

Übung 3

Aufgabe 1

Erstellen Sie ein FX-Programm, dass ein oder mehrere grafische Objekte anzeigt. (Sie können dazu z.B. eine passende Lösung aus der letzten Übung weiter nutzen) Fügen Sie jedem Objekt einen Schatteneffekt (*DropShadow*) hinzu, welcher ausgelöst wird, wenn sich der Mauszeiger darüber befindet.

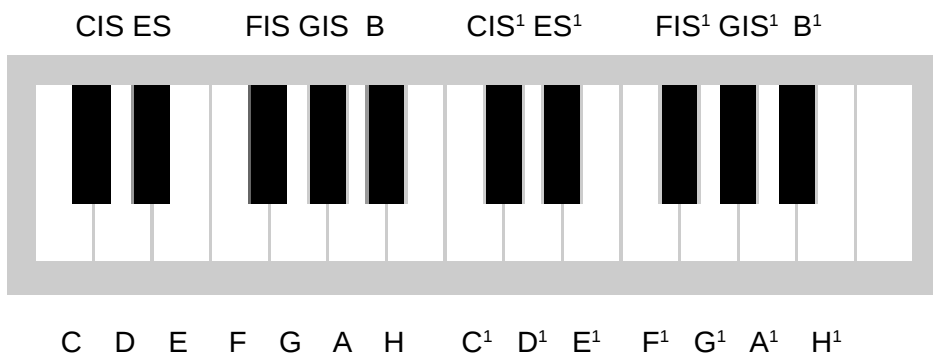
Implementieren Sie dies

- a) mittels Maus-Ereignissen und
- b) indem Sie diesen Effekt an die Hover-Property des Objektes binden.

Aufgabe 2

Schreiben Sie ein FX-Programm, dass ein elektronisches Piano imitiert. Es soll sich sowohl mit der Maus, als auch mit der Tastatur spielen lassen. Zu dieser Aufgabe finden Sie passende Audio-Samples auf dem [S:\-Laufwerk](http://www.apronus.com) (Quelle: <http://www.apronus.com>).

Die Zuordnung der Samples zur Klaviatur können Sie dem folgenden Bild entnehmen.

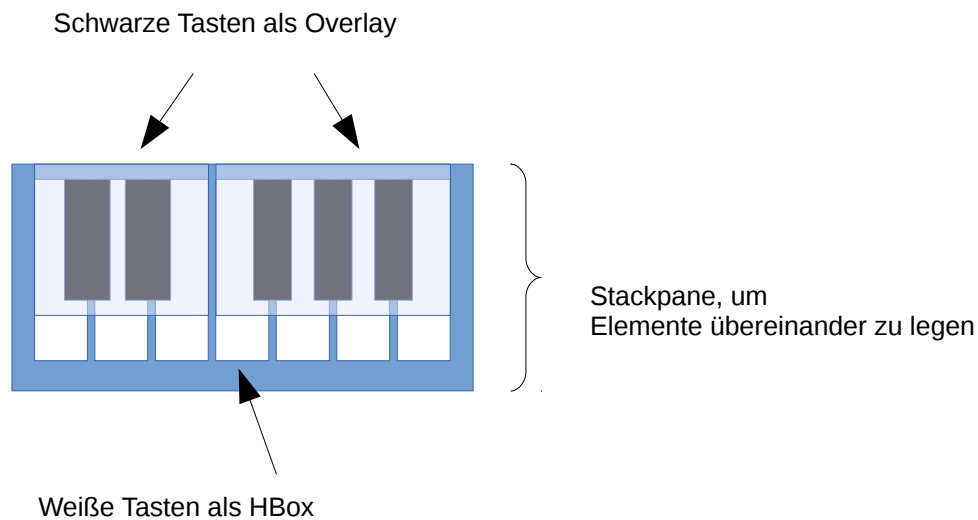


Für die Zuordnung der Tasten der Computer-Tastatur zu den einzelnen Tönen gibt es verschiedene Möglichkeiten. (z.B. Weiße Tasten: y-x-c-v-b-n-m + w-e-r-t-z-u-i, Schwarze Tasten: s-d-g-h-j + 3-4-6-7-8 oder Weiße Tasten: a-s-d-f-g-h-j-k-l-ö-ä, Schwarze Tasten: w-e-t-z-u-o-p oder Weiße Tasten: 1-2-3-4-5-6-7-8-9-0-q-w-e-r, Schwarze Tasten: [SHIFT] + (1-2-4-5-6-8-9-q-w-e) oder gemischt: 1-²-3-⁴-5-6-⁷-8-⁹-0-^q-w-e-^r-t-^z-u-i-^o-p-^ü-a-^s-d)

Hilfreich wäre es, wenn Sie die gewählte Tastaturbelegung am Bildschirm direkt auf den entsprechenden Tasten anzeigen. Beim Anschlagen der entsprechenden Taste auf der Computertastatur soll neben dem Abspielen des richtigen Samples auch ein visuelles Feedback in Ihrer GUI erkennbar sein.

Hinweise: Erstellen Sie sich am besten eine Klasse „Taste“, welche die grafische Repräsentation einer Taste, das zugehörige Audio-Sample (Objekt der Klasse *Media*), sowie den passenden *KeyCode* für das Ansteuern mit der PC-Tastatur als Attribute besitzt. In Ihrer Applikation können Sie sich dann einfach ein Array von Tasten anlegen. Eventuell kann ein „enum Toene“ (mit Symbolen für die Einzeltöne in der Reihenfolge: C-CIS-D-ES-E-F-FIS-G-GIS-A-B-H-C¹- ...) für den Arrayzugriff und das Laden der Audiodateien etwas Programmieraufwand einsparen. Für die Anordnung der weißen Tasten eignet sich eine *HBox*, die Schwarzen Tasten kann man ebenfalls als eine (oder mehrere) *HBox* darüberlegen (*StackPane*). Siehe Skizze. **Wichtig:** Das Overlay muss **setPickOnBounds(false)**; gesetzt haben, sonst empfangen die Objekte darunter keine Maus-Ereignisse. Das Einbinden externer Ressourcen machen Sie wie in Aufgabe 7 aus Übung 2.

Beispiellayout für Aufgabe 2:



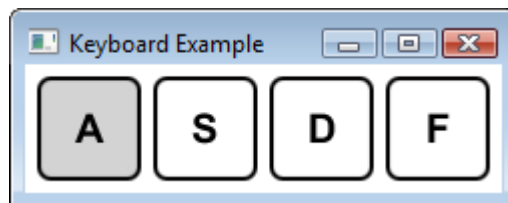
Frage: Wo (in welchem Objekt) fangen Sie am besten die KeyEvents ab? Beachten Sie, dass das Target eines KeyEvents immer nur der Knoten des Szenegraphen ist, welcher aktuell den Fokus hat.

Aufgabe 3

Erweitern Sie das Programm aus Aufgabe 2 so, dass sich mittels einer Full-Drag-Operation mehrere Töne nacheinander abspielen lassen, ohne jede Klaviertaste einzeln klicken zu müssen.

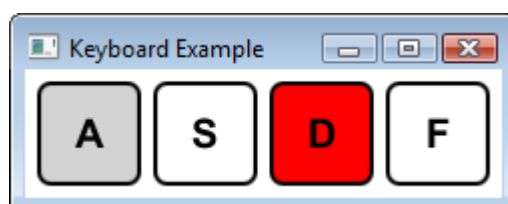
Aufgabe 4

Erstellen Sie ein JavaFX-Programm, welches die vier Buchstaben A-B-C-D anzeigt, wobei jedes in einem Rechteck dargestellt wird. (Quelle: KeyEvent-Beispiel aus der JavaFX Online Dokumentation) Am besten Sie erzeugen sich dazu wieder eine eigene Klasse. Diesmal empfiehlt es sich diese mittels Vererbung z.B. von *Group* erben zu lassen. So können Sie Ihr eigenes Objekt direkt in den Szenegraphen einfügen.



Das Programm soll nun folgendes Verhalten zeigen:

- Drückt man eine der Tasten A-S-D-F, dann wird das zugehörige Rechteck so lange rot eingefärbt dargestellt, bis die Taste wieder losgelassen wird. Nach Loslassen erscheint das Objekt wieder in der Ausgangsfarbe.



- b) Das Objekt, welches den Eingabefokus besitzt, soll grau hervorgehoben werden. Zu Beginn soll immer das Objekt A den Eingabefokus bekommen. Um den Fokus anschließend zu einem anderen Objekt zu bewegen, können die Pfeiltasten \leftarrow , \rightarrow verwendet werden.
- c) Drückt man die Return-Taste so färbt sich das Objekt rot, welches gerade den Eingabefokus besitzt. Bei Loslassen der Taste soll wieder die vorige Farbe angezeigt werden.

Hinweise: Den Fokus können Sie einem Knoten selbst zuordnen mittels `node.requestFocus()`; Ein Knoten nimmt am Fokuszyklus teil, wenn `node.setFocusTraversable(true)`; gesetzt ist.

Beachte: In JavaFX gibt es keinen extra Fokus-Event, welcher anzeigt, ob ein Objekt den Fokus gerade bekommt oder verliert. Es gibt jedoch eine *focusProperty*, auf die man lauschen bzw. an die man ein Verhalten binden kann.

Aufgabe 5

Erinnern Sie sich noch an die Aufgaben 4-9 aus Übung 1? Es ging um ein Programm zur Erfassung von Personendaten. Sie können es für diese Aufgabe wieder her nehmen und erweitern.

Erstellen Sie eine Klasse Person, welche passende Attribute entsprechend der bereits vorhandenen Eingabefelder Ihrer UI aufweist. Verwenden Sie für diese Attribute observierbare Properties. Fügen Sie auch Methoden zum Laden und Speichern eines Objektes in die Klasse ein. Binden Sie nun die Attribute eines Personenobjektes bidirektional an die Eingabefelder. Testen Sie das Programm und vergleichen Sie es mit Ihrer anfänglichen Lösung aus Übung 1.

Aufgabe 6

Erweitern Sie das Programm aus Aufgabe 5 so, dass

- a) anstelle von Textfeldern für Geschlecht und Alter entsprechend ein DropDown und ein Slider -Kontrollelement eingesetzt werden.
- b) dem Feld Postleitzahl eine Prüfroutine hinzugefügt wird, welche sicherstellt, dass nur Zahlen eingegeben werden können und insgesamt nur 5 Zeichen. Verletzt der Anwender diese Bedingungen soll er mittels *ToolTip* auf die korrekte Eingabefunktion hingewiesen werden.