

Design, Implementierung und Evaluation einer Frameworks für die Integration von Eye-Tracking-Signalen in Entwicklungsumgebungen.

Thema:

Design, Implementierung und Evaluation einer Frameworks für die Integration von Eye-Tracking-Signalen in Entwicklungsumgebungen.

Art:

MA

BetreuerIn:

Alexander Bazo

BearbeiterIn:

Jan Rankenhohn

ErstgutachterIn:

Christian Wolff

ZweitgutachterIn:

N.N.

Status:

in Bearbeitung

Stichworte:

Software Engineering, Eye-Tracking, Entwicklungsumgebungen

angelegt:

2016-07-11

Beginn:

2020-03-16

Textlizenz:

Unbekannt

Codelizenz:

Unbekannt

Hintergrund

Die Messung von Blickdaten ist eine verbreitete und akzeptierte Methodik zur Evaluation von und Interaktion mit Computersystemen. Die zunehmende Miniaturisierung der Geräte und die stetig sinkenden Anschaffungskosten lassen eine wachsende Integration dieser Technologie in klassische Anwendungsumgebungen realistisch erscheinen. Ein potentiell Szenario stellt dabei die Softwareentwicklung dar. Obwohl Eye-Tracking-Experimente bereits interessante Ergebnisse im Zusammenhang mit Programmierern und deren Tätigkeiten feststellen, fehlen konkrete Vorschläge für die Eye-Tracking-basierte Tool-Unterstützung des Entwicklungsprozess. Ein möglicher Grund dafür ist die heterogene Systemlandschaft die neben unterschiedlichen Eye-Tracker-Herstellern natürlich auch die Vielzahl unterschiedlicher Entwicklungsumgebungen und individueller *Toolchains* umfasst. Diese Arbeit hat das Ziel, das Potenzial der Eye-Tracking-Integration durch die Schaffung eines

nutzbaren *Plugin Frameworks* für *IDEs* zu fördern. Auf der Basis dieses Frameworks können unterschiedliche integrierte Tools entstehen, die den Programmierer innerhalb seiner gewohnten Arbeitsumgebung unterstützen.

Konkrete Problemstellung

Moderne Entwicklungsumgebungen bieten Plugin-Schnittstellen an, mit deren Hilfe die grundsätzliche Funktionalität der Anwendungen beliebig erweitert werden kann. Dadurch lässt sich die Arbeitsweise von Programmierern an individuelle oder andere Bedürfnisse anpassen. Eye-Tracking-Daten können vielfältig in bestehende Entwicklungsumgebungen integriert werden und als Grundlage von Plugins genutzt werden. Neben der intuitiven Gestaltung der tatsächlichen Interaktion des Programmierers mit der Software können z.B. der Reviewprozess unterstützt werden oder das individuelle Verhalten von Entwicklern ausgewertet werden. Grundlage für eine einfache und zielführende Implementierung entsprechender Plugins ist dabei die Integration der eigentlichen Eye-Tracking-Daten in die Entwicklungsumgebungen bzw. deren Plugin-Schnittstelle. Problematisch ist hierbei die heterogene Gestaltung der Hardware-Anbindung Eye-Tracker verschiedener Anbieter sowie die nicht einheitliche Repräsentation der Daten durch die unterschiedlicher Hersteller. Die Arbeit entsteht als Vorbereitung eines größeren Forschungsprojekts. Wird dessen Förderung bewilligt, ist bei entsprechender Eignung unter Umständen eine anschließende Anstellung im Rahmen des Projekts möglich.

Zielsetzung der Arbeit

Ziel der Arbeit ist der Entwurf und die Implementierung eines Frameworks, das die Entwicklung beliebiger, Eye-Tracking-basierter Plugins für Entwicklungsumgebungen erlaubt. Hierzu soll ein mehrschichtiges System konzipiert werden, dass ein einheitliches und konvergentes Format für Eye-Tracking-Daten unterschiedlicher Hersteller anbietet und die konkreten Signale aufbereitet anbietet. Dazu ist auch eine umfassende Analyse existierender Standards und Frameworks nötig. Zur Anbindung dieser Daten an die unterschiedlichen Entwicklungsumgebungen wird eine einheitliche Schnittstelle definiert. Eine konkrete Implementierung des konzipierte Systems auf der Basis eines vorhanden Eye-Trackers, einer konkreten Entwicklungsumgebungen und eines einfachen Use-Cases zur Verwendung der Daten soll die tatsächliche Einsatzfähigkeit des Konzept prüfen. Dabei soll neben der funktionalen Anforderungserfüllung auch die Sinnhaftigkeit und Verwendbarkeit des erarbeiteten Konzepts durch entsprechende Methoden validiert werden.

Konkrete Aufgaben

- Übersicht über verbreitete IDEs und die mögliche Nutzung/Entwicklung von Plugins
- Recherche von Eye-Tracking-Standards und Plugin-Schnittstellen
- Recherche/Konzeption möglicher Anwendungsszenarien für den Einsatz von Eye-Tracking in Entwicklungsumgebungen
- Aufstellung allgemeiner Anforderungen für diese Szenarien hinsichtlich der Eye-Tracking-Hardware
- Übersicht vorhandener und nutzbarer Eingangs-Signale [x,y,Radius,Dauer,...] verschiedener Eye-Tracker
- Implementierung eines Systems mit möglichst unabhängigen, erweiterbaren Komponenten
 - Hardware-Anbindung
 - Aufbereitung der Eingangs-Signale

- Allgemeine Schnittstelle zum Bereitstellen der Daten
- Plugin-Framework für eine IDE
- Entwicklung einer konkreten Demo-Anwendung auf der Basis des erstellten Systems

Erwartete Vorkenntnisse

- Sehr gute Kenntnisse des Software Engineerings
- Sehr gute Programmierkenntnisse
- Erfahrung mit C, C++ oder C#
- Gute Kenntnisse im Umgang mit IDEs
- Wünschenswert: Erfahrung in der Entwicklung von IDE-Plugins
- Wünschenswert: Hintergrundwissen zu Eye-Tracking

Weiterführende Quellen

- Rachel Turner, Michael Falcone, Bonita Sharif, and Alina Lazar. 2014. An eye-tracking study assessing the comprehension of c++ and Python source code. In Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '14). ACM, New York, NY, USA, 231-234. DOI=<http://dx.doi.org/10.1145/2578153.2578218>
- Hartmut Glücker, Felix Raab, Florian Echtler, and Christian Wolff. 2014. EyeDE: gaze-enhanced software development environments. In CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14). ACM, New York, NY, USA, 1555-1560. DOI=<http://dx.doi.org/10.1145/2559206.2581217>
- Bonita Sharif and Huzefa Kagdi. 2011. On the use of eye tracking in software traceability. In Proceedings of the 6th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE '11). ACM, New York, NY, USA, 67-70. DOI=<http://dx.doi.org/10.1145/1987856.1987872>
- S. Yusuf, H. Kagdi and J. I. Maletic, „Assessing the Comprehension of UML Class Diagrams via Eye Tracking,“ 15th IEEE International Conference on Program Comprehension (ICPC '07), Banff, Alberta, BC, 2007, pp. 113-122. DOI=<http://dx.doi.org/10.1109/ICPC.2007.10>

From:

<https://wiki.mi.ur.de/> - **MI Wiki**

Permanent link:

<https://wiki.mi.ur.de/arbeiten/eye-tracking-ides-plugin-framework?rev=1584351170>

Last update: **16.03.2020 09:32**

