

Protokoll: 01.06.25

Erstellt von: Carolin Scheffler

Inhaltsverzeichnis

1 Daily	2
1.1 Grand Koordinator	2
1.2 Master Gitlab	2
1.3 Arch of Architektur	2
1.4 Double Doku	2
2 Besprochenes	2
3 Github Branch hinter der Main	2
4 Github origin weiter als lokal auf aktuellem Branch	2
5 Methoden mit MoveTiles sind im Konflikt	3
6 Wie testen Components	3
7 Testklassen mit Entitäten erstellen	3
8 PullRequest	3
9 Multiplayer Issues	3
10 TODO:	3
10.1 Grand Koordinator	3
10.2 Master Gitlab	4
10.3 Arch of Architektur	4
10.4 Double Doku	4
10.5 Alle sind Robust	4
11 Fragen an Iglar	4

1 Daily

1.1 Grand Koordinator

- ProgressBar, ChangeMountainLayer Review
- NEUES: HealthBar, durchsichtige Berge

1.2 Master Gitlab

- Methode für Request vom Client
- Client sendet Request, die im Server ankommt

1.3 Arch of Architektur

- APComponent weiterimplementiert

1.4 Double Doku

- eine main.yml die testete ob auf die Main pusht
- eine debug.yml die testet ob Debug im code steht
- eine Testmethode für CLient Request
- Ordnerstruktur und vorgehen zum schreiben von Testklassen

2 Besprochenes

3 Github Branch hinter der Main

In IntelliJ auf Remote;Main;merge "main"into "lokal-branch"und dann in der Maske merge...

In jeder Datei klicken was rein soll und was raus soll.

4 Github origin weiter als lokal auf aktuellem Branch

Falls wir zusammen auf einem branch arbeiten. Vor dem push "branch update". Ansonsten workaround neuer Branch.

5 Methoden mit MoveTiles sind im Konflikt

Problem: Es wurde auf zwei unterschiedliche Weise auf MoveTiles Zugriffen, das wird noch einheitlich zwischen Nico und Burak besprochen.

Entscheidung:

6 Wie testen Components

Problem: Components die funktionieren müssen nicht getestet werden. Components welche erweitern und die Funktionalität einer weiteren Components einschränken oder verändern, sollten die benutzen components auf ihre Funktionalität testen. **Entscheidung:** Zu mocken macht keinen Sinn. Änderungen könnten getestet werden. Wenn Test nicht zu viel Aufwand und manuell ausgeführt.

7 Testklassen mit Entitäten erstellen

Entitäten sind den Componenten bekannt, aber nicht den Testklassen. So können Componenten nicht getestet werden.

Können Services getestet werden.

Ki Tools können uns die Teststruktur liefern, Probleme von oben bestehen jedoch und müssen anders gelöst werden.

Entscheidung: Konzentration auf ServerTests

8 PullRequest

Es wurden alle abgearbeitet und kurz besprochen wie wir vorgehen. Github Meister bittet: nur dann wirklich PR, wenn Änderung final sein soll.

9 Multiplayer Issues

Passen in ihrem jetzigen Konzept nicht zusammen. **Entscheidung:** Werden neu geschrieben. Müssen darauf Roadmap und Zeitplan, sowie Sprint3 darauf anpassen.

10 TODO:

10.1 Grand Koordinator

- Rundenmodus lokal
- MoveTiles synchronisieren mit Burak

10.2 Master Gitlab

- Serverimplementation anfangen. Client stellt Request, Server antwortet.
- Multiplayer Issues zerpfücken

10.3 Arch of Architektur

- HP in der UI sichtbar
- MoveTiles synchronisieren mit Nico

10.4 Double Doku

- yml auch force push?
- KI Tool sinnvoll?
- wie bekomme ich die Entitäten in die Tests?
- welche Services kann ich testen?
- Multiplayer Issues zerpfücken

10.5 Alle sind Robust

11 Fragen an Iglar