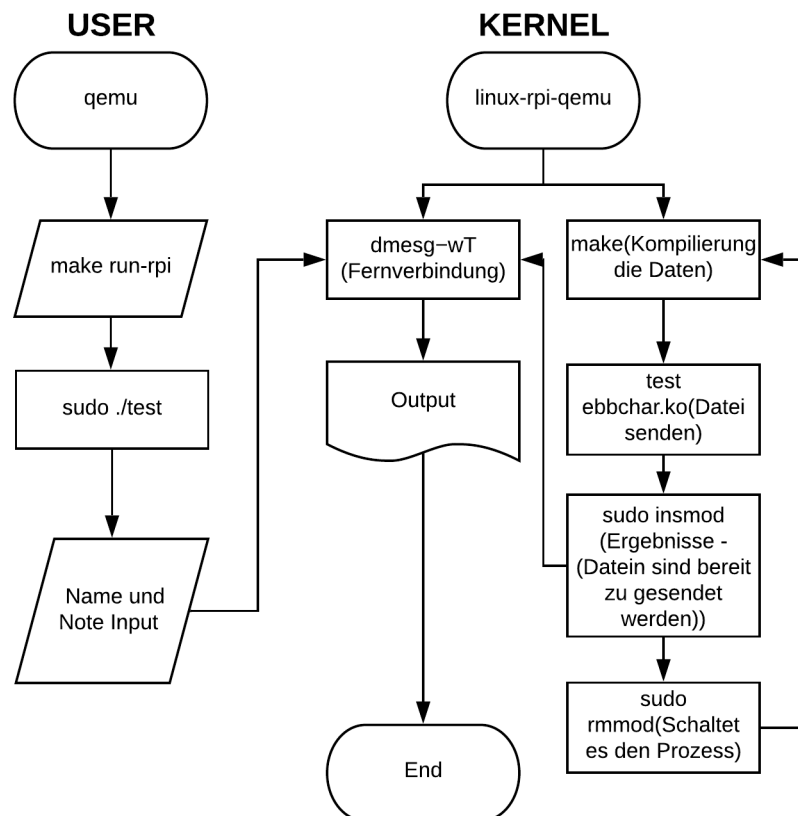


Ich habe nicht verstanden ob wir Code oder User-Kernel FlowChart gefragt werden, weil im Beispiel gibt es Code aber wir User-Kernel Relation gefragt haben, so habe ich beiden gemacht.

1)



Wenn wir mit Kernel arbeiten, müssen wir zuerst qemu-dev-rpi öffnen, so machen wir `cd rpi-qemu-dev`, danach `make run`.

`cd rpi-qemu-dev`: Es öffnet die rpi-qemu-dev Feile
`make run`: Es fängt unsere qemu Umgebung an.

Unsere Program ist in der linux-rpi-qemu-dev Feile, so machen wir fast gleiche Sache hier auch.

`cd linux-rpi-qemu-dev`: Es öffne die linux-rpi-qemu-dev Feile.
 Wenn wir in dieser Feile müssen wir auch 002-Kernel-Module-Character öffnen.

`cd 002-Kernel-Module-Character` öffnen

Wir sind in diser Feile. Wenn rpi-qemu-dev gestratet haben, teilen wir den Bildschirm ins zwei.

Rechte Teile: `sshpass -p raspberry ssh -p 5022 -o StrictHostKeyChecking=no -o UserKnownHosts`
`File=/dev / null pi@127.0.0.1 "dmesg -wT"`

Wir haben verbinden mit rpi-qemu.

Linke Seite:

make: Es kompiliert notwendige Sache.

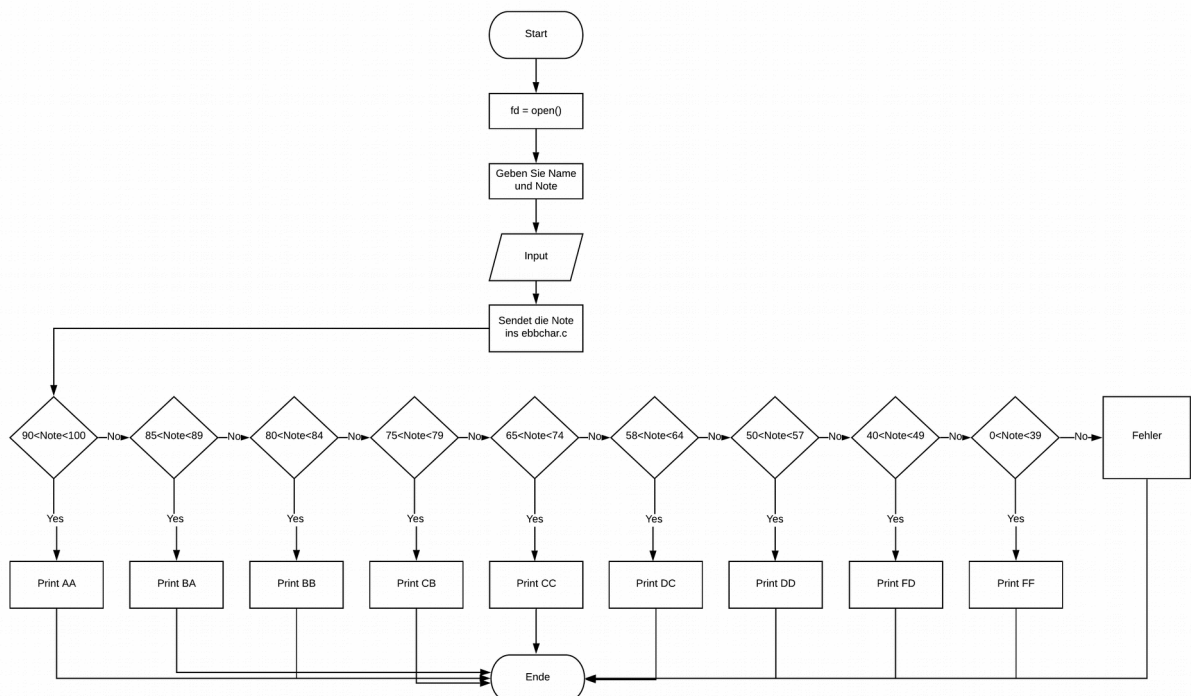
```
sshpass -p raspberry scp -P 5022 -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null test ebbchar.ko pi@127.0.0.1:/home/pi/: Senden wir Daten
```

```
sshpass -p raspberry ssh -p 5022 -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null pi@127.0.0.1 "sudo insmod /home/pi/ ebbchar.ko" : Das Programm hat angefangen.
```

Wenn wir in der rpi-qemu Umgebung `sudo ./test` schreiben, können wir das Programm verwenden.

Um zu beenden schreiben wir:

```
sshpass -praspberry ssh -p 5022 -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null pi@127.0.0.1 "sudo rmmod /home/pi/ebbchar.ko 2>/dev/null"
```



Wenn wir das Programm anfangen, macht es zuerst `fd = open()` und es öffnet eine neue Datei um unsere Daten zu geschrieben werden. Dann möchte es unsere Eingaben. Eigentlich wir schreiben unsere Eingaben in `testebbchar` aber es sendet diese Eingaben in `ebbchar`. In `ebbchar` rechnet es die Noten aus und prints das genügende Ergebnis.

```
dev@tau: ~/rpi-qemu-dev 112x48
The received message is: [56]
The received message is: [43]
The received message is: [34]
The received message is: [aysu aksu]
The received message is: [65]
The received message is: [98]
The received message is: [54]
The received message is: [umut yesildal]
The received message is: [45]
The received message is: [23]
The received message is: [12]
End of the program
pi@raspberrypi:~$ sudo ./test
Starting device test code ...
Student 1/3:
ugur sarp
Writing message to the device [ugur sarp].
INF208: 45
MEC308: 23
MEC324: 65
Student 2/3:
aysu aksu
Writing message to the device [aysu aksu].
INF208: 77
MEC308: 80
MEC324: 64
Student 3/3:
umut yesildal
Writing message to the device [umut yesildal].
INF208: 88
MEC308: 45
MEC324: 67
Haben sie alle Werte gegeben?:

The received message is: [ugur sarp]
The received message is: [45]
The received message is: [23]
The received message is: [65]
The received message is: [aysu aksu]
The received message is: [77]
The received message is: [80]
The received message is: [64]
The received message is: [umut yesildal]
The received message is: [88]
The received message is: [45]
The received message is: [67]
End of the program
pi@raspberrypi:~$
```

In rpi-qemu haben wir sudo ./test schreiben und haben wir das Programm angefangen. Schreiben wir unsere Values.

```

dev@tau: ~/linux-rpi-qemu-dev/002-Kernel-Module-Character 104x47
30 01:38:02 2020] CPU: 0 PID: 461 Comm: rmmod Tainted: G      0      4.19.50+ #1
30 01:38:02 2020] Hardware name: ARM-Versatile (Device Tree Support)
30 01:38:02 2020] [<c001d230>] (unwind_backtrace) from [<c00190ac>] (show_stack+0x10/0x
30 01:38:02 2020] [<c00190ac>] (show_stack) from [<c0026150>] (__warn+0xc4/0xf0)
30 01:38:02 2020] [<c0026150>] (__warn) from [<c0025dac>] (warn_slowpath_fmt+0x44/0x70)
30 01:38:02 2020] [<c0025dac>] (warn_slowpath_fmt) from [<c026ef08>] (refcount_sub_and_
/0xbc)
30 01:38:02 2020] [<c026ef08>] (refcount_sub_and_test_checked) from [<c053cc74>] (kobje
30 01:38:02 2020] [<c053cc74>] (kobject_put) from [<bf00026c>] (ebbchar_exit+0x28/0xdc
30 01:38:02 2020] [<bf00026c>] (ebbchar_exit [ebbchar])) from [<c007b544>] (sys_delete_m
)
30 01:38:02 2020] [<c007b544>] (sys_delete_module) from [<c0009000>] (ret_fast_syscall+
30 01:38:02 2020] Exception stack(0xccd3dfa8 to 0xccd3dff0)
30 01:38:02 2020] dfa0:                00041eb8 bea11c00 00041ef4 00000800 0ec4b000
30 01:38:02 2020] dfc0: 00041eb8 bea11c00 00000002 00000081 bea11ee0 00041eb8 00000000
30 01:38:02 2020] dfe0: 0003ff88 bea11bb4 00022bb8 b6ccffcc
30 01:38:02 2020] ---[ end trace e3feb778a9c9e5a2 ]---
30 01:38:02 2020] EBBChar: Goodbye from the LKM!
30 01:38:55 2020] EBBChar: Initializing the EBBChar LKM
30 01:38:55 2020] EBBChar: registered correctly with major number 250
30 01:38:55 2020] EBBChar: device class registered correctly
30 01:38:55 2020] EBBChar: device class created correctly
30 01:38:59 2020] EBBChar: Device has been opened 1 time(s)
30 01:39:05 2020] EBBChar: Received 1096
30 01:39:09 2020] FD
30 01:39:09 2020] EBBChar: Received 45
30 01:39:15 2020] FF
30 01:39:15 2020] EBBChar: Received 23
30 01:39:23 2020] CC
30 01:39:23 2020] EBBChar: Received 65
30 01:39:37 2020] CB
30 01:39:37 2020] EBBChar: Received 77
30 01:39:40 2020] BB
30 01:39:40 2020] EBBChar: Received 80
30 01:39:52 2020] DC
30 01:39:52 2020] EBBChar: Received 64
30 01:39:57 2020] BA
30 01:39:57 2020] EBBChar: Received 88
30 01:40:00 2020] FD
30 01:40:00 2020] EBBChar: Received 45
30 01:40:06 2020] CC
30 01:40:06 2020] EBBChar: Received 67
30 01:40:06 2020] EBBChar: Devices successfully closed

```

In der Kernel Umgebung haben wir empfangen.

```

int main(){
    int ret, fd;
    char son[BUFFER_LENGTH];
    char name1[BUFFER1];
    char name2[BUFFER2];
    char name3[BUFFER3];
    char inf208_1[BUFFER_LENGTH];
    char inf208_2[BUFFER_LENGTH];
    char inf208_3[BUFFER_LENGTH];
    char mec308_1[BUFFER_LENGTH];
    char mec308_2[BUFFER_LENGTH];
    char mec308_3[BUFFER_LENGTH];
    char mec324_1[BUFFER_LENGTH];
    char mec324_2[BUFFER_LENGTH];
    char mec324_3[BUFFER_LENGTH];

    printf("Starting device test code ...\n");
    fd = open("/dev/ebbchar", O_RDWR);           // Open the device with read/write access
    if (fd < 0){
        perror("Failed to open the device...");
        return errno;
    }

    printf("Student 1/3: \n");
    scanf("%[^\n]*c", name1);                    // Read in a string (with spaces)
    printf("Writing message to the device [%s].\n", name1);

    printf("MEC324: ");
    scanf("%[^\n]*c", mec324_1);
    ret = write(fd, mec324_1, strlen(mec324_1));
    if (ret < 0){
        perror("Sıkıntı burada da olabilir abi ama");
        return errno;
    }

    printf("Student 2/3: \n");
    scanf("%[^\n]*c", name2);                    // Read in a string (with spaces)
    printf("Writing message to the device [%s].\n", name2);

    printf("INF208: ");
    scanf("%[^\n]*c", inf208_2);
    ret = write(fd, inf208_2, strlen(inf208_2));
    if (ret < 0){
        perror("Sıkıntı burada da olabilir abi ama");
        return errno;
    }

    printf("MEC308: ");
    scanf("%[^\n]*c", mec308_2);
    ret = write(fd, mec308_2, strlen(mec308_2));
    if (ret < 0){
        perror("Sıkıntı burada da olabilir abi ama");
        return errno;
    }

    printf("MEC324: ");
    scanf("%[^\n]*c", mec324_2);
    ret = write(fd, mec324_2, strlen(mec324_2));
    if (ret < 0){
        perror("Sıkıntı burada da olabilir abi ama");
        return errno;
    }
}

```

Senden wir die Noten als char.

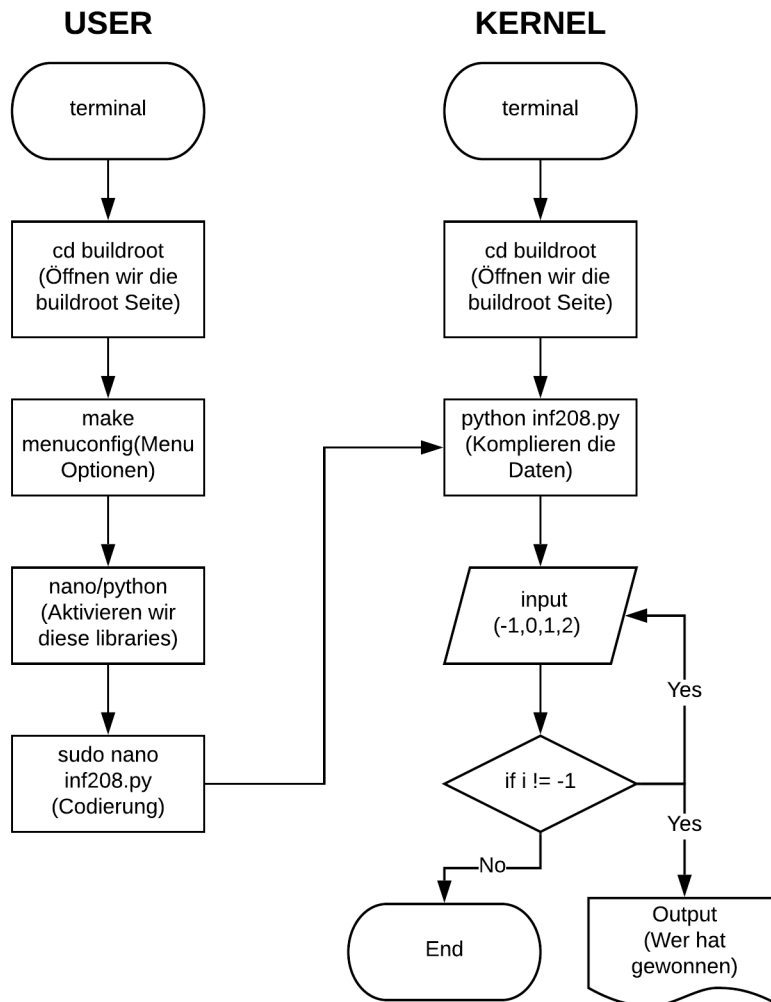
```

e > dev > linux-rpi-qemu-dev > 002-Kernel-Module-Character > C ebbchar.c
static ssize_t dev_write(struct file *filep, const char *buffer, size_t len, loff_t *offset){
    int error_count = 0;
    char* kbuffer;
    int gecici;
    sscanf(message, "%d", &gecici);
    if(gecici>=90 && gecici<=100){
        printk(KERN_INFO "AA");
    }
    if(gecici>=85 && gecici<=89){
        printk(KERN_INFO "BA");
    }
    if(gecici>=80 && gecici<=84){
        printk(KERN_INFO "BB");
    }
    if(gecici>=75 && gecici<=79){
        printk(KERN_INFO "CB");
    }
    if(gecici>=65 && gecici<=74){
        printk(KERN_INFO "CC");
    }
    if(gecici>=58 && gecici<=64){
        printk(KERN_INFO "DC");
    }
    if(gecici>=50 && gecici<=57){
        printk(KERN_INFO "DD");
    }
    if(gecici>=40 && gecici<=49){
        printk(KERN_INFO "FD");
    }
    if(gecici<=39){
        printk(KERN_INFO "FF");
    }
    kbuffer=kmalloc(len, GFP_KERNEL);
    error_count = copy_from_user(kbuffer,buffer,len);
    sprintf(message, "%s(%zu letters)", kbuffer, len); // appending received string with its length
    size of message = strlen(message); // store the length of the stored message
    printk(KERN_INFO "EBBChar: Received %.2d\n", gecici);
    kfree(kbuffer);
    return len;
}

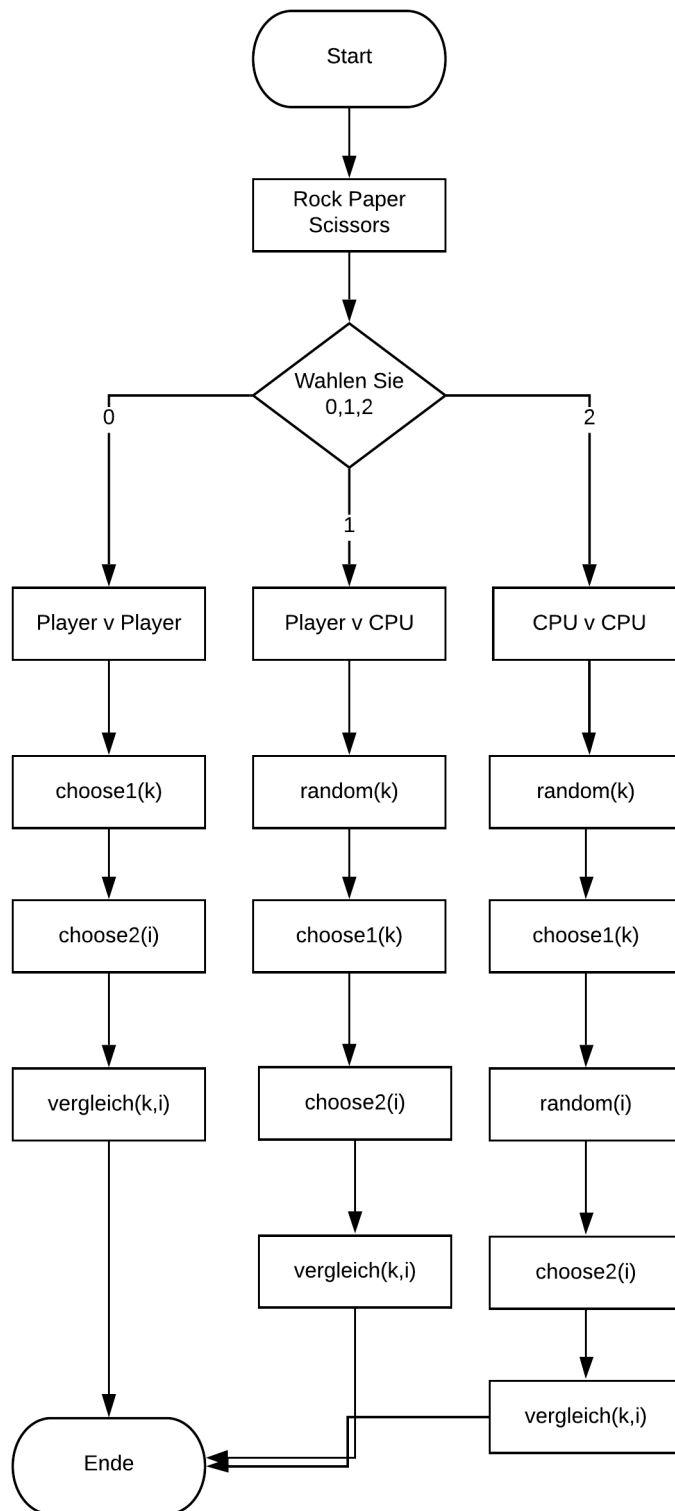
```

In ebbchar machen wir unsere char int, deshalb können wir if statements verwenden und print Briefnotizen.

2)



Schreiben wir `cd buildroot`, wenn wir diese Feile öffnen möchten. Wir brauchen `nano` und `python` libraries um dies Programm zu verwenden, so schreiben wir `make menuconfig` und wählen wir diese zwei und speichern. Ins `sudo nano` schreiben wir unsere Programm. Wenn wir es verwenden möchten schreiben wir `python inf208.py` in terminal. Das Programm funktioniert bis -1 geschrieben wird.



Es gibt 3 Möglichkeiten. Bei der ersten Option spielen zwei echte Spieler das Spiel. In der zweiten Option spielen Sie es gegen die CPU und in der letzten Option spielen es zwei CPU-Spieler. Wenn die erste Option ausgewählt ist, spielen zwei Spieler das Spiel, bis einer der Spieler -1 drückt. Wenn die zweite Option ausgewählt ist, wählen Sie Stein, Papier oder Schere aus, und die CPU wählt zufällig eine davon aus. Sie werden es spielen, bis Sie -1 drücken. Wenn die dritte Option ausgewählt ist, bestimmen Sie, wie viele Spiele gespielt werden. Jeder CPU-Spieler wählt zufällig eine Option aus. Am Ende des Spiels werden aufgelistet, wie viele Spiele jeder Spieler gewonnen hat.


```
dev@tau:~/buildroot$ sudo nano inf208.py
dev@tau:~/buildroot$
```

Schreiben wir es um unsere Code zu schreiben.

```
dev@tau: ~/buildroot 88x43
dev@tau:~/buildroot$ python inf208.py
0 - Spielen Sie das Spiel mit einem anderen Spieler
1 - Spielen Sie das Spiel gegen CPU
2 - Lassen Sie die beiden CPU-Spieler das Spiel spielen
-1 - Beende das Spiel
3-Rock
4-Papier
5-Schere
█
```

Wenn wir unsere Code anfangen möchten, schreiben wir python inf208.py

```
Player1 hat Schere gewählt
Player2 hat Rock gewählt
Schere gegen Rock, Rock gewinnt
Player1 hat Papier gewählt
Player2 hat Schere gewählt
Papier gegen Schere, Schere gewinnt
Player1 hat Papier gewählt
Player2 hat Papier gewählt
Unentschieden Spielen
Player1 hat Schere gewählt
Player2 hat Papier gewählt
Schere gegen Papier, Schere gewinnt
Player1 hat Papier gewählt
Player2 hat Papier gewählt
Unentschieden Spielen
Player1 hat Rock gewählt
Player2 hat Schere gewählt
Rock gegen Schere, Rock gewinnt
Player1 hat Schere gewählt
Player2 hat Papier gewählt
Schere gegen Papier, Schere gewinnt
('Spieler1: ', 3)
('Spieler2: ', 2)
('Total: ', 7)
dev@tau:~/buildroot$
```

Beispiel

```

import random
print("0 - Spielen Sie das Spiel mit einem anderen Spieler")
print("1 - Spielen Sie das Spiel gegen CPU")
print("2 - Lassen Sie die beiden CPU-Spieler das Spiel spielen")
print("-1 - Beende das Spiel")
print("3-Rock")
print("4-Papier")
print("5-Schere")
i = 10
p1 = 0
p2 = 0
g = 0
def choose1(k):
    if(k == 3):
        print("Player1 hat Rock gewählt")
    if(k == 4):
        print("Player1 hat Papier gewählt")
    if(k == 5):
        print("Player1 hat Schere gewählt")

def choose2(i):
    if(i == 3):
        print("Player2 hat Rock gewählt")
    if(i == 4):
        print("Player2 hat Papier gewählt")
    if(i == 5):
        print("Player2 hat Schere gewählt")

def vergleich(k,i):
    global p1
    global p2
    global g
    if(k == 3 and i == 3):
        print("Unentschieden Spielen")
        g +=1
    if(k == 4 and i == 4):
        print("Unentschieden Spielen")
        g += 1
    if(k == 5 and i == 5):

```

[Read 95 lines]

In der choose1 und choose2 Funktionen bestimmen wir, dass was das Spieler gewählt hat.
 In der vergleich Funktion vergleichen wir die gewählte Optionen von beiden Spielern.

```

def vergleich(k,i):
    global p1
    global p2
    global g
    if(k == 3 and i == 3):
        print("Unentschieden Spielen")
        g +=1
    if(k == 4 and i == 4):
        print("Unentschieden Spielen")
        g += 1
    if(k == 5 and i == 5):
        print("Unentschieden Spielen")
        g += 1
    if(k == 3 and i == 4):
        print("Rock gegen Papier, Papier gewinnt")
        p2 +=1
    if(k == 3 and i == 5):
        print("Rock gegen Schere, Rock gewinnt")
        p1 += 1
    if(k == 4 and i == 3):
        print("Papier gegen Rock, Papier gewinnt")
        p1 += 1
    if(k == 4 and i == 5):
        print("Papier gegen Schere, Schere gewinnt")
        p2 += 1
    if(k == 5 and i == 3):
        print("Schere gegen Rock, Rock gewinnt")
        p2 += 1
    if(k == 5 and i == 4):
        print("Schere gegen Papier, Schere gewinnt")
        p1 += 1

```

```

while i != -1:
    i = int(input())
    if(i == 0):
        while i != -1:
            i = int(input())
            k = i
            if(k == -1):
                break
            choose1(k)
            i = int(input())
            if(i == -1):
                break
            choose2(i)
            vergleich(k,i)
    if(i == 1):
        while i != -1:
            k = random.randint(3,5)
            choose1(k)
            i = int(input())
            choose2(i)
            if(i == -1):
                break
            vergleich(k,i)
    if(i == 2):
        for k in range(7):
            x = random.randint(3,5)
            choose1(x)
            y = random.randint(3,5)
            choose2(y)
            vergleich(x,y)
        break

print("Spieler1: ",p1)
print("Spieler2: ",p2)
print("Total: ", p1+p2+g)

```

Wenn beide Player gleiche wählen $g += 1$

Wenn Player 1 gewonnen hat $p1 += 1$

Wenn Player 2 gewonnen hat $p2 += 1$

Total = $p1+p2+g$

Wenn $i = 0$ Beide Player wählen eine Nummer.

Wenn $i = 1$ Player wählen eine Nummer, CPU random

Wenn $i = 2$ Beide CPU random

Wenn $i = -1$ ist wird das Programm beendet.