**Lösungsskizze BSRN Werkstück A**

Allgemeines:  
Unsere Gruppe hat sich für das Werkstück A die Alternative 3 – Buzzword-Bingo-Spiel mit Interprozesskommunikation – ausgesucht. Zur Lösung der Aufgabenstellung wird die Programmiersprache Python verwendet.

Die folgende Lösungsskizze beschreibt einen ersten Entwurf für das Vorgehen unserer Gruppe.

Lösungsansatz in grafischer Form:

Spieler (Prozess)

Host (Prozess)

Bingo-Card (Class)

Creates Bingo-Card mit Worddatei.txt

Creates Logfile

Bingo-Game (Class)

Message Queues/ Pipes

Game Start

Game End

Pro Prozess:

Abbildung in einer Shell

Message Queues/Pipes

n

1

Worddatei.txt

reads

Game Start

Creates Roundfile

Ausformulierung der grafischen Lösungsskizze:

Der Lösungsansatz verwendet Objektorientierung, Dateizugriffe bzw. Serialisierung, und das Modul POSIX-IPC für Python (Message Queue/Pipes).

Erläuterungen der einzelnen Bausteine:

* Klasse **„BingoCard“:** für die Bingo-Karten und deren Verwaltung, ankreuzen von Feldern über Anklicken der Zelle in der Shell, enthält auch das Rückgängig machen von Markierungen
* Klasse **„BingoGame“:** für die Steuerung des Spiels und der Interprozesskommunikation (POSIX-IPC: Verwendung von Messages Queues), enthält auch Methoden zur Überprüfung der Siegbedingung bzw. der Spiellogik
* Für jede Bingo-Karte wird eine **Logdatei** erstellt welche Startzeit, Größe des Feldes, Markierungen und Spielende aufzeichnet
* Bei jedem Spielstart wird eine **Roundfile** erstellt. Diese enthält die Runden-ID (in Form des Zeitpunkts des Erstellens des Spiels), maximale Spieleranzahl, Größe des Feldes
* Die **Worddatei** liegt im Verzeichnis und wird beim Erstellen der Bingo-Karten eingelesen.
* Die Grafische Ausgabe erfolgt in der Shell

Ablauf des Spiels:

Der Host – wird automatisch als Spieler 1 initialisiert– startet ein Spiel. Beim Start des Spiels wird die Datei Roundfile erstellt. Dabei legt der Host die Parameter (Größe der Bingo-Karte, maximale Anzahl an Spielern) und damit den Inhalt der Datei fest. Anschließend ist es verschiedenen Spielern möglich über eine eigene Shell auf dem gleichen Computer dem Spiel beizutreten. Dafür muss die Roundfile eingelesen werden können, um die Spiel-ID herauszufinden. Bei jedem Eintritt erhalten alle schon beigetretenen Spieler über die Interprozesskommunikation die Nachricht, dass ein weiterer Spieler beigetreten ist. Außerdem wird bei jedem Spieleintritt für jeden Spieler ein Objekt der Klasse BingoCard instanziiert und eine Logfile erstellt. Der Host kann jederzeit das Spiel starten und abbrechen, sobald mindestens ein weiterer Spieler beigetreten ist. Jeder Spieler hat seine BingoCard in seiner Shell offen und kann die Felder bei Gegebenheit ankreuzen. Über die Interprozesskommunikation bekommt jeder Spieler den Hinweis, wenn ein Spieler gewonnen hat und/oder das Spiel vorbei ist.

Bisherige Arbeitsumgebung:

* Allgemeine Informationen und Koordination der Gruppe über Textdateien im Github Repository
* Der Code im GitHub Repository wird in PyCharm erstellt. Python Interpreter Python 3.12.10 WSL eingerichtet. Posix und Pip packages in Pycharm installiert. Bisherige Test-Codes mit Message Queues waren erfolgreich.

Aufgabenverteilung in der Gruppe:

* Bilal, Marvin und Bene:
  + Erstellen die Klasse BingoGame und implementieren angegebene Methoden mit Schwerpunkt IPC
* Robin und Jamie:
  + Erstellen die Klasse BingoCard und implementieren angegebene Methoden mit Schwerpunkt auf dem Erstellen der IPC und Handling in der Shell