

Medientechnik – Übungsblatt 5

Lernziele:

- Programmieren mit Sound in Java

Aufgabe 1: ClipPlayer

10 Punkte

Erweitern Sie den SimpleClipPlayer aus der Übung. Erstellen Sie ein Fenster mit dessen Hilfe das gleichzeitige Abspielen von mindestens drei Clips gesteuert werden kann. Für jeden Clip sollte es folgende Steuerungsmöglichkeiten geben:

- **Play/Stop-Button:** Per Klick auf den Play/Stop-Button kann der Clip in einer Endlosschleife abgespielt werden. Durch ein erneutes Klicken wird das Abspielen wieder gestoppt.
- **Volume/Gain:** Die Lautstärke soll über einen JSlider einstellbar sein. Recherchieren Sie zuerst was der Unterschied zwischen MASTER_GAIN und VOLUME ist (z.B. mit Hilfe der Java API). Welche Werte akzeptieren die entsprechenden FloatControls und wie wirken sich die Werte auf die Lautstärke aus? Entscheiden Sie sich auf Grund Ihrer Recherche für eine der beiden Controls (MASTER_GAIN oder VOLUME). Schreiben Sie Ihre Begründung als Kommentar in Ihren Quelltext (über dem Aufruf der Funktion `getControl()`). Wählen Sie dementsprechend eine sinnvolle Skala und zeigen Sie diese unter dem JSlider an.
- **Sample Rate:** Die Abspielrate soll ebenfalls durch einen JSlider einstellbar sein. Dabei soll die Skala von 1/4 (Abspielrate entspricht einem Viertel der originalen Abspielrate) bis 4 (vierfache Abspielrate) reichen. Zeigen Sie Skala unter dem JSlider an (siehe Abbildung 1).

Tip: Überlegen Sie welche mathematische Funktion geeignet ist um die lineare Skala eines JSliders auf die oben genannte Skala abzubilden. Da die angezeigten Werte (1/4, 1/2, 1, 2 und 4) nicht den linearen Werten des JSlider entsprechen müssen die Labels überschrieben werden (`JSlider.setLabelTable(Dictionary labels)`).

So sollte Ihr Ergebnis in etwa aussehen (wobei die Skala für den Volume Slider hier noch fehlt):

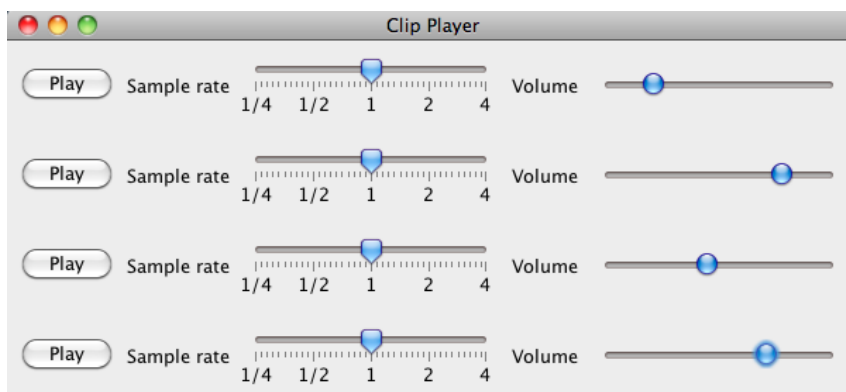


Abbildung 1: Steuerungsfenster für den ClipPlayer

Hinweis: Achten Sie auf saubere Programmierung! Nicht alle Controls werden in allen Systemen und von allen Java-Versionen unterstützt. Fügen Sie deshalb nur dann ein Steuerelement zu Ihrem Fenster dazu, wenn es auch tatsächlich unterstützt wird (z.B. durch Anfrage bei der entsprechenden Line, ob das Control unterstützt wird: `isControlSupported(Control.Type control)`)

Geben Sie sowohl Ihren src-Ordner mit dem Namen „Aufgabe 1“ als auch ein Runnable jar mit dem Namen ClipPlayer.jar ab.

Aufgabe 2: MIDI-Orgel

7 Punkte

Schreiben Sie ein Programm, das durch Druck auf verschiedene Tasten der Computer-Tastatur verschiedene Töne erzeugt. Dabei soll eine volle Oktave (d.h. die Töne C, C#, D, D#, E, F, F#, G, G#, A, A#, H und C) abspielbar sein. Die Zuordnung von Tasten auf MIDI-Noten ist dabei fest vorgegeben (siehe Abbildung 2). Drückt man also beispielsweise auf die Taste ‚A‘ soll der Ton C (MIDI-Nummer 24) abgespielt werden. Die Note soll dabei solange gehalten werden bis die Taste wieder losgelassen wird.

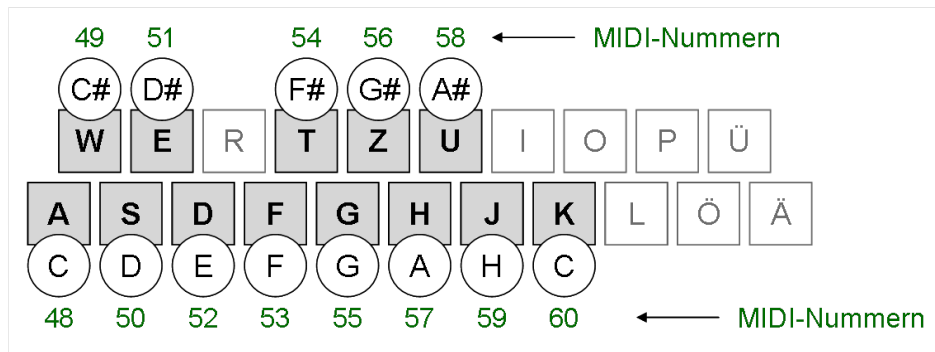


Abbildung 2: Zuordnung von Tasten auf MIDI-Noten

Tipp: In der Vorlage auf der Vorlesungshomepage befindet sich die Java-Klasse *SynthNote.java*, die demonstriert wie man mit Hilfe des `javax.sound.midi`- Pakets Töne erzeugen kann.

Erstellen Sie außerdem einen JFrame, das die Ereignisse (Tastendrucke und gespielte Noten) anzeigt (siehe Abbildung 3).

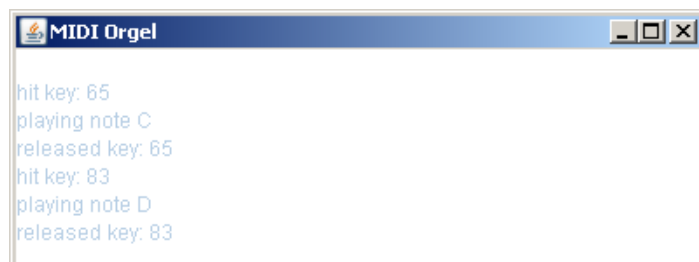


Abbildung 3: Ausgabefenster der MIDI-Orgel

Geben Sie sowohl Ihren src-Ordner mit dem Namen „Aufgabe 2“ als auch ein Runnable jar mit dem Namen MidiOrgel.jar ab.

Abgabe

1 Punkt

Noch einige Hinweise zur Abgabe (um die Korrektur zu erleichtern):

- Zu den Programmieraufgaben sollen nur die src-Dateien und ein runnable jar abgegeben werden.
- Packen Sie dann alle Dateien dieses Übungsblatts in eine Datei („blatt5.zip“) und geben Sie diese bis spätestens **07.07.2010, 11:59 Uhr (MEZ)** über Uniworx ab! Eine spätere Abgabe oder eine Abgabe per E-Mail ist nicht möglich!

Bei Problemen oder Fragen kann die Programmierberatung im CIP-Pool der Amalienstraße 17 aufgesucht werden. Die entsprechenden Termine befinden sich auf der Homepage zur Vorlesung.