Programmierung

- Scratch
- Turtle-Grafiken mit Python
- Einführung in Python
- Programmierung einfacher Spiele

Scratch

Adresse zu Scratch

Eine Anmeldung ist nicht erforderlich, um mit Scratch zu programmieren. Die eigenen Programme können heruntergeladen und auch wieder hochgeladen werden. Damit ist eine Anmeldung auch nicht empfohlen.

Online Scratch Editor

Kurzanleitung

Das Prinzip von Scratch ist mit dem von Lego zu vergleichen. Verschiedene Bausteine der Programmierung können zusammengesteckt werden. Ob verschiedene Elemente zusammen passen wird durch die Form deutlich gemacht. Die Grundelemente von Programmiersprachen gibt es auch in Scratch. So gibt es Schleifen, Bedingungen, Variablen und auch Zufallszahlen. Besonders praktisch für die Einführung in die Robotik ist, dass es auch virtuelle Sensoren gibt. So kann z.b. der Bildschirmrand oder Hindernisse erkannt werden. Damit können wir viele Funktionen der Roboter hier virtuell darstellen.

Einführungskurs auf appcamps

Turtle-Grafiken mit Python

Um mit Python Turtle-Grafiken zu zeichnen, muss man nur die Bibliothek "turtle" importieren: `from turtle import *` Beispiel für eine Grafik:

```
from turtle import *
for i in range(0,4):
    forward(100)
    right(90)
```

Dies zeichnet ein Quadrat auf dem Bildschirm. Etwas interessanter ist die Spirale:

```
from turtle import *
for steps in range(100):
    for c in ('blue', 'red', 'green'):
        color(c)
        forward(steps)
        right(30)
```

Die offizielle Dokumentation der Turtle

Einführung in Python

Die Einführung findet ihr hier.

Programmierung einfacher Spiele

Wörterraten

Kopiere den Quellcode in Thonny und führe ihn aus. Bearbeite folgende Arbeitsaufträge:

1. Finde heraus, was du tun musst, um das Spiel zu spielen.
2. Schau dir den Quellcode an und vollziehe nach, wie das Programm abläuft.
3. Verändere das Programm nach deinen Vorstellungen.

```
# Quelle: https://www.geeksforgeeks.org/python-program-for-word-guessing-game/?ref=lbp
import random
# library that we use in order to choose
# random words from a list of words

name = input("What is your name? ")

# Here the user is asked to enter the name first

print("Good Luck ! ", name)

words = ['rainbow', 'computer', 'science', 'programming',
        'python', 'mathematics', 'player', 'condition',
        'reverse', 'water', 'board', 'geeks']

# Function will choose one random
# word from this list of words
word = random.choice(words)



print("Guess the characters")

guesses = ''

# any number of turns can be used here
```

```
turns = 12


while turns > 0:

    # counts the number of times a user fails
    failed = 0

    # all characters from the input
    # word taking one at a time.
    for char in word:

        # comparing that character with
        # the character in guesses
        if char in guesses:
            print(char, end=" ")

        else:
            print("_")

            # for every failure 1 will be
            # incremented in failure
            failed += 1

    if failed == 0:
        # user will win the game if failure is 0
        # and 'You Win' will be given as output
        print("You Win")

        # this print the correct word
        print("The word is: ", word)
        break

    # if user has input the wrong alphabet then
    # it will ask user to enter another alphabet
    print()
    guess = input("guess a character:")

    # every input character will be stored in guesses
    guesses += guess
```

```
    # check input with the character in word
    if guess not in word:

        turns -= 1

        # if the character doesn’t match the word
        # then “Wrong” will be given as output
        print("Wrong")

        # this will print the number of
        # turns left for the user
        print("You have", + turns, 'more guesses')

        if turns == 0:
            print("You Loose")
```

Erweiterungen

Wortliste in eigener Datei

```
# definiere die Datei, die du verwenden möchtest.
file = "wordlist.txt"
with open(file, "r") as f: # Öffne die Datei im Nur-Lesen-Modus
    wordlist = f.read().splitlines() # Teile die Datei zeilenweise.
print(wordlist)
```

Ausgabe der Wörter besser formatieren

Die Wörter werden in diesem Programm von oben nach unten geschrieben. Das liegt daran, dass der `print`-Befehl mit einem *carriage return* endet. Speichere die Läsungen zunächst in einer Variablen und gib diese am Schluss mit dem `print`-Befehl aus.

Anzahl Runden

Definiere die Anzahl der Fehlversuche in Abhängigigkeit der Wortlänge.

Speichern der Gewinner

```
from datetime import datetime
with open("winner.txt", "a") as text_file:
            text_file.write(f"{name} hat gewonnen. {datetime.now()}\r\n") # oder nur \n, falls
das nicht funktioniert.
```

Mastermind

Kopiere den Quellcode in Thonny und führe ihn aus. Bearbeite folgende Arbeitsaufträge:

1. Finde heraus, was du tun musst, um das Spiel zu spielen.
2. Schau dir den Quellcode an und vollziehe nach, wie das Programm abläuft.
3. Verändere das Programm nach deinen Vorstellungen.

```
# Quelle: https://www.geeksforgeeks.org/mastermind-game-using-python/?ref=lbp
import random



# the .randrange() function generates a
# random number within the specified range.
num = random.randrange(1000, 10000)

n = int(input("Guess the 4 digit number:"))

# condition to test equality of the
# guess made. Program terminates if true.
if (n == num):
    print("Great! You guessed the number in just 1 try! You're a Mastermind!")
else:
    # ctr variable initialized. It will keep count of
    # the number of tries the Player takes to guess the number.
    ctr = 0

    # while loop repeats as long as the
    # Player fails to guess the number correctly.
    while (n != num):
        # variable increments every time the loop
        # is executed, giving an idea of how many
        # guesses were made.
```

```
        ctr += 1

        count = 0

        # explicit type conversion of an integer to
        # a string in order to ease extraction of digits
        n = str(n)

        # explicit type conversion of a string to an integer
        num = str(num)

        # correct[] list stores digits which are correct
        correct = ['X']*4
        print(correct)
        # for loop runs 4 times since the number has 4 digits.
        for i in range(0, 4):

            # checking for equality of digits
            if (n[i] == num[i]):
                # number of digits guessed correctly increments
                count += 1
                # hence, the digit is stored in correct[].
                correct[i] = n[i]
            else:
                continue

        # when not all the digits are guessed correctly.
        if (count < 4) and (count != 0): #- this condition is not needed as we are starting with the condition, n!=num, which is, count<4
            print("Not quite the number. But you did get ",
                    count, " digit(s) correct!")
            # second code is not supposed to print the guessed numbers, from the sample output, here I get we are not recording the position of the guess,but count. But as per the explanation, the code should not print the guessed numbers, rather give their count.
                # print("Also these numbers in your input were correct.")
                # for k in correct:
                # print(k, end=' ')
            print('\n')
            print('\n')
```

```
            n = int(input("Enter your next choice of numbers: "))

        # when none of the digits are guessed correctly.
        elif (count == 0):
            print("None of the numbers in your input match.")
            n = int(input("Enter your next choice of numbers: "))

    # condition for equality.
    if n == num:
    # ctr must be incremented when the n==num gets executed as we have the other incrmentation
in the n!=num condition
        ctr+=1
        print("You've become a Mastermind!")
        print("It took you only", ctr, "tries.")
```

Siegerehrung

Die Siegerehrung kann in jedes Spiel eingebaut werden. Der Import muss natürlich an den Anfang der Datei.

```
import time
width=70
speed=0.015
for x in range(0,3):
    for i in range(width,0,-1):
        out = ""
        for j in range(width,i,-1):
            out += " "
        out += "You win"
        print(out)
        time.sleep(speed)

    for i in range(0,width):
        out = ""
        for j in range(i,width):
            out += " "
```

```
        out += "You win"
        print(out)
        time.sleep(speed)
```