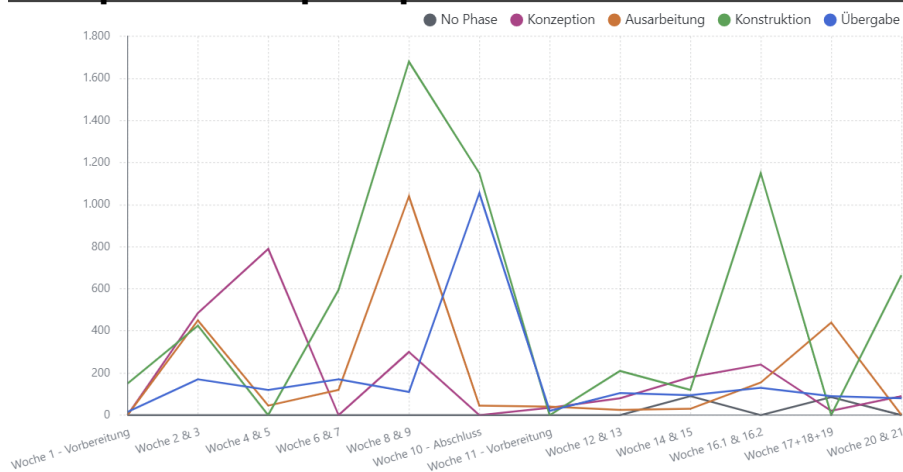




Statistik

Teammitglied	Arbeitsaufwand	Hauptbeitrag
Nunn, Jannik	38 Stunden	Entity Event Handling
Nuss, Kai	59 Stunden	Persistieren der meisten Daten
Schnur, Tim	95 Stunden	Entwicklung des Frontend
Solfronk, Felix	41 Stunden	Input Handling
Steinbring, Christian	110 Stunden	Entwicklung des Rendering

Zeit pro Phase pro Sprint



Phasen

Konzeption:
37 Stunden

Ausarbeitung:
40 Stunden

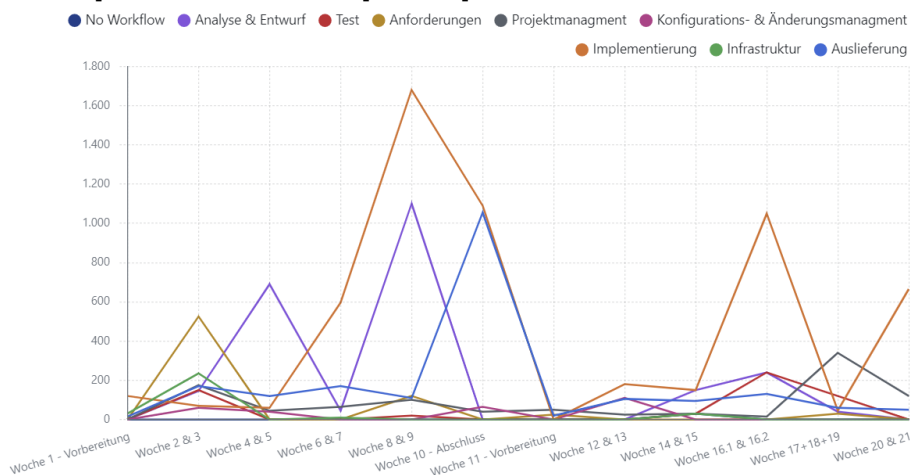
Konstruktion:
102 Stunden

Übergabe:
36 Stunden



Statistik

Zeit pro Workflow pro Sprint



Workflows

Anforderungen:
11 Stunden

Analyse & Entwurf:
40 Stunden

Implementierung:
95 Stunden

Test:
34 Stunden

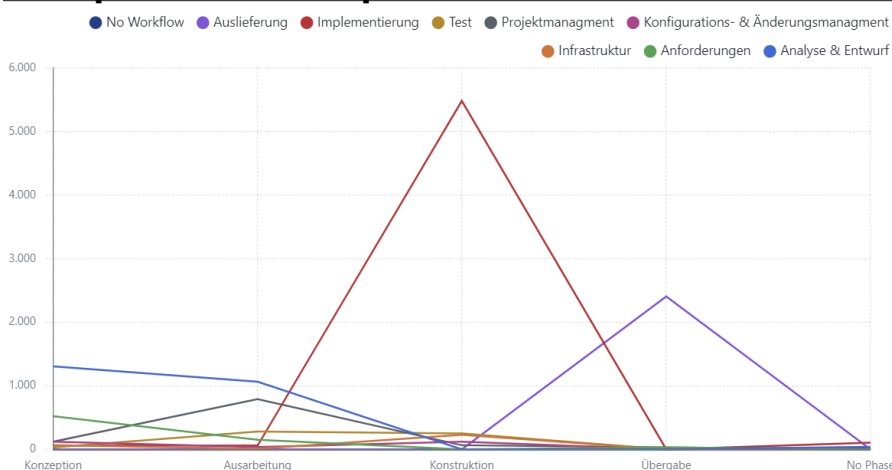
Auslieferung:
35 Stunden

Änderungsmanagement:
5 Stunden

Projektmanagement:
17 Stunden

Infrastruktur:
5 Stunden

Zeit pro Workflow pro Phase



Fun Fact

Wir haben 2.500 Minuten (das sind fast 17 % unserer Arbeitszeit) benötigt, um nach außen zu anderen Teams und dem Dozenten zu kommunizieren.



Highlights

Projektvision und Ziele

Es soll eine Videospielengine und ein Videospieleditor entwickelt werden, die das einfache Entwickeln von Videospielen ermöglichen

Verwendete Softwaretools, Plattformen und Techniken

- Entwicklung der Software mit **Maven**
- Distribution des Quellcodes unterhalb des Entwicklungsteam mittels **GitHub**
- Planung und Management mittels **GitHub-Projects**
- Realisierung des Test-Driven-Development über Unittests durch **JUnit**

Tests

- Es wurden **62 Unit-Test** durchgeführt
- Dabei wurden **2 Fehler** gefunden mit einem durchschnittlichen **Schweregrad 2**
- Die durchschnittliche Zeit zur Fehlerbehebung beträgt **4 Minuten**
- Insgesamt wurde eine Testabdeckung von **25 %** erreicht



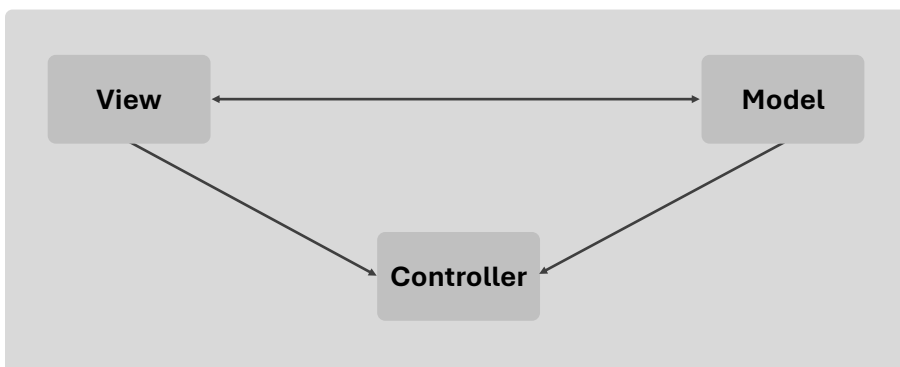
Highlights

Metrik	Wert (15.05.2024)	Wert (13.06.2024)
Anzahl Codezeilen	~ 8.900	~ 11.700 ▲
Checkstyle Fehler	1.900	439 ▼
Durchschnittliche zyklomatische Komplexität	11,6	11,1 ▼
Anzahl Abhängigkeiten anderer Bibliotheken	1	1 ▬

CI / CD

- GitHub Actions CI-Pipeline baut das Projekt und führt Unit-Tests durch
- Teststand wird mittels einer Status-Badge in der README angezeigt

Architektur



Argumente

- Einheitliche Trennung zwischen Darstellung und Geschäftslogik
- Leichtere Weiterentwicklung der Software
- Unterstützt die Erzeugung übersichtlicher Quellcodes