

2. Praktikum

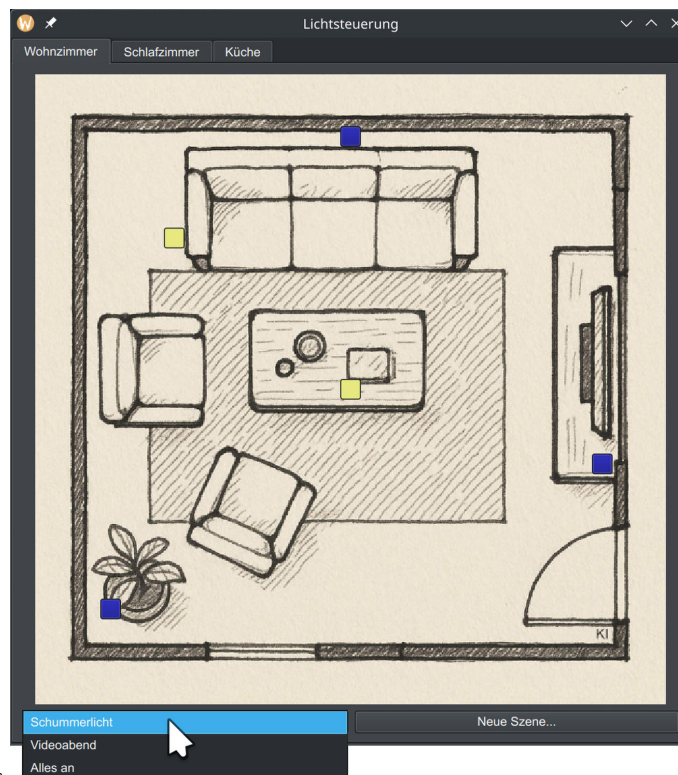
Jörn Loviscach

Versionsstand: 24. Mai 2025, 23:58



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Ein PySide-Programm soll Lichtszenen mehrerer Räume verwalten:



Screenshot unter Linux im Dark Mode

Einige Dateien sind dazu vorbereitet: Die Datei `configuration.json` enthält Informationen zu den Räumen und ihren Leuchten. (Die Positionsangaben darin sind Bruchteile von Breite und Höhe, gemessen von der linken oberen Ecke aus.) Außerdem gibt es Bilddateien mit den Grundrissen der Räume. Das Programm soll diese Dateien benutzen und die grafische Oberfläche daraus erzeugen. Die Räume und Leuchten sollen also *nicht* fest einprogrammiert sein, sondern konfigurierbar sein!

Gehen Sie in diesen Schritten vor:

1. Legen Sie eine Klasse `LightConfig` an, deren Instanzen Name und Koordinaten einer einzelnen Leuchte speichern.

2. Legen Sie eine Klasse `RoomConfig` an, deren Instanzen für einen Raum dessen Namen, den Namen der Grundriss-Datei und die Liste seiner `LightConfig`-Objekte speichern.
3. Laden Sie zu Programmbeginn die Datei `configuration.json` und werten Sie die aus:

```
with open('configuration.json', 'r') as file:  
    rooms_config_raw = json.load(file)
```

Sehen Sie sich im Debugger an, was dann in `rooms_config_raw` steht. Darin kommen nicht nur Listen [...], sondern auch Dictionaries {...} vor (nächste Woche mehr dazu in der Veranstaltung, aber lesen Sie schon [hier](#)). Werten Sie diese Konfigurationsdaten aus, um `RoomConfig`-Objekte zu erzeugen, die ihrerseits `LightConfig`-Objekte erzeugen. Dazu sind in beiden Klassen entsprechende Methoden nötig.

4. Legen Sie eine Klasse `MainWindow` an, die von `QMainWindow` erbt. Diese erhält als zentrales und einziges Widget ein `QTabWidget`, also einen Rahmen mit Karteireitern darin. Für jeden Raum erzeugen Sie eine Instanz der – erst im nächsten Schritt anzulegenden – Klasse `RoomTab` und fügen diese dem `QTabWidget` mit `addTab` hinzu.
5. Legen Sie eine Klasse `RoomTab` an, die von `QWidget` erbt. Die Instanzen dieser Klasse werden die Karteireiter der einzelnen Räume. Die Initialisierungsmethode dieser Klasse erzeugt ein `QVBoxLayout` mit einem `QLabel` oben (das im nächsten Schritt für das Bild verwendet werden wird) und einem `QHBoxLayout` unten, in dem eine `QComboBox` (das ist die aufklappbare Auswahlliste) und ein `QPushButton` stecken. Außerdem legt die Initialisierungsmethode `QPushButton`-Objekte für die Leuchten des jeweiligen Raums an und verwandelt sie mit `setCheckable(True)` von Tastern zu An/Aus-Schaltern.
6. Vorläufige, noch nicht saubere Darstellung: Erzeugen Sie in der Initialisierungsmethode des `RoomTab` eine `QPixmap` mit dem jeweiligen Grundriss und weisen Sie diese dem `QLabel` mit dessen Methode `setPixmap` zu. Erzeugen Sie für jede Leuchte des Raums einen `QPushButton` und fügen Sie ihn dem Layout mit `addWidget` hinzu.
7. Legen Sie eine Klasse `LightScene` an, deren Instanzen die An/Aus-Einstellungen der Leuchten eines Raums sowie einen Namen für die jeweilige Szene speichern. Sorgen Sie dafür, dass beim Klick auf den Knopf zum Anlegen einer neuen Szene Folgendes passiert: Das Programm bittet mit `QInputDialog.getText` um die Eingabe eines Namens; es legt eine Instanz von `LightScene` mit diesem Namen und den (booleschen) Zuständen der Buttons der Leuchten an; es erzeugt mit `addItem` einen neuen Eintrag in der Combobox mit dem Namen der Szene; es hängt mit `setItemData` an diesen Eintrag die eben erzeugte Instanz von `LightScene` an.

8. Verbinden Sie das Signal `activated` der Combobox mit einer neuen Methode, in der die an den aktuellen Eintrag angehängte `LightScene` in die Buttons der Leuchten übernommen wird.
9. Sorgen Sie dafür, dass der Grundriss immer maximal groß (aber mit seinem ursprünglichen Seitenverhältnis) im Fenster der Anwendung angezeigt wird und dass die Buttons der Leuchten immer an den richtigen Stellen platziert sind. Sie müssen dazu die Methoden `resizeEvent` und `showEvent` des `RoomTab` überschreiben.

Mögliche Erweiterungen:

- Ergänzen Sie einen Button zum Löschen der aktuell angewählten Szene.
- Ergänzen Sie einen Button zum Speichern einer geänderten Szene unter dem alten Namen.
- Sorgen Sie dafür, dass die Buttons gelb und blau statt hellgrau oder dunkelgrau werden.
- Benutzen Sie für die Leuchten Kreisscheiben statt der herkömmlichen Buttons.
- Speichern und laden Sie die Szenen. (Die üblichen Techniken dafür folgen später im Semester.)
- Invertieren Sie die Farben der Grundrissbilder, wenn der Computer im Dark Mode läuft.

Diese Anleitung ist bewusst nicht vollständig – als Übung dafür, die automatische Code-Vervollständigung zu nutzen und die richtigen Fragen an Suchmaschinen oder KI zu stellen und deren Antworten zu verstehen und zu bewerten.