

## Protokoll Soundmaschine

Das Ziel unseres Projektes "Soundmaschine" war es, ein Programm zu schreiben, das uns, per Zufall oder unter bestimmten Regeln der Harmonie, eine kleine Melodie ausgibt und diese auch abspielt. Lösen wollten wir das mit Hilfe eines Lindemeyer-Systems.

Wir haben uns also zu erst eingelesen in die Theorie und Praxis eines sogenannten L-Systems und haben selbst angefangen ein bisschen damit auszuprobieren. Gleichzeitig haben wir auch mit dem "random" Paket ausprobiert, vor allem mit den "choice" Funktionen daraus, die wir mit einbauen wollten. Diese Befehle wählen aus einer Liste zufällig Daten aus. Interessant ist auch der Befehl von "choice" bei dem eine Wahrscheinlichkeit für jedes Element der Liste festgelegt werden kann (weights) und nach dieser Wahrscheinlichkeit dann ein Element ausgegeben wird. Für unser L-System haben wir aber einfach mit den if und elif und else Schleifen gearbeitet, da es mit den "choice" Befehlen innerhalb des L-Systems nicht funktionierte. Wahrscheinlichkeiten für einen Ton haben wir dann mit einer if-Funktion definiert: Zuerst sucht sich unser Programm mit "random.randint(1, 100)/100" eine Integer Zahl zwischen Null und Eins aus, jeder mögliche Ton hat dann einen bestimmten Bereich zwischen zwei Integer Zahlen, je größer, desto größer die Wahrscheinlichkeit, dass sich die zufällig gewählte Integer darin befindet. Gewählt wurde also der Ton in dessen Bereich die Integer liegt. Eine höhere Wahrscheinlichkeit wollten wir den Tönen geben, die direkt neben dem Ton davor liegen, damit unsere Melodie nicht zu große Sprünge hat. Den selben Ton haben wir ausgelassen, so konnte direkt hintereinander nicht der selbe Ton gespielt werden. Das Anfangsaxiom, in unserem Fall also der Anfangston, haben wir dann aber über den "choice" Befehl aussuchen lassen. So beginnt das Programm jeden Durchgang mit einem anderen Ton. Wir konnten uns also eine Tonfolge, bis jetzt nur in Form von aneinander gereihten Buchstaben, erfolgreich ausgeben lassen.

Um die Töne dann auch zu hören, wollten wir sie abspielen lassen. Wir haben Pygame für Python installiert. Wir probierten es erstmal mit "pygame.mixer", dabei sollten unsere Tonfolge als von uns festgelegten Töne interpretiert und abgespielt werden. Doch dabei stürzte jedes Mal beim Durchlauf unser Kernel ab. Vermutlich wegen zu geringer Leistung. Bei Spyder, was wir dachten hätte mehr Leistung, war leider dasselbe Problem. Wir blieben trotzdem erstmal bei pygame und erstellten kurze Tondateien, luden die bei Github hoch und definierten eine if Schleife, welche diese Dateien abspielen sollte. Aber das funktionierte nicht. Bei unsere Recherchen zur Problemlösung, stießen wir auf Pyknon, eine Musik library für Python, die wir dann uns bei Python installierten. In ihr sind bereits Töne gespeichert, allerdings als Zahlen, das heißt z.B. 1 für C. Also haben wir unser Programm abgeändert, sodass uns eine Zahlen-statt Buchstabenfolge ausgegeben wird. Da wir diese als array erhalten und bei Pyknon Midi (das Paket zum Töne abspielen) mit Listen arbeitet, mussten wir das noch umformen lassen, bevor unsere Tonfolge von Pyknon Midi interpretiert werden konnte. Ohne Probleme spielte unser Programm "Soundmaschine" jetzt eine kleine Melodie ab.

Moric Noatsch und Veza Schlüpen