

# Dokumentation der Morse-Aufgabe

Björn Guth

19. April 2016

## 1 Abhängigkeiten

- *Python 2.7*<sup>1</sup>: Die gesamten automatisierten Teile der Verarbeitung sind in Python geschrieben.
- *pygame*<sup>2</sup>: Da dies bei diesen Programmen auch graphischer Output und Audio gebraucht werden, wird pygame genutzt um all dies zu managen.
- *PyQt4*<sup>3</sup>: Dies ist nötig um Dialogfenster darzustellen.

## 2 Installation

Bevor mit der Installation des eigentlichen Programms begonnen werden kann, müssen zuerst die in Abschnitt 1 beschriebenen Abhängigkeiten installiert werden. Da es sich bei pygame um Paket für Python selbst handelt, muss Python auch schon vor der Installation von pygame installiert sein.

Sind die Abhängigkeiten installiert, kann der Quellcode auch schon von einem Git-Repository<sup>4</sup> heruntergeladen und an einem beliebigen Ort entpackt werden. Anschließend ist das Programm startbereit und sollte getestet werden.

### Definition 1 (*Top-Level*)

*Im weiteren Verlaufe wird der Ordner, in dem sich das eigentliche Programm, also die Pythondateien morse1.py usw., befindet, als Top-Level bezeichnet.*

Später kann Top-Level nach belieben verschoben werden, da ja keine Installation im eigentlichen Sinne stattgefunden hat. Jedoch sollte darauf geachtet werden, dass die Ordnerstruktur in Top-Level nicht verändert wird.

## 3 Allgemeine Informationen

### 3.1 Start der Programme

Um eines der Programme zu starten, muss nur die entsprechende Datei auf dem Top-Level ausgeführt werden. Dies kann entweder von einer Kommandozeile oder durch ein graphisches Programm wie beispielsweise den Arbeitsplatz unter Windows geschehen. Am generellen Ablauf des Programms ändert sich hierbei nichts, allerdings haben graphische Programme gegen über einer Kommandozeile den klaren Nachteil, dass gegeben Falls auftretende Fehlermeldungen zwar „angezeigt“ werden, aber nicht lesbar sind, da sie sofort nach anzeigen der Fehlermeldung verschwinden.

Anschließend erscheint ein Dialog, indem ein Probandencode abgefragt wird. Dieser muss nicht ausgefüllt werden, um mit dem Programm fortzufahren, da im Falle eines leeren Eingabefeldes der Probandencode „test“ angenommen wird. Anschließend wird überprüft, ob der genannte Probandencode

---

<sup>1</sup><http://python.org/>

<sup>2</sup><http://pygame.org/download.shtml>

<sup>3</sup><http://www.riverbankcomputing.co.uk/software/pyqt/download>

<sup>4</sup><https://github.com/scattenlaeufer/monster>

schon in den Log-Files existiert und, sollte dies der Fall sein, gefragt, ob die Daten überschrieben werden sollen. Bei beiden Dialogen führt eine negative Antwort zum Beenden des Programms. Ansonsten startet der eigentliche Teil des Programms mit einem Sartscreen. Von diesem aus kann mit einem Druck auf <ENTER> mit den Aufgaben begonnen werden.

### 3.2 Hilfreiche Tastenbefehle

Es gibt ein paar Tastenbefehle in den Programmen, die dem Nutzer die Arbeit ein wenig erleichtern können. Die wichtigsten seien hier aufgezählt:

- <F12>: Einige der Instruktionen lassen sich durch einen Druck auf <F12> beenden. Das Programm läuft danach normal weiter als wäre die Instruktion beendet.
- <ESC>: Hiermit lässt sich das Programm zu jeder Zeit beenden. An manchen Stellen, besonders bei bestimmten Instruktionen, kann das Programm nicht sofort beendet werden, sollte jedoch während dessen <ESC> gedrückt werden, wird zum nächsten möglichen Zeitpunkt beendet.

## 4 Morse 1

Morse 1 ist wiederum ein komplett anderes Szenario. Zur Durchführung ist neben einem separaten Programm zur Aufnahme von Audio (hier sei Audacity<sup>5</sup> empfohlen) auch ein Mikrofon und ein Setup, welches sowohl Audio ausgeben als auch aufnehmen kann, von Nöten.

### 4.1 Durchführung

Da es während der Durchführung des Programms nicht möglich ist, zu einem anderen Programm zu wechseln, muss die Audioaufnahme vor Beginn gestartet werden. Anschließend ist das Programm wie in Unterabschnitt 3.1 beschrieben zu starten. Anschließend gibt es eine kleine Einführung in den Test und danach sollen durch den Probanden die auf dem Bildschirm präsentierten Zeichen vorgelesen werden.

Zwischen den einzelnen Trials wird mit <SPACE> und <ENTER> die Aussage des Kindes bestätigt, wobei ersteres eine falsche und letzteres eine richtige Aussage markieren.

### 4.2 Auswertung

Die Auswertung von Morse 1 birgt noch einige Probleme. Da es keine Möglichkeit gibt, Audio direkt im Programm Morse 1 aufzuzeichnen, sondern ein externes Programm genutzt werden muss, müssen die Zeiten der Audioaufnahme und die von Morse 1 ermittelten im Nachhinein synchronisiert werden. Dies geschieht über die Vorstellung des Zeichen *ta*. Aber die bisherigen Durchführungen haben gezeigt, dass dies problematisch ist, da die Zeit in unterschiedlichen Programmen unterschiedlich schnell gezählt werden.

Im eigentlich geplanten Verlauf der Auswertung müssen die Zeit des Punktes der Synchronisation und der Beginn der finalen Aussagen des Probanden ausgelesen werden und anschließend werden letztere in eine Datei in im Ordner `helpers/trial_log/<PROBANDENCODE>/morse1` abgespeichert. Anschließend wird das Programm `auswertung.py` ausgeführt, welches die Synchronisationszeit abfragt und anschließend eine Datei `test_resptime` in genanntem Ordner anlegt, in welcher dann folgende Daten abgespeichert werden:

- *trial*: die laufende Nummer des Trials
- *image*: das präsentierte Bild
- *correct*: der kodierte Response
- *time\_m*: ???

---

<sup>5</sup><http://audacity.sourceforge.net/>

- *time\_r*: ???
- *time\_response*: die Zeit zwischen Präsentation des Stimulus und der finalen Äußerung

Generell ist das Problem der unzuverlässigen Synchronisation aber ein Problem, welches vor einer erneuten Verwendung nochmals überdacht und nach Möglichkeit gelöst werden sollte.