

Software Ergonomie und Usability Engineering

Arbeitsblatt zur Hausarbeit

Thema 3.1: Testumgebung für Fitts' Law mit Qt Quick (20 Punkte)

Schreiben Sie mit QML ein kleines Programm mit dem Ziel eine eigene Untersuchung zu Fitts' Law durchzuführen. Der Ablauf könnte etwa wie folgt aussehen:

- Der Bildschirm zeigt ein Rechteck und einen Button.
- Die Testperson wird aufgefordert, den Mauszeiger in das Rechteck zu bewegen.
- Dann wird die Testperson aufgefordert, in normaler Arbeitsgeschwindigkeit den Button anzuklicken.
- Verlässt der Mauszeiger das Start-Rechteck, wird ein Timer gestartet. Dieser wird gestoppt, wenn die Testperson den Button anklickt.
- Dann startet das Szenario erneut. Dabei werden verschiedene Kombinationen aus Größe des Buttons und Abstand des Rechteck-Buttons durchgeführt, wobei alle Testpersonen die gleichen Szenarien durchlaufen.
- Protokolliert werden Probandennummer, Buttongröße, Abstand Rechteck-Button und die zugehörige gemessene Zeit, ggf. weitere notwendige Daten.

Bei der weiteren Ausgestaltung des Experiments machen Sie sich Gedanken über Reihenfolgeeffekte und die Frage, ob eine Testrunde erforderlich ist.

Führen Sie den Versuch mit mindestens 10 Testpersonen durch. Bieten Sie sich auch anderen Studierenden als Testperson an. Pro Testperson sind mindestens 10 Szenarien vorzusehen.

Werten Sie die gemessenen Zeiten aus und vergleichen diese mit Fitts' Law.

Abzugeben im Rahmen der Hausarbeit sind:

- Die *Testsoftware* als gezipptes QML-Projekt
 - Benennung: **Matrikelnummer_Vorname_Name_Fitts_Law.zip**
- Die *Messwerte als Rohdaten*, wie sie von Ihrem Programm ausgegeben werden

- Benennung: **Matrikelnummer_Vorname_Name_Results.csv**
- Eine genaue *Beschreibung und Begründung des Testablaufs inkl. Fotos von der Durchführung*
- Eine *Auswertung der Messwerte* (z.B. als Diagramm)
- *Vergleich mit Fitts' Law* (Entsprechen die gemessenen Werte den theoretischen? Wo gibt es Abweichungen?)
- Die vorherigen drei Punkte bitte gebündelt in einer PDF-Datei abgeben
 - Benennung: **Matrikelnummer_Vorname_Name_Fitts_Law.pdf**

Arbeitsschritte

Gemeinsame Code-Basis

Im Praktikum wird eine Code-Basis zur Verfügung gestellt und erläutert

- Machen Sie sich mit den neuen Mechanismen und den bereitgestellten Bausteinen vertraut.
- Das bereitgestellte Programm ist eine Kombination von C++ und QML/JavaScript. Der Grund ist, dass aus QML/JavaScript keine Dateioperationen möglich sind. Für die Lösung Ihrer Aufgabe ist jedoch nur eine Bearbeitung der QML/JavaScript-Teile notwendig, also keine Programmierung in C++.
- Die QML bzw. JavaScript-Dateien befinden sich unter den Ressourcendateien. Hier sind von Ihnen Änderungen bzw. Erweiterungen vorzunehmen.

Erste Schritte

Anpassungen in FittsLaw.js

- In der Datei FittsLaw.js sind bereits einige Funktionen vorbereitet.
- Es wird reichlich Gebrauch vom Math-Object gemacht (siehe z.B. http://www.w3schools.com/jsref/jsref_obj_math.asp)
- Im Praktikum wird gemeinsam die Methode `getDistance` erarbeitet.
- Vervollständigen Sie als nächstes die Methode „`getDifficultyIndex`“ (weitere Hinweise im Quelltext)
- Die Methode `calculateResults` soll alle zu protokollierenden Daten eines Versuchsdurchlaufs in einen String speichern. Einzelne Werte sind dabei durch Komma zu trennen. Von anderen Programmteilen wird dieser String später in eine Datei geschrieben. Diese Datei enthält dann alle Messerergebnisse im CSV Format (siehe z.B. http://de.wikipedia.org/wiki/CSV_%28Dateiformat%29). Dadurch kann die Datei später in die gängigen Tabellenkalkulationsprogramme eingelesen und weiterverarbeitet werden. Die Datei wird bei Ausführung des Programms im Ausführungsverzeichnis unter dem Namen `Results.csv` abgelegt. Je nach Wunsch oder Bedarf können Sie auch die Parameter dieser Funktion anpassen, weitere Funktionen aufrufen, etc.
 - Bitte schreiben Sie folgende **Werte in dieser Reihenfolge** in den String
 - Probandennummer
 - Versuchsbedingung
 - Entfernung (Start-Ziel)
 - Winkelabhängige Zielbreite (Real Width)
 - Zeit zur Zielerreichung

- Difficulty Index
- Optional: weitere Daten

Anpassungen in main.qml

- Hier werden die GUI-Elemente des Testprogramms sowie das Verhalten definiert.
- Zunächst wird ein FileIO-Objekt angelegt. Dies stellt die Schnittstelle zum C++-Teil dar. Hier brauchen Sie nichts zu machen.
- Nun sollten Sie das Text-Objekt erweitern, in dem die Anweisungen für die ProbandInnen erscheinen sollen.
- Als nächstes sollten Sie das Start-Rechteck weiter definieren. Beispielsweise können Sie die Anweisung ändern, wenn die Maus über das Rechteck bewegt wird. Beim Verlassen des Rechtecks können Sie die Stoppuhr, die in `FittsLaw.js` programmiert ist, starten. Dazu können Sie den Befehl `FittsLaw.StopWatch.start();` verwenden
- In der Button-Definition sollten Sie den Timer wieder stoppen. Die `stop()`-Methode liefert die verstrichene Zeit in Millisekunden zurück.
- Ebenfalls sind beim Click alle relevanten Daten zu speichern und ein neues Test-Szenario zu starten.
- Die verschiedenen Test-Szenarien können Sie in dem ListModel ergänzen.

Zur Bewertung:

- SW-Prototyp: 9 Punkte nach folgender Skala:
 - „Geht nicht“: 0
 - Programmcode fehlt
 - „Eingeschränkt“: 3
 - Programmcode kompiliert nicht
 - „1-Weg-Programm“: 5
 - Programmcode kompiliert, und das Programm tut, was es soll
 - allerdings nur wenn man es so bedient wie vorgesehen
 - „Mehrwegprogramm“: 7
 - Programmcode kompiliert, und das Programm tut, was es soll
 - auch wenn man es nicht so bedient wie vorgesehen
 - „Schönes Mehrwegprogramm“: 9
 - Programmcode kompiliert, und das Programm tut, was es soll
 - auch wenn man es nicht so bedient wie vorgesehen, Programmcode ist flexibel
- Dokumentation Prototyp: 2 Punkte
- Versuchsplanung: 1 Punkt
- Versuchsdurchführung (durch Fotos nachzuweisen): 4 Punkte
- Versuch auswerten (nur wenn Durchführung nachgewiesen): 4 Punkte