

# Projekt Buzzword Bingo

Ioana-Carmen Moldovan (1514892)

Vanessa Nguyen (1513741)

Pantea Dolatabadian (1414653)

Aliia Nurbekova (1526039)

Thi Song Thu Pham (1413382)

Frankfurt University of Applied Sciences

June 30, 2024



- 1 Einführung in das Projekt
- 2 Gesamtüberblick über das System
  - Systemarchitektur
  - Technologien und Werkzeuge
  - Logdatei
- 3 Detaillierte Beschreibung der Komponenten
  - Server-Komponente
  - Client-Komponente
- 4 Demonstration der Funktionalität
- 5 Herausforderungen und Lösungen
- 6 Fazit und zukünftige Arbeiten
  - Erreichte Ziele
  - Lerneffekte
  - Zukünftige Arbeiten

# Table of Contents

- 1 Einführung in das Projekt
- 2 Gesamtüberblick über das System
  - Systemarchitektur
  - Technologien und Werkzeuge
  - Logdatei
- 3 Detaillierte Beschreibung der Komponenten
  - Server-Komponente
  - Client-Komponente
- 4 Demonstration der Funktionalität
- 5 Herausforderungen und Lösungen
- 6 Fazit und zukünftige Arbeiten
  - Erreichte Ziele
  - Lerneffekte
  - Zukünftige Arbeiten

- **Projektziel:** Entwicklung eines interaktiven Buzzword Bingo Spiels
- **Motivation:**
  - **Relevanz und Anwendbarkeit:** Buzzword Bingo ist eine kreative und praktische Anwendung für Meetings und Präsentationen, um Aufmerksamkeit und Unterhaltung zu steigern.
  - **Technische Herausforderung:** Beherrschung verschiedener Technologien und Konzepte wie Interprozesskommunikation, Ereignissteuerung und GUI-Entwicklung.
  - **Lernmöglichkeiten:** Vertiefung in Python, Nutzung von TermTk für GUI-Entwicklung und Implementierung von Named Pipes für Prozesskommunikation.
  - **Teamarbeit und Zusammenarbeit:** Erfordert parallele Entwicklung und Integration mehrerer Komponenten.
  - **Spaßfaktor:** Spielerischer Charakter des Projekts sorgt für Motivation und Spaß.

# Table of Contents

- 1 Einführung in das Projekt
- 2 Gesamtüberblick über das System
  - Systemarchitektur
  - Technologien und Werkzeuge
  - Logdatei
- 3 Detaillierte Beschreibung der Komponenten
  - Server-Komponente
  - Client-Komponente
- 4 Demonstration der Funktionalität
- 5 Herausforderungen und Lösungen
- 6 Fazit und zukünftige Arbeiten
  - Erreichte Ziele
  - Lerneffekte
  - Zukünftige Arbeiten

- **Beschreibung der Architektur:**

- **Server:** Verwaltet das Spiel und die Kommunikation zwischen den Clients.
- **Clients:** Anzeigen der Bingo-Karte und Interaktion mit dem Benutzer.
- **Kommunikation:** Verwendung von Named Pipes zur Datenübertragung.
- Darstellung des Datenflusses zwischen den Komponenten

- **Verwendete Technologien:**

- Programmiersprache: Python
- Framework: pyTermTk
- Kommunikation: Benannte Pipes

- **Entwicklungswerkzeuge:**

- Entwicklungsumgebung: Visual Studio Code
- Versionskontrolle: GitHub
- Weitere: Ubuntu für Terminal, YouTube zum Lernen

- **Aufgaben:**

- Erstellung einer Logdatei mit dem Namen im Format YYYY-MM-DD-HH-MM-SS-bingo-SpielerNummer.txt
- Datei ausfüllen mit zeitgestempelten Ereignissen während des Spiels
- Beispiele: Start/Ende des Spiels, Streichen eines Wortes, Sieg etc.



# Logdatei - Beispiel

```
2024-06-30 19:11:46,516 - Start des Spiels
2024-06-30 19:11:46,517 - Größe des Spielfelds: 3 x 3
2024-06-30 19:11:48,898 - Best Practice (1/0)
2024-06-30 19:11:49,816 - Game-changing (0/2)
2024-06-30 19:11:50,447 - Qualität (0/1)
2024-06-30 19:11:52,669 - Szenario (2/1)
2024-06-30 19:11:53,223 - Sieg
2024-06-30 19:11:53,224 - Ende des Spiels
```

# Table of Contents

- 1 Einführung in das Projekt
- 2 Gesamtüberblick über das System
  - Systemarchitektur
  - Technologien und Werkzeuge
  - Logdatei
- 3 Detaillierte Beschreibung der Komponenten
  - Server-Komponente
  - Client-Komponente
- 4 Demonstration der Funktionalität
- 5 Herausforderungen und Lösungen
- 6 Fazit und zukünftige Arbeiten
  - Erreichte Ziele
  - Lerneffekte
  - Zukünftige Arbeiten

- **Funktionalität:**

- Verwaltung der Spielinstanzen
- Kommunikation mit den Clients
- Senden von Benachrichtigungen (Spielstart, Spielende, Beitreten von Spielern)
- Sicherstellung der Kommunikation zwischen den Spielern

- **Technische Details:**

- Implementierung der benannten Pipes für die Kommunikation
- Handhabung von Nachrichten und Spielstatus

# Server-Prozess Teil 1

```
def server_process(pipe_name, pos, size):  
    """  
    Server-Prozess zur Verwaltung des Spiels und der Bingo-Karte des Servers  
  
    pipe_name: Name der benannten Pipe zur Kommunikation  
    pos: Position des Fensters  
    size: Größe des Fensters  
    """  
    print("Das Bingospiel wurde gestartet.")  
  
    if not os.path.exists(pipe_name):  
        # Erstellen mit First in First Out  
        os.mkfifo(pipe_name)  
  
    # Spieler in einer Liste speichern  
    clients = []
```

# Server-Prozess Teil 2

```
# Benannte Pipe im read()-Modus öffnen
while True:
    try:
        with open(pipe_name, 'r') as pipe:
            while True:
                message = pipe.readline().strip()
                if message:
                    logging.info(f"{datetime.now().strftime('%Y-%m-%d-%H-%M-%S')} Nachricht empfangen: {message}")
                    print(f"{message}")
                    if "ist beigetreten" in message:
                        client_name = message.split()[0]
                        clients.append(client_name)
                    elif "Kein Gewinner" in message:
                        broadcast_message(clients, message)
                        break
                    elif "hat gewonnen" in message:
                        broadcast_message(clients, message)
                        break
            except Exception as e:
                logging.error(f"Error reading from pipe: {e}")
```

```
# Für jeden Client wird eine Pipe erstellt
def broadcast_message(clients, message):
    for client in clients:
        client_pipe_name = f"/tmp/{client}_pipe"
        if not os.path.exists(client_pipe_name):
            os.mkfifo(client_pipe_name)
        with open(client_pipe_name, 'w') as pipe:
            pipe.write(f"{message}\n")
            pipe.flush()

    print("Das Spiel ist beendet.")
    sys.exit()
```

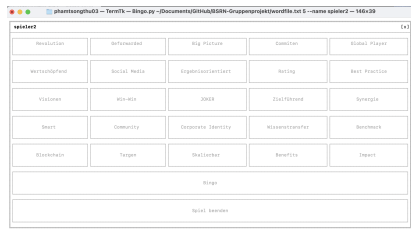
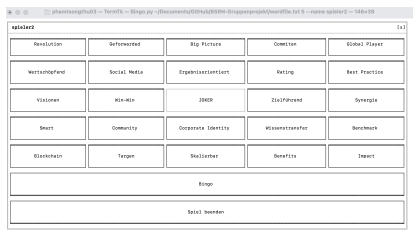
## • Funktionalität:

- Namenseingabe: Spieler können ihren Namen eingeben, um dem Spiel beizutreten.
- Bingo-Karte: Anzeige der Bingo-Karte mit frei wählbaren Feldern.
- Interaktive Buttons:
  - Bingo: Meldet Bingo, wenn der Spieler glaubt, gewonnen zu haben.
  - Spiel beenden: Ermöglicht es dem Spieler, das Spiel zu verlassen.
- Wörter Auswahl:
  - Spieler können Wörter auf der Karte markieren und demarkieren.
  - Meldet Bingo, wenn alle Wörter in einer Reihe/Spalte/Diagonale markiert sind.
- Automatische Sperre: Die Bingo-Karte wird automatisch gesperrt, wenn das Spiel endet.

# Client-Komponente - Teil 2

## ● Technische Details:

- Implementierung der Benutzeroberfläche mit pyTermTk.
- Verwendung von benannten Pipes zur Kommunikation mit dem Server.
- Ereignisgesteuerte Programmierung zur Handhabung der Benutzerinteraktion.





# Client: Kommunikation über Pipes

```
def lese_pipe():
    client_pipe_name = f"/tmp/{name}_pipe"
    if not os.path.exists(client_pipe_name):
        os.mkfifo(client_pipe_name)
    with open(client_pipe_name, 'r') as pipe:
        message = pipe.readline().strip()
        if message:
            if "hat gewonnen" in message and name not in message:
                spieler.zeige_verloren_nachricht()
                spieler.logger.info(f"Ende des Spiels")
                spieler.lock_bingo_card()
            elif "Kein Gewinner" in message:
                spieler.lock_bingo_card()
                spieler.logger.info(f"Ende des Spiels")
```

# Table of Contents

- 1 Einführung in das Projekt
- 2 Gesamtüberblick über das System
  - Systemarchitektur
  - Technologien und Werkzeuge
  - Logdatei
- 3 Detaillierte Beschreibung der Komponenten
  - Server-Komponente
  - Client-Komponente
- 4 Demonstration der Funktionalität**
- 5 Herausforderungen und Lösungen
- 6 Fazit und zukünftige Arbeiten
  - Erreichte Ziele
  - Lerneffekte
  - Zukünftige Arbeiten

- Demonstration der Spieloberfläche
- Interaktion zwischen Client und Server
- Beispiel für eine Spielrunde

# Table of Contents

- 1 Einführung in das Projekt
- 2 Gesamtüberblick über das System
  - Systemarchitektur
  - Technologien und Werkzeuge
  - Logdatei
- 3 Detaillierte Beschreibung der Komponenten
  - Server-Komponente
  - Client-Komponente
- 4 Demonstration der Funktionalität
- 5 Herausforderungen und Lösungen
- 6 Fazit und zukünftige Arbeiten
  - Erreichte Ziele
  - Lerneffekte
  - Zukünftige Arbeiten

- **Herausforderungen:**

- Synchronisation der Kommunikation zwischen Server und Clients
- Verwaltung der Benutzerinteraktionen in Echtzeit

- **Lösungen:**

- Verwendung von benannten Pipes für zuverlässige Kommunikation
- Ereignisgesteuerte Programmierung zur Handhabung der Benutzerinteraktionen

# Table of Contents

- 1 Einführung in das Projekt
- 2 Gesamtüberblick über das System
  - Systemarchitektur
  - Technologien und Werkzeuge
  - Logdatei
- 3 Detaillierte Beschreibung der Komponenten
  - Server-Komponente
  - Client-Komponente
- 4 Demonstration der Funktionalität
- 5 Herausforderungen und Lösungen
- 6 Fazit und zukünftige Arbeiten
  - Erreichte Ziele
  - Lerneffekte
  - Zukünftige Arbeiten

- **Erreichte Ziele:**

- Erfolgreiche Entwicklung eines interaktiven Buzzword-Bingo Spiels.
- Positive Rückmeldungen während der Tests.

- **Lerneffekte:**

- **Technische Kenntnisse:** Vertiefte Kenntnisse in Python und Kommandozeilen, Verständnis der Named Pipes für Interprozesskommunikation.
- **Werkzeuge und Technologien:** LaTeX, GitHub, Kommandozeilentools.
- **Projektarbeit:** Verbesserte Teamarbeit, Projektplanung und Aufgabenverteilung.
- **Selbststudium:** Eigenständiges Erlernen neuer Bibliotheken und Konzepte.



- **Zukünftige Arbeiten:**

- Verbesserung der Benutzeroberfläche
- Unterstützung für mehrere Spielmodi
- Integration von Web-Sockets für eine bessere Skalierbarkeit

# Vielen Dank für Ihre Aufmerksamkeit!

Wir freuen uns auf Ihre Fragen und Ihr Feedback.