



Semesterprojekt

Implementierung eines Brettspiels (inklusive computergesteuerter Spieler)

Wintersemester 18/19

Einführungsveranstaltung

Patrick Schäfer

patrick.schaefer@hu-berlin.de

Ulf Leser

leser@informatik.hu-berlin.de

Semesterprojekt

- Kernidee: Erstellen einer Software mittlerer Komplexität im Team über ein komplettes Semester hinweg
- Ziele:
 - Arbeiten in Teams: Gruppendynamik, Meetings, Überflieger, ...
 - Softwareentwicklungsprozess von Anfang bis Ende miterleben
 - Anwenden von Tools und Methoden der professionellen Softwareentwicklung
- Zu erwartender Aufwand:
 - 12 LP; 25 bis 30 Zeitstunden pro LP (<https://hu.berlin/creditpoints>)
 - ergibt (bei insgesamt 15 Wochen) über **20 Zeitstunden pro Woche**
 - gleichwertig mit Bachelorarbeit (abzüglich der Verteidigung)
 - in etwa die Hälfte des Gesamt-Arbeitsaufwandes des Semesters
 - mehrere Treffen pro Woche zwingend notwendig

Semesterprojekt

Modul SP: Semesterprojekt		Leistungspunkte: 12	
Lern- und Qualifikationsziele: Studierende üben die Fähigkeit, sich in einem Team in neue Problemstellungen einzuarbeiten, indem sie ein komplexes System entwickeln, welches eine gegebene Aufgabenstellung löst. Hierdurch schulen sie zugleich ihre Fähigkeit, Hard- und/oder Software zu entwickeln, zu testen und zu dokumentieren, sowie die Ergebnisse in geeigneter Form zu präsentieren. Sie üben die praktische Anwendung von Techniken des Software Engineering und erlangen Kenntnisse über die typischen Probleme bei Projekten mit mehr als zwei Beteiligten. Sie erhalten die Fähigkeit zur selbstkritischen Präsentation des Erreichten und der vorgenommenen Entscheidungen.			
Voraussetzungen für die Teilnahme am Modul: Erfolgreicher Abschluss der Module „Grundlagen der Programmierung“ und „Software Engineering“ oder vergleichbare Leistungen. Einzelne Projekte können, soweit vor Veranstaltungsbeginn angekündigt, Kenntnisse über spezielle Lerninhalte voraussetzen.			
Lehrveranstaltungsart	Präsenzzeit in SWS, Workload in Stunden	Leistungspunkte und Voraussetzung für deren Erteilung	Themen, Inhalte
Projektseminar	4 SWS 45 Stunden Präsenz, 315 Stunden Projektarbeit und spezielle Arbeitsleistungen	12 LP, Vorstellung von Zwischenstand und Endergebnissen in Vorträgen (i.d.R. ca. 20 Minuten) sowie Projektzwischen- und Abschlussberichte (ein angemessener Textumfang für einen Bericht kann entweder durch die/den Lehrenden vorgegeben werden [i.d.R. max. 40 Seiten] oder seine Festlegung ist Teil der zu erbringenden Arbeitsleistung)	Planung, Organisation und Durchführung von Systementwicklungen in Teams, unter Anleitung durch und in Interaktion mit Lehrenden. Anhand einer konkreten, für die Studierenden neuen Problemstellung sollen Modellierung, Entwurf, Implementierung und Test eines komplexen Programms „erfahren“ werden.
Dauer des Moduls	<input checked="" type="checkbox"/> 1 Semester <input type="checkbox"/> 2 Semester		
Beginn des Moduls	<input checked="" type="checkbox"/> WS <input checked="" type="checkbox"/> SS		

Zug um Zug Europa / Semesterprojekt WS 16/17



Existierende Steam- / iOS-Version



<https://itunes.apple.com/us/app/ticket-to-ride/id432504470?mt=8>

<http://store.steampowered.com/app/108200/>

Heute

13:15 bis 13:30	Einführung in das Semesterprojekt
13:30 bis 13:45	Einführung der Spielregeln
13:45 bis 14:45	Zug um Zug spielen <ul style="list-style-type: none">• Erfassung Teams / Termine
14:45 bis 15:15	Pause / Überlauf
15:15 bis 15:30	Einführung in Game Development
15:30 bis 16:00	Einführung in Scrum
16:00 bis 16:45	Scrum Kickoff: <ul style="list-style-type: none">• Scrum-Teams bilden• Trello vorstellen• Backlog-Grooming

Semesterprojekt WS 16/17



Semesterprojekt WS 16/17: Lessons Learned

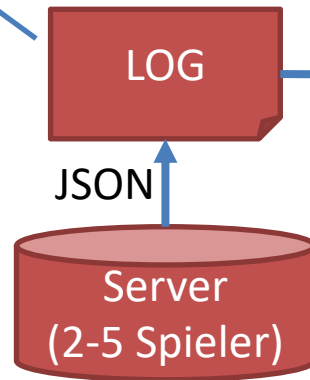
- Gut
 - Scrum
 - GUI-/KI-Wettbewerb
 - Unity
- Nicht so gut
 - Größere Probleme in der JavaFX-Gruppe
 - Prototypen nicht kompatibel
 - Großer Stress bei Gewinner-Team
 - Großer Frust bei anderen Teams
- Daher diesmal:
 - Alle programmieren gegen die gleiche API

Anforderungskatalog

- Implementierung des Brettspiels „Zug um Zug Europa“
- Wir stellen einen Spiele-Server + Protokoll zur Verfügung
- 1. Meilenstein:
 - **Observer-Modus:** grafische **Spieloberfläche** mit Spielbrett, Handkarten, aufgedeckten Karten, etc.
 - **Single-Player-Modus:** Client/Server, vollständige Unterstützung sowie Sicherstellung des Einhaltens aller **Spielregeln**
- 2. Meilenstein:
 - **KI-Wettbewerb:** unterschiedlich agierende **KI-Spieler (einer pro Gruppe)**
- wünschenswerte weitere Features:
 - Sound
 - Animationen
 - Android-Client?
 - ...?

Anforderungskatalog

```
[{"type": "Info", "player": "Blue", "success": true, "turnType": "Join", "playerName": "Q"}, {"type": "Info", "player": "Red", "success": true, "turnType": "BoardState", "destinationTicketsCount": 24, "d"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "BoardState", "destinationTicketsCount": 24, "d"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "ClaimDestinationTickets", "drawCards": [{"c"}, {"type": "Info", "player": "Red", "success": true, "turnType": "BoardState", "destinationTicketsCount": 25, "d"}, {"type": "Info", "player": "Red", "success": true, "turnType": "ClaimDestinationTickets", "drawCards": [{"c"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "BoardState", "destinationTicketsCount": 26, "d"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "ClaimRoute", "dt": "OklabonCity", "dd": "Oall"}, {"type": "Info", "player": "Red", "success": true, "turnType": "BoardState", "destinationTicketsCount": 26, "d"}, {"type": "Info", "player": "Red", "success": true, "turnType": "DrawPassengerCars", "drawCard": "Rainbow", "f"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "BoardState", "destinationTicketsCount": 26, "d"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "DrawPassengerCars", "drawCard": "Rainbow", "f"}, {"type": "Info", "player": "Red", "success": true, "turnType": "BoardState", "destinationTicketsCount": 26, "d"}, {"type": "Info", "player": "Red", "success": true, "turnType": "ClaimRoute", "dt": "NewYork", "dd": "Boston", "p"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "BoardState", "destinationTicketsCount": 26, "d"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "DrawPassengerCars", "drawCard": "Red", "f"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "DrawPassengerCars", "drawCard": "Red", "f"}, {"type": "Info", "player": "Red", "success": true, "turnType": "BoardState", "destinationTicketsCount": 26, "d"}, {"type": "Info", "player": "Red", "success": true, "turnType": "DrawPassengerCars", "drawCard": "Orange", "f"}, {"type": "Info", "player": "Red", "success": true, "turnType": "DrawPassengerCars", "drawCard": "Orange", "f"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "BoardState", "destinationTicketsCount": 26, "d"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "ListAllRoutes", "routes": [{"c": "color", "Rainbow"}, {"type": "Info", "player": "Blue", "success": true, "turnType": "DrawPassengerCars", "drawCard": "Black", "f"}]
```



Playback

JSON

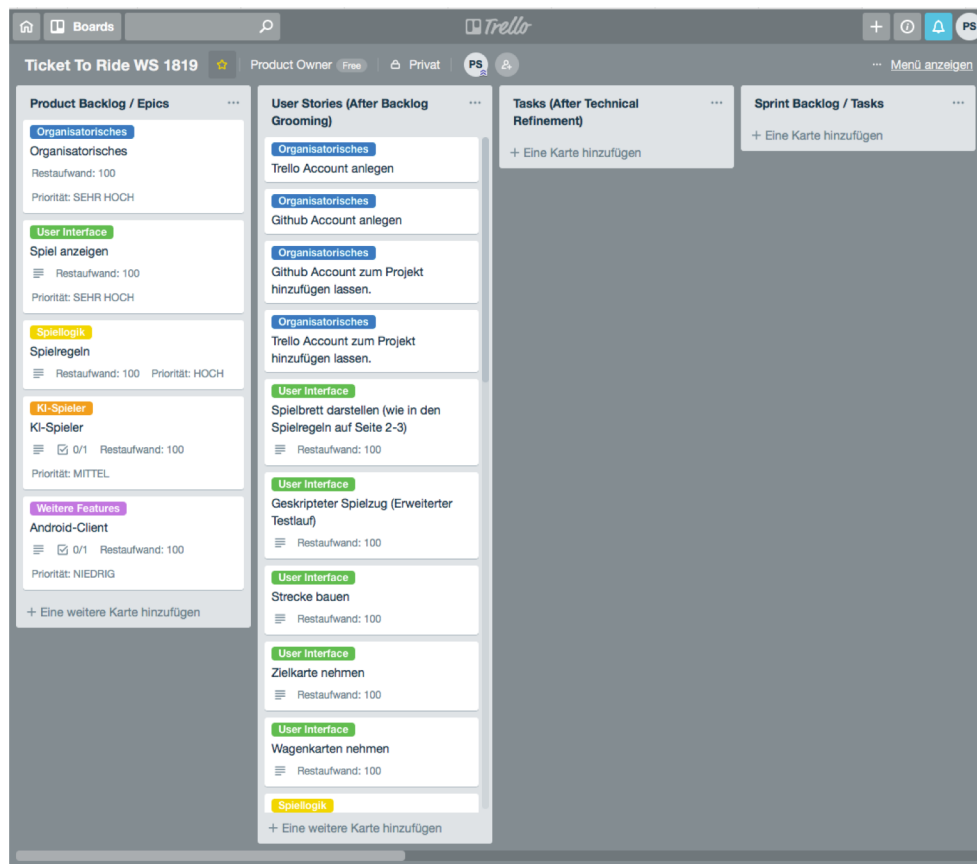
1a) Meilenstein
Observer-GUI

1b) Meilenstein
Single-Player

2. Meilenstein
KI-Spieler /
Wettbewerb



Anforderungskatalog: “Product Backlog”



Organisatorisches

- Wir übernehmen die Rolle des **Product Owners**:
Angabe, **was** wir wollen, **nicht wie** es umgesetzt wird.
- Sie übernehmen die Entwicklerrolle:
Definition von Teilaufgaben, Prototyping, Test, Präsentation.
- Vier Scrum-Teams à 5 Personen
- Ein Scrum-Master (Marc Seibert) für alle Teams
- Website: <https://hu.berlin/tickettoride1819>
- Versionierung und Issue-Tracking via **GitHub**:
<https://github.com/hu-berlin-semesterprojekte>
- Product Backlog („Anforderungskatalog“) via **Trello**:
<https://trello.com/b/e8GlcYp4/ticket-to-ride-ws-1819>

Pflichttermine

- Mindestens **zwei mal pro Woche, besser öfter (!)**:
 - „Daily“ Scrum: Kurze Treffen im Stehen, in denen sich das Team updatet und koordiniert
- **Wöchentlich**:
 - Scrum of Scrums (Mo): Kommunikation / Abstimmung der Scrum-Teams
- **Zweiwöchentlich** / innerhalb eines Sprints:
 - In der Gruppe:
 - Backlog Grooming (1. Mo): Klärung fachlicher Fragen; Aufteilen komplexer Stories; Backlog für den nächsten Sprint vorbereiten
 - Sprint Review Meeting (3. Mo): das aktuelle Produkt wird vorgestellt und mit Backlog verglichen.
 - Im Team;
 - Daily Scrum (Mo, Di-Fr): Kurz; Was wurde und was wird erreicht, Hindernisse?
 - Technical Refinement (Di-Fr): Erstellung technischer Tasks aus User Stories und Abschätzung.
 - Sprint Planning (2. Mo): Verteilung der Tasks an die Scrum-Teams.

Ablauf eines Sprints

2 wöchiger Rhythmus:

1. Woche: Backlog und Technical Refinement
2. Woche: Sprint Review (vorheriger Sprint) und Sprint Planning (nächster Sprint)

1. Mo	1. Di-Fr	2. Mo	2. Di-Fr
Daily Scrum (15min) Backlog Grooming	Daily Scrum (15min) Technical Refinement	Daily Scrum (15min) Sprint Review (Demo!) Sprint Planning	Daily Scrum (15min)

Erster Meilenstein (Observer-Modus + Single-Player)

43 KW: 22.10.	Vorbereitend	Backlog Grooming, pro Scrum Team: Wahl einer Spiele-Engine und Technical Refinement
44 KW: 29.10.	Erster Sprint	Sprint Planning
45 KW: 05.11.	Vorbereitend	Backlog Grooming und Technical Refinement
46 KW: 12.11.	Zweiter Sprint	Demo und Sprint Planning
47 KW: 19.11.	Vorbereitend	Backlog Grooming und Technical Refinement
48 KW: 26.11.	Dritter Sprint	Demo und Sprint Planning
49 KW: 03.12.	Vorbereitend	Backlog Grooming und Technical Refinement
51 KW: 17.12.	Ende 1. Meilenstein	Präsentation & Wahl der Sieger-Engine

Fragen?



Organisatorisches (Fortsetzung)

- verfügbare Zeiten (von Marc Seibert):
 - (Montag, 13 bis 17 Uhr, RUD 25, 4.113)
 - **Mittwochs, ab 17 Uhr, Team Grün (R 4.113)**
 - **Donnerstags ab 17 Uhr, Team Schwarz (R 4.113)**
 - **Donnerstags ab 13 Uhr, Team Rot (R 4.410)**
 - **Freitags, ab 11 Uhr, Team Blau (R 4.410, bis auf 16.11.)**
 - Weitere Termine gewünscht?
- begleitende Vorträge:
 - GitHub
 - Unit-Tests
 - Continuous Integration
 - Coding Guidelines
 - Einführung zu KI in Spielen

Teilnehmer

Name	Scrum Team	Name	Scrum Team
Arndt, Tobias	Do ab 13 Uhr	Liepe, Jonas	Do ab 17 Uhr
Behnke, Tim	Mi ab 17 Uhr	Malsam, Georg	Fr ab 11 Uhr
Brand, Jakob	Do ab 17 Uhr	Möhring, Maximilian	Mi ab 17 Uhr
Breitkopf, Tom-Lukas Johann	Do ab 17 Uhr	Müller, Steven	Fr ab 11 Uhr
Brunkow, Alexander	Do ab 17 Uhr	Quiter, Malte	Mi ab 17 Uhr
Döring, Jacob	Do ab 13 Uhr	Röhr, Carl	Fr ab 11 Uhr
Dröse, Adrian	Fr ab 11 Uhr	Schneider, Nicolai	Do ab 17 Uhr
Hildebrand, Lydia	??	Seibert, Marc	Scrum Master
Kremer, Julian	Mi ab 17 Uhr	Tavakoli, Pedram	Do ab 13 Uhr
Leppich, Birk	Fr ab 11 Uhr	Toth, Andras	Mi ab 17 Uhr

Next Steps

- Jedes Team erhält eigenen Product Backlog.
 - bis **morgen**: **Trello**-Account erstellen und Trello-Account unter <https://goo.gl/UyAAj8> eintragen (oder per E-Mail mitteilen)
- **diese Woche** (ca. 15-20 Stunden Arbeitsaufwand pro Woche!):
 - **Technical Refinement**: Jedes Team trifft sich, um fehlende Tasks für die User Story „Spiel anzeigen“ zu identifizieren und Aufwand abzuschätzen.
 - jede Gruppe legt den Termin für das Technical Refinement fest
 - jede Gruppe beschäftigt sich mit den vorgeschlagenen Sprachen / **Engines** und entscheidet sich für eine
 - jeder arbeitet sich in die Sprache/Engine seiner/ihrer Gruppe ein
- nächster Termin: Montag ab 13 Uhr: Vortrag „Server-Protokoll“, „Versionierung und Bug Tracking mit GitHub, Evtl. Continuous Integration“; Sprint Planning