1. **Vorwort**
2. **Projektplanung**
   1. **Projektvorbereitung HENRY**
   2. **Phasen**
      1. **Menüdesign OMAR**
      2. **Spieldesign TOM**
      3. **Ressourcenplanung HERING**
      4. **Zeitplanung SVEN**
      5. **Qualitätsmanagement OMAR**
3. **Projektdurchführung**
   1. **Vorgehensmodell TOM**
   2. **Verwendete Software HENRY**
   3. **Phasen**
      1. **Pflichtenhefterstellung TOM**
      2. **Benutzeroberflächen-/Datenbank-Design CHINA**
      3. **Datenbankerstellung CHINA**
      4. **Programmierung**
         1. **Benutzeroberfläche TOM**
         2. **Menüsteuerung HERING**
         3. **Spiellogik OMAR**
         4. **Spielfeldgestaltung SVEN**
         5. **Datenbankanbindungen HERING**
      5. **Testphase und Bugfixing SVEN**
      6. **Technische Dokumentation CHINA**
   4. **Zeitmanagement SVEN**
4. **Projektreflexion**
   1. **Fachlicher Soll-/Ist-Vergleich CHINA**
   2. **Zeitlicher Soll-/Ist-Vergleich CHINA No.1**
   3. **Persönliche Reflexion**
      1. **Tom**
      2. **Omar**
      3. **Henry**
      4. **China Ping Pong**
      5. **Sven**
      6. **Herrrrrrrrring**
5. **Anhang**
   1. **Glossar JEDER**
   2. **Quellenverzeichnis JEDER**

**Spiellogik**

Die Spiellogik wird in der Klasse „Settler“ im Package „game.Settler“ festgelegt. Die Klasse ist

abgeleitet von der allgemeinen Spielklasse „Game“.

Die Spiellogik-Klasse steuert den gesamten Spielablauf, vom Initialisieren des Spiels über Ablauf der Spielzüge und Ereignisse bis hin zum Spielende. Diese Klasse ist sehr groß, obwohl alle Funktionen, die das Fürstentum eines Spielers betreffen, bereits in eine eigene Klasse „Princedom“ ausgelagert wurden, die allerdings ebenfalls sehr umfangreich geworden ist. Grund für die sehr großen Spiellogik-Klassen ist das umfangreiche und sehr detaillierte Regelwerk des Spiels.

**Qualitätsmanagement**

Zur Erreichung der angestrebten Codequalität mit hohem Modularisierungsgrad haben wir uns im Vorfeld auf einen Qualitätssicherungsworkflow (Abb. 2, Seite 6) geeinigt, um den Code aller Projektmitglieder einer Prüfung unterziehen zu können. Dabei wird im speziellen auf folgende qualitätsschiernde Maßnahmen geachtet:

* Hoher Modularisierungsgrad.
* Hoher Abstraktionsgrad(Portabilität).
* Einhaltung des HHBK-Standards.

Code wird in GIT Featurebranch gepusht

Merge Request wird gestellt

Merge durchführen

Request ablehnen und Feedback an Entwickler

**If**

Code-Qualität Ok ?

**Menüdesign:-**

Am Anfangen haben wir das Startmenü erstellt, womit man sich anmelden kann als neuen Spieler oder schon gespielt hat.

In dem *Benutzerregistrierungsmenü kann man sich als neuen Spieler anmelden mit dem Nutzername, Nachname, Vorname, und wiederholenden Passwort.*

*Wenn man sich angemeldet hat, kann man seinen Benutzername bei dem Benutzeranmeldmenü eingeben.*

*Mit dem Hauptmenü kann man sich abmelden, die Regeln lesen, sich die Bestenliste angucken das Spiel beenden, und das Spiel starten.*

*Unter die Regeln Menü findet man die Regeln des Spieles.*