Schweizer Zeitmesser App

Martin Reiche, Version vom 3.5.2024

Inhaltsverzeichnis

1 Einführung....2
2 Zeitgeber (Timer)....2
2.1 Äußere Erscheinung und Funktion....2
2.2 Formale Spezifikation....4
2.3 Implementierung....5
2.4 Hilfe....6
2.5 Test....6
3 Zeiterfassung....6
3.1 Äußere Erscheinung und Funktion....6
4 Wecker (alarm clock)....7

1 Einführung

Der Titel ist ein Wortspiel: Ein Schweizer Taschenmesser hat viele einzelne Werkzeuge, also Klingen, Schraubendreher usw. Entsprechend hat diese App mehrere Teile, die sich mit Zeitmessung beschäftigen: Einen Wecker, eine Stoppuhr, einen Zeitgeber, eine Zeiterfassung und eine Weltzeit-Anzeige. Diese Bestandteile werden durch Auswahl des Reiters (Tab) aktiviert:



Wie man sieht, ist aktuell nur der Zeitgeber implementiert. Die übrigen Bestandteile sollen von den Kursteilnehmern entwickelt werden. Dies hat die Vorteile, dass einerseits die Umgebung schon präpariert ist und andererseits eine funktionierende Vorlage existiert, aus der man Teile wiederverwenden oder einfach abschauen kann. Man kann also beliebig kreativ sein und dabei möglichst originelle bzw. benutzerfreundliche Elemente hinzufügen. Natürlich darf man sich auch die Aufgabe stellen, eine eigene Implementierung des Zeitgebers zu entwickeln.

2 Zeitgeber (Timer)

2.1 Äußere Erscheinung und Funktion

Die Benutzeroberfläche zeigt folgende Elemente:

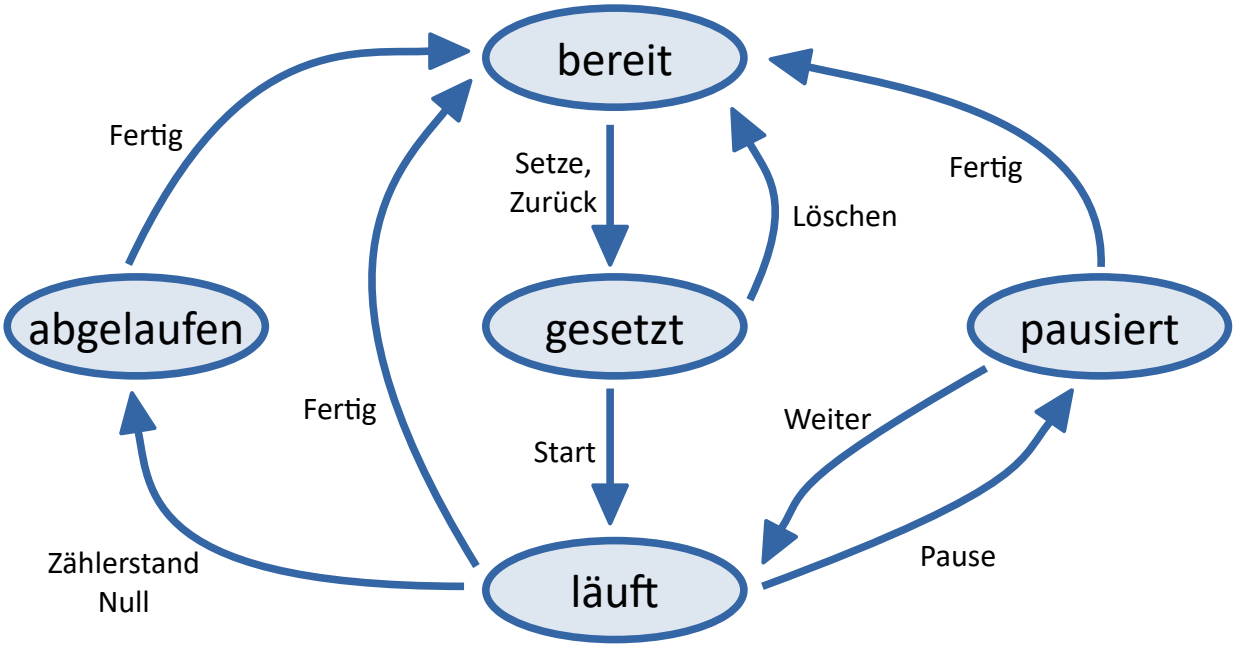
- In der Mitte wird die noch verbleibende Zeit angezeigt, im Bild 29 Minuten und 11 Sekunden.
- Zu Beginn kann man durch Klicken auf die kleinen Auf- und Ab-Dreiecke die jeweiligen Ziffern erhöhen oder erniedrigen, um die Laufzeit zu programmieren.
- Läuft der Zeitgeber, so verkleinert sich der blaue Kreisumfang gegen den Uhrzeigersinn, bis

er am Ende ganz verschwunden ist. Dann ertönt ein akustisches Signal.

- Die vier Knöpfe darunter haben Funktionen, die weiter unten beschrieben werden. Wichtig: Diese Funktion – und damit die Beschriftung der Knöpfe – kann sich ändern.
- Unten links ist ein Feld, in das der Benutzer einen Titel d.h. Zweck des Zeitgebers eingeben kann: Im Bild ist es die verbleibende Garzeit der Schweinshaxe im Backofen. Dieses Feature macht Sinn, wenn man mehrere Zeitgeber parallel laufen lässt, indem man die App mehrfach startet. Dann weiß man immer genau, welcher Zeitgeber welchen Zweck erfüllt.
- Unten rechts wird die Uhrzeit angezeigt, zu der der Zeitgeber abgelaufen sein wird. Im Bild also ungefähr um viertel vor Eins.

2.2 Formale Spezifikation

In der Software-Entwicklung hat es sich bewährt, vorab genau festzulegen, wie sich ein Programm verhalten soll. Ein wichtiges Hilfsmittel kann ein Zustandsdiagramm wie das folgende sein:



Das Diagramm liest sich so:

- Der Zeitgeber befindet sich immer in genau einem der Zustände, die durch die Ovale gekennzeichnet sind. Nach Programmstart ist er dazu ***bereit,*** durch Benutzeraktivität eine Ablaufzeit anzunehmen.
- Durch Wahl der Ablaufzeit mithilfe der kleinen Dreiecke (im Diagramm „Setze“) ändert sich der Zustand nach ***gesetzt***. Falls vorher schon einmal eine Ablaufzeit festgelegt wurde, kann man auch auf diese *zurück*greifen. Alternativ kann man die Ablaufzeit wieder *löschen*.
- Nach betätigen des *Start*-Knopfes ***läuft*** der Zeitgeber. Der innere Zählerstand wird im Sekundentakt erniedrigt.
- Sobald der Zeitgeber läuft, kann er angehalten d.h. ***pausiert*** werden und später *weiter*laufen.
- Erreicht der Zählerstand den Wert Null ist der Zeitgeber ***abgelaufen***. Ein akustisches Signal ertönt, bis der Benutzer mit dem *Fertig*-Knopf quittiert. Dies bringt den Zeitgeber zurück in den Zustand ***bereit***.

Je nach Zustand zeigen die Knöpfe verschiedene Aufschriften (Labels). Außerdem sind nicht alle Knöpfe immer aktiv. Z.B. macht es keinen Sinn zu pausieren, wenn noch gar keine Ablaufzeit eingestellt wurde. Nicht-aktive Knöpfe werden „ausgegraut“, wie man sagt. Im Bild betrifft dies die Knöpfe „Pause“ und „Zurück“. Zur Festlegung dieser Verhaltensweisen kann eine Tabelle wie die folgende dienen. Die Knöpfe sind hier von links nach rechts nummeriert. Für deaktivierte Knöpfe ist die Aufschrift in Klammern gesetzt. Die Dreiecke zum Einstellen der Startzeit haben ihre eigene

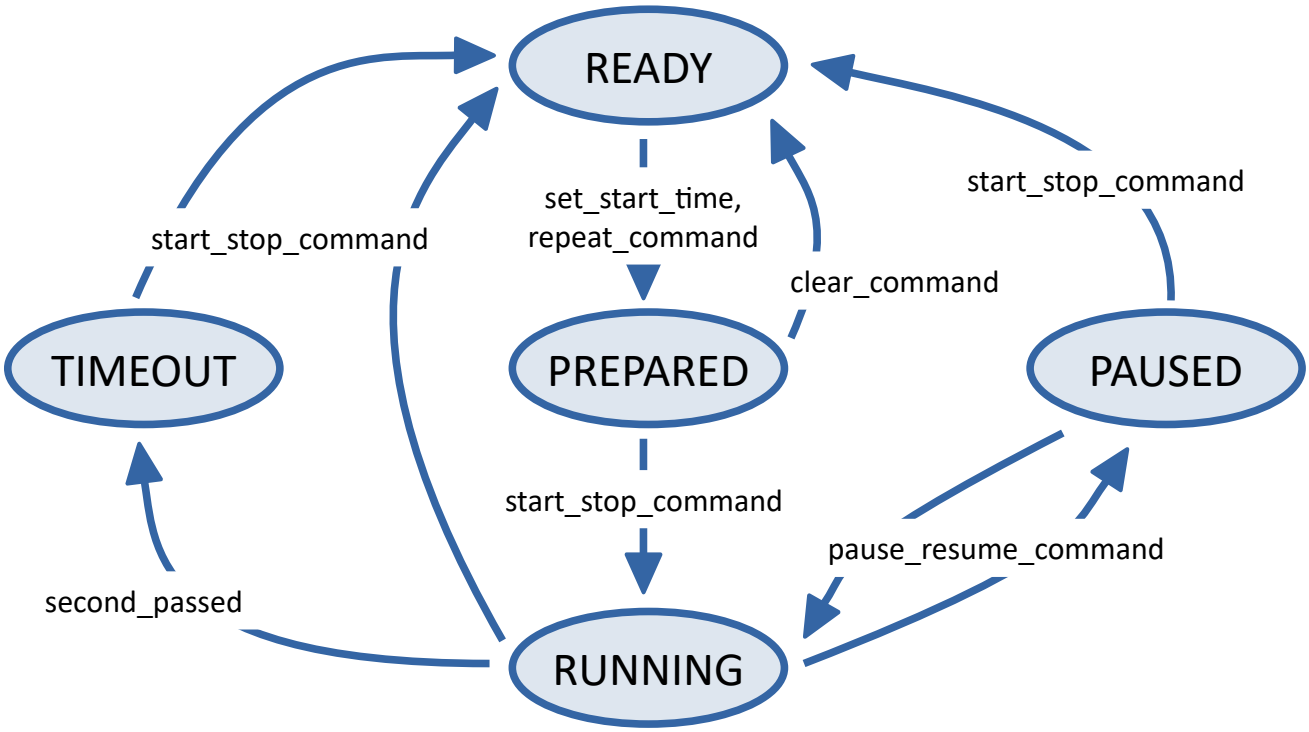
Spalte. Falls die Betätigung einer Funktion einen Zustandsübergang auslöst, ist dies durch einen Pfeil (→) markiert.

Funktion → ↓ Zustand	Dreieck	Knopf 1	Knopf 2	Knopf 3	Knopf 4
bereit	aktiv → gesetzt	(Start)	(Pause)	Zurück → gesetzt	(Löschen)
gesetzt	aktiv	Start → läuft	(Pause)	Zurück	Löschen → bereit
läuft	nicht aktiv	Fertig → bereit	Pause → pausiert	(Zurück)	(Löschen)
pausiert	nicht aktiv	Fertig → bereit	Weiter → läuft	(Zurück)	(Löschen)
abgelaufen	nicht aktiv	Fertig → bereit	(Pause)	(Zurück)	(Löschen)

2.3 Implementierung

In vielen, ja vermutlich den meisten Arbeitsumgebungen wird die Verwendung der englischen Sprache verlangt. Dies hat zusätzlich den Vorteil, dass Beschreibungen kürzer ausfallen.

Das folgende *state diagram* zeigt die Zustände auf Englisch und benennt die Python-Funktionen, welche an die *tkinter-widgets* d.h. die Elemente der grafischen Benutzeroberfläche gebunden sind:



Somit hat man eine Orientierung für die Implementierung besagter Funktionen. Falls es mehrere Pfade gibt, die zu einem Zustand führen, sollte man die zugehörigen Aufgaben in separate

Funktionen packen, um Wiederholungen von Befehlen zu vermeiden. Die Implementierung des Autors folgt diesem Vorschlag zugunsten der Einheitlichkeit für alle Zustände: *enter_ready()*, *enter_prepared()* usw.

Deutsch:	bereit	gesetzt	läuft	abgelaufen	pausiert
Englisch:	READY	PREPARED	RUNNING	TIMEOUT	PAUSED

Weiterhin ergeben sich die Korrespondenzen:

Deutsch:	Start	Zurück	Löschen	Fertig	Pause	Weiter
Englisch:	start	repeat	clear	stop	pause	resume

2.4 Hilfe

Das Paket Pmw (Python Megawidgets) bietet Hilfe für Knöpfe (buttons) an. Befindet sich der Mauszeiger darüber, erscheint eine Erläuterung der Funktion. Weil sich diese bei manchen Knöpfen beim Gebrauch der App ändert, muss auch der jeweilige Hilfetext angepasst werden. Zusätzlich kann beim Alarm Knopf 1 die Hilfe „Stoppt den Alarm“ zugewiesen bekommen.

2.5 Test

Bei Apps solcher Komplexität bietet es sich an, einen Testplan zu schreiben. Darin sollten Beobachtungen und Aktionen enthalten sein, welche einerseits der „normalen Erwartung“ entsprechen und sich andererseits aus der Spezifikation ableiten lassen.

Das Testen von Software ist eine eigene Disziplin und hat viele Aspekte. Siehe z.B.:

- https://de.wikipedia.org/wiki/Softwaretest
- https://en.wikipedia.org/wiki/Software_testing

3 Zeiterfassung

3.1 Äußere Erscheinung und Funktion

Dieses Programm dokumentiert die Zeitspannen, die man selbst (oder ein beobachtetes Objekt) für bestimmte Aktivitäten gebraucht hat. Dazu kann man eine Liste von z.B. 5 Feldern für deren Bezeichnungen anbieten. Die Aufzeichnung für eine Aktivität beginnt, sobald man den nebenstehenden Radio-Button anklickt und endet, sobald eine andere Aktion gewählt wird. Das Programm startet mit der Leerlaufaktion. Beim Beenden des Programms werden alle erfassten Zeitspannen in einer Datei gespeichert. Das bedeutet auch, dass diese Datei zum Programmstart gelesen werden muss. Eine Zeile der Datei enthält 3 Felder: Aktivität, Start, Ende. Als Format für die Zeit kann man das numerische Format von time.time() verwenden, dann entfällt aufwändiges Parsen.

4 Wecker (alarm clock)

… hier kann es weiter gehen ...