

Konzept 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:

Unser Version-Control erfolgt über GitLab. Um eine saubere und strukturierte Entwicklung sicherzustellen, werden wir ein Branching-Modell verwenden (z. B. Git-Flow oder Feature-Banches). 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 Nestings zu vermeiden, um die Code-Lesbarkeit zu gewährleisten.

Fazit

Unser Ansatz zur Software-Qualitätssicherung kombiniert effizientes Projektmanagement, strukturierte Code-Verwaltung, regelmäßige Reviews und gezieltes Testing. Ziel ist eine stabile, performante und wartbare Software, die ein reibungsloses Spielerlebnis gewährleistet, ohne die Entwicklung unnötig zu verlangsamen.