

Das Spiel lebt

Progress Report

</> Reviewing Jars we receive	● Milestone 4	Medium
</> WinState und mainMenu	● Milestone 4	Low
</> GUI verbessern (JavaFX)	● Milestone 4	High
</> Grafiken für GUI	● Milestone 4	High
</> High Scores einfügen und globale Liste führen	● Milestone 4	Medium
</> Unit Tests schreiben und ausführen	● Milestone 4	High
</> Connection Loss: Reconnect	● Milestone 4	Low
</> implement useEntity Methods (notes on txt)	● Milestone 4	Medium
</> Fertigstellen der Verarbeitung von Network Befehlen ClientSide	● Milestone 4	High
</> Überprüfen von Funktionalität (build files, commands, chat, gameLogic ru	● Milestone 4	Low
</> Optimisierung für Nicht-Funktionale Anforderungen (z.B. Ladezeiten)	● Milestone 5	Medium
</> "Marketing" (Screenshots, Videos, etc.)	● Milestone 5	High
</> Playtesting (auch außerhalb der Entwicklergruppe)	● Milestone 5	High
</> Alle Docs aktualisieren (Überwachen)	● Milestone 5	Low
</> QA Report	● Milestone 5	Medium
</> Bonus Points (Website, Cheatcodes, Mods, etc.) → tbd	● Milestone 5	Low
</> Präsentation erstellen 3	● Milestone 5	Low
</> External Libraries managed by gradle via maven central	● Milestone 5	Medium
</> Jar-Export und Testen auf Uni-Server	● Milestone 5	High
</> Unit Tests überprüfen	● Milestone 5	Low

Rules to Code

I *GameLogicInterface*

- `startGame(players: String[])`
- `endGame()`
- `nextTurn()`
- `endTurn()`
- `buyTile()`
- `buyStatue()`
- `buyStructure()`
- `placeStructure()`
- `useStructure()`
- `upgradeStatue()`
- `useStatue()`
- `useFieldArtifact()`
- `usePlayerArtifact()`



Technology!



Design Pattern:

Observer

Mediator

Factory

Builder



Libraries:

Mockito

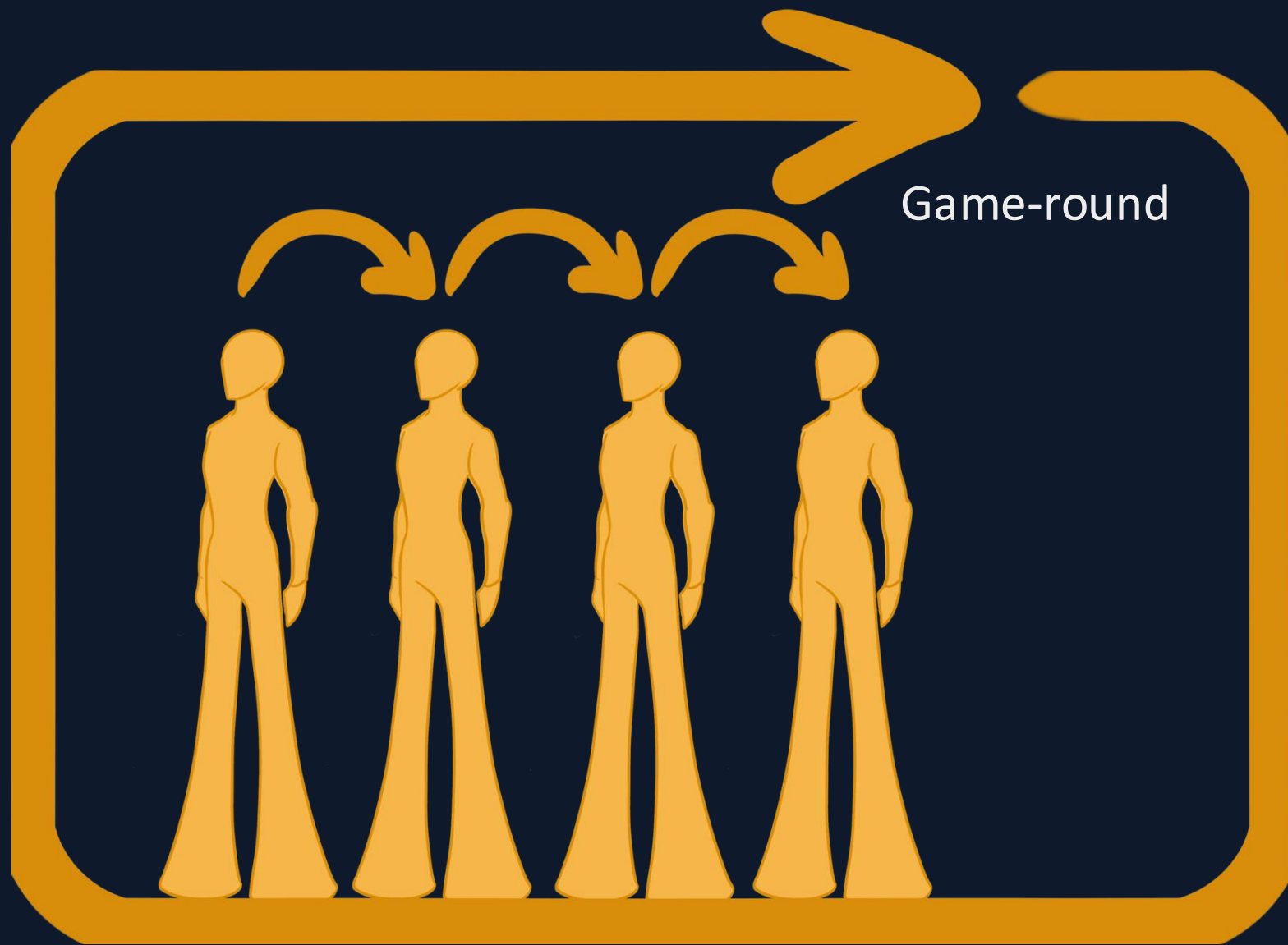
Gson

QA



About the Game

Player-Round





Demo

Next Level: Wir sind bereit