option „Öffnen“ unter dem Reiter „Datei“ oder über den linken Button auf dem bereits vorher beschriebenen Startbildschirm geöffnet werden.

5 Methoden

5.1 Programmstruktur

Die Programmstruktur lässt sich wie folgt veranschaulichen. Zur Vereinfachung werden deutsche Übersetzungen verwendet, die so nicht im Quellcode wiederzufinden sind. Eckige Kästen symbolisieren Python-Dateien die mehrere Funktionen beinhalten können. Runde Kästchen symbolisieren tatsächliche Objekte.

5.2 Benutzeroberfläche

Die grafische Benutzeroberfläche wurde mit dem Programm „Qt Designer„ erstellt und mit Hilfe des Tools „pyuic5“ in eine Python-Datei konvertiert und später über den Programmcode noch modifiziert.

5.3 PyQt5 Grafiksystem

5.4 Dokumentaufbau

5.5 Algorithmen

5.5.1 Notationssystem

5.5.2 Positionierung

5.5.3 Speichereinheit

5.5.4 Editierungseinheit

5.6 Kompilierung

6 Diskussion

6.1 Diskussion des Ergebnisses

6.2 Erweiterungsmöglichkeiten

7 Quellen

7.1 Bibliotheken

7.2 Schriftarten

7.3 Textquellen und Links

1 <https://github.com/NO411/notensatz>

2 <https://en.wikipedia.org/wiki/SMuFL>

3 <https://github.com/steinbergmedia/bravura/releases>

4 <https://w3c.github.io/smufl/latest/index.html>

7.4 Abbildungen

1 <https://w3c.github.io/smufl/latest/media/precomposed-notes-unicode.png>

2 <https://w3c.github.io/smufl/latest/media/notehead-stem-flag.svg>

Alle anderen Abbildungen sind Bildschirmaufnahmen der erstellten Software oder wurden anderweitig selbst erstellt.

8 Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen Hilfsmittel als die angegebenen verwendet habe. Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht habe.

Ort: Leipzig

Datum: 28.05.2023

Unterschrift: