

Campus Conquest

Zwischenpräsentation

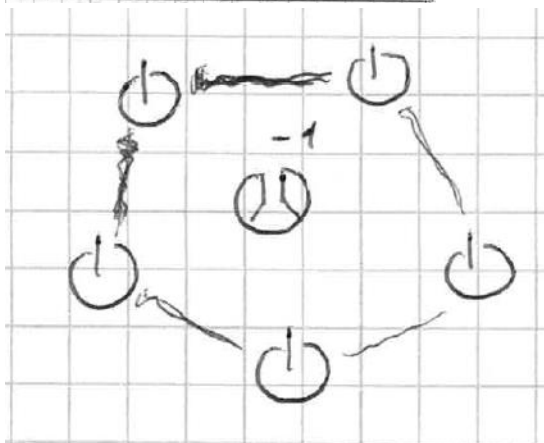
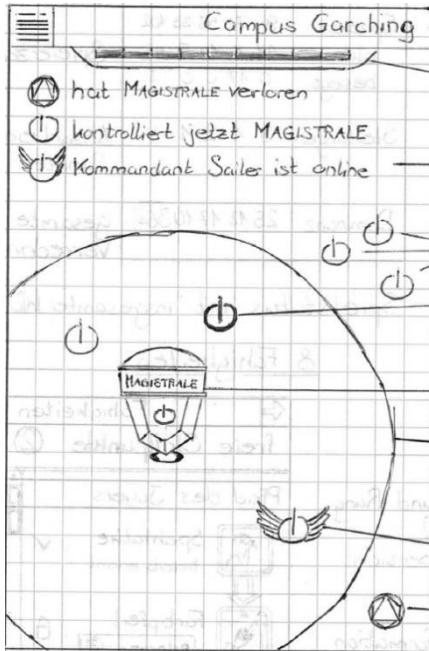
Gruppe 8

Michael Sailer, Paul Preißner, Jonas Mayer,
Benedict Drechsler, Jean-Paul Vieira, Julian Frattini

[IN 0036] Praktikum Social Gaming

Übersicht

- **Gamekonzept** Idee, Inspiration und Hintergrund des Spiels
- **Pipeline** Arbeitsweise und Teamaufstellung
- **Projektplan** Meilensteine, Zeitplan und Entwicklungsoptionen
- **Clientstruktur** Aufgabenbereiche und Layout des Clients
- **Long-Term-Social-Context** Ideen zur Verbesserung des Long-Term Social-Context



Gamekonzept

Inspiration - Problem - Lösung

Inspiration:

Bekanntes Konzept aus RTS und FPS

Neu:

location-based, mobile und soziale Aspekte

Problem:

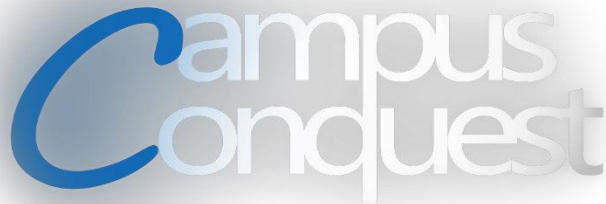
Methode zum Neutralisieren von Gegnern
nicht trivial

Lösung:

Zeichnen von Polygonen aus verbündeten
Spielern

Gamekonzept

Warum es sich gegen die anderen Konzepte durchsetzte



Campus
Conquest

Vorteile in der Entwicklung:

- Überschaubare Menge an Art-Assets
- Map-Design und Design von Fraktionen trivial

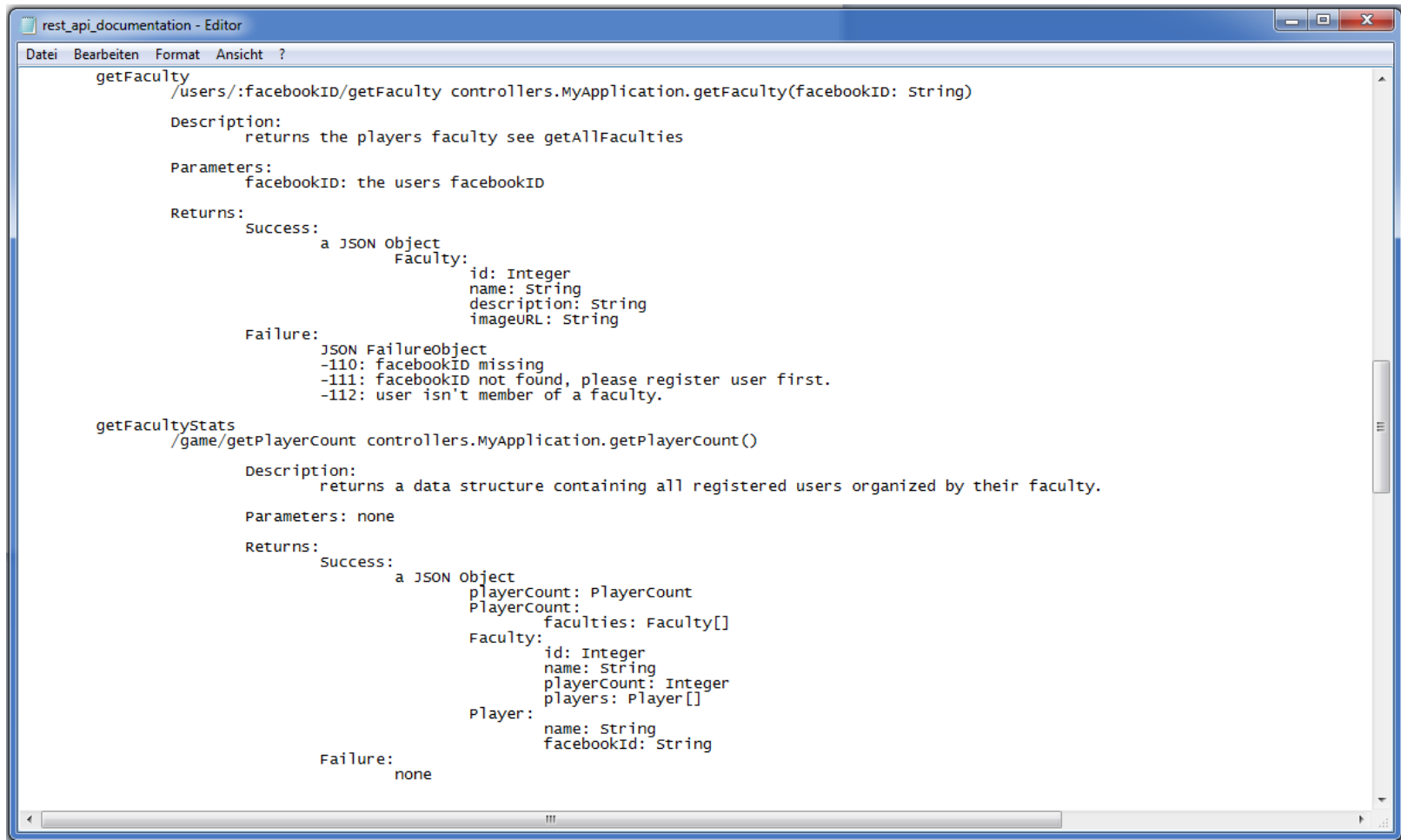
Playability:

- Im Grunde einfach verständliche Spielmechanik
- Gruppeneinteilung fest durch äußere Gegebenheiten
- Langzeitspielmotivation gegeben durch „Rivalität der Fakultäten“



Pipeline





```
rest_api_documentation - Editor
Datei Bearbeiten Format Ansicht ?

getFaculty
/users/:facebookID/getFaculty controllers.MyApplication.getFaculty(facebookID: String)

Description:
  returns the players faculty see getAllFaculties

Parameters:
  facebookID: the users facebookID

Returns:
  Success:
    a JSON object
    Faculty:
      id: Integer
      name: String
      description: String
      imageURL: String

  Failure:
    JSON FailureObject
    -110: facebookID missing
    -111: facebookID not found, please register user first.
    -112: user isn't member of a faculty.

getFacultyStats
/game/getPlayerCount controllers.MyApplication.getPlayerCount()

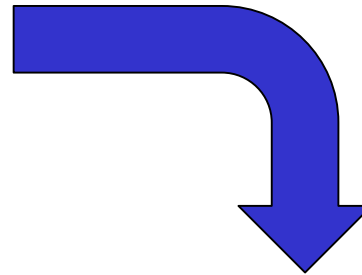
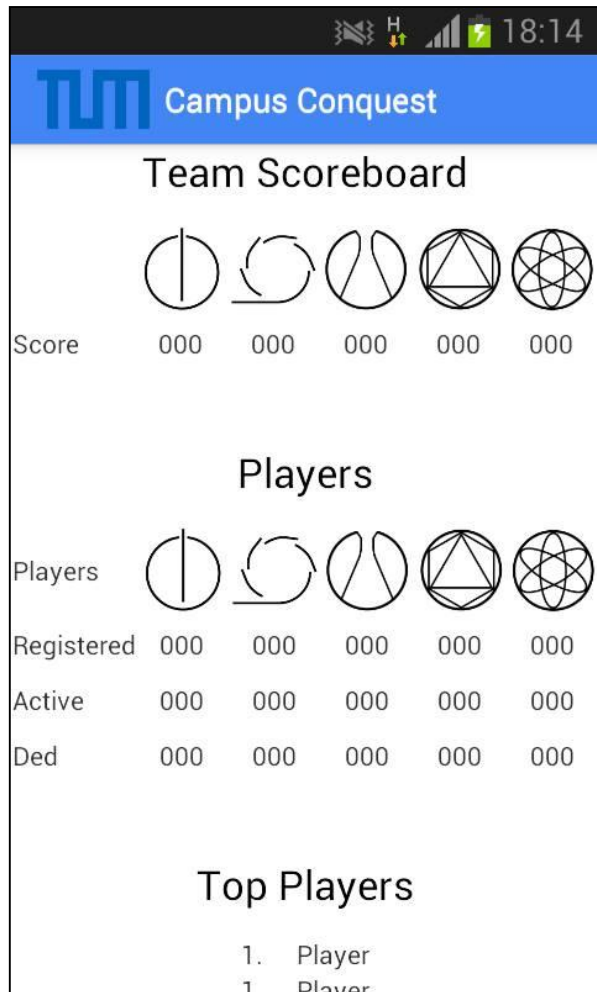
Description:
  returns a data structure containing all registered users organized by their faculty.

Parameters: none

Returns:
  Success:
    a JSON object
    playerCount: PlayerCount
    PlayerCount:
      faculties: Faculty[]
    Faculty:
      id: Integer
      name: String
      playerCount: Integer
      players: Player[]
    Player:
      name: String
      facebookId: String

  Failure:
    none
```

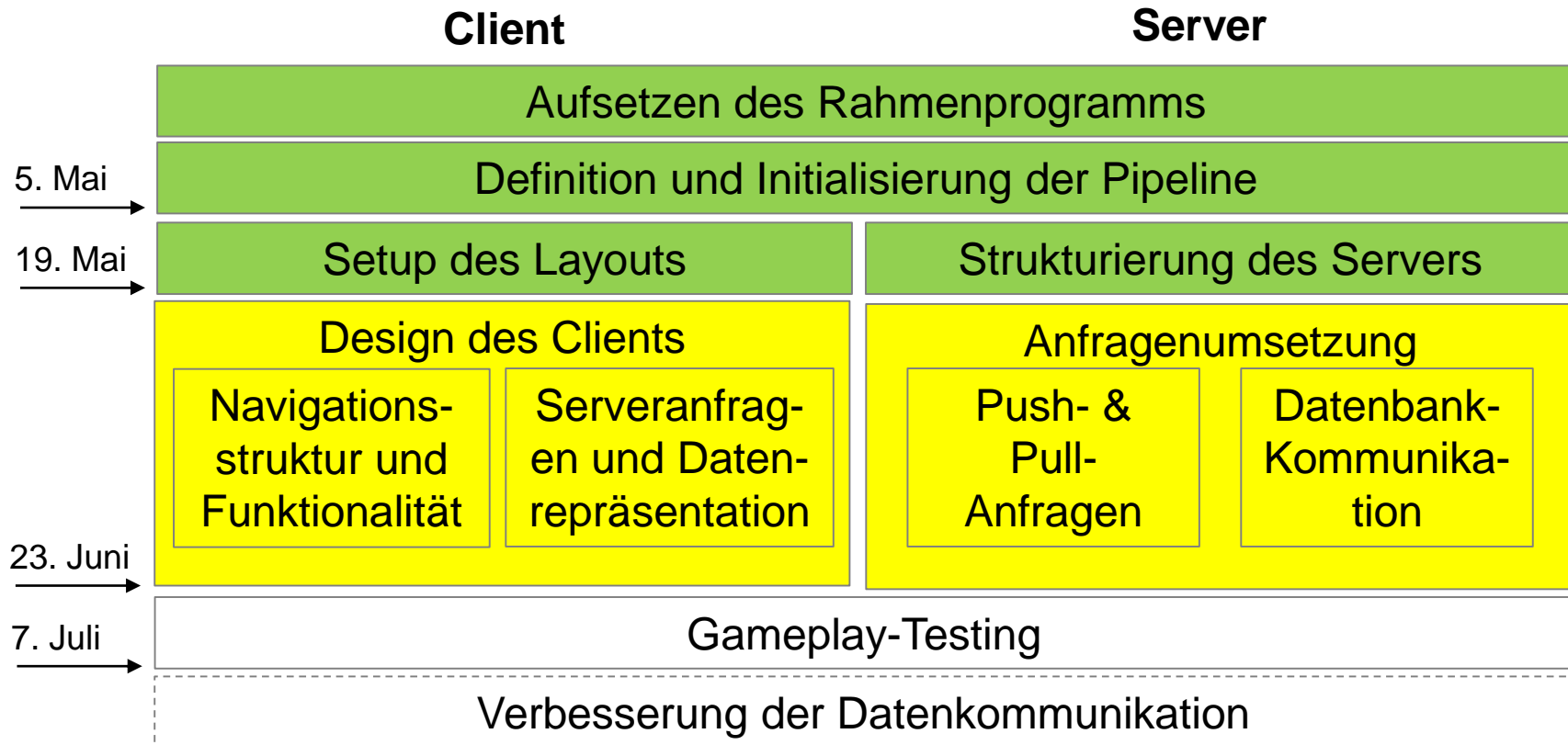
Dokumentation auf Server-Seite



Neue Anforderungen

```
what_we_need - Editor
Datei Bearbeiten Format Ansicht ?
For the fragment "Team Scoreboard" we need a Http get() method. It should return all
values that are needed to fill the Team Scoreboard fragment:
-getTeamScoreBoard (bzw getFacultyStats):
  returns:
    Success:
      a JSON object:
        faculties: Faculty[]
        Faculty:
          id:integer
          score:integer
          players_registered: integer
          players_active: integer
          players_dead: integer
          topPlayers: Players[10]
          facebookID:???
          name:String
For the "Statistics" Fragment we need a Http get() method. It should return all
values that are needed to fill the Statistics fragment:
-getUserStats:
  returns:
    Success:
      a JSON object:
        playtime: Playtime:      hours:integer
                                minutes:integer
        score:integer
        kills:integer //number of kills the user has assisted at
        deaths:integer
        captures:integer //number of flag captures the user has assisted at
        faculty:integer
        polygons:integer //number of polygons the user has drawn
```

Projektplan



Clientstruktur

Tasks/Upload:

- Constantly updated sensor data + player moves/polygons
 - (rarely) new user data

The Interface

- Three main Views (Map, Team, Player)
- Switch between Views through dropdown menu (Navigation Drawer use planned)

The *Thin Client* and how it works

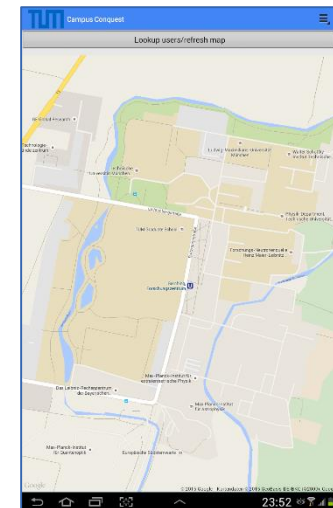
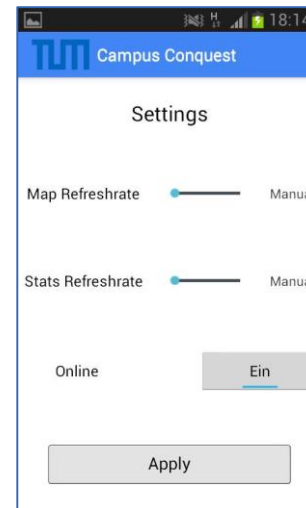
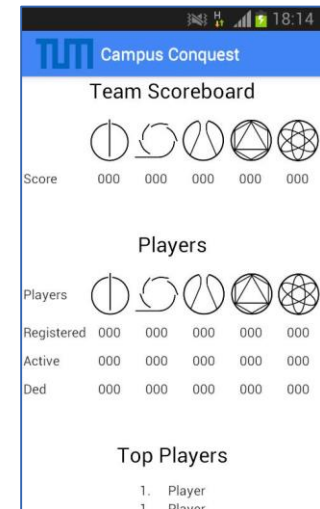
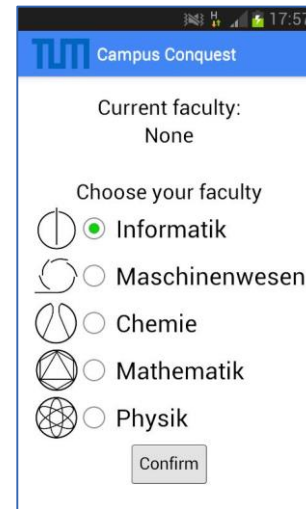
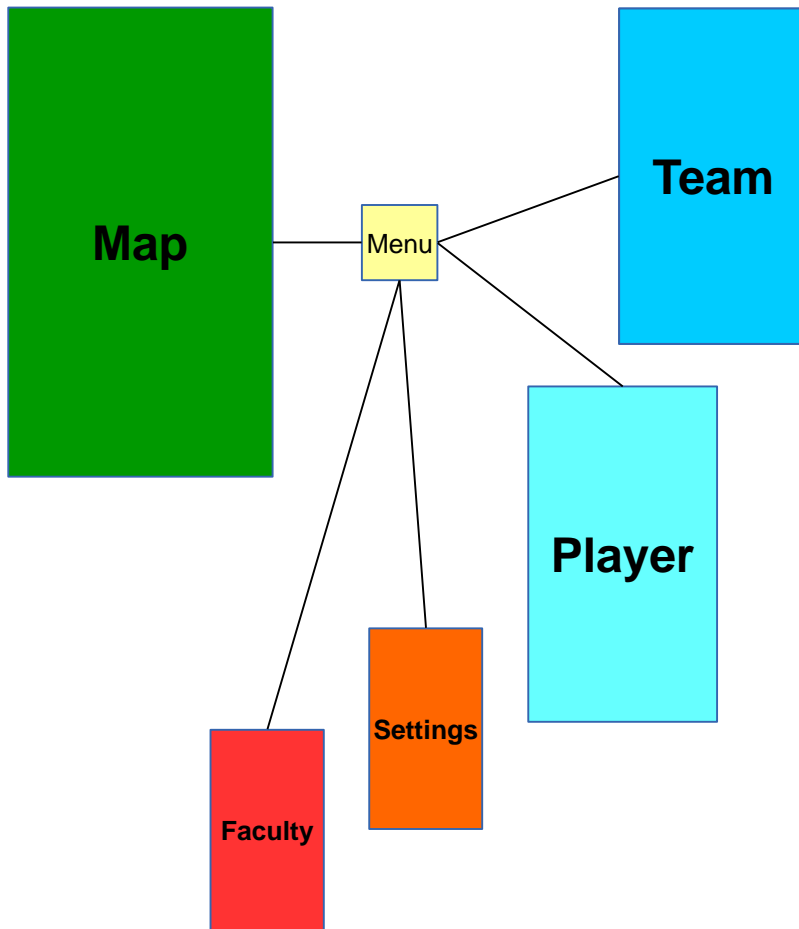
- Login through Facebook API at startup
- Load straight into the Map View
- Upon switching to a different View → server request
- Frequently trade user data with the server

Tasks/Download:

- Own user data and statistics
 - Other players' data
 - Team statistics

Vision (layout)

- Modern design
- Minimal latency
- Fluid navigation



Long-Term-Social-Context

- Verbesserung des Long-Term-Social-Context durch ein Hierarchie-System (vgl. Gilden)
- Untergruppierungen innerhalb der Fakultäten
- Interaktion via Foren, Umfragen, Events, etc.
- Mitglieder-Ranglisten zur Übersicht von Aktivität
- Wettbewerb zwischen den Gruppierungen als Ansporn zum Einsatz für die Fakultät



Vielen Dank
Für ihre
Aufmerksamkeit