



FSST-PROJEKT

SOCKET UND THREAD

Höhere Technische Bundeslehr- und Versuchsanstalt Anichstraße

Abteilung

ELEKTRONIK UND TECHNISCHE INFORMATIK

Schüler:

Metin Doganer 4aHel

Lehrperson:

Mag. Jäger Alexander

Abgabe:

Innsbruck, am 25.06.2021



Inhaltsverzeichnis

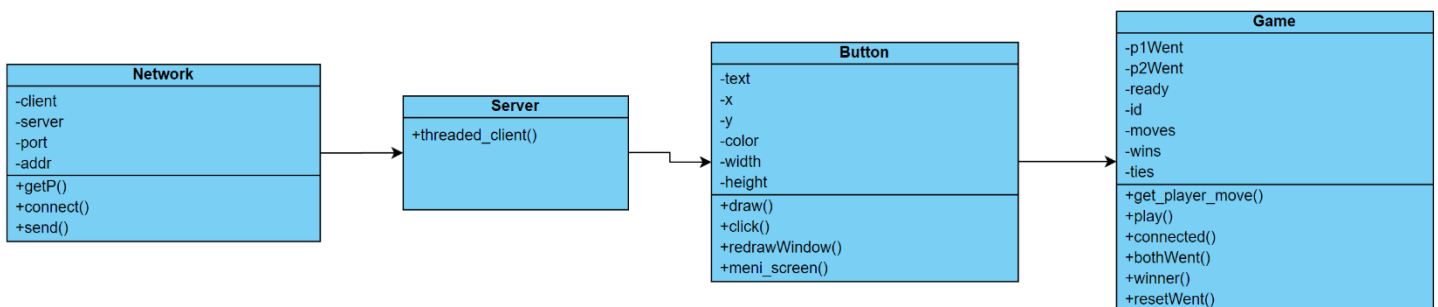
1	Vorgangsweise	1
2	Klassendiagramme	1
3	Beschreibung wichtiger Code – Teile.....	2
4	Kurzes Benutzerhandbuch.....	4
5	Reflexion	4



1 VORGANGSWEISE

Das Projekt basiert sich auf Socket und Thread. Ziel ist es, das Spiel „Rock-Paper-Scissors“ via Socket und Thread zu steuern und während dabei können zwei Spielern gegenseitig das Spiel spielen. Das Spiel wird über eine Oberfläche „pygame“(Modul) ausgeführt. Für die Ausführung am „Pygame“ wurde zusätzlich das Modul „Pickle“ (dient für die Oberfläche-Darstellung eines Objektes) inkrementiert. Als Erstes wurde das Network (Netzwerk) des Systems hergestellt, das heißt, die Ports, IP und Connection definiert. Dann wurde der Server programmiert. Der Server dient dazu, dass es mit dem Network verbindet. Anschließend wurde die Oberfläche des Spieles programmiert (Button, Größe der Fläche...). Zum Schluss wurde das Spiel mit einem Algorithmus programmiert.

2 KLASSENDIAGRAMME



3 BESCHREIBUNG WICHTIGER CODE – TEILE

Das Projekt befasst sich mit vier Python Dateien, diese lauten: „*client network server und game*“. In den unterliegenden Punkten werden die Dateien genauer erklärt.

- **network**: Die Networkkonfiguration wurde eingestellt, diese sind Ports, Server, Client, Adresse und Konnektivität (ON oder OFF). Zusätzlich wurde die Datei mit den Dateien Client und Server importiert bzw. verknüpft.

```
import socket
import pickle

class Network:
    def __init__(self):
        self.client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.server = "10.0.0.22"
        self.port = 5555
        self.addr = (self.server, self.port)
        self.p = self.connect()
```

- **server**: Hier wurde der Port, Server und die Anzahl der Teilnehmer definiert.

```
import socket
from _thread import *
import pickle
from game import Game

server = "10.0.0.22"
port = 5555

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try:
    s.bind((server, port))
except socket.error as e:
    str(e)

s.listen(2)
print("Waiting for a connection, Server Started")
```

- **client:** Die Oberfläche des Spieles wurde programmiert, diese waren die Farbe, der Button und die Länge bzw. die Breite der Oberfläche.

```
import pygame
from network import Network
import pickle
pygame.font.init()

width = 700
height = 700
win = pygame.display.set_mode((width, height))
pygame.display.set_caption("Game")

class Button:
    def __init__(self, text, x, y, color):
        self.text = text
        self.x = x
        self.y = y
        self.color = color
        self.width = 150
        self.height = 100
```

- **game:** Der Algorithmus des Spieles wurde hier programmiert. (Definition der Spieler, Win oder Lose)

```
def winner(self):

    p1 = self.moves[0].upper()[0]
    p2 = self.moves[1].upper()[0]

    winner = -1
    if p1 == "R" and p2 == "S":
        winner = 0
    elif p1 == "S" and p2 == "R":
        winner = 1
    elif p1 == "P" and p2 == "R":
        winner = 0
    elif p1 == "R" and p2 == "P":
        winner = 1
    elif p1 == "S" and p2 == "P":
        winner = 0
    elif p1 == "P" and p2 == "S":
        winner = 1

    return winner
```



4 KURZES BENUTZERHANDBUCH

ACHTUNG!!! Bevor man das Projekt startet, sollen die Module pickle und pygame installiert werden. Als Nächstes muss die lokale IP-Adresse geändert, damit der Server funktioniert, um das zu ändern, muss derjenige bzw. diejenige unter Windowssuche CMD eingeben. Nach der Eingabe muss folgender Befehl geschrieben werden: ipconfig. Zum Schluss muss die Adresse kopiert werden, danach wird die Adresse in den folgenden Python Dateien „Server und Network“ kopiert.

1. Zuerst muss der Server gestartet werden
2. Dann wird der Client startet => erscheint eine Oberfläche (pygame)
3. Gleiches Prinzip für Spieler 2 wie im Nummer 1 und 2 steht
4. Zum Schluss kann das Spiel gespielt werden

5 REFLEXION

Schuld Ich:

Selber habe ich Schuld, dass ich wenig Stunden fürs Projekt investiert habe, dagegen hat mein Partner Elias mehr Stunden investiert, aber normalerweise habe ich mit meinem Partner folgendes ausgemacht: „Dass er das Projekt fertig macht dafür mache ich die Dokumentation fertig. Zum Schluss muss ich noch erwähnen, dass ich bei jeder Recherche den Elias geholfen habe und habe ständig die Lösungen geschickt. Nicht im Leben kann man alles haben. Uns ist es auch klar, dass wir nicht Babys sind, sondern wir sind HTL Schüler.

Schuld Partner:

Mein Partner Elias hat mir nie zurückgeschrieben, wobei ich bei jedem Schreiben höflich geschrieben habe, aber er hat mich ständig ignoriert. Elias hat mich immer in stichgelassen und nie unterstützt.



Schuld Lehrer:

Für mich ist der Lehrperson ein Neutraler. Leider habe ich das bei der Lehrperson nicht gefühlt, ständig hat der Lehrperson mich beschuldigt, so habe ich das gefühlt. Ich habe gegen den Professor kein Problem, sogar ich mag ihn sehr gern. Für die Zukunft soll der Lehrer die Gruppen so einteilen, dass die Schüler gut arbeiten bzw. gut auskommen können und der Lehrer soll neutral sein.

QUELLENVERZEICHNIS:

1. <https://realpython.com/python-sockets/>
2. <https://www.youtube.com/watch?v=T0rYSFPAR0A>
3. https://openbook.rheinwerk-verlag.de/python/34_001.html
4. https://www.tutorialspoint.com/python/python_networking.htm
5. <https://realpython.com/python-rock-paper-scissors/>
6. <https://realpython.com/pygame-a-primer/>
7. <https://www.python-lernen.de/pygame-tutorial.htm>
8. https://www.python-kurs.eu/tkinter_buttons.php
9. <https://www.youtube.com/watch?v=yuuDJ3-EdNQ>
10. https://www.tutorialspoint.com/python/tk_button.htm