THU

Technische
Hochschule
Ulm

Vorkurse der Fakultät Informatik
Projekt Tic-Tac-Toe

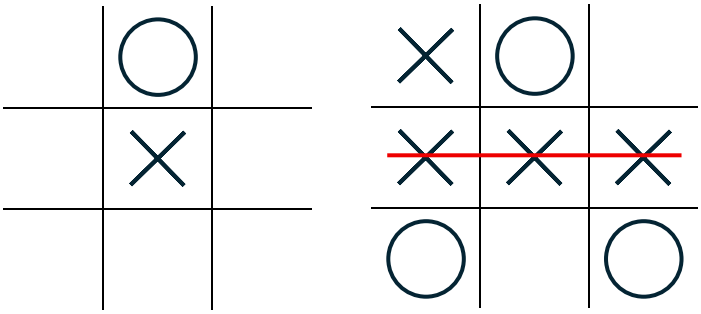
Projekt Tic-Tac-Toe

Allgemein

Ziel dieses Projekts ist es ein lauffähigs Python-Programm zu implementieren. Die komplette Benutzerführung und Spielsteuerung (Ein- und Ausgabe) soll konsolenbasiert sein. Der Spielverlauf ist möglichst originalgetreu und benutzerfreundlich umzusetzen bzw. zu gestalten. Ein Spieler der den Quellcode bzw. die Besonderheiten Ihrer Applikation nicht kennt, soll trotzdem in der Lage sein, sofort losspielen zu können. Das Programm soll möglichst stabil laufen, d.h. auch eventuelle Falscheingaben durch den Spieler müssen abgefangen werden.
Ein weiterer wichtiger Aspekt ist Ihre Programmarchitektur. Versuchen Sie Ihren Quellcode möglichst logisch und wiederverwendbar zu strukturieren.

Spielverlauf

„Auf einem quadratischen, 3×3 Felder großen Spielfeld setzen die beiden Spieler abwechselnd ihr Zeichen (ein Spieler Kreuze, der andere Kreise) in ein freies Feld. Der Spieler, der als Erster drei Zeichen in eine Zeile, Spalte oder Diagonale setzen kann, gewinnt. Wenn allerdings beide Spieler optimal spielen, kann keiner gewinnen, und es kommt zu einem Unentschieden. Das heißt, alle neun Felder sind gefüllt, ohne dass ein Spieler die erforderlichen Zeichen in einer Reihe, Spalte oder Diagonalen setzen konnte.“
Quelle: Wikipedia (https://de.wikipedia.org/wiki/Tic-Tac-Toe | Zugriff am 09.09.2025).



Aufgabe 1

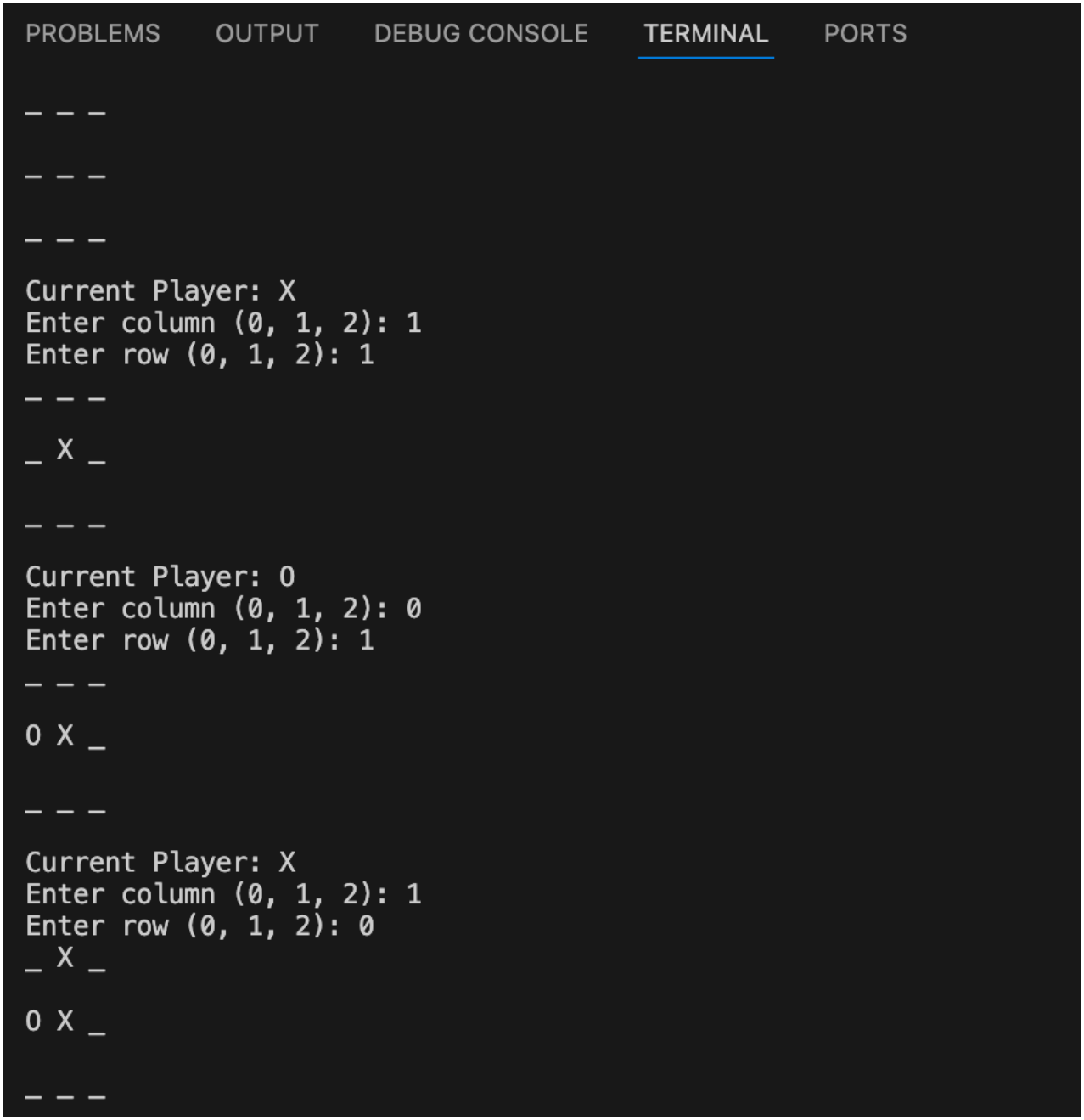
Überlegen Sie sich zunächst eine Struktur für Ihr Programm. Dabei sollten Sie folgende Fragen beantworten:

- In welche Teile können Sie den Spielablauf zerlegen?
- Wo bietet es sich an eine Funktion zu erstellen?
- Wie kann die Gewinnüberprüfung durgeführt werden?
- Wie kann ein Unentschieden erkannt werden?
- Wie kann das Spielfeld abgespeichert werden?

Aufgabe 2

Implementieren Sie nun Ihre geplante Struktur in Python. Wenn Sie in einer Gruppe arbeiten, können Sie die Implementierung der Funktionen auch auf Teammitglieder aufteilen. Es hilft dabei sehr, wenn vorher die

Schnittstellen (Name der Funktion, Rückgabewert, Parameter) klar definiert sind. Ein möglicher Spielablauf könnte wie folgt aussehen:



Tipp: Eine Ausgabe ohne Zeilenumbruch kann wie folgt ausgeführt werden:

```
print(text, end=" ")
```