Digital Input Computer Keyboard

F.Zeller und E.Hubenschmid

5. Juli 2015

Inhaltsverzeichnis

1 Einleitung

1.1 Vorstellung des Prokeltes

Unser Projekt das Digital-Input Computer-Keyboard, ist wie der Name schon vermuten lässt ein Digitales Keyboard. Man kann damit mithilfe seines PCs und einer Tastatur Keyboard spielen. Es kann allerdings auch noch mehr. Der Benutzer hat die Möglichkeit verschiedene Samples (Tonspuren die Instrumente darstellen) einzustellen. Somit kann er theoretisch jedes Musikinstrument über seine Tastatur spielen. Außerdem kann der Benutzer gespieltes aufnehmen und abspielen lassen. Somit kann er mit verschiedenen Instrumenten und Tonspuren eigene Lieder zusammensampeln.
Dieses Projekt hat Emanuel und mir sehr gut gefallen, da wir beide ein Musikinstrument spielen und auch allgemein Musik begeistert sind. Ein eigenes kleines Sample-Programm zu gestalten lag also nahe.

1.1.1 Unser Team

Bei unserem IT-Projekt hatten wir die Möglichkeit Teams zu bilden um größere Projekte umzusetzen. Das Team des Projektes Digital Input Computer Keyboard, besteht aus Emanuel Hubenschmid und Fabian Zeller. Diese beiden sind musikalisch aktiv und hatten somit Interesse daran den Computer mit musikalischer Kreativität zu verbinden. Das Arbeitsklima untereinander war durchgehend angenehm, da jeder seine Arbeit hatte, diese erledigt hat und alles dann gemeinsam zusammengeführt wurde. Es kam also nie zu Streit oder Uneinigkeit über den weiteren Verlauf des Projekts. Da wir verschiedene Stärken haben haben wir unser Projekt auch entsprechend aufgeteilt. Emanuel hat sich größtenteils um Wiedergabe von Samples gekümmert, während Fabian die GUI entsprechend angepasst hat.

1.2 Das Projektmanagement

1.2.1 SCRUM

Definition:
Scrum wurde ursprünglich in der Softwaretechnik entwickelt, ist aber auch in anderen Bereichen einsetzbar. Mittlerweile wird Scrum in sehr vielen Firmen eingesetzt. Scrum basiert darauf das Projekt oder auch die Projektplanung kontinuierlich zu verbessern. Es ist also sehr flexibel und passt sich ständig an. Ziel von dieser kontinuierlichen Verbesserung ist die schnelle und kostengünstige Entwicklung hochwertiger Produkte.
Anfangs hat man nur eine Vorstellung seines Projekts die durch den Kunden oder den Entwickler gegeben wird. Man versucht dann diese Vorstellung anhand von einer Liste mit Eigenschaften festzuhalten. Anders als bei anderen Methoden fertigt man kein Lasten- oder Pflichtenheft an, sondern eben jene Liste. Die Liste dieser Eigenschaften oder auch Anforderungen an das Endprodukt nennt man Backlog.
Diese Anforderungen werden dann intervallartig abgearbeitet. Diese Intervallartigen umsetzungen werden Sprints genannt. Nach Abschluss eines Sprints hat man dann also ein fertiges Teilprodukt des ganzen, dass man testen oder dem Kunden präsentieren kann. Hier fallen dann schon eventuelle Änderungen auf,die dann in das Backlog, bzw. in das nächste Sprint einfließen. Somit verändert sich dass Backlog ständig und das Endprodukt wird optimiert.
Zusammengefasst basiert Scrum auf 3 SSäulen":

- **Transparenz:** Der Fortschritt und die Verzögerungen werden kontinuierlich festgehalten und man hat immer eine Vorstellung davon wie weit das Projekt fortgeschritten ist.
- **Überprüfung:** Durch die Sprints werden regelmäßig Teilprodukte geliefert die man testen, überprüfen und optimieren kann.
- **Anpassung:** Dadurch dass nicht von Anfang das Endprodukt festgelegt ist, sondern kontinuierlich angepasst wird, entstehen aus einem meist komplexen System viele kleinere Teilsysteme die einfacher zu bewältigen sind.

Wen dieses Konzept interessiert kann sich hier weiter über Scrum erkundigen: Link

Unsere Erfahrung mit Scrum

Wir haben für unser Schulprojekt Scrum verwendet. Wir haben ein Backlog erstellt und wöchentlich Sprints abgeschickt. Scrum hat uns sehr geholfen unser Projekt zu strukturieren, allerdings waren wöchentliche Sprints nicht optimal. Da wir als Schüler nicht Vollzeit an unserem Projekt arbeiten und die Sprints wöchentlich abzuschicken waren, haben sich manche Stories über mehrere Sprints angestaut. In einem Betrieb oder einer Entwicklerfirma in welcher nur an einem Projekt entwickelt wird sind häufige Sprints sinnvoll, bei nur 2 Leuten und eine quasi Nebenbeschäftigung waren diese doch in zu kurzen Intervallen. Ansonsten war Scrum eine große Hilfe.

1.2.2 GitHub

"Build software better, together."(Motto von GitHub)

Für unsere Projektarbeiten verwendeten wir GitHub. GitHub ist ein webbasierter Hosting-Dienst für Softwareprojekte. Damit Emanuel und ich also bequem von zuhause aus zusammen arbeiten konnten haben wir uns ein Repository in GitHub eingerichtet. Zusammen mit dem Eclipse-Plugin EGit konnten wir unsere Arbeit austauschen und vergleichen. Obwohl es Anfangs Probleme mit der Bedienung und den verschiedenen Funktionen GitHubs und EGits gab, hat es uns doch sehr geholfen und vieles vereinfacht. So konnten wir zum Beispiel gleichzeitig an verschieden Problemen arbeiten, indem wir verschiedene Branches (Pfadeälso Ableger des Projekts) erstellt haben und dann an diesen Branches gearbeitet haben. Sobald dann ein Problem behoben war hat man den Pfad wieder dem Hauptprojekt hinzugefügt und konnte besprechen was genau gemacht wurde und was vielleicht noch verbessert werden muss.

GitHub war uns im allgemeinen eine sehr große Hilfe, aber zu Anfang auch eine große Hürde. Bis wir zurechtkamen mit den Branches, Commits ect. hat es eine Weile gedauert. Man kann aber durchaus behaupten das sich der Aufwand gelohnt hat. Ich würde jedem der ein Softwareprojekt entwickelt empfehlen mit GitHub zu arbeiten, auch wenn er alleine daran arbeitet. Denn EGit zwingt einen dazu seine Änderungen zu dokumentieren und zwischenzuspeichern. Dies ist zwar etwas nervig, aber man kann sein Projekt immer wieder auf einen beliebigen Standpunkt zurücksetzten wenn etwas komplett schiefgelaufen ist. Dies hat mir oft sehr viel Arbeit erspart.

Gerade jetzt für diese Dokumentation verwenden wir auch GitHub for Windows. Ein Programm für Windows welches einem erlaubt jede Art von Datei über GitHub zu veröffentlichen und zu bearbeiten. GitHub hat uns also sehr viel Arbeit erspart und es uns ermöglicht obwohl wir weit auseinander wohnen ein Projekt auch zusammen zu erarbeiten.

2 Technische Dokumentation

2.1 Dokumentation der Storys

2.1.1 Der Benutzer möchte ein Fenster

1. Das Fenster ist sichtbar
2. Das Fenster hat eine festgelegte größe
3. Das Fenster kann verschoben werden

Storypoints: 2
Bearbeitet von: Fabian Zeller

Beschreibung: Ein einfaches Fenster ohne Inhalt war nicht schwer zu verwirklichen. Das Fenster ist ein einfaches JPanel.

2.1.2 Der Benutzer möchte 3 Sektionen in dem Fenster

1. Die Sektionen sollen beschriftet sein
2. Sektionen untereinander"

Storypoints: 2
Bearbeitet von: Fabian Zeller

Beschreibung: Das ursprüngliche JPanel wurde zu der contentpane. Drei weitere Panels wurden untereinander angeordnet hinzugefügt.

2.1.3 Der Benutzer möchte, dass in der unteren Sektion Klaviertasten angezeigt werden

1. Die Tasten haben 2 Farben (schwarz/weiß)
2. Die Tasten können angeklickt werden
3. Die Tasten sind wie auf einem Klavier angeordnet

Storypoints: 5
Bearbeitet von: Fabian Zeller

Beschreibung: Die Klaviertasten bestehen aus jeweils bis kleineren Toggle Buttons. Dies wurden so konfiguriert dass sie wie Klaviertasten aussehen.

2.1.4 Der Benutzer möchte mittels Tasten (Tastatur) einen Klavierton abspielen

1. Die Tasten spielen jeweils genanu einen Ton wenn sie geklickt werden
2. Die Tasten können einen Ton halten

Storypoints: 5
Bearbeitet von: Emanuel Hubenschmid

Beschreibung: Zu Begin versuchten wir dies über eine Synthesizer-Klasse der Javabibliothek. Dies erwies sich allerdings als äußerst umständlich. Deswegen entschlossen wir uns die Samples selber zusammen zu stellen und diese als Clips aufzurufen.

2.1.5 Der Programmierer arbeitet sich in Git-Hub ein

1. Das Git-Hub Repository kann von allen Programmierern benutzt werden

Storypoints: 3
Bearbeitet von: Fabian Zeller, Emanuel Hubenschmid

Beschreibung: Um die Zusammenarbeit zu erleichtern verwendeten wir GitHub zur Verwaltung unseres Projektes (Siehe **1.2.2 GitHub**)

2.1.6 Der Programmierer arbeitet sich in Javadocs ein

1. Die Javadoc-Seite soll alle wichtigen Informationen zu dem Programmquellcode enthalten
2. Die Javadoc-Seite soll in der Dokumentation verlinkt werden

Storypoints: 3
Bearbeitet von: Fabian Zeller, Emanuel Hubenschmid

Beschreibung: Wir entschieden uns unseren Quelltext mittels Javadocs auszukemmentieren. Dies bot uns die möglichkeit den überblick zu behalten und dem jeweils anderem eine einfache möglichkeit zu bieten den Quelltext besser zu verstehen.

2.1.7 Der Benutzer möchte, dass die gespielte Taste hervorgehoben wird

1. Wenn eine Taste auf der Tastatur betätigt wird, färbt sich die entsprechende Taste auf dem Bildschirm

Storypoints: 5
Bearbeitet von: Fabian Zeller

Beschreibung: Der Taste auf der Tastatur wird eine Gruppe von Toggle Buttons zugewiesen, die dann wie eine taste von einem Klavier aussehen. Diese werden dann betätigt.

2.1.8 Der Benutzer möchte voreingestellte Samples auswählen können

1. Die Sample sollen die Tasten mit unterschiedlichen Tönen belegen
2. Die voreingestellten Samples sollen Klaviertöne und Schlagzeugtöne abspielen

Storypoints: 5
Bearbeitet von: Emanuel Hubenschmid

Beschreibung: Das Programm sollte eine möglichkeit bieten, direkt etwas spielen zu können. Da die Java Sound API nur .wav Dateien mit 8 oder 16 Bit, mussten wir auch solche Sounddateien als Samples verwenden. Die Samples können über Radiobutton ausgewählt werden.

2.1.9 Der Benutzer möchte mehrere Töne gleichzeitig spielen können

1. Die Töne sollen gleichzeitig abgespielt werden können
2. Die Töne sollen nicht durch andere Töne unterbrochen werden
3. Die Töne, die gleichzeitig angespielt wurden, sollen auch gleichzeitig abgespielt werden
4. Die anzahl der Töne die gleichzeitig gespielt werden können ist auf 3 beschränkt

Storypoints: 5
Bearbeitet von: Emanuel Hubenschmid

Beschreibung: Damit Töne gleichzeitig abgespielt werden können müssen diese über einen Thread in einem Vektor erzeugt und gestartet werden. Der Key Buffer kann jedoch nicht mehr als 3 Tasten gleichzeitig erfassen.

2.1.10 Der Benutzer möchte, dass in der oberen Sektion Notenlinien angezeigt werden

1. Die Notenlinien bestehen aus fünf Linien
2. Die Notenlinien beginnen mit einem Notenschlüssel

Storypoints: 5
Bearbeitet von: Fabian Zeller

Beschreibung: Mehrere Images von leeren Noten werden der oberen Sektion hinzugefügt.

2.1.11 Der Benutzer möchte Notenlblätter, auf denen Noten angezeigt und abgespeichert werden

1.

2.

Storypoints: 5
Bearbeitet von: Emanuel Hubenschid

Beschreibung:

2.1.12 Der Benutzer möchte das gespielte Töne als Note angezeigt werden

1. Noten laufen die Notenlinie entlang

2. Vorzeichen bei bestimmten Noten

Storypoints: 8
Bearbeitet von: Fabian Zeller

Beschreibung: Ein leeres Bild wird durch das Bild der entsprechend gespielten Note ersetzt.

2.1.13 Der Benutzer möchte das gespeicherte Notenlinien in einem externen Fenster aufgerufen werden können

1. Die Notenlinien werden in einem externen Fenster geöffnet

Storypoints: 5
Bearbeitet von: Emanuel Hubenschmid

Beschreibung: Wird eine Datei abgespielt öffnet sich ein seperates Fenster, in welchem der Player die gerade abgespielte Note abbildet.

2.1.14 Der Benutzer möchte gespeicherte Notenlinien abspielen

1. Die Notenlinien können abgespielt, pausiert und gestoppt werden

Storypoints: 8
Bearbeitet von: Emanuel Hubenschmid

Beschreibung: Die gespeicherten Noten werden aus einer Textdatei ausgelesen und abgespielt. Ein Button kann diese nun pausieren. Wird das Fenster geschlossen, wird auch die Datei gestoppt.

2.1.15 Der Benutzer möchte Noten, die auf den Notenlinien angezeigt und abgespeichert werden

1. Die Noten, die gespielt wurden, sollen auf den richtigen Notenlinien angezeigt werden
2. Die Noten sollen die Tonhöhe speichern

Storypoints: 8
Bearbeitet von: Fabian Zeller, Emanuel Hubenschmid

Beschreibung: Wird während der Aufnahme eine Taste gedrückt wird diese in ein Textdokument geschrieben.

2.1.16 Der Entwickler möchte eine Latex Dokumentation

1. Deckblatt
2. Inhaltsverzeichniss
3. Link auf Javadocs
4. Quellen

Storypoints: 5
Bearbeitet von: Fabian Zeller, Emanuel Hubenschmid

Beschreibung: Ich arbeite daran.

2.1.17 Der Benutzer möchte ein ToolBar-Menü

1. Die ToolBar enthält ein Hilfe-Menü
2. Die ToolBar enthält ein Projekt Öffnen-Menü
3. Die ToolBar enthält ein Neues Projekt-Menü

Storypoints: 3
Bearbeitet von: Fabian Zeller, Emanuel Hubenschmid

Beschreibung: Wir entschieden uns dazu keine Untermenüs zu verwenden, da bei einer so geringen anzahl an Optionen, dies viel umständlicher für den Benutzer währe.

2.1.18 Der Benutzer möchte, dass in der Mittleren Sektion Buttons zur Steuerung angezeigt werden

1. Samples können über Radiobutton ausgewählt werden

Storypoints: 3
Bearbeitet von: Emanuel Hubenschmid

Beschreibung: Ürsprünglich waren die Button zur Aufnahme und Wiedergabe ebenfalls in diesem Bereich geplant. Diese sind jedoch nun Projektgebunden und deswegen in dem Projektfenster. Es finden sich hier nurnoch die Radiobutton zur Auswahl der Sample.

2.1.19 Der Benutzer möchte mit der Leertaste alle geöffneten Aufnahmen abspielen

1. Die Leertaste startet oder Stoppt alle geöffneten Aufnahmen

Storypoints: 8
Bearbeitet von: Emanuel Hubenschmid

Beschreibung: Damit alle Aufnahmen gleichzeitig abgespielt werden können und als "ganzes Lied" gehört werden können. Ermöglicht es die Leertaste alle geöffneten Aufnahmen gleichzeitig abzuspielen.

2.1.20 Der Benutzer möchte eingene Samples benutzen können

1. Die Samples sollen in eine Bibliothek geladen werden können
2. Die Samples sollen vom Benutzer ausgewählt und abgespielt werden können

Storypoints: 5
Bearbeitet von: Emanuel Hubenschmid

Beschreibung: Am einfachsten und übersichtlichsten erwies es sich, eigene Samples nur mit einem Projekt verwenden zu können. Diese sind dabei immer in dem Projekt-Ortner unter Samples in einem Ortner abzuspeichern.

2.1.21 Der Benutzer möchte einen Projektordner erzeugen können

1. Der Projektordner enthält eine Sample- und einen Aufnahme-Ordner

Storypoints: 8
Bearbeitet von: Emanuel Hubenschmid

Beschreibung: Wird ein neues Projekt erstellt speichert dies nur den Ordnerpfad in dem sich die Samples und Aufnahmen befinden.

2.1.22 Der Benutzer möchte das die Note einer Taste auf der Taste angezeigt wird

1. Sichtbare Note auf den Tasten

Storypoints: 5
Bearbeitet von: Fabian Zeller

Beschreibung: Eine Glasspane mit entsprechender Beschriftung wurde über die GUI gelegt.

2.1.23 Der Benutzer möchte die Samples wechseln

1. Die Samples sollen mittels RadioButton ausgewählt werden können,
2. Die RadioButton sollen für jede Taste das gleiche Sample auswählen
3. Die Samples sollen mit Tasten gewächselt werden können"

Storypoints: 5
Bearbeitet von: Emanuel Hubenschmid

Beschreibung: Ist ein Radiobutton ausgewählt wird der Ordnerpfad des jeweiligen Samples benutzt um den Clip zu starten.

2.2 Klassendiagramme

2.3 Sequenzdiagramme

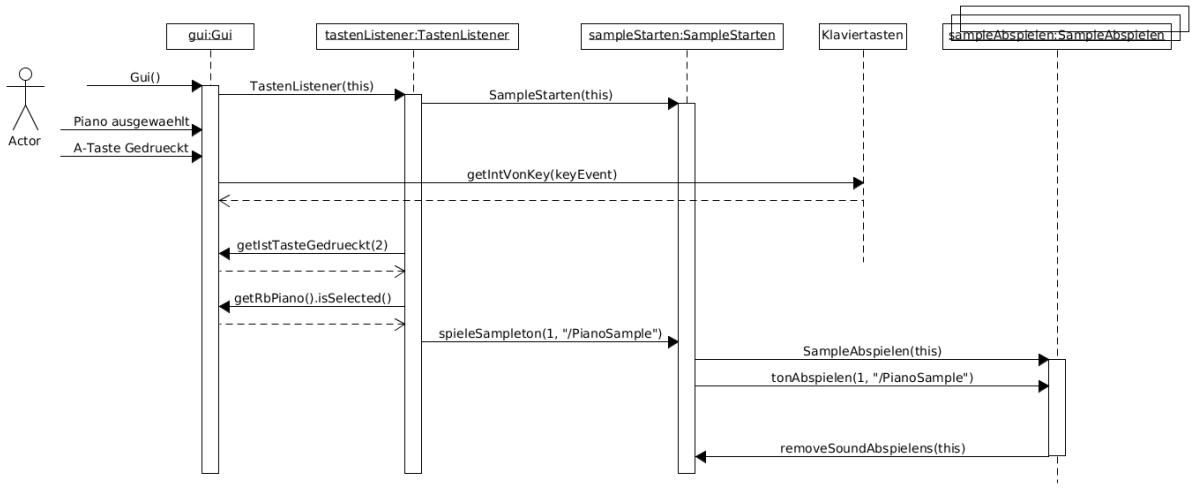


Abbildung 1: Sequenzdiagramm für das Abspielen eines Tones

fehler verbessern

Beschreibung: In diesem Szenario will der Benutzer das untere Cis des Piano Samples abspielen. Als erstes startet dieser das Programm und erzeugt dadurch ein Gui-Objekt. Als nächstes wählt er das Piano Sample über die Radiobutton aus (Piano ist zu Begin ausgewählt). Drückt der Benutzer nun die A-Taste wird der Ton abgespielt.
Damit der Ton abgespielt werden kann benötigt das Programm ein TastenListener-Objekt welches ständig abprüft, ob eine der, mit Tönen belegten Tasten, gedrückt wurde. Dieses Objekt wird im Konstruktor der Gui erzeugt.
Um nun einen Ton starten zu können, wird ein SampleStarten-Objekt benötigt. Dies wird im Konstruktor des TastenListener-Objektes erzeugt.
Wird nun die A-Taste gedrückt, sucht die Methode getIntVonKey(KeyEvent keyEvent) in der Klasse Klaviertasten, den der Taste zugewiesene Integer-Wert (die A-Taste hat den Wert 1). Mit diesem Wert kann man nun das Array istTasteGedrückt an der stelle 2 auf true setzten.
Der TastenListener prüft dabei Ständig ab welche Tasten gedrückt wurde. Wurde eine Taste gedrückt frägt er noch das dabei ausgewählte Sample ab.
Diese Informationen schickt der TastenListener dann zu dem SampleStarten-Objekt. Welches wiederum, in einem Vektor, ein SampleAbspielen-Objekt erzeugt und anschließend über die Methode tonAbspielen den Ton abspielt.
Sobald der Ton abgespielt wurde löscht sich das SampleAbspielen-Objekt wieder aus dem Array. Wie der Ton in dem SampleAbspielen-Objekt funktioniert können sie der **Abbildung 4** entnehmen.

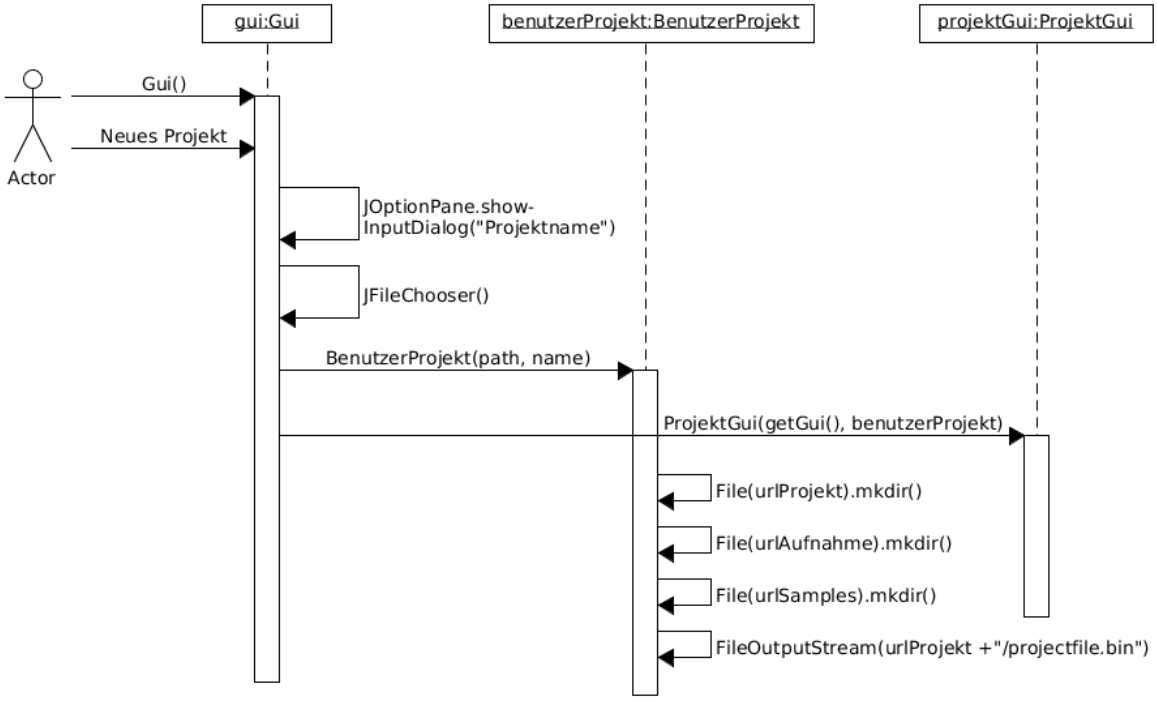


Abbildung 2: Sequenzdiagramm für erstellen eines neuen Projektes

Beschreibung: In diesem Szenario will der Benutzer ein neues Projekt erstellen. Nach dem Starten des Programmes und erzeugen eines Gui-Pbjektes klickt der Benutzer auf das Menü "Neues Projekt" in der ToolBar. Als nächstes gibt dieser den Namen des Projektes in einem Dialog ein und wählt anschließend den Dateipfad über einen FileChooser aus. Mit diesen Informationen kann nun ein BenutzerProjekt-Objekt erzeugt werden. Der Konstruktor der BenutzerProjekt-Klasse erzeugt dann einen Projekt-Ordner und in diesem ein Aufnahme-Ordner, Samples-Ordner und eine projectfile.bin datei. mit dem BenutzerProjekt-Objekt kann nun die ProjektGui gestartet werden.

2.4 Struktogramme

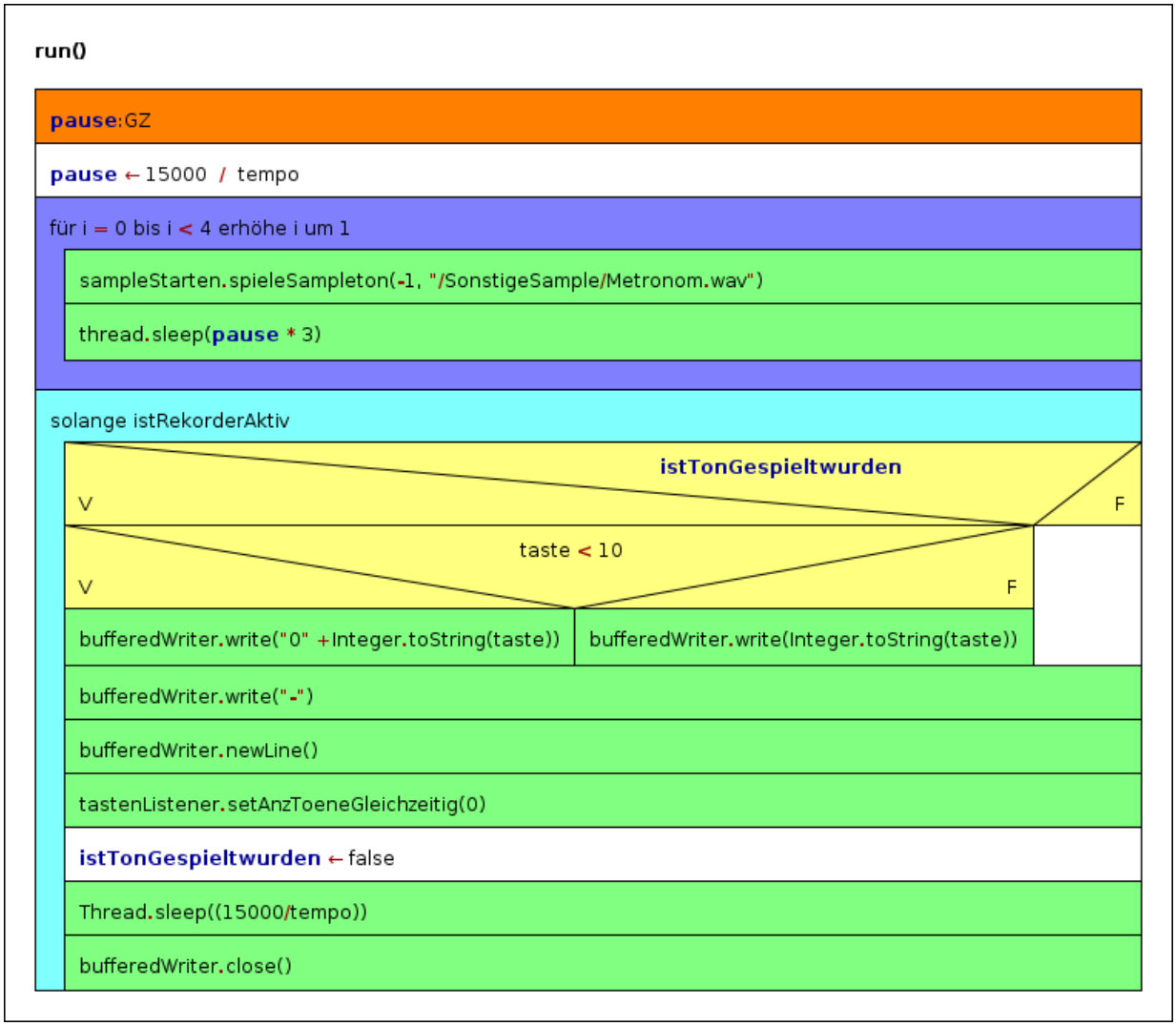


Abbildung 3: Struktogramm für das Aufnehmen eines Tones

Beschreibung: Am Anfang der run()-Methode wird die Zeit für die Pause zwischen zwei 16tel berechnet. Anschließend spielt die Methode 4 Metronomschläge in 4tel abständen ab. Wird kein Ton abgespielt, schreibt die Methode einen Bindestrich in die Aktuelle Zeile des Textdokumentes und wartet eine 16tel-Pause.
Wird ein Ton gespielt, schreibt die Methode dessen ID-Nummer in das Textdokument (bei einstelligen Zahlen eine Null voran gestellt werde).

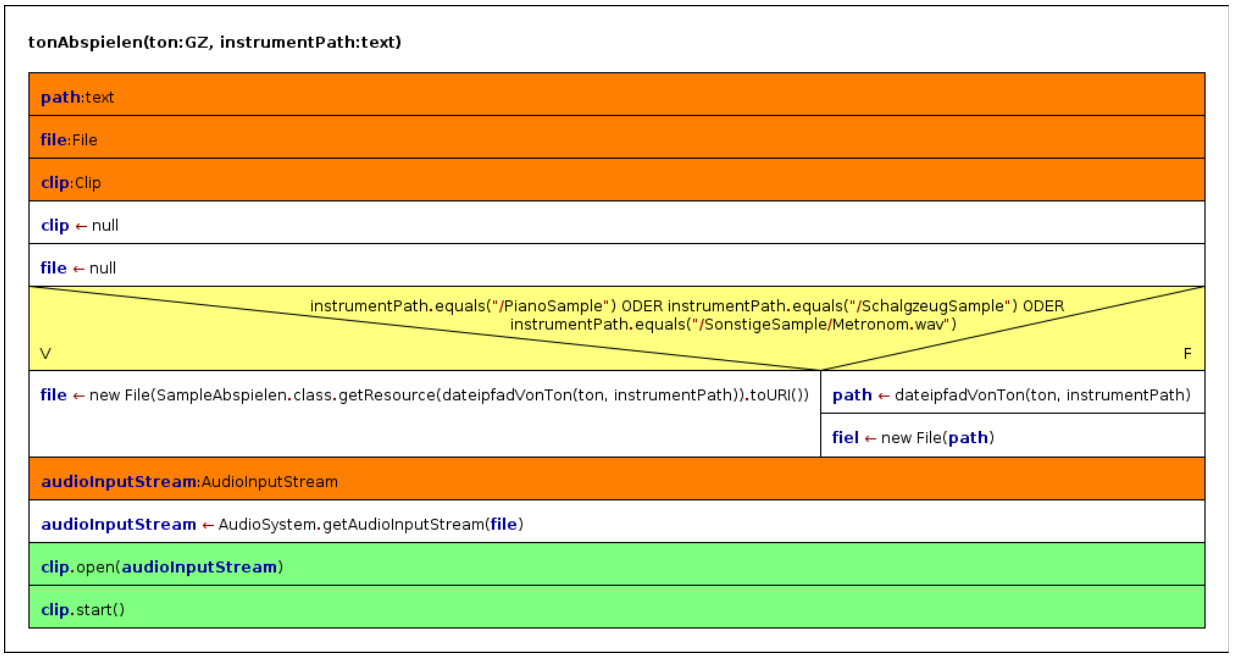


Abbildung 4: Struktogramm für das Abspielen eines Tones

Beschreibung: In dieser Methode wird ein Clip erzeugt und gestartet. Allerdings müssen Dateien aus dem Javaprojekt-Ordner anderst geöffnet werden als Dateien die auserhalb des Javaprojekts.

2.5 Beschreibung einiger Methoden

Wir haben während des Projekts alle Methoden mithilfe von Javadocs dokumentiert. Unser komplettes Project ist also hier einsehbar:

Einige größere Methoden werden wir in dieser Dokumentation allerdings etwas genauer beschreiben.

2.6 Javadocs

Der komplette Quelltext wurde mittels Javadocs auskommentiert. Hier kann nachgelesen werden, welche funktion die einzelnen Methoden erfüllen, sowie ihre Übergabe- und Rückgabeparameter, bzw. Exeptions eingesehen werden.
Die komplette Javadocs Dokumentation kann hier geöffnet werden:
`./JavaDocs/index.html`

3 Benutzerhandbuch

3.1 Einleitung

Das Digital Input Computer Keyboard ist eine Art elektronisches Keyboard, dass verschiedene Samples abspielen und aufnehmen kann. Man kann mithilfe der gegebenen Samples (Piano und Schlagzeug) spielen und Tonspuren aufnehmen. Aufgenommene Tonspuren können dann auch parallel abgespielt werden. Es können aber auch eigene Samples in das Programm geladen werden. Mit diesen kann man ebenso verfahren wie mit den gegebenen.

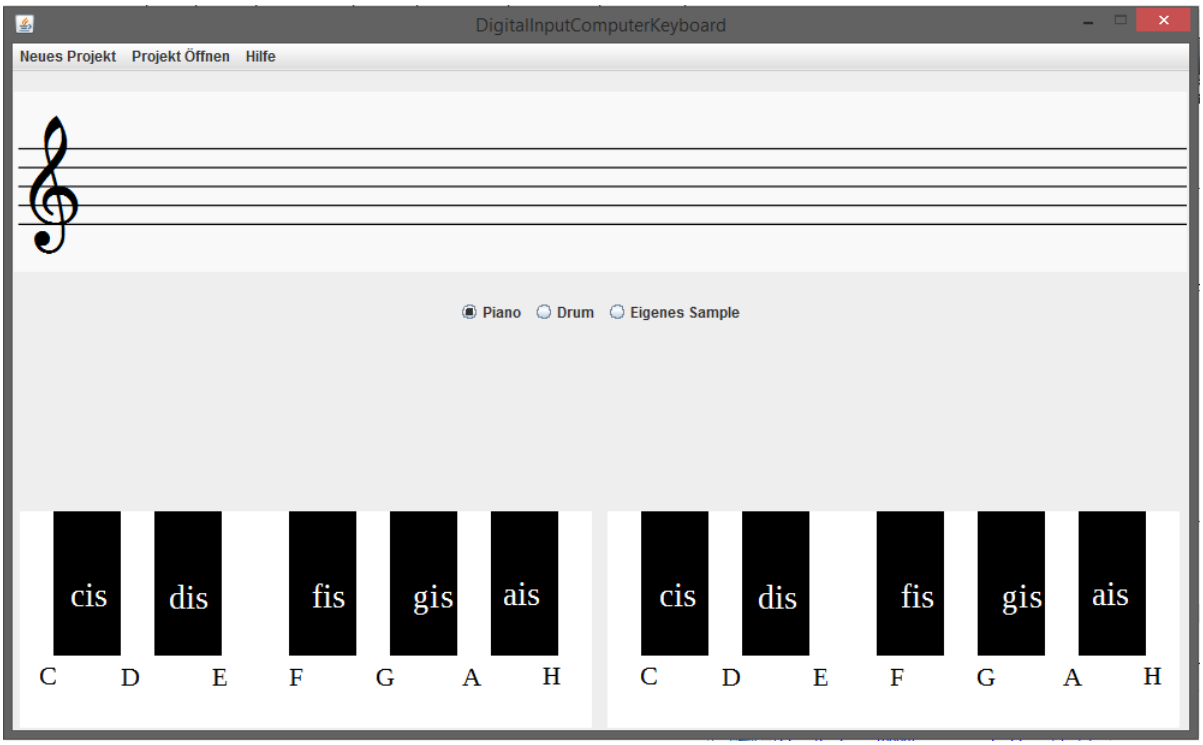


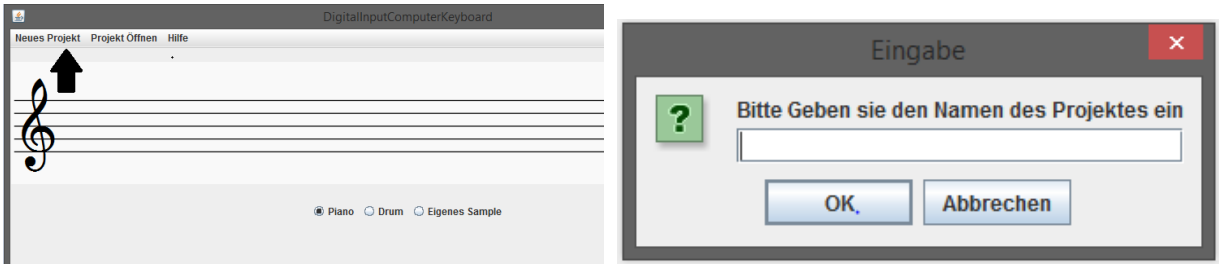
Abbildung 5: Bild der Benutzeroberfläche

3.2 Projekt

Mithilfe des DICKs hat man die Möglichkeit Projekte zu erstellen. Ein Projekt enthält dann ein Projektfile, einen Ordner in dem eigene Aufnahmen abgespeichert sind und einen Sample Ordner in den man eigene Samples hinein laden kann.

3.2.1 Erstellen

Mithilfe des Buttons Neues Projekt"kann man einen Projekt Ordner erstellen. Dieser Ordner muss erst benannt und dann ein Speicherort ausgewählt werden. Während der Namensgebung aktivieren sich die Tasten, dies kann aber ignoriert werden.



(a) Neues Projekt -Button (b) Eingabe-Feld für den Projektname

Abbildung 6: Projekt Erstellen

Sobald das Projekt benannt wurde muss nur noch ein Speicherort ausgewählt werden.

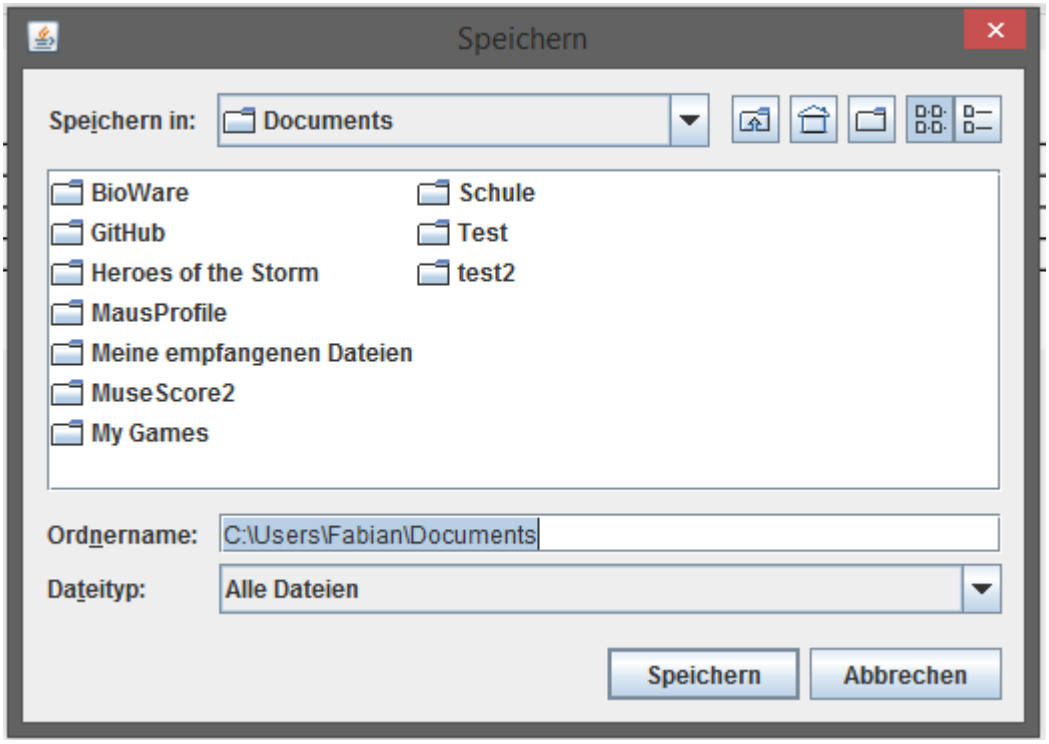


Abbildung 7: Speicherplatzauswahl

3.2.2 Öffnen

Wenn man bereits ein Projekt erstellt hat kann man dieses mithilfe des Buttons Projekt Öffnen"öffnen. Man sucht den Projekt Ordner und wählt die Projektfile.bin aus. Sobald dies erledigt wurde, öffnet sich ein separates Fenster in dem man weitere Funktionen zur Auswahl hat.

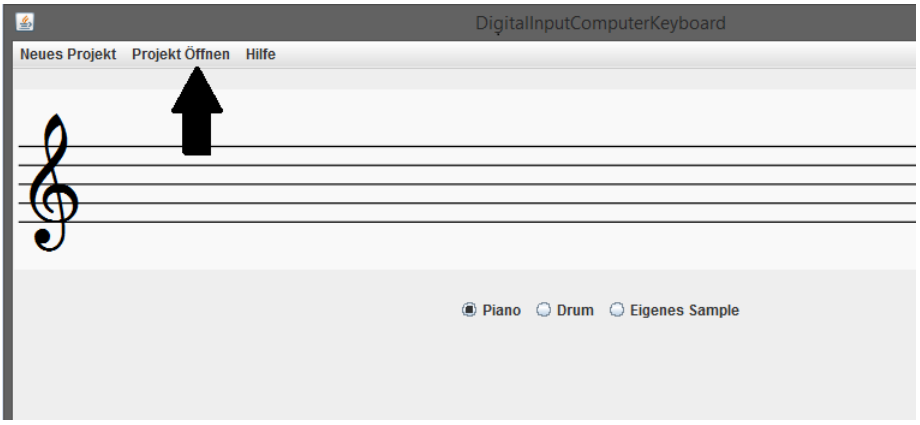
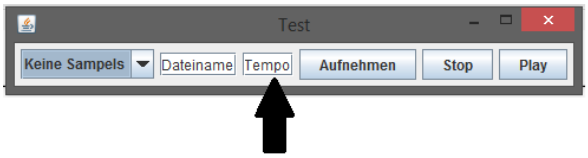


Abbildung 8: Der Button Projekt Öffnen

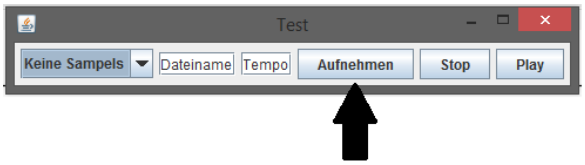
3.2.3 Aufnehmen

Sobald man ein Projekt erstellt und geöffnet hat, kann man gespieltes Aufnehmen. Dazu muss man das Tempo und den Namen der Datei eingeben die aufgenommen werden soll. Zu beachten ist hierbei dass man sobald der Name oder das Tempo eingegeben wurde Enter gedrückt wird, da diese während dem spielen sonst verändert werden.

(a) Eingabe-Feld für den Dateiname



(b) Eingabe-Feld für das Tempo



(c) Aufnehmenbutton

Abbildung 9: Aufnahmespezifische Bedienelemente

Sobald alles ausgefüllt wurde muss man den Aufnehmenbutton betätigen. Es folgen vier einleitende Schläge in dem gewählten Tempo bevor aufgenommen wird. Um die Aufnahme wieder zu stoppen einfach den Stopbutton betätigen.

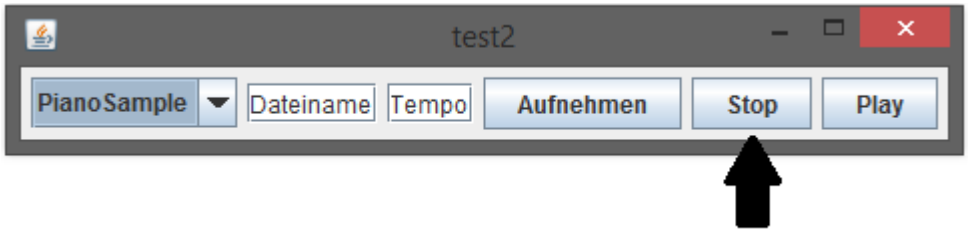


Abbildung 10: Der Stopbutton

3.2.4 Abspielen

Um ein aufgenommenes Sample abzuspielen betätigt man den PlayButton. Dadurch öffnet sich ein Fenster in dem man seine aufgenommenen Samples sieht.

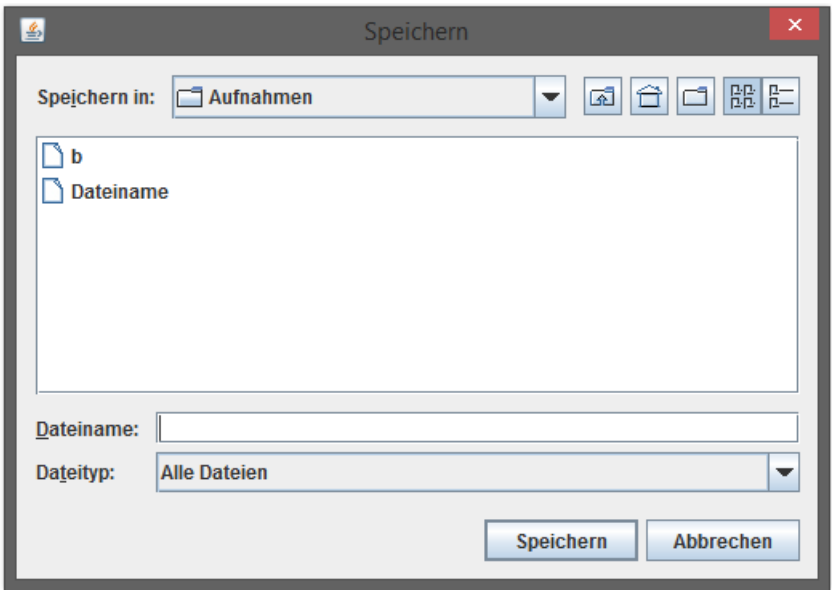


Abbildung 11: Das Fenster in dem gespeicherte Samples angezeigt werden

Durch Auswahl einer Aufnahme wird ein Fenster geöffnet das einen Button enthält. Durch betätigen des Buttons wird das Sample abgespielt. Dabei werden die aufgenommenen Noten angezeigt. Hier wurde beispielhaft die Aufnahme b geöffnet.

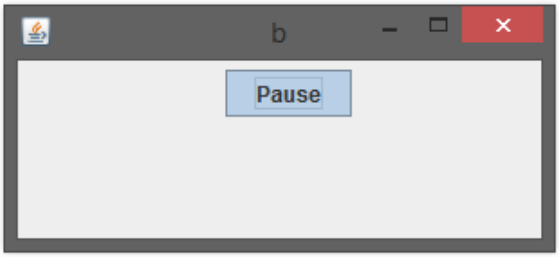


Abbildung 12: Das Abspielen Fenster eines Samples

Um die Aufnahme nun abzuspielen muss man nur noch den Pausebutton betätigen. Durch nochmaliges betätigen wird das abspielen angehalten und kann durch erneutes betätigen fortgesetzt werden.

Anmerkung: Wenn mehrere Aufnahmen geöffnet sind und gleichzeitig abgespielt werden wollen, kann man dies mithilfe der Leertaste tun.

3.2.5 Eigene Samples

Um eigene Samples in das Programm einzufügen muss man diese in den Sample Ordner des Projekt einfügen. Dabei muss man einige Dinge beachten:

- Die Töne des Samples müssen 8 oder 16 bit lang sein.
- Es müssen wav Dateien sein.
- Alle Töne eines Samples müssen in einem eigenem Ordner abgelegt sein
- Die Töne müssen wie folgt bezeichnet sein:
  - A4
  - A5
  - Ais4
  - Ais5
  - B4
  - B5
  - C4
  - C5
  - Cis4
  - Cis5
  - D4
  - D5
  - Dis4
  - Dis5
  - E4
  - E5
  - F4
  - F5
  - Fis4
  - Fis5
  - G4
  - G5
  - Gis4
  - Gis5

Wenn diese Bedingungen erfüllt sind sollte das Sample in dem Dropdownmenü auswählbar sein.

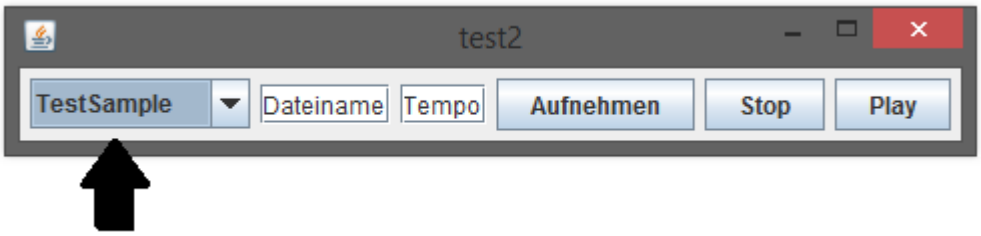


Abbildung 13: Dropdownmenü

Anmerkung:

Ein Sample im Dropdownmenü auszuwählen reicht noch nicht aus. Zusätzlich muss in der Benutzeroberfläche Eigene Samples ausgewählt sein, da sonst das entsprechende schon in dem Programm integrierte Sample abgespielt wird.

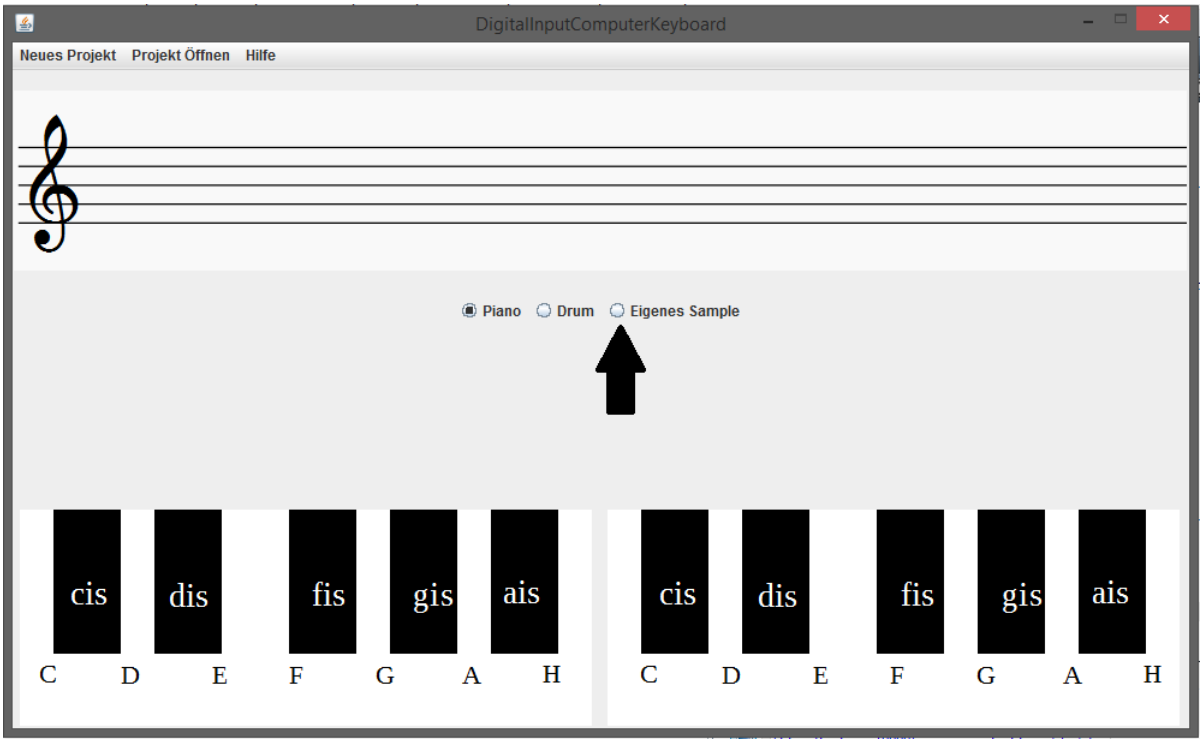


Abbildung 14: Eigenes Sample auswählen