



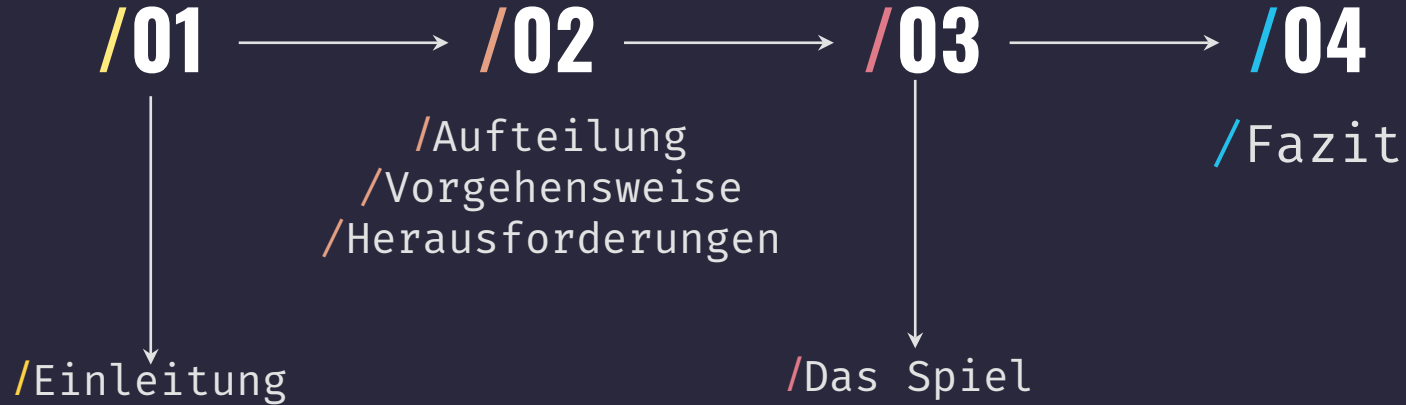
/ Buzzword-Bingo- Spiel in Python

Fabian Matzollek, Jakub Naruszko, Louisa Bahr,
Michael Nguyen, Mustafa Islek





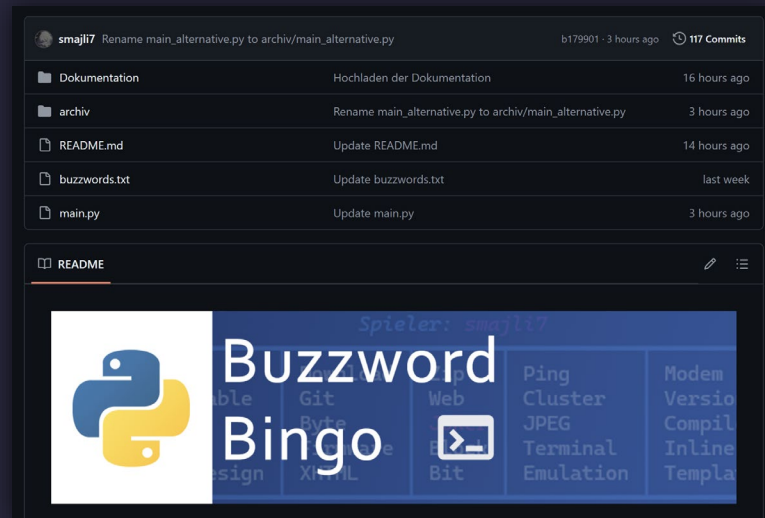
/Agenda



/Basics

- **GitHub:** github.com/smajli7/BingoSpielBSRN
- **Programmart:** Spiel
- **Sprache:** Python
- **Projektart:** Gruppenarbeit
- **Gruppengröße:** 5

- -rich
- main.py
- Buzzwords.txt





/Buzzword-Bingo

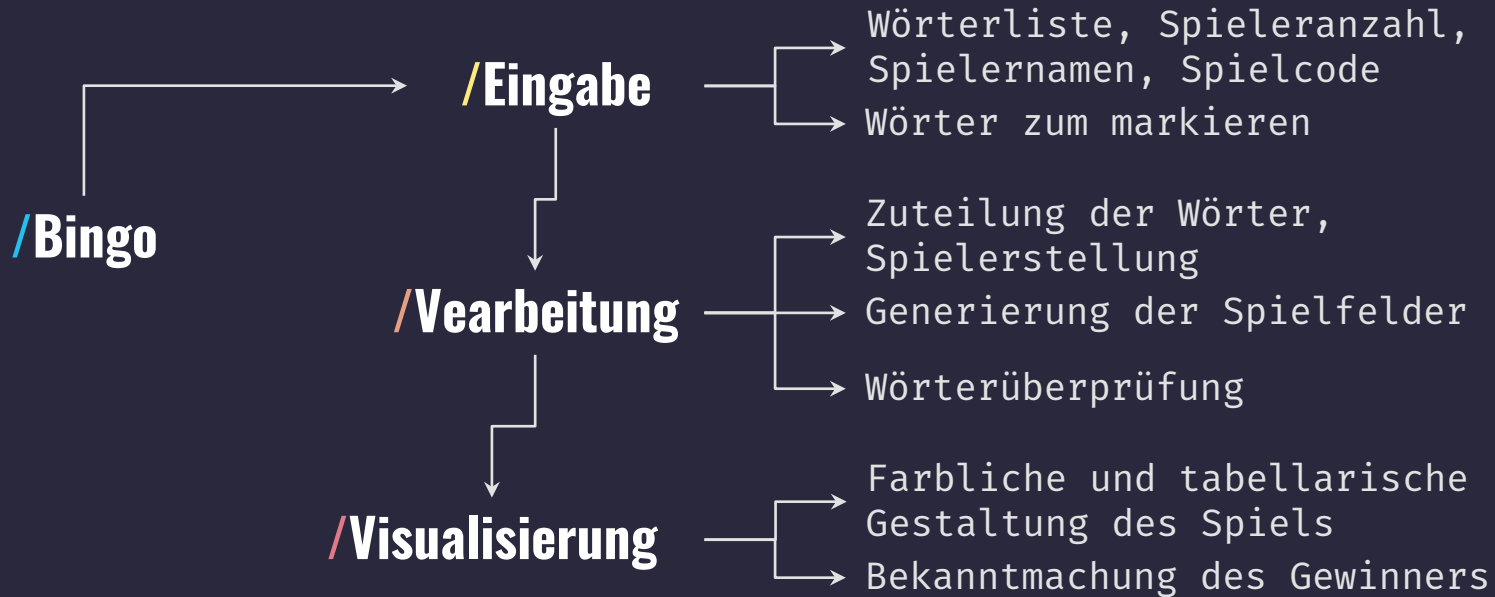


- **Spielprinzip:** Humoristische Variante des Bingo-Spiels
- **Einsatzort:** Vorträge, Präsentationen, Besprechungen
- **Spielkarten:** Statt Zahlen sind Schlagwörter (Buzzwords) auf den Karten
- **Regeln:** Wörter werden gestrichen, wenn sie genannt werden
- **Gewinnbedingung:** Eine vollständig gefüllte Reihe, Spalte oder Diagonale
- **Gewinnausruf:** Spieler ruft «Bingo» oder «Bullshit»
- **Kritik:** Übermäßige und inhaltsleere Verwendung von Schlagwörtern



Quelle: <http://de.wikipedia.org/wiki/Buzzword-Bingo> Abgerufen: 20.06.24

/Spielfunktionen





/Arbeitsaufteilung



Gruppenmitglieder

Aufgaben

/Basis-Code

Fundament und
Spielfunktionen



/Fabian + Micheal

/IPC

IPC + Helfer bei
Spielfunktionen



/Mustafa

/Visualisierung

GUI und
Logdateien



/Jakub + Louisa





/Probleme & Herausforderungen

/GUI

Bibliothek, Ideen,
Umsetzung, User-Input

/Spielfeld

Größe, Joker,
Generierung



/Wörter

Auswahl, Duplikate,
Anzahl

/Pipes & Logdateien

IPC, falsche Logs,
Spielende, Gewinner





/Live-Demonstration



```
>bingo:~$ main.py
```





/Gui >>> python -m pip install rich



```
145     table = Table(show_header=False, box=HEAVY_EDGE, border_style="bold blue",
146                   title=f"[bold blue]Spieler:[/bold blue] [magenta]{playernamelist[i]}[/magenta]")
147
148     for _ in range(len(matrix[0])): # Spaltenanzahl
149         table.add_column()
150
151     for row in matrix: # row ist eine Zeile der Bingokarte
152         table.add_row(*[f"[red]{cell}[/red]" if cell in marked_words or cell == "Joker" else str(cell) for cell in row])
153
154     console.print(table)
```

def display_bingo_cards()





/Spielfeld



```
115     for k in playernamelist: # Schleife, die die Bingokarten für jeden Spieler generiert
116         if len(buzzwords_list) < xsize * ysize: # Überprüfung, ob genug Buzzwörter vorhanden sind
117             raise ValueError(
118                 "Nicht genug Buzzwords, um die Bingokarten zu füllen") # Fehlermeldung, wenn nicht genug Buzzwörter vorhanden sind
119     used_words = set() # Set für verwendete Wörter
120     matrix = [] # Liste für die Bingokarte
121     for l in range(ysize): # Zeilen
122         b = [] # Liste für die Zeile
123         for j in range(xsize): # Spalten
124             if xsize % 2 != 0 and ysize % 2 != 0 and l == middle_y and j == middle_x: # Joker in der Mitte, nur wenn xsize und ysize ungerade sind
125                 b.append("Joker") # Fügt den Joker in die Mitte der Bingokarte ein
126             else:
127                 while True:
128                     random_word = buzzwords_list.pop(0) # Nimmt das erste Element aus der Liste und entfernt es, damit kein Wort doppelt verwendet wird
129                     if random_word not in used_words: # Überprüfung, ob das Wort bereits verwendet wurde
130                         used_words.add(random_word) # Fügt das Wort zu den verwendeten Wörtern hinzu
131                         b.append(random_word) # Fügt das zufällige Wort in die Bingokarte ein
132                         break # Beendet die Schleife, wenn ein neues Wort gefunden wurde
133                 buzzwords_list.append(random_word) # Füge das Wort zurück zur Liste, damit es nicht verloren geht
134             matrix.append(b) # Fügt die Zeile der Bingokarte in die Bingokarte ein
135         matrixlist.append(matrix) # Fügt die Bingokartenmatrix einer Person in die Bingokartenliste ein
136     return matrixlist # Gibt die Liste der Bingokarten zurück
```

def generate_bingo_cards()





/Wörter

```
20  ▾ def initialize_file(filename): # Funktion, die die Datei einliest
21      global buzzwords_list
22      try:
23          with open(filename) as file: # Öffnet die Datei
24              reader = file.readlines() # Liest die Datei ein
25              buzzwords_list = [i.strip() for i in reader] # Entfernt die Zeilenumbrüche
26              random.shuffle(buzzwords_list) # Mischt die Liste, um zufällige Auswahl zu gewährleisten
27      except FileNotFoundError:
28          console.print(f"Fehler: Datei '{filename}' nicht gefunden.", style="bold red")
29          return False
30      return True
```

```
128         random_word = buzzwords_list.pop(0) # Nimm ein Wort
129         if random_word not in used_words: # Überprüfe, ob das Wort bereits verwendet wurde
130             used_words.add(random_word) # Fügt das Wort zur Liste hinzu
131             b.append(random_word) # Fügt das Wort zur Zeile hinzu
132             break # Beendet die Schleife, wenn ein Wort gefunden wurde
133         buzzwords_list.append(random_word) # Füge das Wort zurück zur Liste hinzu
134     matrix.append(b) # Fügt die Zeile der Bingokarte in die Matrix hinzu
135     matrixlist.append(matrix) # Fügt die Bingokartenmatrix zur Liste hinzu
136     return matrixlist # Gibt die Liste der Bingokarten zurück
```





/Pipes (Spielbeitritt)

- Runde beitreten möglich, wenn 'Spieler < Spieleranzahl'
- Für das Beitreten wird der Spielcode benötigt

```
Geben Sie die Spieleranzahl ein: 3
Geben Sie den Namen des Spielers 1 ein: Mustermann
```

| Spieler Nummer | Spieler Name |
|----------------|--------------|
| 1 | Mustermann |

```
Wie viele Spalten sollen die Bingokarten haben? (mindestens 2): 2
Wie viele Zeilen sollen die Bingokarten haben? (mindestens 2): 2
Es fehlen: 2 Spieler
Spielcode: 420
_
```

- Der code ist die FIFO-Pipe mit dem der Prozess eine Kommunikation aufbaut

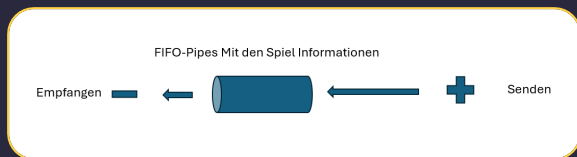
```
joinround / newround? joinround
Spielcode: 420_
```



/Pipes (Datenübertragung)



- Man erhält vom Gegner den Namen seiner FIFO-Pipe, über die wir dann die Spielinformation übertragen
- Der Gegner erhält diese Informationen und verarbeitet sie so das er damit ein Spiel Starten kann



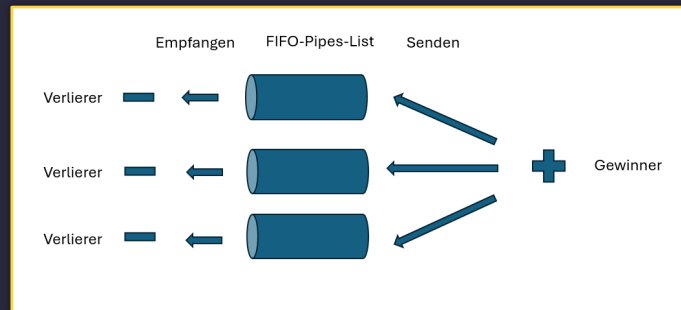
Spiel Informationen

| Position in list | Type | Result |
|---------------------|----------|-----------------|
| 0 | "String" | "buzzwords.txt" |
| 1 | "String" | "5" |
| 2 | "String" | "5" |
| 3 | "String" | "8945" |

// und so weiter

/Pipes (Gewinner)

- Wie überprüft man, wer gewonnen hat
- Was passiert, wenn man gewonnen hat
- Was passiert, wenn man verloren hat





/Log-Dateien



```
196  ✓ def create_log_file(pid, player_name):  
197      timestamp = datetime.now().strftime("%Y-%m-%d-%H-%M-%S")  
198      filename = f"{timestamp}-bingo-Spieler{pid}.txt"  
199      log_files[pid] = open(filename, 'w')  
200      log_files[pid].write(f"Spieler: {player_name}\n")  
201      log_files[pid].write(f"{timestamp} Start des Spiels\n")  
202      return log_files[pid]  
203  
204  
205  
206  def log_event(pid, event):  
207      timestamp = datetime.now().strftime("%Y-%m-%d-%H-%M-%S")  
208      log_files[pid].write(f"{timestamp} {event}\n")
```

```
    log_event(pid, f"Wort demarkiert: {word_to_mark}")  
else: # Wenn das Wort nicht markiert ist  
    marked_words.add(word_to_mark) # Fügt das Wort zu d  
    log_event(pid, f"Wort markiert: {word_to_mark}")
```





/Verbesserungen



IPC

- +Schneller
- +Zuverlässiger
- +Übersichtlicher

.....

Spielfeld

- +Möglichkeiten
- +Modifikationen

.....

GUI

- +Moderner
- +Maus-Input
- +Diverser

.....





/Fazit



- Solide Umsetzung der Aufgabenstellung
- Funktionsfähiges Spiel, welches jedoch nicht perfekt ist
- Vieles wurde neu erlernt
- I. Python Bibliotheken und Importe, Python an sich
- II. Umgang mit PowerShell, WSL und Ubuntu
- III. PyCharm als IDE und Erweiterungen
 - Umgang mit Gruppenarbeiten und Aufteilungen
 - Dokumentationen und Umgang mit LaTeX





/Quellen



<http://de.wikipedia.org/wiki/Buzzword-Bingo> Abgerufen: 20.06.24





/Danke



>>> Für Ihre Aufmerksamkeit!

