

WaterWizards

Justin Dewitz, Erick Zeiler, Max Kondratov, Julian jNachnamej

17. Juli 2025



Inhaltsverzeichnis

Einleitung

Organisation

Rollen des Projektes

Ansprechpartner

Technologie + Architektur

UI

GameScreen

Server/Backend

Shared

Backend - Client Kommunikation

Technische Erfahrungen

Analyse

Wiki

Erfahrungen und Fazit

Einleitung

- ▶ Was ist WaterWizards?
 1. Multitplayer Real-Time Schiffe versenken
 2. Angriff durch Zauber, die durch Karten repräsentiert werden
 3. Ziel: Zerstörung der gegnerischen Schiffe
- ▶ Warum WaterWizards?
 1. Schiffe versenken ist ein Klassiker
 2. Durch Real-Time wird es dynamischer
 3. Für jede Altersgruppe interessant
- ▶ Was macht WaterWizards besonders?
 1. Kombination aus Strategie und schnellen Entscheidungen
 2. Zauber und Karten bringen neue Dynamik ins Spiel
 3. Multiplayer und Echtzeit sorgen für Spannung
- ▶ Für wen ist das Spiel gedacht?
 1. Strategie-Fans
 2. Familien und Freunde
 3. Alle Altersgruppen

Organisation

- ▶ git über GitHub für die Versionsverwaltung
- ▶ Scrum mit 2-Wochen-Sprints
- ▶ Kanban/Issue-Board über GitHub
- ▶ Kommunikation über Discord-Server

Rollen des Projektes

Ansprechpartner

Technologie + Architektur

- ▶ Programmiersprache: C#
- ▶ Raylib für die grafische Darstellung
- ▶ LiteNetLib für die Client-Server-Verbindung
- ▶ Nuke für das Build-System
- ▶ CodeQL für die statische Code-Analyse
- ▶ GitHub Actions für die CD-Pipeline
- ▶ GitHub Pages für die Dokumentation
- ▶ Event-Driven Architektur
- ▶ Docker für die Containerisierung
- ▶ Hetzner Server für das Hosting

- ▶ Raylib für die grafische Darstellung
- ▶ GameScreen als Hauptkomponente
- ▶ GameBoard für die Spielbretter
- ▶ GameShip für die Schiffe
- ▶ GameCard für die Karten
- ▶ GameHand für die Hand der Spieler

GameScreen

- ▶ GameScreen enthält alles, was auf dem Bildschirm gezeigt wird
 1. Die beiden Spielbretter (GameBoard) mit Schiffen (GameShip) und Steinen
 2. Die Karten (GameCard) auf der Hand (GameHand) der Spieler
 3. Die Kartenstapel (CardStack) der einzelnen Kartentypen (CardType)

Server/Backend

- ▶ Das Backend ist in C# mit der Library LiteNetLib geschrieben
- ▶ Ein globaler Server der eine Lobby auf Hetzner bereitstellt
- ▶ Server wird in Docker-Containern ausgeführt

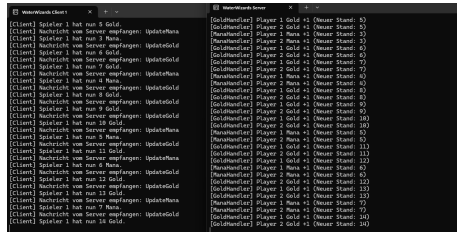
Shared

- ▶ Shared enthält alle Klassen, die sowohl im Client als auch im Server verwendet werden
- ▶ Enthält die Definitionen der Karten und der Kartentypen
- ▶ Enthält die Definitionen der Schiffe und der Schiffs-Typen

Backend - Client Kommunikation

Event-Driven Architektur

- ▶ UDP für die Echtzeit-Kommunikation
- ▶ Nachrichten basiertes Protokoll für beidseitige Kommunikation



The image shows two terminal windows side-by-side. The left window is titled 'WurmWitch Client' and the right window is titled 'WurmWitch Server'. Both windows display a series of log messages indicating network communication. The client messages show actions like 'Spieler 1 hat nun 8 Gold' and 'Nachricht vom Server empfangen: UpdateMana'. The server messages show corresponding updates for 'Player 1 Gold' and 'Player 2 Gold', along with 'Mana' updates for both players. The logs show a sequence of events where the client's state is updated by the server and vice versa.

```
WurmWitch Client
[Client] Spieler 1 hat nun 8 Gold.
[Client] Nachricht vom Server empfangen: UpdateMana.
[Client] Spieler 1 hat nun 3 Mana.
[Client] Nachricht vom Server empfangen: UpdateGold.
[Client] Spieler 1 hat nun 6 Gold.
[Client] Nachricht vom Server empfangen: UpdateMana.
[Client] Spieler 1 hat nun 7 Gold.
[Client] Nachricht vom Server empfangen: UpdateGold.
[Client] Spieler 1 hat nun 8 Gold.
[Client] Nachricht vom Server empfangen: UpdateMana.
[Client] Spieler 1 hat nun 9 Gold.
[Client] Nachricht vom Server empfangen: UpdateGold.
[Client] Spieler 1 hat nun 10 Gold.
[Client] Nachricht vom Server empfangen: UpdateMana.
[Client] Spieler 1 hat nun 8 Mana.
[Client] Nachricht vom Server empfangen: UpdateGold.
[Client] Spieler 1 hat nun 9 Gold.
[Client] Nachricht vom Server empfangen: UpdateGold.
[Client] Spieler 1 hat nun 10 Gold.
[Client] Nachricht vom Server empfangen: UpdateMana.
[Client] Spieler 1 hat nun 8 Mana.
[Client] Nachricht vom Server empfangen: UpdateGold.
[Client] Spieler 1 hat nun 9 Gold.
[Client] Nachricht vom Server empfangen: UpdateGold.
[Client] Spieler 1 hat nun 10 Gold.
[Client] Nachricht vom Server empfangen: UpdateMana.
[Client] Spieler 1 hat nun 7 Mana.
[Client] Nachricht vom Server empfangen: UpdateGold.
[Client] Spieler 1 hat nun 16 Gold.

WurmWitch Server
[GoldHandler] Player 1 Gold +1 (Neuer Stand: 8)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 8)
[ManaHandler] Player 2 Mana +1 (Neuer Stand: 3)
[ManaHandler] Player 2 Mana +1 (Neuer Stand: 3)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 6)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 7)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 7)
[ManaHandler] Player 2 Mana +1 (Neuer Stand: 6)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 8)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 8)
[GoldHandler] Player 1 Gold +1 (Neuer Stand: 9)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 8)
[GoldHandler] Player 1 Gold +1 (Neuer Stand: 10)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 10)
[ManaHandler] Player 2 Mana +1 (Neuer Stand: 3)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 10)
[ManaHandler] Player 2 Mana +1 (Neuer Stand: 6)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 11)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 11)
[GoldHandler] Player 1 Gold +1 (Neuer Stand: 12)
[ManaHandler] Player 2 Mana +1 (Neuer Stand: 6)
[ManaHandler] Player 2 Mana +1 (Neuer Stand: 6)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 12)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 13)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 13)
[ManaHandler] Player 2 Mana +1 (Neuer Stand: 7)
[ManaHandler] Player 2 Mana +1 (Neuer Stand: 7)
[GoldHandler] Player 1 Gold +1 (Neuer Stand: 14)
[GoldHandler] Player 2 Gold +1 (Neuer Stand: 14)
```

Technische Erfahrungen

Analyse



Erfahrungen und Fazit