**Anforderungsanalyse:**

Das fertige Programm soll eine Kommandozeilenanwendung sein.

• Der Quellcode soll durch Kommentare verständlich sein.

• Die Spieler (mindestens zwei Spieler) sollen auf einem Computer gegeneinander

spielen können.

• Jeder Spieler ist ein eigener Prozess. Nutzen Sie zur Interprozesskommunikation entweder Nachrichtenwarteschlangen (Message

Queues nach den Standards System V oder POSIX), Anonyme Pipes

und/oder Benannte Pipes.

• Jeder Spieler, der am Spiel teilnimmt, bekommt vom Spiel eine eigene Bingokarte in der Shell generiert.

• Die Anzahl der Felder in der Breite und Höhe der zu generierenden Bingokarten legt der Benutzer, der das Spiel eröffnet, per Kommandozeilenargument

fest. Also z.B. -xaxis 5 -yaxis 5

• Die Werte der Felder auf der Bingokarte sollen aus einer Textdatei eingelesen und per Zufall verteilt werden. Konkret soll der Benutzer, der das Spiel

eröffnet, die Möglichkeit haben, per Kommandozeilenargument den Pfad und

Dateinamen der Textdatei zu definieren, also z.B. -wordfile <dateiname>.

• Die Textdatei kann auch gerne mehr Wörter enthalten und es werden entsprechend so viele Wörter zufällig vom Buzzword-Bingo-Spiel aus der Datei

importiert und auf der bzw. den Bingokarte(n) verteilt, wie es die definierte

Höhe und Breite vorgibt.

• Die Spieler wählen mit der Tastatur (und evtl. auch mit der Maus – das ist

aber nicht zwingend) einzelne Felder aus, um diese zu streichen (bzw. zu markieren). Alternativ ist eine Auswahl der Felder durch Eingabe der Koordinaten

mit der Tastatur auch denkbar. Verwenden Sie dafür eine geeignete Bibliothek

oder eine vergleichbare Lösung, die es ermöglicht, grafisch ansehnliche Ausgaben auf der Kommandozeile zu realisieren. Beispiele für Bibliotheken, die

„grafische Darstellung“ und Bedienung in der Shell ermöglichen, sind ncurses1 2 (für C-Programme), Termbox3

(für C-Programme oder Python-Scripte

- wird nicht mehr weiterentwickelt), Textual4

(für Python-Scripte), Typer5

(für Python-Scripte), Asciimatics6

(für Python-Scripte), pyTermTk7

(für

Python-Scripte), dialog8 9 10 (für Shell-Scripte) oder Whiptail11 12 13 (für

Shell-Scripte).

• Fehler bei der Bedienung (z.B. fehlerhaftes Streichen eines Feldes sollen die

einzelnen Benutzer auch rückgängig machen können.

• Fehlerhafte Kommandozeilenargumente soll das Programm erkennen und

durch Fehlermeldungen und/oder Abbruch des Programms sinnvoll behandeln

können.

• Hat ein Spieler oder ein Gegenspieler eine komplette Spalte, Zeile oder Diagonale seiner Bingokarte an Feldern gestrichen bzw. markiert, gilt das Spiel

als gewonnen, was bei allen Spielern angezeigt wird. Das kann beispielsweise

durch eine Laufschrift geschehen, durch ein Blinken oder durch ein Invertieren

der Farben in der Shell, etc.

• Bei 5x5 oder 7x7 Feldern bleibt das Feld in der Mitte üblicherweise frei (Joker).

• Zur Dokumentation des Spiels soll das Programm für jede generierte Bingokarte eine Logdatei mit folgendem Dateinamen anlegen:

YYYY-MM-DD-HH-MM-SS-bingo-SpielerNummer.txtenthalten. Die Logdatei soll während des Spiels mit sinnvollen Daten (Zeilen)

befüllt werden. Sinnvolle Beispiele sind:• Ein Spieler definiert und eröffnet eine Partie bzw. eine Bingo-Runde, und die

anderen Benutzer auf dem gleichen Computer können beitreten. Sie haben die

freie Wahl, wie sie die Propagierung bzw. Definition von Spielrunden und den

Beitritt zu einer Partie bzw. zu einem Spiel realisieren. Dieses ist zum Beispiel

über eine (als Kommandozeilenargument) definierbare lokale Datei möglich,

die einen Namen für die Spielrunde und die beteiligten Prozess-ID-Nummern

(PIDs) enthält.