

Übungsblatt 5 – Mensch-Maschine-Interaktion 2

Achtung: Dieses Übungsblatt kann in Gruppen mit bis zu zwei Studenten bearbeitet werden. Pro Team ist nur eine Abgabe nötig.

Aufgabe 1: Gestures with MT4j (6 Punkte)

In dieser Aufgabe geht es darum selbstdefinierte Gestensets zu unterstützen. Dafür müssen sie einen Gestern-Recorder implementieren und diese aufgenommenen Gesten dann mit Hilfe des 1\$ Gesten-Erkenner [2] interpretieren.

Erstellen Sie nun eine simple Anwendung, die folgende Aufgaben erfüllt:

1. Es gibt eine Scene, in der ein Nutzer 3 eigene Gesten definieren kann:
 - a. Zum Löschen von Bildern.
 - b. Um ein Bild auf die gesamte Bildschirmfläche zu maximieren.
 - c. Um ein Bild zu skalieren (runter).
2. Sobald das Einstellen der Gesten fertig ist sollten die Nutzer in der Lage sein zu der Scene mit den Bildern zu wechseln (vorherige Übung).
3. Die Bilder in dieser Szene sollen jetzt auf die Gesten reagieren, die zuvor definiert wurden (löschen, maximieren und runterskalieren).

Erstellen Sie ein package `de.lmu.mmi2` im Verzeichnis `MT4j-desktop/examples` und arbeiten Sie in diesem. Weitere technische Hinweise gibt es am Ende dieses Übungsblatts.

Erstellen Sie in Ihrer Abgabe einen Ordner „aufgabe1“, der die Lösung zu dieser Aufgabe enthält. Geben Sie eine ausführbare JAR Datei ab und eine zip Datei, welche den Sourcecode zu Ihrer Abgabe enthält.

Aufgabe 2: \$1 Recognizer and Rubine (4 Punkte)

In der Übung wurden der Rubine Classifier und der 1\$ Gesten-Erkenner vorgestellt. Nennen Sie jeweils einen Vorteil ODER einen Nachteil für jeden. Geben Sie außerdem jeweils eine Situation an, in der sie benutzt werden könnten.

Erstellen Sie in Ihrer Abgabe einen Ordner „aufgabe2“, der die Lösung zu dieser Aufgabe enthält. Die Lösung muss als PDF, PNG oder TXT abgegeben werden. Kombinationen der Formate sind möglich.

Referenzen

1. MT4j Multitouch Framework for Java. <http://www.mt4j.org>
2. \$1 Unistroke Recognizer. <http://depts.washington.edu/aimgroup/proj/dollar/>
3. Wobbrock, J.O., Wilson, A.D. and Li, Y. (2007). Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes. Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '07). Newport, Rhode Island (October 7-10, 2007). New York: ACM Press, pp. 159-168.

Abgabe

Bitte geben Sie Ihre Lösung als ZIP-Datei bis zum 26.11.2012 14:00 Uhr im UniWorx Portal (<https://uniworx.ifi.lmu.de/?action=uniworxCourseWelcome&id=109>) ab.

Technische Hinweise #2

Ersetzen Sie den Inhalt des Packages `unistrokeProcessor` mit dem Inhalt der angehängten zip Datei. Der exakte Ort in `mt4j-core`:

```
org.mt4j.input.inputProcessors.componentProcessors.unistrokeProcessor
```

Definieren Sie Ihre eigenen Gesten // in `CreateTemplateScene.java`

Der `unistroke gesture manager` des `MT4j toolkits` basiert auf dem `1$ Erkenner`, der in der Übung vorgestellt wurde. Um Ihre eigenen Gesten zu definieren müssen Sie ein `Template` erstellen und zu der Liste existierender `Templates` hinzufügen. Um das zu tun, erstellen Sie eine `Scene` wenn die Anwendung gestartet wird. Fertiger Code findet sich in `GestureExercise.java`

Wir werden die Klasse `CreateTemplateScene` verwenden, um `custom gesture templates` zu erzeugen.

1. Erzeugen Sie ein Rechteck, in dem Sie die Geste malen.
2. Erstellen Sie eine Reihe von `Buttons`, um dem System mitzuteilen:
 - a. Wann die Aufnahme der Geste gestartet und beendet werden soll.
 - b. Die Aufnahmezone zu resettet, falls ein Fehler passiert ist.
 - c. Die Geste zu speichern.

Um einen Knopf zu erzeugen verwenden Sie ein einfaches Rechteck, gefüllt mit Text und einem `tap listener`, z.B.: `addGestureListener(TapProcessor.class, new IGestureEventListener() {})`...

3. Benutzen Sie einfache Gesten für bessere Ergebnisse bei der Erkennung.
4. Fügen Sie einen `RawFingerProcessor` zur `Scene` hinzu und hören Sie auf `touch events` (eine Methode dafür ist schon verfügbar). Sie können `MTEllipse` verwenden, um die Liste von Punkten auf dem Bildschirm zu visualisieren.
5. Wenn der `Speichern Knopf` berührt wird, verwenden Sie die `liste von Touchpunkten` um ein neues `Template` für eine Geste zu erzeugen.
6. Fügen Sie dieses `Template` zur `Liste bekannter Templates` hinzu.

Dazu verwenden Sie `GestureExercise`, das `initGestureProcessors()`; aufruft und `addGestureProcessors(points, gestureID);`

Eigene Gesten verwenden // in `ImageScene.java`

7. Sobald Sie mit der Definition von Gesten fertig sind wechseln Sie zur `Scene` mit `dne Bildern`.
8. Fügen Sie hier eine Reihe von `Bildern` und einen „`command`“ `Button` hinzu.
9. Ein `tap` oder Drücken des „`command`“ `Buttons` deaktiviert alle `Standardaktionen` (`dragging`, `scaling` und `rotation`) und aktiviert Ihre `Gesten`.