

Plattformübergreifende interaktive Tischanwendungen mittels TUIOFX

Mirko Fetter, David Bimamisa, Tom Gross

Lehrstuhl für Mensch-Computer-Interaktion, Otto-Friedrich-Universität Bamberg

Zusammenfassung

Die Wahl eines Toolkits für die Entwicklung von Multi-Touch-Anwendungen für interaktive Tische ist häufig eine Kompromissentscheidung. TUIOFX bündelt die unterschiedlichen Stärken verschiedener Toolkits für die plattformübergreifende Entwicklung von Anwendungen in Mehrbenutzerkontexten.

1 Einführung

Interaktive Tische versprechen schon lange neue Möglichkeiten der örtlichen Zusammenarbeit in Gruppen und sind daher ein langjähriges Forschungsthema im Bereich der computer-gestützten Gruppenarbeit und der Mensch-Computer-Interaktion. Doch selbst mit der Einführung der Microsoft Surface Hardware (heute PixelSense), dem ersten kommerziell erfolgreichen und massenproduzierten interaktiven Tisch, fristen interaktive Tische noch ein Nischendasein im Arbeitsalltag. Der Einsatzbereich solcher Tische reicht heute leider oft nur vom interaktiven Exponat im Museum über den effekthaschenden Showcase für das Unternehmensportfolio auf Messen zum Restaurantführer in der Hotellobby (Fetter et al. 2013).



Abbildung 1. Drei kleine Beispielanwendungen die in nur wenigen Zeilen Code mit TUIOFX verwirklicht wurden: Ein einfacher, mittels CSS gestylter Bildbetrachter (1), ein Moodboard (2) und ein simples Malprogramm (3).

Ein Grund für dieses Nischendasein ist mitunter der Mangel an erhältlicher Produktiv-Software für interaktive Tische. Die Ursache für diesen Mangel liegt wiederum in der einge-

schränkten Verfügbarkeit ausgereifter Toolkits für die Diversität an verfügbarer Tisch-Hardware der verschiedenen Hersteller. TUIOFX (Abb. 1) hat das Ziel diese Lücke auszufüllen, indem es die plattformübergreifende Entwicklung für interaktive Tische unterstützt, wie wir im Folgenden näher erläutern.

2 Anforderungen und Konzept

Die Entwicklung von umfangreichen Anwendungen für interaktive Tische wird leider immer noch unzureichend durch Toolkits unterstützt. So gingen zwar zahlreiche hervorragende, teilweise plattformunabhängige Toolkits—und in diesem Sinne verwandte Arbeiten—aus der Forschungsgemeinde hervor (z.B. PyMT (Hansen *et al.* 2009), libTisch (Echtler & Klinker 2008), MT4j (Ruff & Laufs 2014)), viele davon besitzen aber einen oder mehrere der folgenden Nachteile: langsamer Einstieg durch neues Erlernen; zu geringer Umfang an grafischen Standard-Bedienelementen; unzureichende Dokumentation; keine aktive Weiterentwicklung des Toolkits; kleine Entwickler-Communities bei Problemen und Fragen; keine Werkzeugunterstützung; etc. Das umfangreichere, kommerzielle MS Surface SDK (Microsoft Corporation 2014) auf der andere Seite lässt viele dieser Probleme vermissen, ist aber dafür wiederum auf nur eine Hardwareplattform beschränkt.

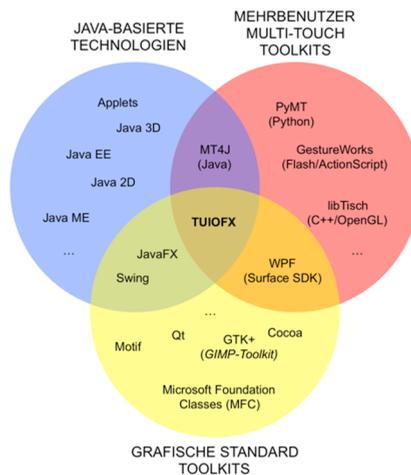


Abbildung 2. TUIOFX kombiniert die Vorteile eines Mehrbenutzer-, Multi-touch-Toolkits, mit denen eines umfangreichen grafischen Standard Toolkits, für die plattformübergreifende Programmiersprache Java.

Das Konzept von TUIOFX ist es, die plattformübergreifende Entwicklung für interaktive Tische zu ermöglichen. Wir kombinieren hierfür die Vorteile (vgl. Abb. 2) einer plattformunabhängigen Programmiersprache (Java) und einer umfangreichen Grafikbibliothek zur Programmierung von grafischen Benutzeroberflächen (JavaFX) mit den Konzepten von Mehrbenutzer-, Multi-touch-Toolkits. TUIOFX erweitert hierfür das Aussehen und Verhalten von JavaFX-Komponenten, ändert aber nichts an deren API. Die Vorteile von TUIOFX

sind daher die folgenden: schneller Einstieg durch die Gemeinsamkeiten mit JavaFX; sehr großer Umfang an grafischen Standard-Bedienelementen (auch Charts, Tabellen, HTML Editor, etc.); Dokumentation zum Großteil durch JavaFX gegeben; Weiterentwicklungen für JavaFX können oft einfach integriert werden; stetig wachsende JavaFX Entwickler-Community bei Problemen und Fragen; Werkzeugunterstützung, wie z.B. durch JavaFX Scene Builder zur visuellen Entwicklung von TUIOFX- Anwendungen (vgl. Abb. 3).

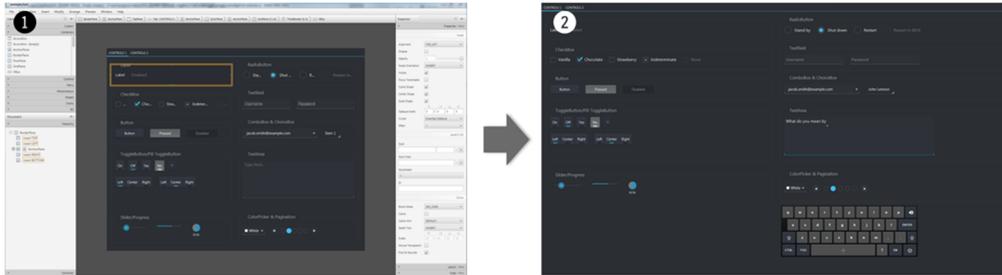


Abbildung 3. Visuelles Editieren von Komponenten in Scene Builder (1) und finale Anwendung auf dem Tisch (2).

3 Aufbau des TUIOFX Toolkits

TUIOFX besteht aus zwei zentralen Bausteinen, den Bibliotheken TUIOFX-Core und TUIOFX-WidgetToolkit. TUIOFX-Core fungiert als Abstraktionsschicht, welche die auf Basis des TUIO-Protokolls (Kaltenbrunner *et al.* 2005) einkommenden Nachrichten in JavaFX konforme Berührungs- und Gestenereignisse übersetzt. Core benutzt hierzu JavaFXs Event-Handling Infrastruktur und die vordefinierten Eventtypen `TouchEvent` und `GestureEvent`, um Benachrichtigungen über erkannte Berührungen und Gesten an den betreffenden Node (d.h. das Bedienelement) zu schicken. Zu diesem Zweck analysiert TUIOFX-Core alle einkommenden TUIO Nachrichten gruppiert nach Node und erkennt so pro Bedienelement mögliche Gesten, die dann als `RotateEvent`, `ScrollEvent`, `SwipeEvent` oder `ZoomEvent` weitergeleitet werden.

Das TUIOFX-WidgetToolkit hingegen ist für die Anpassung der Oberfläche zuständig. Es überschreibt hierfür das vorgegebene Aussehen und Verhalten von allen JavaFX Standard-Komponenten, und erweitert diese so um Mehrbenutzer-Unterstützung (z.B. durch die Erweiterung von Eingabefeldern mit frei positionierbaren Bildschirmstaturen zur simultanen Text-Eingabe) und Multi-touch-Unterstützung (z.B. durch das Anpassen der Komponenten in der Größe für die bessere Eingabe durch Berührung auf großen Bildschirmen). Alle diese Änderungen erfolgen durch extensives Skinning. JavaFX erlaubt es durch die Implementierung eigener `Skin`- und `Behaviour`-Klassen das vorgegebene Look-and-Feel jeder Komponente anzupassen. Dies gestattet komplexeste Anpassungen eines Bedienelementes—wie das Hinzufügen einer Bildschirmstatur zur `TextField`-Komponente—ohne dass sich deren Verwendung im Code für den Entwickler ändert.

Durch dieses sorgfältige Softwaredesign gestaltet sich die Handhabung von TUIOFX entsprechend einfach für Entwickler. Durch das Hinzufügen der beiden Bibliotheken Core und

WidgetToolkit als JAR zu einem existierendem JavaFX-Projekt kann dieses einfach in ein TUIOFX-Projekt migriert werden. Die beiden Zeilen Code `TuioFX tuioFX = new TuioFX(stage);` und `tuioFX.start();` sind ausreichend, um die Berührungs- und Gestenerkennung auf TUIO-basierter Hardware zu ermöglichen. Mittels der Zeile `tuioFX.enableMTWidgets(true)` ändern die Bedienelemente ihr Aussehen und Verhalten, und erhalten neue Funktionalitäten wie Bildschirmstaturen, etc. Darüber hinaus gibt es in der API nur noch die Klasse `Configuration` zur besseren Anpassung verschiedener Parameter an die aktuelle Tischhardware.

4 Zusammenfassung und Ausblick

In diesem Beitrag haben wir Einblicke in das TUIOFX Toolkit gegeben, ein neues Toolkit für die Entwicklung von plattformübergreifenden Anwendungen für interaktive Tische mit JavaFX. Erste informale Workshops mit einer kleinen Anzahl von Entwicklern haben uns erlaubt, schon erste Ergebnisse in Form kleinerer Anwendungen zu sehen. Aktuell bereiten wir das Toolkit für die Herausgabe an weitere Forschungsgruppen vor, deren Feedback für verschiedenste Hardware (u.a. eine große, interaktive Videowand) es uns erlauben soll, das Toolkit vor dessen Veröffentlichung (unter www.tuiofx.org) noch weiter zu optimieren.

Danksagung

Wir danken den Mitgliedern des Cooperative Media Lab.

Literaturverzeichnis

- Echtler, F. & Klinker, G. (2008). A Multitouch Software Architecture. In *Proceedings of NordiCHI 2008* (Okt. 20-22, Lund, Schweden). New York: ACM Press, S. 463-466.
- Fetter, M., Leicht, S., Bimamisa, D. and Gross, T. (2013). Structuring Interaction in Group Decision Making on Tabletops. In *Mensch & Computer 2013* (Sept. 8-11, Bremen, Deutschland). München: Oldenbourg. S. 277-280.
- Hansen, T.E., Hourcade, J.P., Virbel, M., Patali, S. & Serra, T. (2009). PyMT: A post-WIMP Multi-touch User Interface Toolkit. In *Proceedings of ITS 2009* (Nov. 23-25, Banff, Alberta, Kanada). New York: ACM Press, S. 17-24.
- Kaltenbrunner, M., Bovermann, T., Bencina, R. & Costanza, E. (2005). TUIO - A Protocol for Table-Top Tangible User Interfaces. In *Proceedings of GW 2005* (Mai 18-20, Ile deBerder, Frankreich).
- Microsoft Corporation. (2015). *The Microsoft Surface 2.0 SDK*. <http://www.microsoft.com/en-us/pixelsense/SoftwarePlatform.aspx>, 2014. (Letzter Zugriff: 5/6/2015).
- Ruff, C. & Laufs, U. (2014). *Entwicklung von Multi-Touch-Anwendungen*. In Schlegel, T., (Hrsg.) *Multit-Touch—Interaktion durch Berührung*. Heidelberg: Springer, S. 153-163.

Kontaktinformationen

Mirko Fetter, mirko.fetter@uni-bamberg.de, T. 0951/863-3943