

## Dokumentation Python Projekt:

### Minigame „godoku.py“

Godoku ist eine Variante des bekannten japanischstämmigen Zahlenrätsels Sudoku. Bei Godoku werden Buchstaben anstelle von Zahlen den richtigen Kästchen zugeordnet. Die Vorlage für das Godoku stammt von dieser Seite:

[http://www.math.cornell.edu/~mec/Summer2009/meerkamp/Site/Solving\\_any\\_Sudoku\\_II.html](http://www.math.cornell.edu/~mec/Summer2009/meerkamp/Site/Solving_any_Sudoku_II.html)

Die Vorlage ist ein Sudoku und wurde von Zahlen in Buchstaben „übersetzt“. Eine png-Datei des Sudokus ist beigefügt (sudoku\_vorlage.png). Zu bemerken ist dabei, dass das Sudoku einen Fehler enthält: Die Zahl „3“ in der siebten Zeile, fünfte Spalte muss eigentlich eine „2“ sein. Der Fehler wurde bei der „Übersetzung“ korrigiert und taucht im Spiel nicht auf.

Zu Beginn des Minispiels sollten bereits alle relevanten Module (pygame, sys, pygame.locals import \*) importiert sein. Der erste Teil des Programms besteht aus zwei verschachtelten Listen („nested lists“). Mithilfe der Verschachtelung kann später leicht über Indices auf die Listenelemente zugegriffen werden. Eine der Listen repräsentiert das komplett gefüllte Godoku im Zielzustand (SOLVEDGODOKU), wobei jede Zeile des Godokus wiederum als Liste innerhalb der kompletten Liste (SOLVEDGODOKU) existiert. Dasselbe gilt für die zweite Liste (GODOKU). Diese repräsentiert das Godoku mit einigen Lücken, so wie es dem Spieler dargestellt wird. SOLVEDGODOKU wird im Spielverlauf niemals verändert, während die Liste GODOKU mit dem entsprechenden Input fluktuiert. Leider war es nicht möglich SOLVEDGODOKU als Tupel zu definieren, da Listen und Tupeln nicht verglichen werden können. Das muss jedoch später im Spiel möglich sein, um festzustellen, ob der Spieler alle Felder (richtig) gefüllt hat.

Nach den Listen werden einige Konstanten festgelegt, bevor die Hauptfunktion für das Spiel definiert wird. Das Herzstück dabei ist eine while-Schleife, durch die das Fenster für das Spiel generiert und aktualisiert wird. Der Spielablauf ist relativ simpel: Der Spieler klickt auf das Feld, das er füllen möchte. Danach drückt er die entsprechende Taste auf der Tastatur. Stimmt die Eingabe, so wird die GODOKU-Liste und das angezeigte Spielfenster aktualisiert. Dies

wiederholt sich solange, bis alle Felder gefüllt sind (d.h. die GODOKU-Liste ist komplett gefüllt und stimmt mit der SOLVEDGODOKU-Liste überein). Daraufhin wird über eine entsprechende if-clause der Endbildschirm angezeigt: das gefüllte Godoku und eine Reihe farbiger Felder die das Lösungswort markieren. Schließt der Spieler nun das Minispiel, wird das quit-event von pygame ausgelöst und der Spieler kommt zurück in die Westhöhle.