

Software-Qualitätssicherung

Product Management

Für unser Projektmanagement verwenden wir OpenProject. Dort nutzen wir Timelines, um den Fortschritt zu visualisieren, und organisieren die Aufgabenteilung direkt innerhalb der Timelines. Dies stellt sicher, dass alle Teammitglieder den aktuellen Entwicklungsstand und ihre jeweiligen Aufgaben kennen.

Zusätzlich setzen wir auf regelmäßige Meetings, um den Entwicklungsprozess effizient zu steuern und mögliche Probleme frühzeitig zu erkennen.

Versionskontrolle & Code-Management

Unsere Versionskontrolle erfolgt über GitLab. Um eine saubere und strukturierte Entwicklung sicherzustellen, verwenden wir ein Branching-Modell (z. B. Git-Flow oder Feature-Branched). Dadurch kann jedes Teammitglied ungestört an einzelnen Features oder Bugfixes arbeiten, bevor die Änderungen in den Hauptzweig integriert werden.

Jede Änderung wird erst nach Code-Reviews und Merge Requests in den Hauptzweig übernommen, um Fehler frühzeitig zu erkennen und die Code-Qualität zu gewährleisten.

Code-Reviews & Testing

- Code-Reviews: Jedes Feature oder jeder Fix wird vor dem Merge von mindestens einem weiteren Teammitglied überprüft. Dies verbessert Code-Qualität, Lesbarkeit und Wartbarkeit.
- Automatisierte Tests: Wir planen, Unit-Tests für kritische Spielmechaniken zu implementieren, um Fehler frühzeitig zu erkennen.

- Manuelles Testing: Zusätzlich testen wir regelmäßig Gameplay, UI und Performance, um ein optimales Spielerlebnis sicherzustellen.
- Coding-Style: Wir verwenden eine angepasste Version von Google-Style für Java. Wir versuchen, Verschachtelungen zu vermeiden, um die Code-Lesbarkeit zu gewährleisten.

Tools (Geplante Umsetzung)

Um die Code-Qualität weiter zu verbessern, planen wir folgende Tools einzusetzen:

SonarQube

- SonarQube wird in einem lokalen Docker-Container gehostet.
- Es analysiert unseren Code und liefert Metriken zur Wartbarkeit, insbesondere in Bezug auf die Größe und Komplexität einzelner Klassen.
- Dadurch lassen sich potenzielle Probleme frühzeitig erkennen und die langfristige Code-Qualität verbessern.

log4j2

- Zum Logging verwenden wir log4j2, um eine detaillierte Fehleranalyse zu ermöglichen.
- Die Logs werden in Dateien gespeichert, die zur Fehlerdiagnose genutzt werden können, z. B. zur Überprüfung, ob die Protokoll-Kommandos korrekt gesendet und empfangen wurden.
- Dies erleichtert das Debugging und die Nachverfolgbarkeit von Fehlern.

JaCoCo

- JaCoCo wird als Plugin integriert, um die Testabdeckung (Code Coverage) zu messen.
- Dies zeigt uns, welche Teile unseres Codes durch Tests abgesichert sind und wo noch Lücken bestehen.
- Eine hohe Testabdeckung trägt zur Zuverlässigkeit und Wartbarkeit der Software bei.

Fazit

Unser Ansatz zur Software-Qualitätssicherung kombiniert effizientes Projektmanagement, strukturierte Code-Verwaltung, regelmäßige Reviews und gezieltes Testing.

Mit den geplanten Maßnahmen und Tools stellen wir sicher, dass unsere Software stabil, performant und wartbar ist, ohne die Entwicklung unnötig zu verlangsamen.