

# Meilenstein 3

## Trumpeltier vs Sombreros



### Gruppe 7 - SAND

CS108 Programmierprojekt, FS17

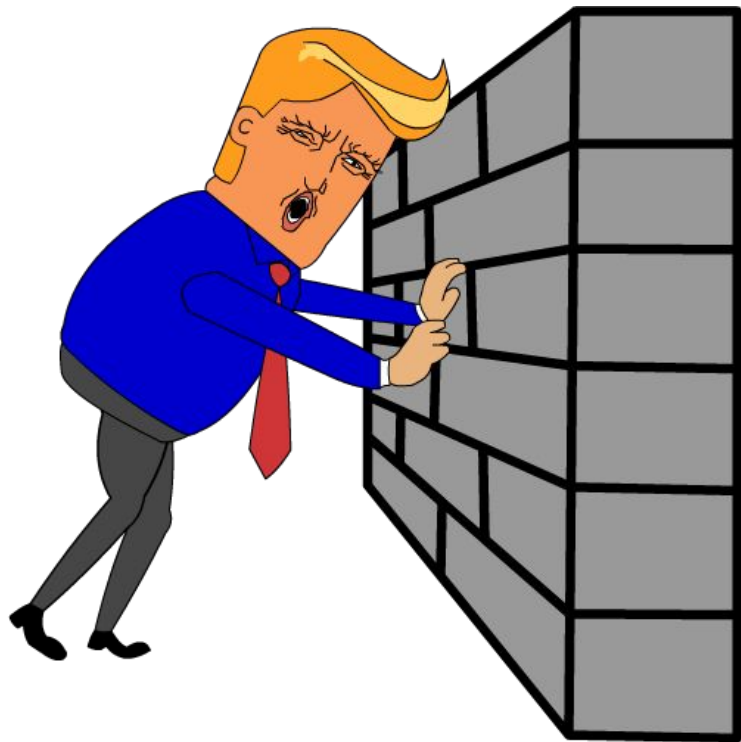
**S**yarif

**A**dalsteinn

**N**ils

**D**amian

# Trumpeltier vs Sombreros



Spieler 1



Spieler 2

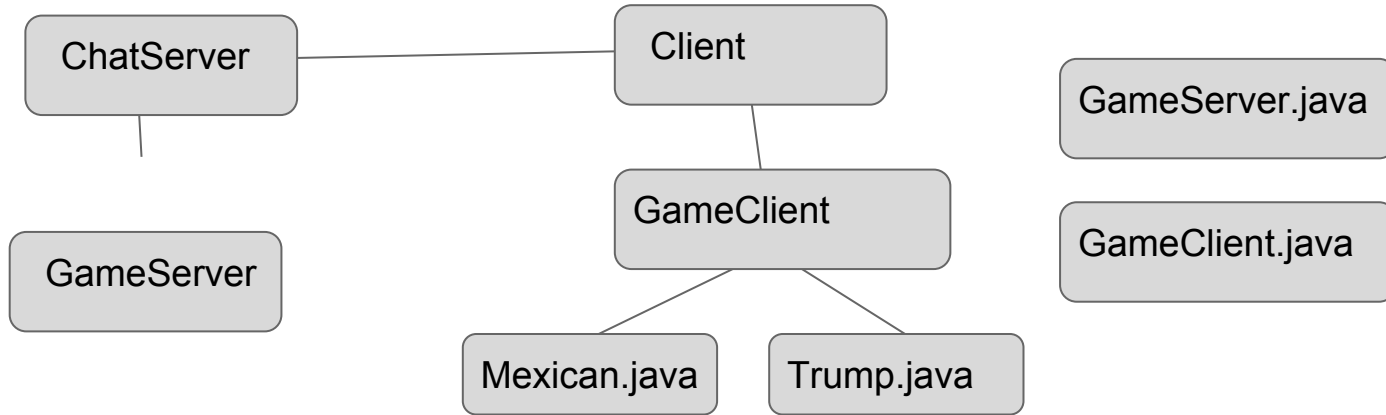
Collected  
Tacos  
40



# Spielstatus

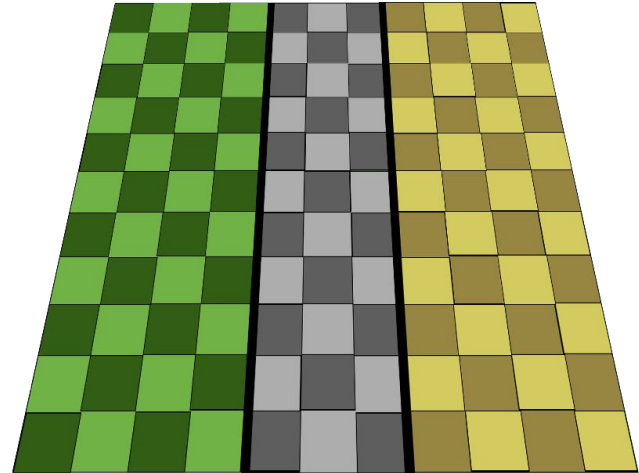
- Spielstatus: Position der Charaktere + vorhandene Ressourcen
- Relevante Daten:
  - Neu gesetzte Figuren
  - Ressourcen
  - Attribute der Figur
  - Wo der Charakter gesetzt wird
- Wird aktualisiert wenn der Client eine Spielfigur setzt
- ConcurrentHashMap mit Positionen
- Server speichert Spiele in HashSet,

# Spielstatus: Hierarchie



# Spielregeln

- Lanes
- Ressourcen
- Spawnbereiche
- Tower / Angreifer
- Verschiedene Charaktere
- Verschiedene Verteidigungsanlagen
- Ressourcen nicht aufsparen
- Basis des Gegners Zerstören
- Nach Zeit Punktesystem



# Technologien

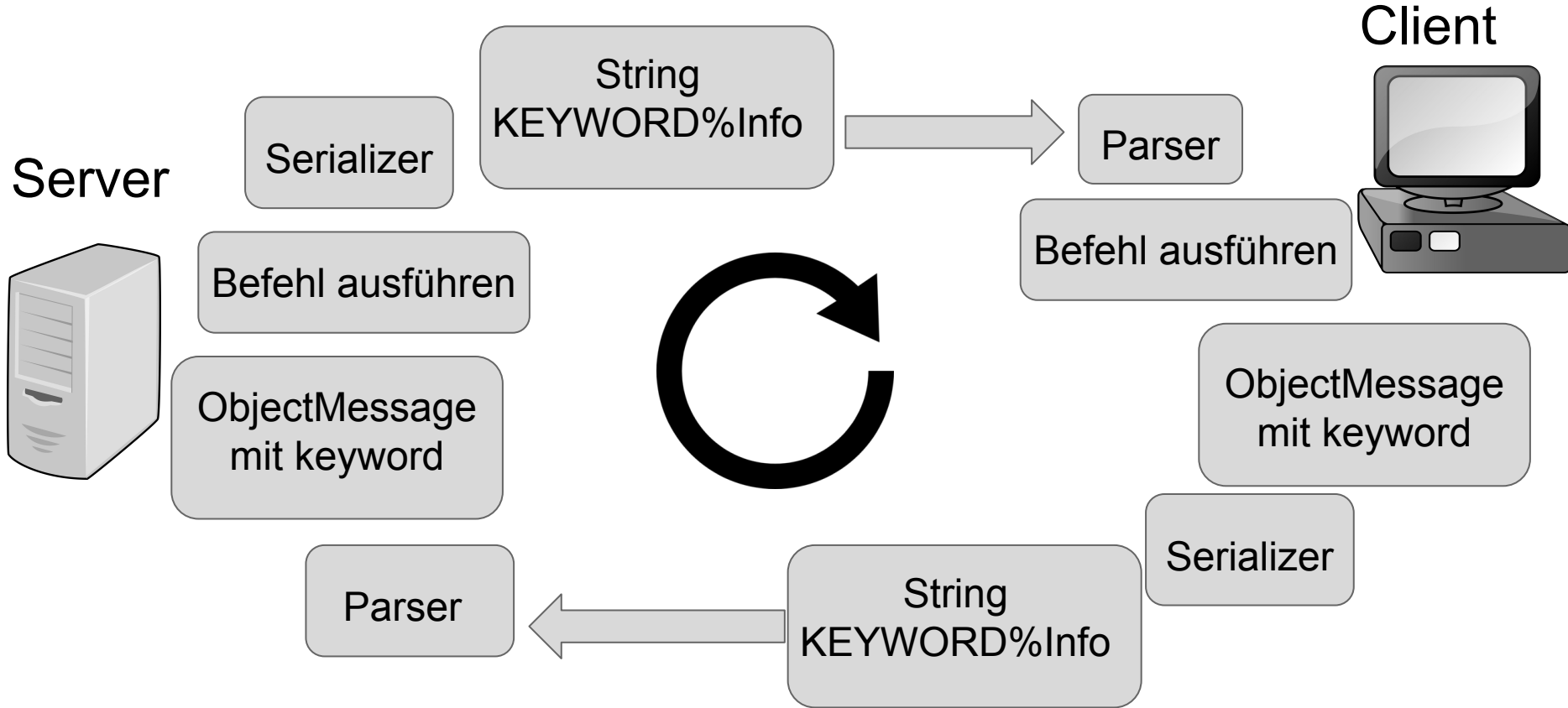
- **JavaFX:**  
Framework zur Applikations-Erstellung  
Für GUI unseres Spieles verwendet
- **JUnit:**  
Testing Framework  
Verwendet für Unit-Tests
- **Regex:**  
Regular Expressions tester
- **Adobe Animate + Illustrator**  
Grafiken

# Netzwerk-Kommunikation

- Für jeden Client wird auf dem Server ein Handler-Objekt erstellt
- Jeder Client und jeder Handler erhalten jeweils ein Message Objekt mit einem Keyword



# Netzwerk-Kommunikation



# Netzwerk-Kommunikation

## Chat:

- Client schickt mit broadcast seine Nachricht + Namen an alle Clients
- Farbe ändert sich wenn Nachricht von einem selbst kommt.
- Whisper Char: Seperator Modus, jeder Client bekommt ID zugewiesen

## Broadcast:

- Liste von output streams, schickt an alle

# Protokol CommandList.java

- **Bestandteile eines Befehls:**

- Keyword

- Seperator: % (nach keyword) danach ein , mehrere Charakter werden mit ; getrennt

- **Beispiele:**

**CHARACTER**%name:mex,xPos:3.5,lane:2,health:10,damage:5;name:trump,xPos:2.5,lane:2,health:10,damage:4;**CHARACTER**%name:trump,xPos:50,lane:4,health:15,damage:8;

**MESSAGE**%Hello, how are you?

**AVAILABLEGAMES**%gameA,gameB

# Protokol CommandList.java

## Server:

IDENTIFY  
WELCOME  
AVAILABLEGAMES  
GETGAME  
NAMECHANGED  
GOODBYE  
PING

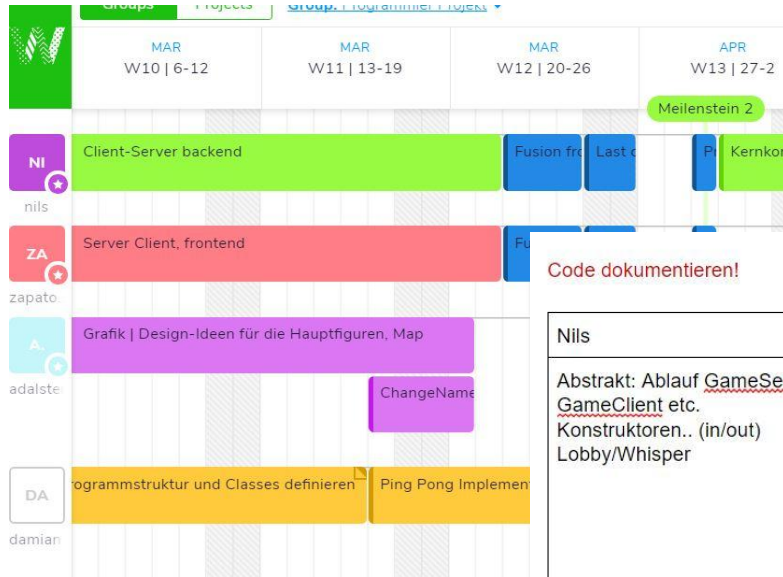
## Client:

IDENTIFYING  
CREATEGAMESERVER  
JOINGAME  
WHISPER  
NEWNAME  
LOGOUT  
PONG

## Beide:

MESSAGE  
CHARACTER

# Arbeitsplan

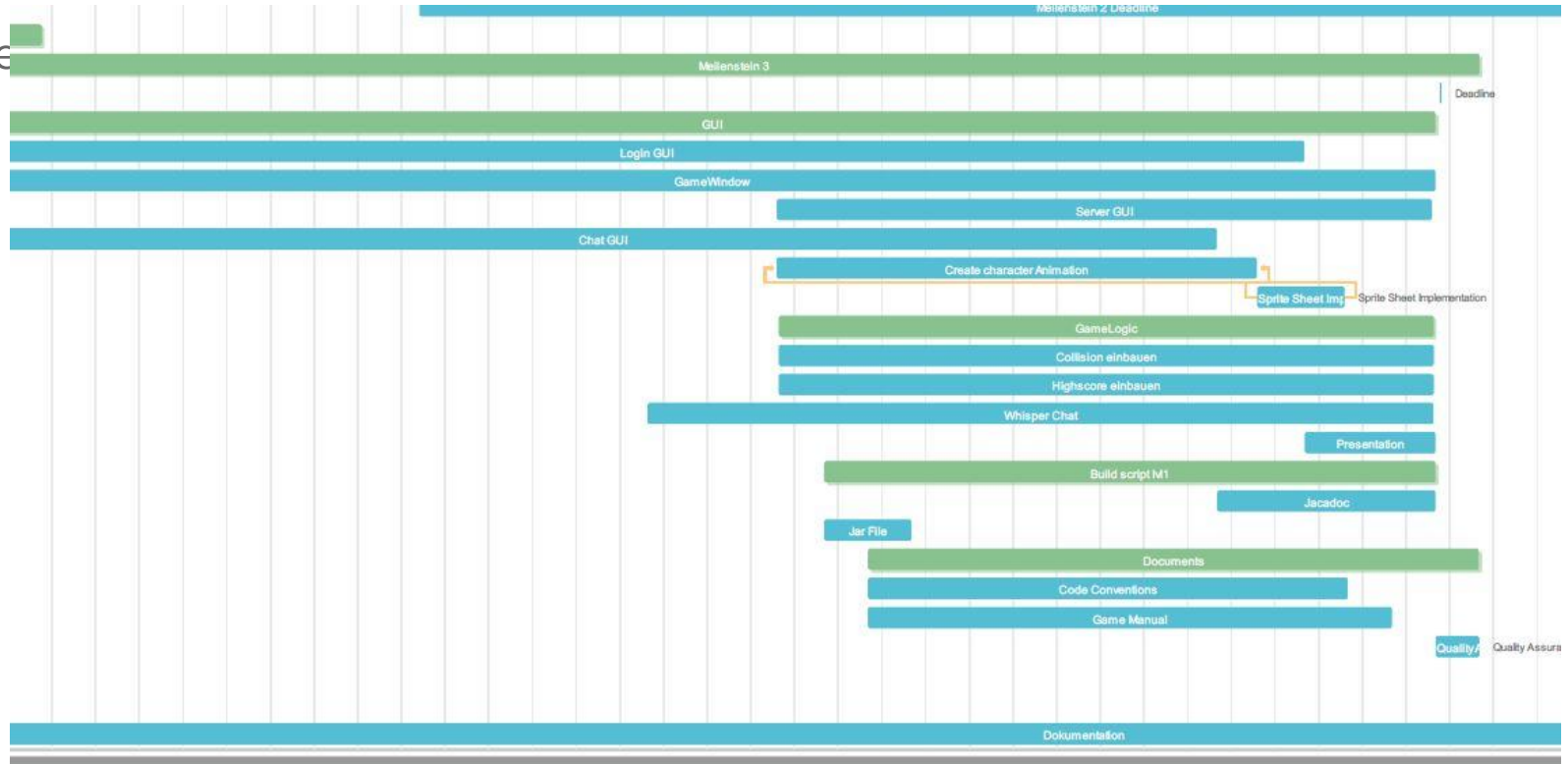


Code dokumentieren!

Nils	Syarif	Adi	Damian
<p>Abstrakt: Ablauf <u>GameServer</u> =&gt; <u>GameClient</u> etc.            Konstruktoren.. (in/out)            Lobby/Whisper</p>	<ul style="list-style-type: none"> <li>• Kollisions Detector Funktion schreiben</li> <li>• Text Area Lobby</li> </ul> <p>GameServer..            Copy paste abstraktes von Nils am richtigen Ort</p>	<ul style="list-style-type: none"> <li>• Sprite sheet implementierung</li> <li>• Manual, Game Tutorial und Ziel des Spiels</li> <li>• Gui verstehen</li> <li>• Dokumentation Enums, Protokoll</li> <li>• Blog</li> <li>• Code Convention</li> </ul>	<ul style="list-style-type: none"> <li>• <u>GameClient</u></li> </ul>

# Neuer Arbeitsplan

Hier



# Qualitätssicherung

- Änderbarkeit

Indikator: Einfach neuen Charakter hinzufügen, anderen Charakter ändern

Masse: 1 Konstruktor mit Attributen + Grafik hinzufügen. 2 Änderungspunkte

- Benutzbarkeit/ Bedienbarkeit:

Indikator: Anzahl Klicks bis zum Spielstart,

Masse: Max Anzahl Klicks bis Spielstart = 5

# Junit Test

- Parser testen
- Gamelogik testen



# Demo

- Server erstellen und chatten,
- Gameserver erstellen und einen Spielzug vorzeigen