Python Kurs, Martin Reiche 2024
Version vom 27.10.2024

Start-Infos für das Projekt „15-Puzzle“

Zur Funktion siehe die Beschreibung in http://tuepy.wikidot.com/exercises.

Tipps:

- Für die Bildbearbeitung musst du zunächst in Thonny das Paket *„pillow“ (*Python Imaging Library (Fork)) installieren.
- Studiere genau den Code in *puzzle-starter.py*! Starte das Programm und klicke auf das Fenster. Welcher Code wird ausgeführt?
- Überlege dir, welche Informationen zu wo speichern willst! Tipps:
  - Halte die Bilder (Rückgabewert von *ImageTk.PhotoImage*) in einer später unveränderlichen Matrix. (Das ist technisch notwendig für *tkinter,* siehe unten!)
  - Speichere die aktuelle Belegung der Felder als Matrix mit den *tkinter*-Handles als Matrix-Elemente. Diese Belegung ändert sich mit jedem Spielzug.
- Mache alle logischen und geometrischen Festlegungen abhängig von einer einzigen Variablen N, die im Programm als Konstante behandelt, also nicht verändert wird. Somit kann man leicht von einem 8-Puzzle auf ein 15-Puzzle, 24-Puzzle usw. umstellen.
- Weil nicht alle Anordnungen der Teilbilder erfolgreich zum Originalbild verschoben werden können (Siehe https://de.wikipedia.org/wiki/15-Puzzle), führe zu Beginn des Spiels zufällige Verschiebungen vom Original aus durch. Dann ist sichergestellt, dass das Problem lösbar ist.
- Der move-Befehl ist hier beschrieben: https://dafarry.github.io/tkinterbook/canvas.htm. Suche nach „move(“!
  Zum Befehl *create_image* steht dort auch: „The application must keep a reference to the image object.“ Das heißt, dass der Rückgabewert von *ImageTk.PhotoImage(image)* dauerhaft in einer Variable (z.B. einem Listen-Element) gespeichert werden muss. Ansonsten verschwindet das Bild, wenn man das *image* später verschiebt.
- Wie schon früher praktiziert, speichere jede funktionierende Zwischenversion deines Programms unter einem anderen Namen, also z.B. puzzle-1.py, puzzle-2.py, puzzle-3.py, usw.

Erweiterungen

Ob alle Elemente am richtigen Platz liegen, zeigt sich an der Belegungsmatrix, siehe oben. Dagegen kann nach jedem Zug geprüft werden und bei Erfolg eine Message-Box erscheinen.

Das Originalfoto, die Konstante N, sowie die Anzahl der Ausgangs-Verschiebungen ließe sich mit *tkinter* Widgets einstellbar machen. Ein GUI könnte also folgende Elemente enthalten:

1. Load Button – zum Laden des Bildes (mit Dateiauswahl Dialog, wie im Starter-Programm schon enthalten, aber auskommentiert, siehe *askopenfilenames*)
2. Spinbox – zum Einstellen von N (Werte von 3 bis maximal 6 wären praktikabel)
3. Spinbox – zum Einstellen der anfänglichen Verschiebung: Von schwach bis sehr stark.

4. Start Button – Zum Erzeugen der (zufallsgestuerten) Anfangsposition
5. Reset Button – Zur Anzeige des Originalbildes
6. Label – Zur Anzeige der Anzahl bislang erfolgter Züge des Spielers

Die Datei *puzzle-num-starter.py* enthält Code, um ein Puzzle mit numerierten Elementen statt Bildausschnitten zu erzeugen:

