First Milestone Review

On a blustery autumn evening five old friends met in the backroom of one of the city’s oldest and most private clubs. Each had traveled a long distance — from all corners of the world — to meet on this very specific day… October 2, 1900 — 28 years to the day that the London eccentric, Phileas Fogg accepted and then won a £20,000 bet that he could travel Around the World in 80 Days.

When the story of Fogg’s triumphant journey filled all the newspapers of the day, the five attended University together. Inspired by his impetuous gamble, and a few pints from the local pub, the group commemorated his circumnavigation with a more modest excursion and wager – a bottle of good claret to the first to make it to Le Procope in Paris.

Each succeeding year, they met to celebrate the anniversary and pay tribute to Fogg. And each year a new expedition (always more difficult) with a new wager (always more expensive) was proposed. Now at the dawn of the century it was time for a new impossible journey. The stakes: $1 Million in a winner-takes-all competition. The objective: to see which of them could travel by rail to the most cities in North America — in just 7 days. The journey would begin immediately…

Ticket to Ride is a cross-country train adventure. Players compete to connect different cities by laying claim to railway routes on a map of North America.

For 2 - 5 players ages 8 and above 30 - 60 minutes

Patrick Schäfer, Ulf Leser
patrick.schaefer@hu-berlin.de

Semesterprojekt: Implementierung eines Brettspiels, WS 18/19
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

1a) Milestone Observer-GUI

1b) Milestone Dummy-AI

2. Milestone Ai-Competition

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