First Milestone Review

Patrick Schäfer, Ulf Leser

patrick.schaefer@hu-berlin.de

Semester Project

- General idea:
  - develop a software of intermediate complexity over the course of a semester
  - Experience software development from start to finish
  - Work in teams: communicate, meetings, integration, deal with slackers & know-it-alls
- Apply tools and methods of professional software development
  - Scrum, Unity3D, Trello (User Stories), GitHub, unit tests, continuous integration, remote procedure calls
- Apply knowledge from university courses in practice
  - Data structures, graph algorithms, artificial intelligence

This Year: Zug um Zug (Amerika)

- Goal:
  - Score most points by connecting cities with railroad tracks

- Score points by:
  1. Connect any two adjacent cities
  2. Fulfil own destination tickets between two far cities
  3. Build the longest connected route

Graph-Concepts

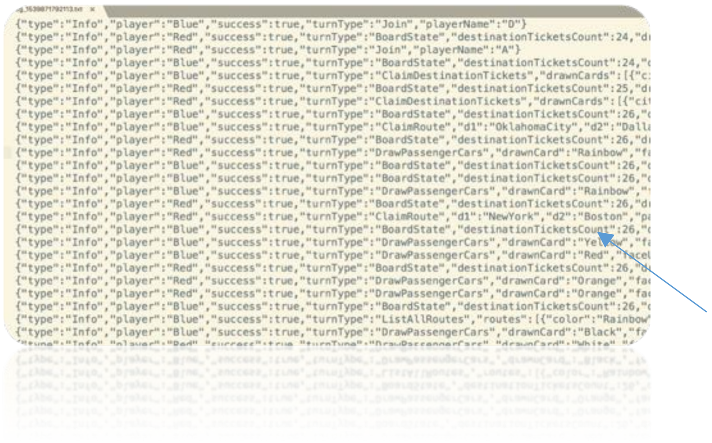
- Representation of the board
  - Adjacency matrix / adjacency list
- Given a destination ticket, find the shortest path
  - Dijkstra
- Fulfill destination tickets with the least amount of trains
  - Minimum spanning tree on subgraph (Minimum Steiner tree)
- Calculating the final score:
  - List of routes claimed by a player
    - Lookup in graph data structure (adjacency matrix or adjacency list)
  - List of destination tickets fulfilled by a player
    - Graph traversal: DFS / BFS
  - 10 point bonus is awarded to player with the longest route
    - Longest path in a tree / graph

Semesterprojekt WS 16/17

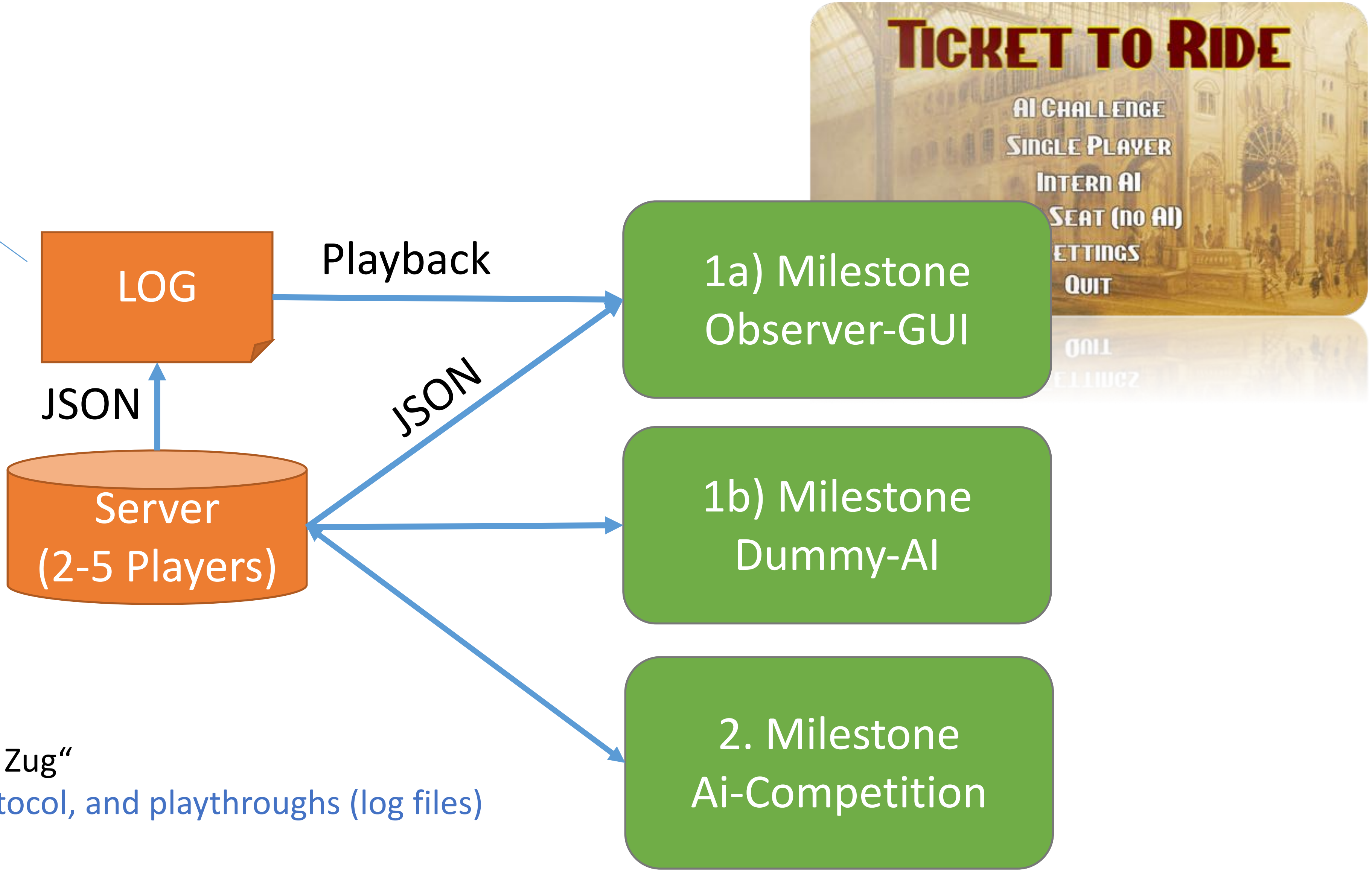
- Zug um Zug Europa
- 2 Unity teams, one Java FX team

- This year: Zug um Zug Amerika

Catalogue of Requirements



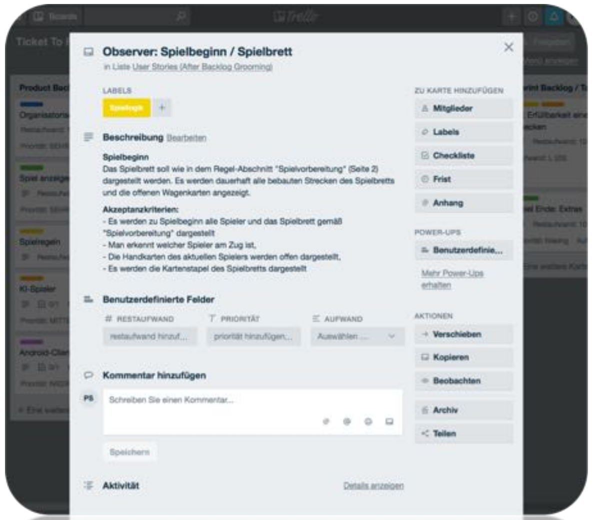
```
{"type":"Info","player":"Blue","success":true,"turnType":"Join","playerName":"D"}
{"type":"Info","player":"Red","success":true,"turnType":"BoardState","destinationTicketsCount":24,"d
{"type":"Info","player":"Red","success":true,"turnType":"Join","playerName":"A"}
{"type":"Info","player":"Blue","success":true,"turnType":"BoardState","destinationTicketsCount":24,"
{"type":"Info","player":"Blue","success":true,"turnType":"ClaimDestinationTickets","drawnCards":[{"c
{"type":"Info","player":"Red","success":true,"turnType":"BoardState","destinationTicketsCount":25,"d
{"type":"Info","player":"Red","success":true,"turnType":"ClaimDestinationTickets","drawnCards":[{"ci
{"type":"Info","player":"Blue","success":true,"turnType":"BoardState","destinationTicketsCount":26,"
{"type":"Info","player":"Blue","success":true,"turnType":"ClaimRoute","d1":"OklahomaCity","d2":"Dall
{"type":"Info","player":"Red","success":true,"turnType":"BoardState","destinationTicketsCount":26,"d
{"type":"Info","player":"Red","success":true,"turnType":"DrawPassengerCars","drawnCard":"Rainbow","f
{"type":"Info","player":"Blue","success":true,"turnType":"BoardState","destinationTicketsCount":26,"
{"type":"Info","player":"Blue","success":true,"turnType":"BoardState","destinationTicketsCount":26,"
{"type":"Info","player":"Blue","success":true,"turnType":"DrawPassengerCars","drawnCard":"Rainbow","
{"type":"Info","player":"Red","success":true,"turnType":"BoardState","destinationTicketsCount":26,"d
{"type":"Info","player":"Red","success":true,"turnType":"ClaimRoute","d1":"NewYork","d2":"Boston","p
{"type":"Info","player":"Blue","success":true,"turnType":"BoardState","destinationTicketsCount":26,"
{"type":"Info","player":"Blue","success":true,"turnType":"DrawPassengerCars","drawnCard":"Yellow","f
{"type":"Info","player":"Blue","success":true,"turnType":"DrawPassengerCars","drawnCard":"Red","face
{"type":"Info","player":"Red","success":true,"turnType":"BoardState","destinationTicketsCount":26,"d
{"type":"Info","player":"Red","success":true,"turnType":"DrawPassengerCars","drawnCard":"Orange","fa
{"type":"Info","player":"Red","success":true,"turnType":"DrawPassengerCars","drawnCard":"Orange","fa
{"type":"Info","player":"Blue","success":true,"turnType":"BoardState","destinationTicketsCount":26,"
{"type":"Info","player":"Blue","success":true,"turnType":"ListAllRoutes","routes":[{"color":"Rainbow
{"type":"Info","player":"Blue","success":true,"turnType":"DrawPassengerCars","drawnCard":"Black","f
"type":"Info","player":"Red","success":true,"turnType":"DrawPassengerCars","drawnCard":"White","
```



Implement board game „Zug um Zug“
We provide a game-server, a protocol, and playthroughs (log files)

(Our Take on) Scrum

- Four scrum teams of five students each
- Scrum master coordinates teams and meetings
- Bi-weekly sprints (prototypes)
- We provide new user stories prior to each sprint
- In every sprint, the developers (students)
  - Disassemble our user stories into technical tasks,
  - estimate the complexity of these task,
  - decide which task to tackle in this sprint, and
  - implement these tasks.
- At the end of each sprint, the teams present their current prototype in a sprint review



Observer: Spielbeginn / Spielbrett

in Liste User Stories (After Backlog Grooming)

LABELS

Spiellogik

Beschreibung Bearbeiten

Spielbeginn
Das Spielbrett soll wie in dem Regel-Abschnitt "Spielvorbereitung" (Seite 2) dargestellt werden. Es werden dauerhaft alle bebauten Strecken des Spielbretts und die offenen Wagenkarten angezeigt.

Akzeptanzkriterien:
- Es werden zu Spielbeginn alle Spieler und das Spielbrett gemäß "Spielvorbereitung" dargestellt
- Man erkennt welcher Spieler am Zug ist,
- Die Handkarten des aktuellen Spielers werden offen dargestellt,
- Es werden die Kartenstapel des Spielbretts dargestellt

Benutzerdefinierte Felder

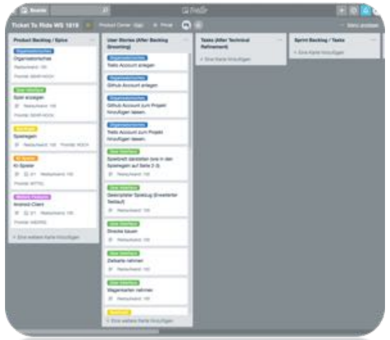
# RESTAUFWAND	PRIORITÄT	AUFWAND
restaufwand hinzuf...	priorität hinzufügen...	Auswählen ...

Kommentar hinzufügen

PS Schreiben Sie einen Kommentar...

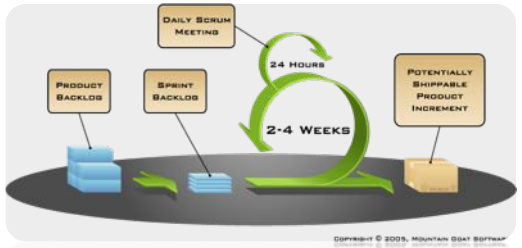
Speichern

Aktivität Details anzeigen



Ticket To Ride WS 1819

Product Backlog / Epics	User Stories (After Backlog Grooming)	Tasks (After Technical Refinement)	Sprint Backlog / Tasks
Organisatorisches	Trello Account anlegen		
Spiel anzeigen	Github Account anlegen		
Spielregeln	Github Account zum Projekt hinzufügen lassen.		
KI-Spieler	Trello Account zum Projekt hinzufügen lassen.		
Android-Client	Spielbrett darstellen (wie in den Spielregeln auf Seite 2-3)		
	Gescripteter Spielzug (Erweiterter Testlauf)		
	Strecke bauen		
	Zielkarte nehmen		
	Wagenkarten nehmen		



Copyright © 2005, Mountain Goat Software

First Milestone

- Plan:
  - Each team builds a prototype over 3.5 sprints; only one selected for AI challenge
  - Use one common API, so all prototypes can be used for playbacks
  - Rationale: Competition raises productivity; 20 developers are too much for working on the core game logic

- Scrum user stories (summarized):
  1. Visualize the game board
  2. Visualize all possible player actions:
     1. build tracks;
     2. draw cards from the two decks
  3. Display one complete (automated) playthrough from a log-file
  4. Display winner
  5. Provide one dummy-AI

Roadmap

- Today:
  - Each of the four teams has 15 minutes to present their prototype
  - Hidden vote for the best prototype
- Until Jan 7:
  - Team with most votes: Set up binary for AI challenge
  - All teams:
    - Finalize first milestone: every team has to submit a binary for playbacks of playthrough logs
    - Familiarize with API for AI development

- Second Milestone (starting Jan 7): AI challenge
  - Each team implements (at least) one AI player
  - Five weekly sprints
  - AI tournament at the end of every sprint (round-robin, 1on1)

	Blue	Red	Green	Black
User-Stories	- Klassisches Design - Menü - Spielbrett - Audio - Animationen - Abspielgeschwindigkeit - Alle Spielerinformationen - Score-Board - Animationen - Playback (Log)	- Klassisches Design - Menü - Spielbrett - Alle Spielerinformationen - Erfüllte/unerfüllbare Strecken - Schraffierung der Strecke - Datenmodell (Längste Strecke, Erfüllbarkeit) - Animationen - Abspielgeschwindigkeit (4 Stufen) - Bug-Tracking des Servers - Playback (Log) - Join der Spieler - Aktiver Spieler hervorgehoben - Schraffierte Strecken - Final Turn - Score-Board	- Hyperloop Design, Neonlichter, Effekte - Menü (Fade-In) - Join von Spielern - Karten ziehen - Aktiver Spieler als Hintergrundfarbe - Zielkarten permanent sichtbar - Fahrende Züge - Animationen - Abspielgeschwindigkeit - Gewinner - Playback (festes Log)	- Eigenes Design - Menü - Abspielgeschwindigkeit - Playback (Log-File) und Server-Modus - Eigenes Design - Animationen - Unbebaute Strecken transparent - Aktiver Spieler unten - Spieler mit längster Strecke - Score-Board - Zustandsautomat (Bug-Tracking des Servers)
Kommentare / Nice-To-Haves	- Server-Socket - Zu erfüllende Strecken auf der Karte markieren - Darstellung der gebauten Strecken	- Server-Socket - Zielkarten nicht immer sichtbar - Zu erfüllende Strecken auf der Karte markieren	- Handycaps, Schwierigkeiten bei Scrum (Scrum-Master) - Final Turn - Score-Board - Log-File auswählen - Länge der Strecke nicht ganz klar	- Schnelleres Playback? - Zu erfüllende Strecken auf der Karte markieren