Protokoll 02.11.18

Beginn: 9:15

Ende: 10:45

Einführung:

- Organisatorisches von Thomas:

- Bewertung in Sternen

- In Relation zu anderen

- 2 oder weniger Sterne muss erneut abgegeben werden

- Wir haben 4 Sterne erhalten

- Makel:

- PNG und keine Vektorgrafik

- Erklärungen zu kurz

- Projektplan stimmte nicht mit Beschreibung überein

- Einführung von Rene

- Vorstellung von Jan zu GIT

- git clone um Repository auf PC zu laden

- git status um Status anzuzeigen

- git add zum hinzufügen

- git commit (Datei) -m " Beschreibung in englisch"

- git push mit GIT zu synchronisieren

- git pull lädt alle Änderungen in lokales Repository runter

Git Craken zum Lösen von Konflikten

- git branch (Branchnamen) bei Erstellung wird alles vom aktuellen Branch mitgenommen

- git checkout (Branchname) setzen neuen Branch als aktiven Branch

- git push origin (Branchname) Branch wird in Origin gepusht

- git tag erstellt tag

- git push origin (tagname) lädt tag ins GIT hub hoch

- Vorstellung von Lukas zur Nutzung von Taiga und GIT

- Slack chanel für Taigaupdates erstellt

- Informationen von Git -> Taiga -> Slack

- master branch protected release branch / öffentliche Version

- development protected

- Review wird von QA und Product Owner bearbeitet

- Houman macht Einführung in Maven

- Siehe Dokument Maven.pdf in Taiga

- Lukas beginnt mit dem Einfügen der User Stories in Taiga

- Spieler Epic erstellt: Beschreibung zu späterem Zeitpunkt

- Bei Erstellung von Epics Namen Farbei und Status einstellen (Status auf New)

- Status mit Vorsicht genießen nie ohne Review auf Done setzen

- erstellen User Stories und zugehöriger Tasks analog zu Epic Erstellung

- Issues mit Namen zuordnen oder mit Tags markieren

- Beispielhafte User Storie hinzugefügt

- Abuse und Misuse cases als User Story mit [ABUSE] / [MISUSE] kennzeichnen

- Alle Gruppen sollen ihre Epics und User Stories eintragen

-