1. Funktionale Anforderungen:

* Accountverwaltung
  + Registrierung
    - Nutzername
    - Passwort
  + Einloggen / Ausloggen
  + Spielhistorie (mit Auswertung)
* Komplettes Spiel SNAKE 2v2 ausdenken
  + Spielfeld
    - Begrenzt/nicht begrenzt
  + Schlangen
    - Farbe
    - länge
  + Brotkrumenverhalten
    - Verschnellert/Verlangsamt Schlange
    - Verlängert / Verkürzt Schlange
    - Schutzschild
    - [kreativ werden]
  + Spielende (Signalton erforderlich)
    - Selbstmord (Wurm beißt sich selbst)
    - X-Fache Länge d. Gegnerischen Wurms
    - [kreativ werden]
  + Sonderregeln
    - Würmer treffen sich unter bestimmtem Umstand (Signalton erforderlich)
  + Level Setup
    - Computerwürmer, mehr Powerups, Spielgeschwindigkeit, Verkehrte Steuerung, Spielfeld …

Konzeptionelle Anforderungen

* Server
  + Framework für das Object-Rational-Mapping: ORM Hybernate
  + API Datenaustausch mit JSON: Jackson
  + JSON Web Tokens zu Authentifizierung v. Anwendern
* Client
  + JavaFX verwenden
    - Log-in / Registrierungs- bildschirm
    - Spielbildschirm
    - Spielhistoriebildschirm
* Entwurfsmuster Composite (Power Ups), Factory(Schlangen, Power Ups) und Observer(Highscore)
* Synchrone und Asynchronisation zwischen Server und Client
  + HTTP
* Parallele Verarbeitung
  + Log in
  + Hintergrundmusik (?)

Git verwenden

Meilensteine

1. UML
   1. **Klassendiagramm** als Datenmodell
   2. Use-Case Diagramm für funktionale Anforderungen
   3. Aktivitätsdiagram für Abläufe
2. Schnittstellenbeschreibung Client-Server
   1. Datenaustausch
      1. Login Daten
      2. Spielhistorie
      3. Laufendes Spiel
   2. UML Verfeinern
3. Prototyp Code für Client und Server
   1. Anwendungsfälle realisieren