



# Kingdom Builder

Ein Projekt von  
Erik Schott, Tom Weggel, Burak Bayram,  
Linda Lang und Julian Kirsch

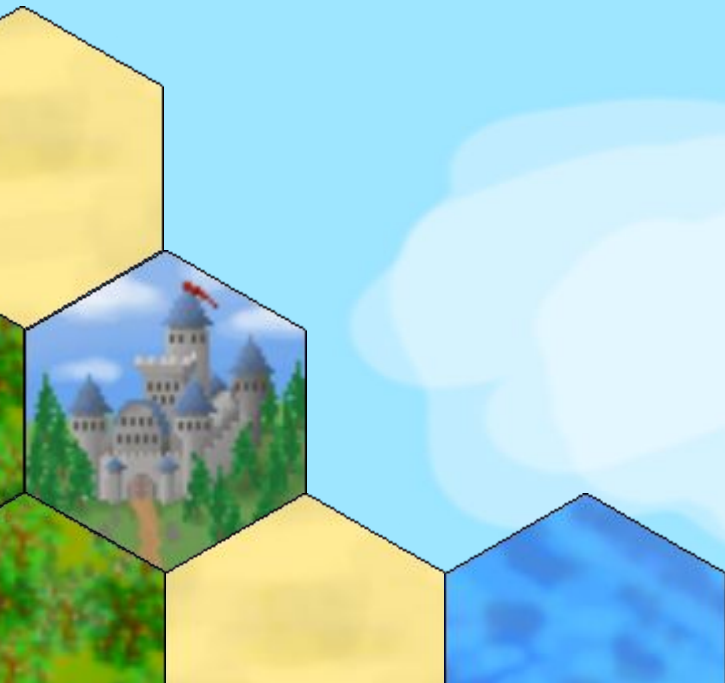
# Gliederung

- Ziel
- Netzwerk
- Redux 1x1
- GUI
- Gamellogic
- Künstliche Intelligenz
- Demonstration



# Ziel

- Implementierung des Brettspiels “Kingdom Builder”
- Rahmenbedingungen:
  - Implementierung unter Verwendung von Java und JavaFX
  - Insbesondere Entwicklung eines Clients zu einem vorhandenen Server





# Netzwerk

- Grundlegende Anforderungen:
  - Umsetzung einer Client-Server-Architektur
  - Modellierung des Spielers als Klient
  - Kommunikation über Textprotokoll



# Netzwerk - Client-Server-Architektur

- Frage in der Planung der Umsetzung:
  - Asynchron oder synchron?
  - Synchron:
    - Einfach zu verstehen und zu implementieren
    - Teile des Programms warten explizit auf das Netzwerk
      - Teile des Programms blockieren die Ausführung
      - Deadlocks möglich
  - Asynchron:
    - Erfordert umdenken
      - Ausführung des Programms wird nicht mehr blockiert
- Interaktionen asynchron
  - Ereignisgesteuerte Architektur:
    - Befehle blockieren nicht
    - Antworten als Event modelliert



# Netzwerk - Lehren aus dem C10K-Problem

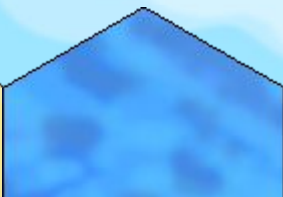
- C10K-Problem:
  - “C10K” - Numeronym:  
Wie  $\geq 10.000$  aktive Netzwerkverbindungen behandeln?
  - Lösung 1:
    - Jede Netzwerkverbindung wird von einem Thread gehandhabt.
    - Leicht zu verstehen
    - Schnell umzusetzen
    - Threads müssen ggf. synchronisiert werden → Hohe Fehleranfälligkeit
  - Lösung 2:
    - Alle Verbindungen werden in einem Thread von einer sog. “Selektor-Event-Loop” behandelt.
    - Kontraintuitiv
    - Schwer umzusetzen, ABER: wird bereits von der Java Library mitgeliefert
    - Effizienter da Threads nicht “unnötig” Rechenzeit mit warten verschwendet
    - Robuster da weniger Synchronisationspunkte



# Netzwerk - Protokoll

- Einfaches Textformat
- Pakete folgen immer dem gleichen Schema:

Header	Action	Payload (Optional)
[SERVER_MESSAGE]	[WELCOME_TO_SERVER]	<[1;Spielername;-1]>



# Netzwerk - Protokoll/Grammatik

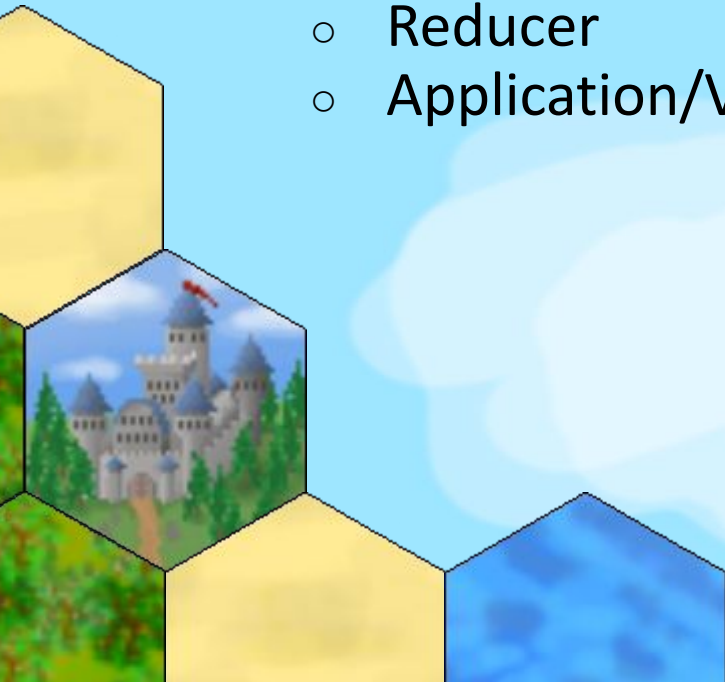
- Payloads folgen einer einfachen Grammatik (Auszug):
  - (natürliche) Zahlen
  - Strings
  - Tupel, z.B.: [1;2;3;4]
  - Liste, z.B.: {1;2;3;4}
- Pakete lassen sich als Objekte modellieren
- Paket-Parser (und Generator) wird automatisch generieren



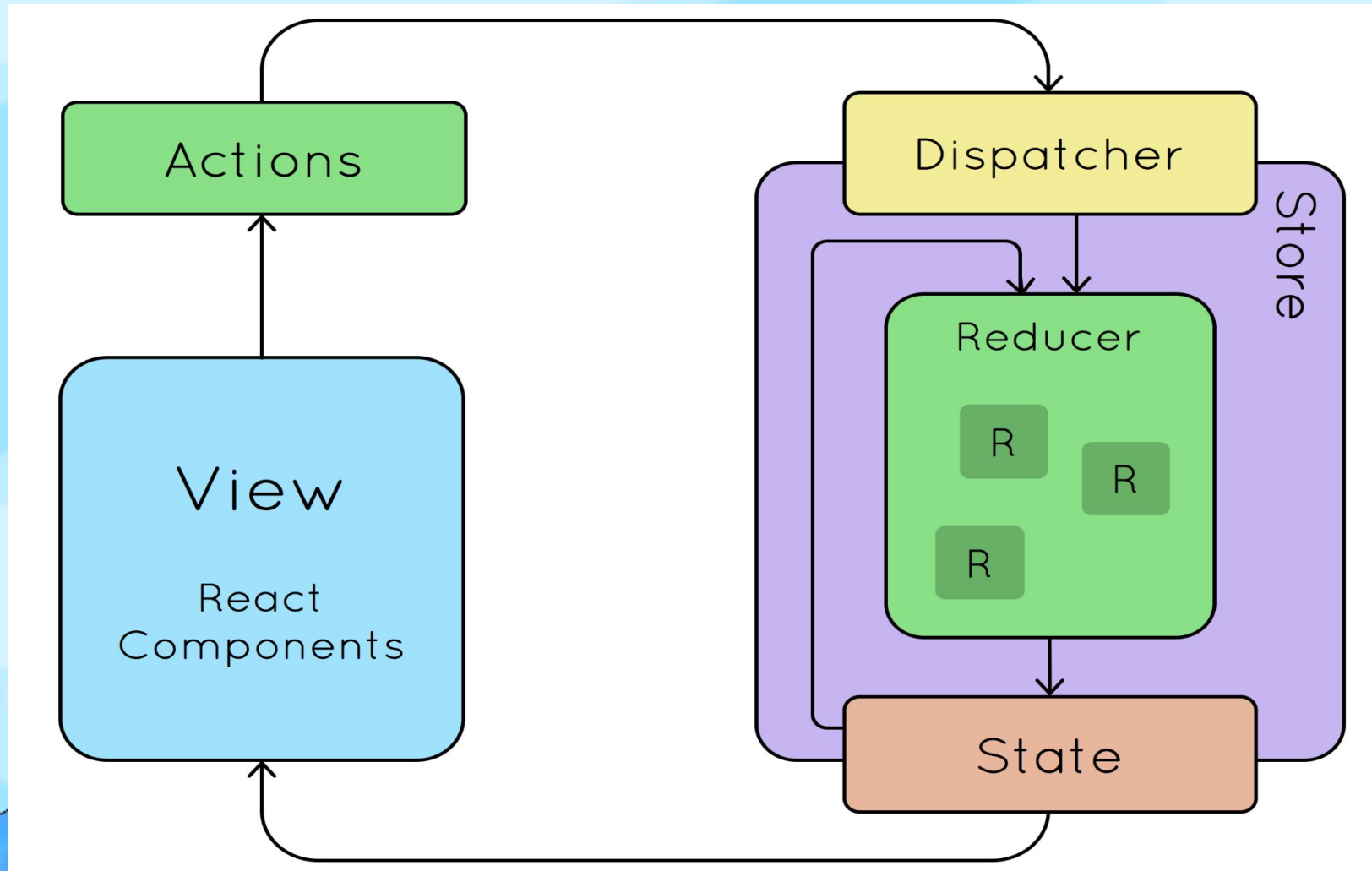


# Redux

- Komponenten
  - Actions
  - Store
  - Dispatcher (in Form einer dispatch(action)-Methode im Store)
  - State
  - Reducer
  - Application/View

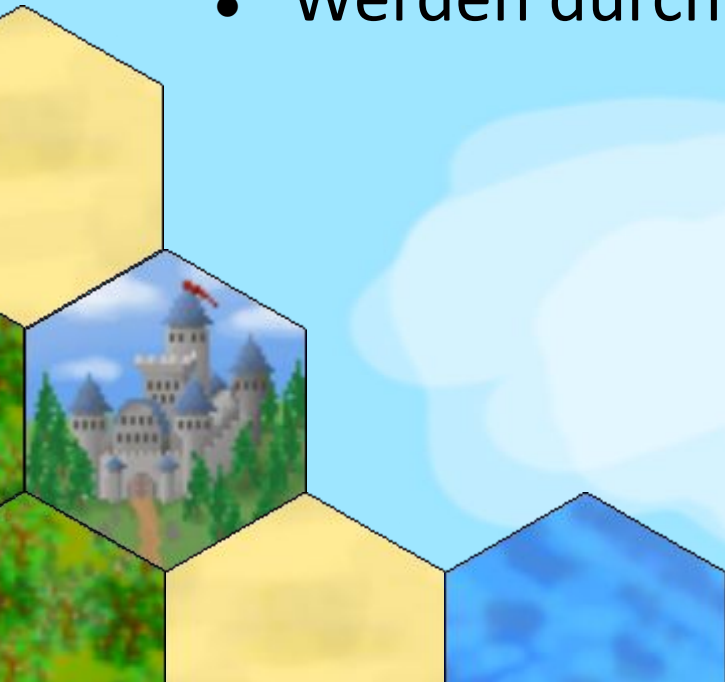


# Redux - Konzept



# Redux - Action

- Lässt sich als Tupel aus Typ/Identifizier und Payload (Parameter der Änderung) darstellen
- Beschreibt, welche Änderungen in der Application/View vorgenommen werden sollen
- Werden durch Änderungen in der Application/View ausgelöst



# Redux - Store & State

- Store:
  - Enthält den Dispatcher & den State.
  - Verwaltet State und seine Änderungen (Aktualisierung des States).
  - Benachrichtigt alle Abonnenten eines Attributs, sobald jenes geändert wurde.
- State:
  - Enthält alle wichtigen Informationen der Application/View.





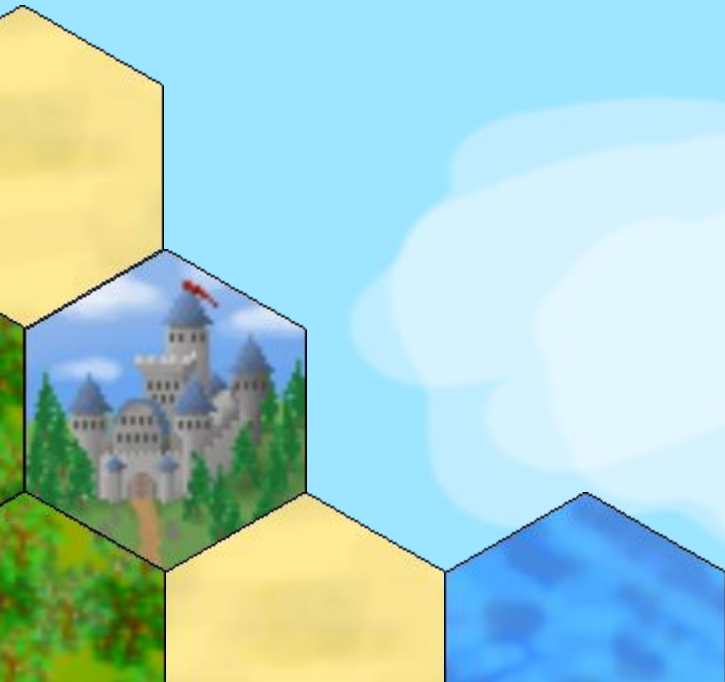
# Redux - Dispatcher

- Ist als `dispatch(Action)`-Methode im Store enthalten.
- Wird aufgerufen, sobald eine Action ausgelöst wird.
- Leitet sozusagen die Action an den Reducer weiter, indem er die `reduce(Store, Action)`-Methode des passenden Reducers aufruft.



# Redux - Reducer

- Enthält die `reduce(Store, Action)`-Methode.
- Basierend auf der Action wird ein neuer State berechnet und zurückgegeben. Der aktuelle State wird im Reducer selbst nicht verändert.



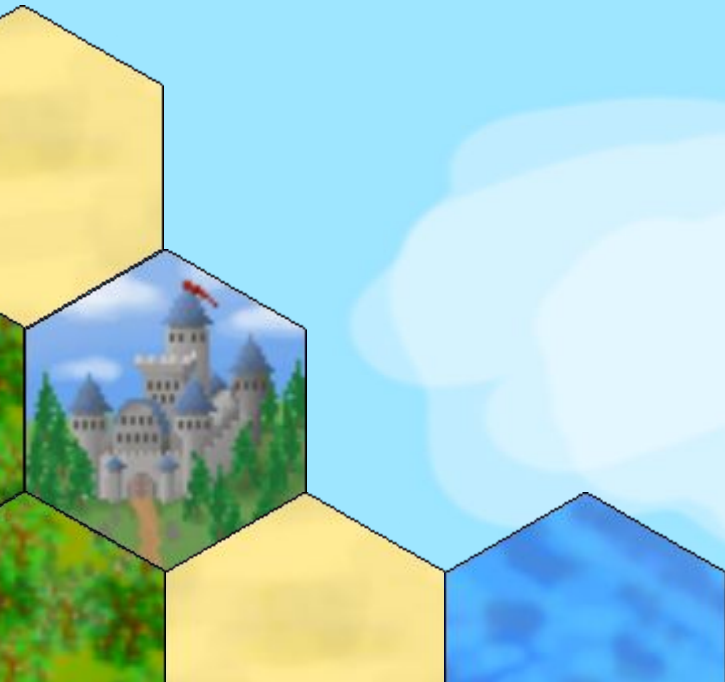
# Redux - View

- Interaktionen des Benutzers mit der Application/View lösen Änderungen des States aus. Dafür löst die Application/View Actions aus.
- Wird vom Store benachrichtigt, sobald Änderungen des States vorgenommen wurden. Dafür werden die subscribe()-Methoden aufgerufen und die UI wird entsprechend aktualisiert.



# Redux - Unsere Umsetzung

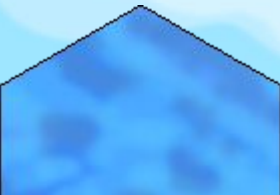
- Dispatcher:
  - dispatchOld(Action action)-Methode
  - dispatch(String action, Object payload)
    - String action als Identifier der Action
    - Object payload als Parameter der Änderung



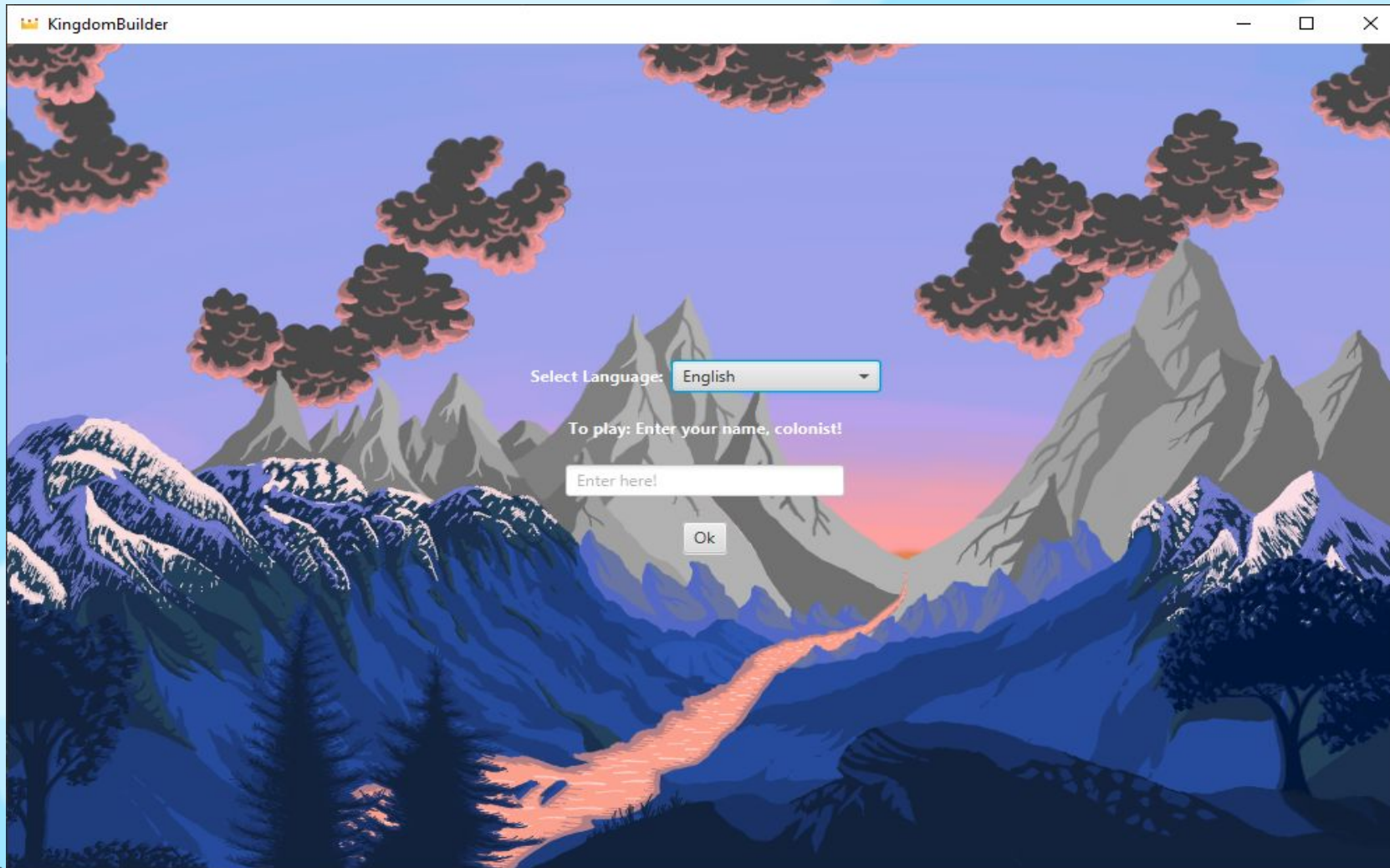


# Redux - Unsere Umsetzung

- Reducer:
  - übergeordnete Reducer-Klasse
  - 5 erweiterte (erbende) Reducer-Klassen:
    - Application-Reducer
    - Bot-Reducer
    - Chat-Reducer
    - Game-Reducer
    - Root-Reducer
  - reduce()-Methode:
    - die richtige Methode des zugehörigen Reducers, um die Action zu verarbeiten, wird über Funktionspointer aufgerufen

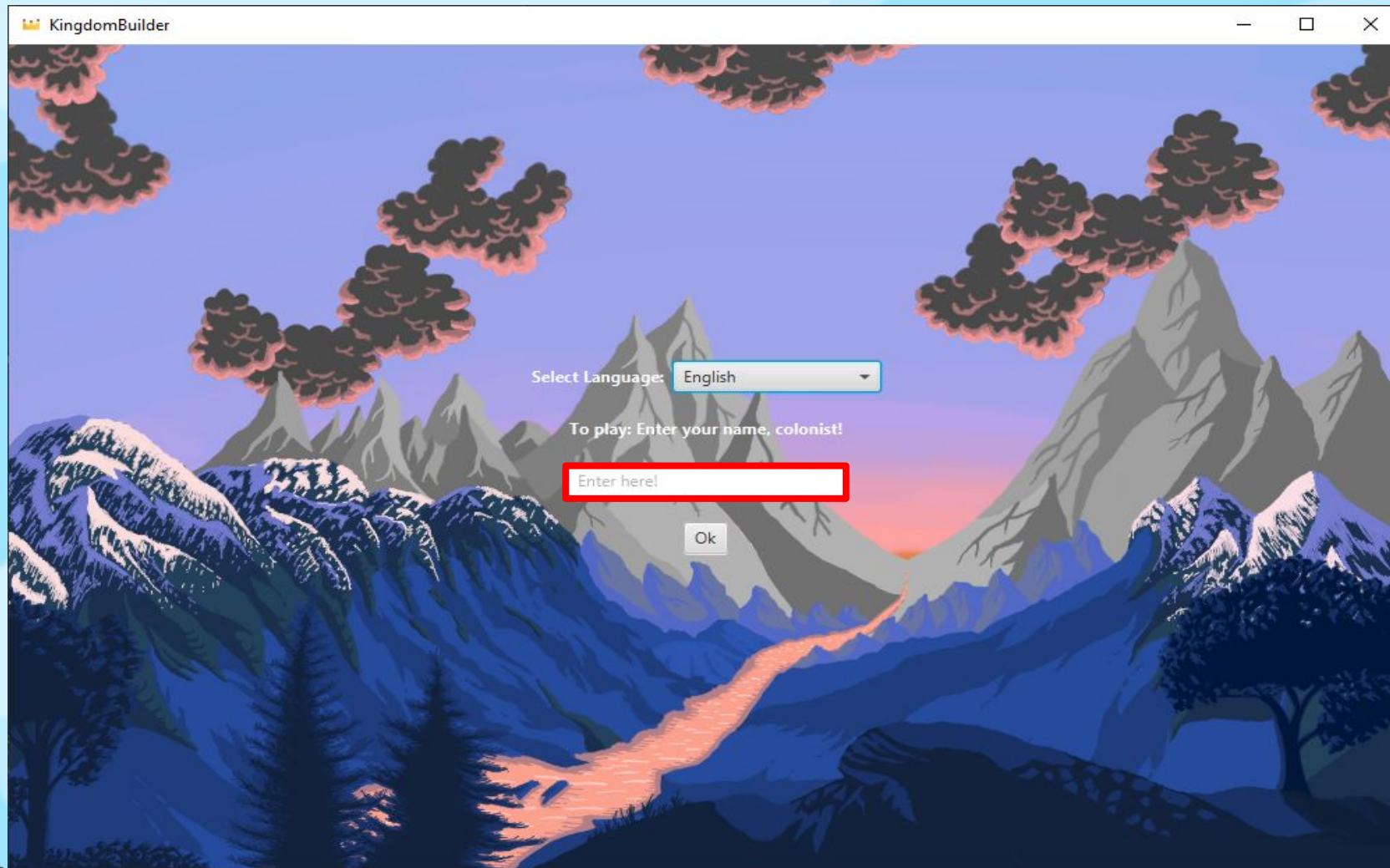


# GUI - Name

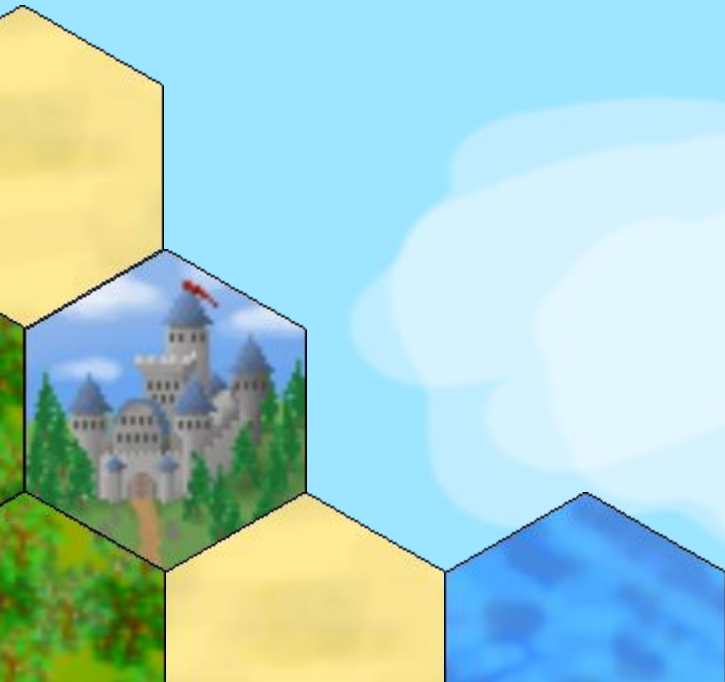
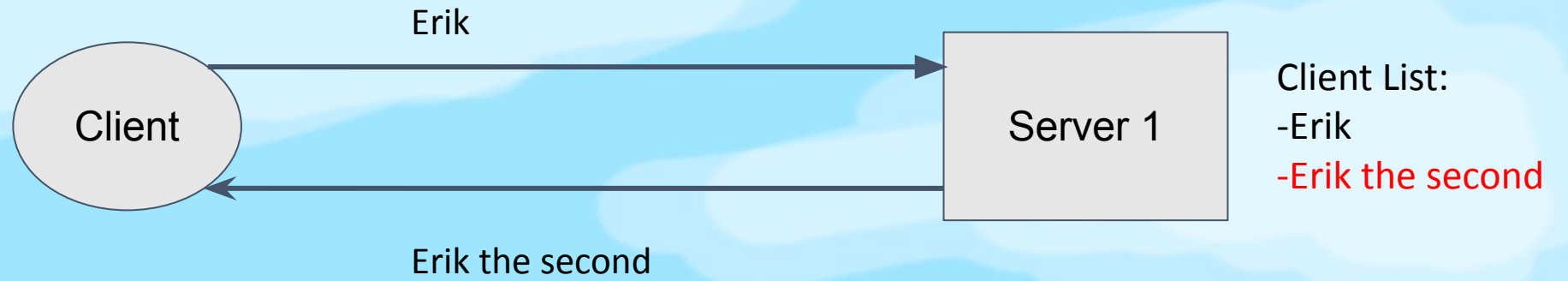




# GUI - Name

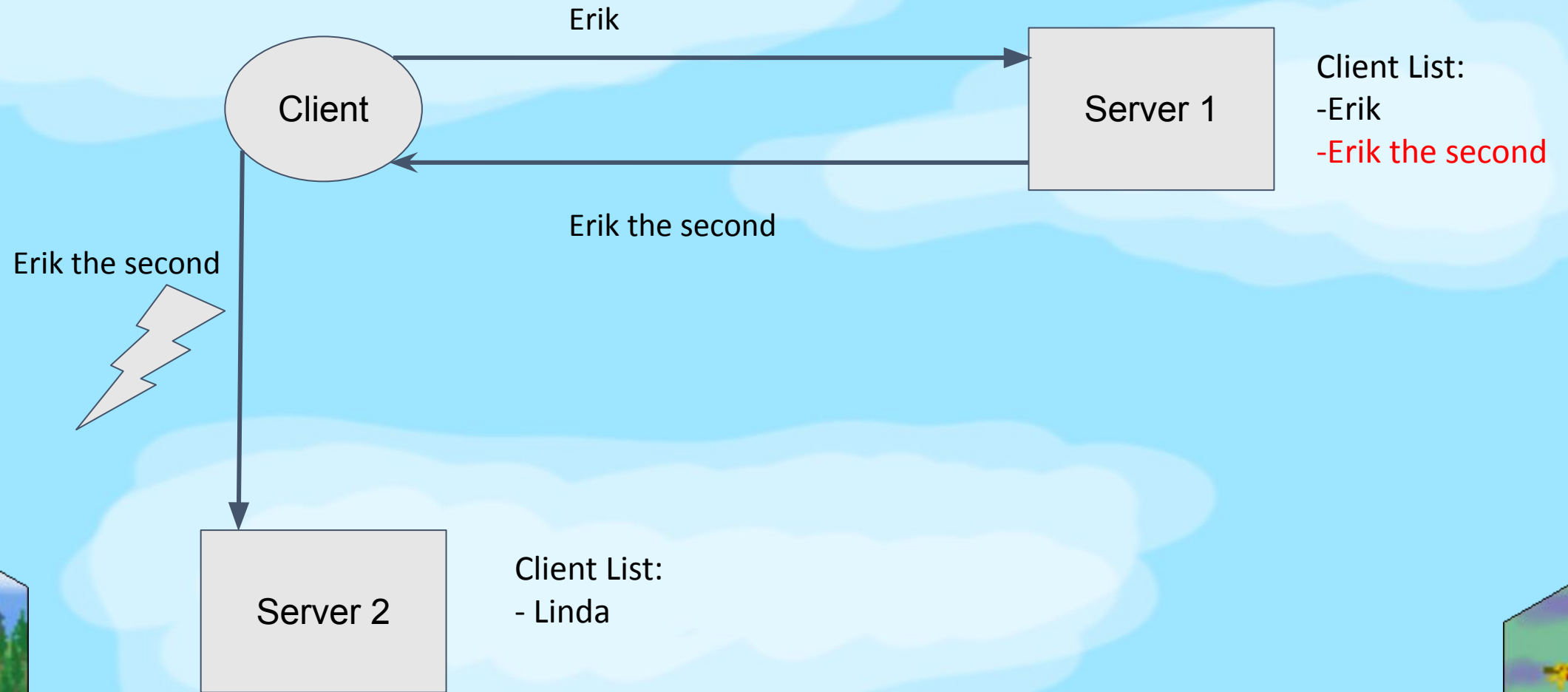


# GUI - Name





# GUI - Name



# Kingdom Builder

Local Game

Online Game

Settings

Exit

# GUI - Einstellungen

Change Language: English ▼

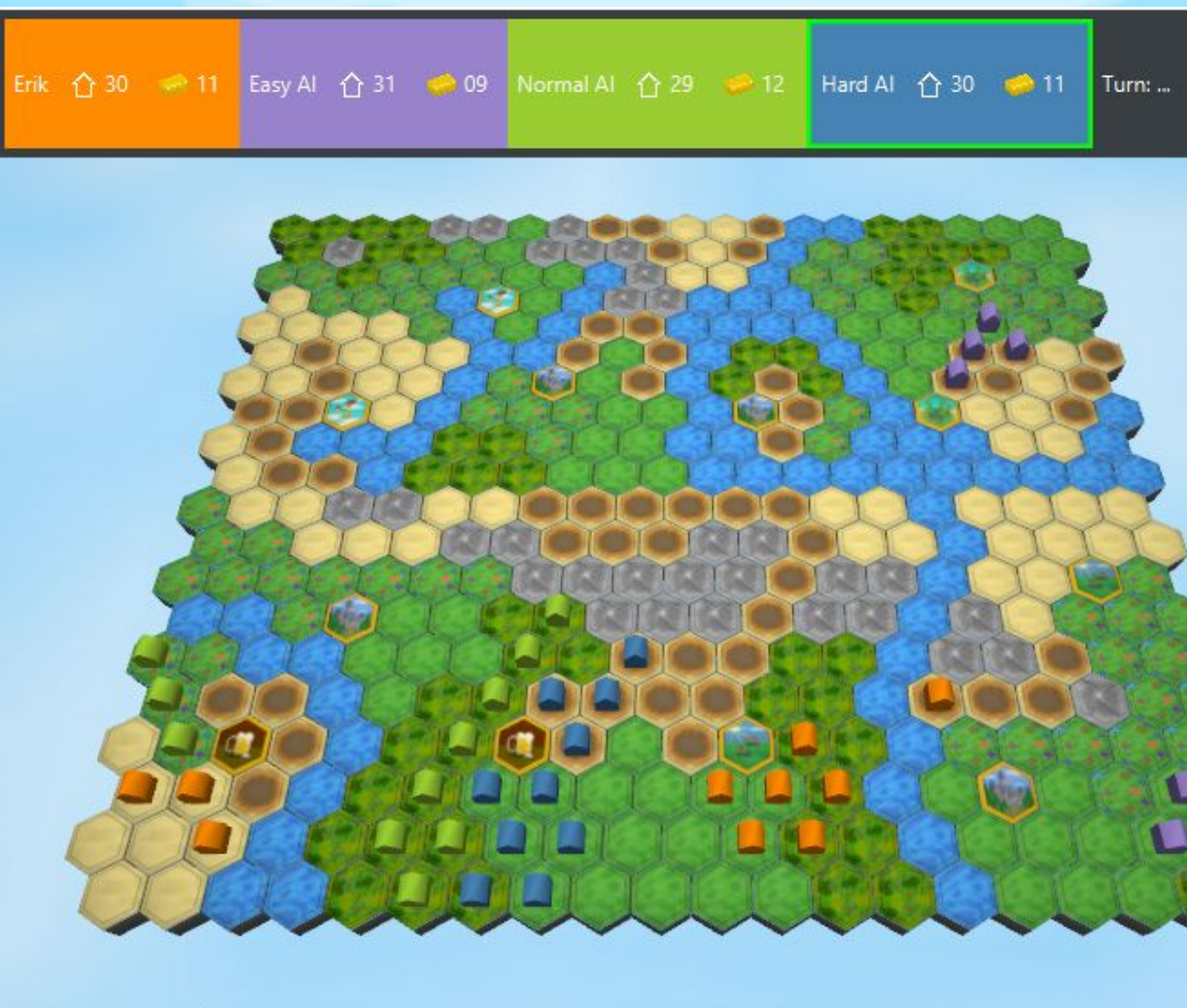
Change your preferred Name:

☐ Changes the colors of the players. Possibly asynchronicity with other players.

Apply



# GUI - Einstellungen





# GUI - Fehlermeldungen

Ändere Sprache: Deutsch ▼

! Invalid Name! Ein valider Name darf keine der folgenden Zeichen enthalten: [ ] ( )

☐ Ändert die Spielerfarben. Mögliche Asynchronitäten mit anderen Spielern entstehen hierdurch.

Anwenden

Abbildungen:  
Übersetzte  
Fehlermeldungen

! Invalid name! A valid name cannot contain any of the following characters: [ ] ( )

KingdomBuilder

Clients

Clear Selection

Back to Menu

Refresh

Filter

ID	Name	Game ID
0	Erik	-1

Global

Game

LOG

<— You are connected to the server —>  
Type #help for further functions

Enter message...

Whisper

Send

ID	Game name	Players	Status
----	-----------	---------	--------

Kein Inhalt in Tabelle

Game name: name

Host: name

Game Description: description

Players players

Level Editor

Create New Game

Join Game

Spectate Game

KingdomBuilder

Clients

Clear Selection

Back to Menu

Refresh

Filter

ID	Name	Game ID
0	Erik	-1

Global

Game

LOG

<— You are connected to the server —>  
Type #help for further functions

Enter message...

Whisper

Send

ID	Game name	Players	Status
----	-----------	---------	--------

Kein Inhalt in Tabelle

Game name: name

Host: name

Game Description: description

Players players

Level Editor

Create New Game

Join Game

Spectate Game

# GUI - Spiel erstellen

Legend:

- Oasis
- Grass
- Paddock
- Mountain
- Barn
- Castle
- Farm
- Water
- Harbor
- Flower
- Oracle
- Forest
- Canyon
- Desert
- Tower
- Tavern

Game name

Game description

Players that can join:

0

☐ Enable Timelimit for game (in min)

5

☐ Enable Turn Limit for Game

50

0 1 2 3



Clients

Clear Selection

Back to Menu

## Game Settings

ID	Name	Game ID
0	Erik	-1

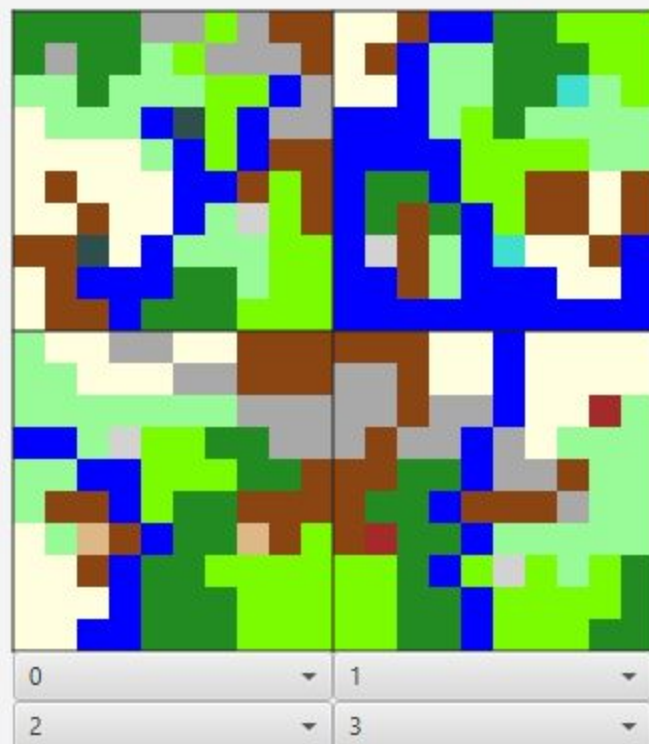
Global Game LOG

<--- You are connected to the server --->  
Type #help for further functions

Enter message...

Whisper

Send



Legend:

- Oasis
- Grass
- Paddock
- Mountain
- Barn
- Castle
- Farm
- Water
- Harbor
- Flower
- Oracle
- Forest
- Canyon
- Desert
- Tower
- Tavern

Game name

Game description

Players that can join:

0

☐ Enable Timelimit for game (in min)

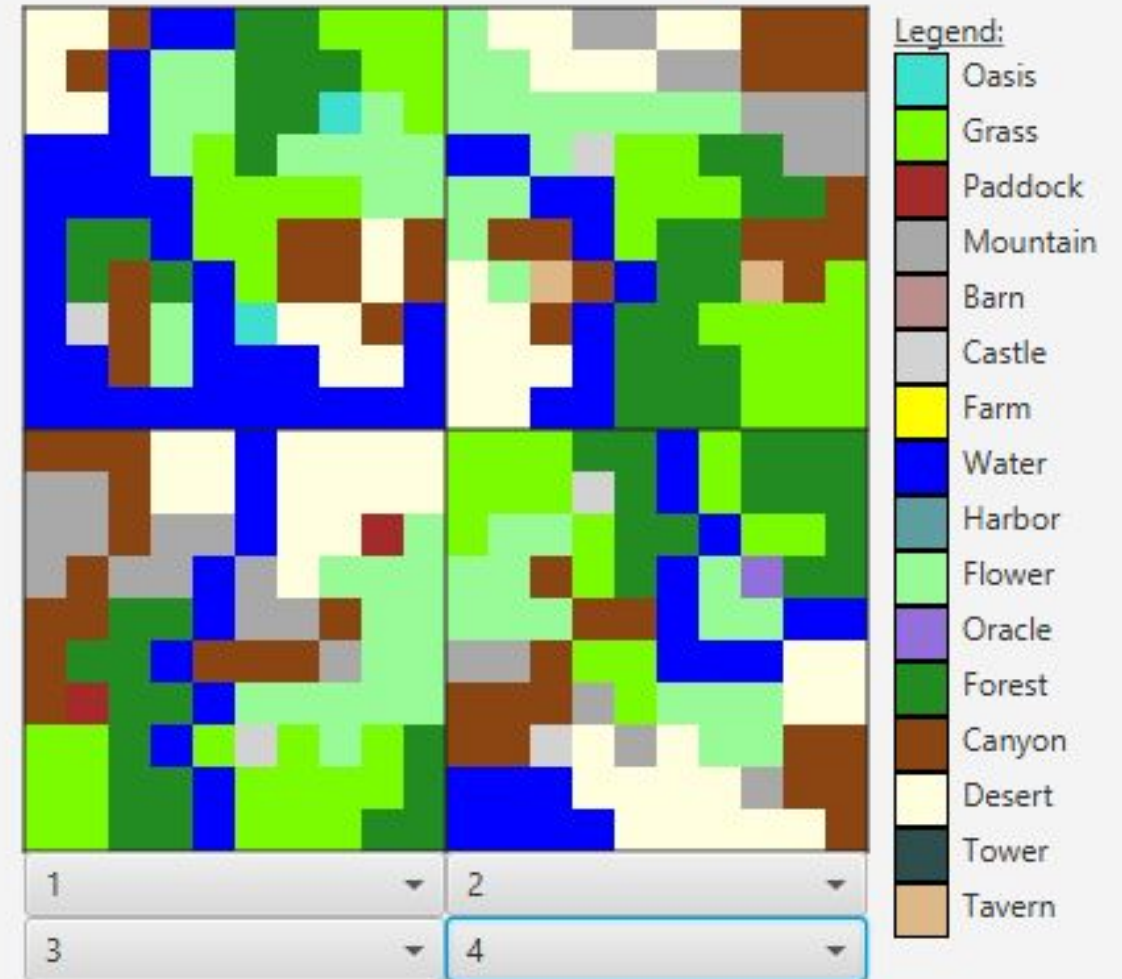
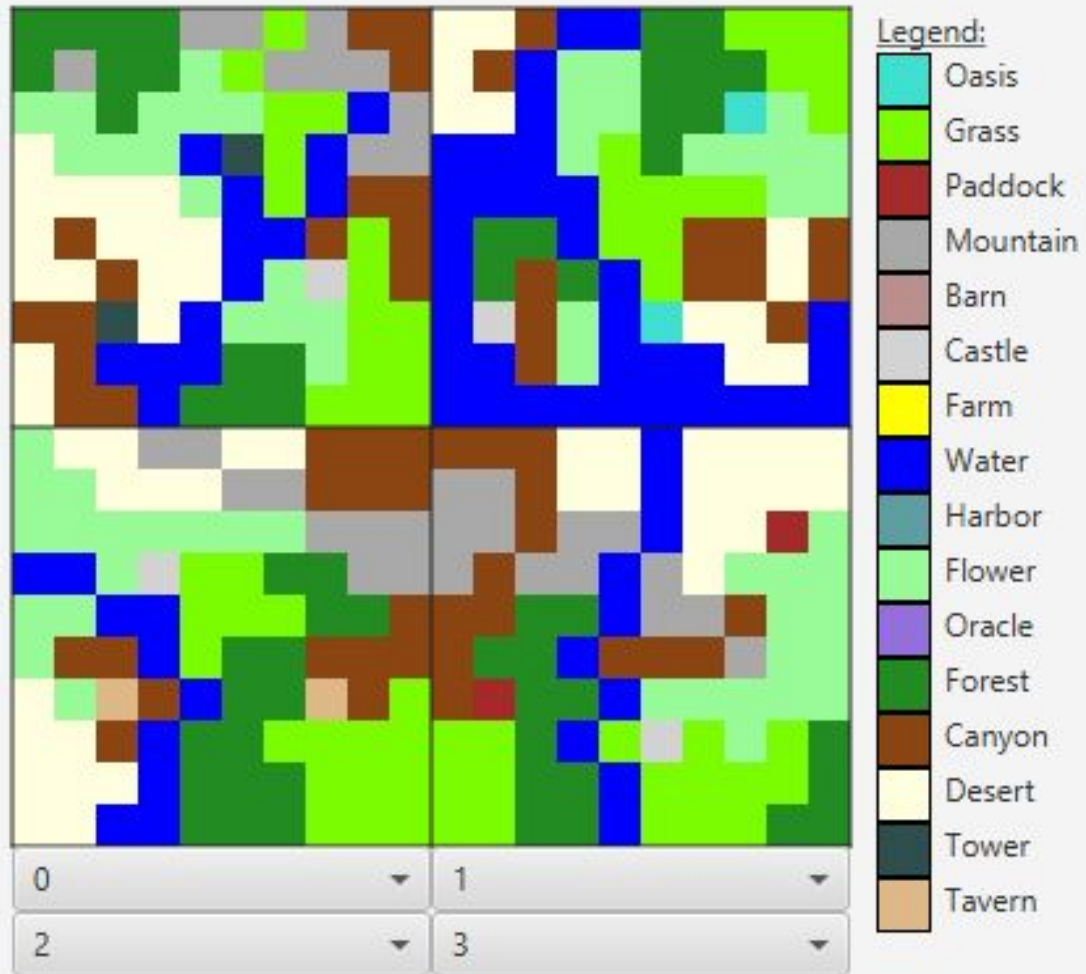
5

☐ Enable Turn Limit for Game

50

Create Game

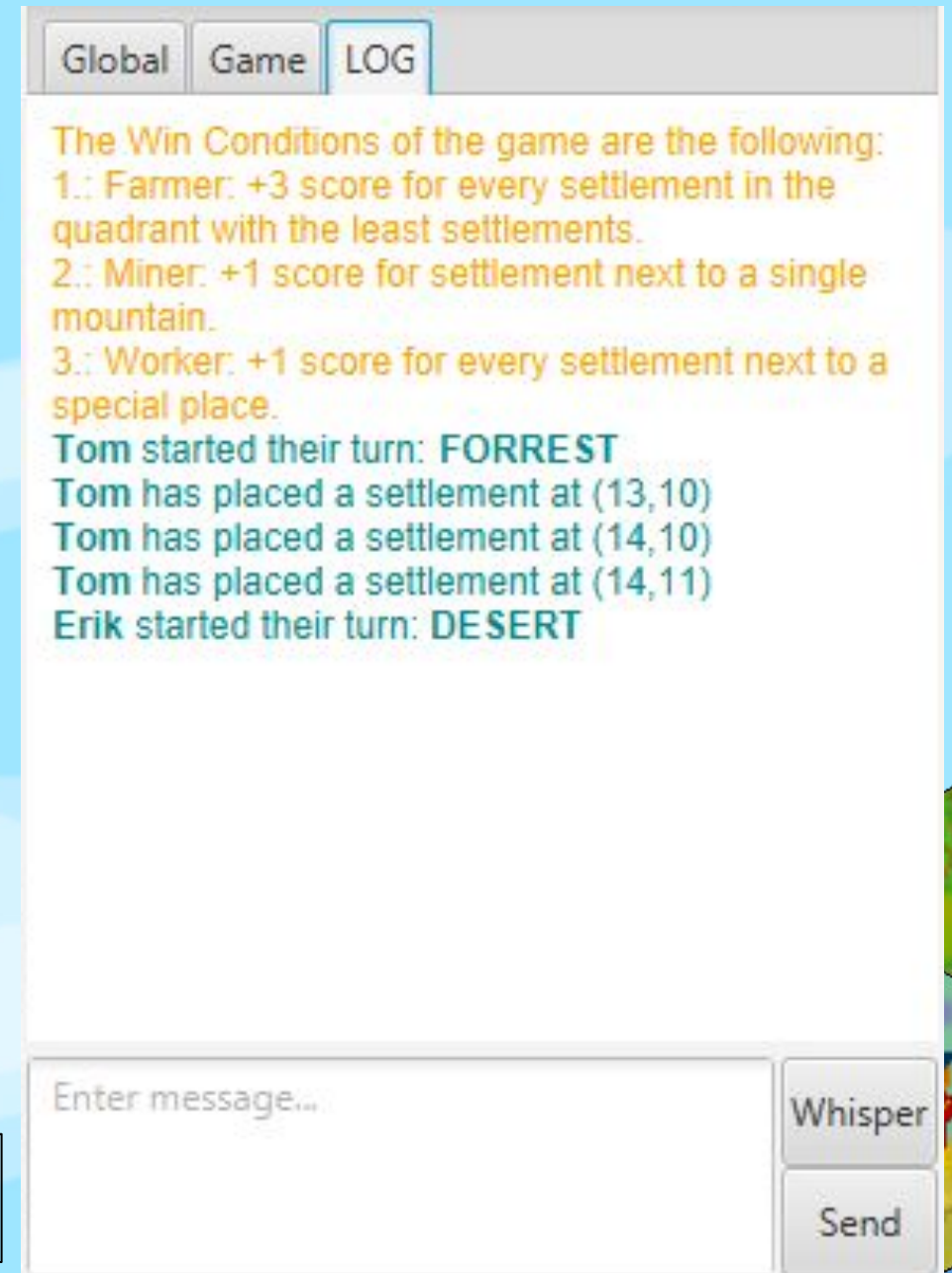
# GUI - Preview



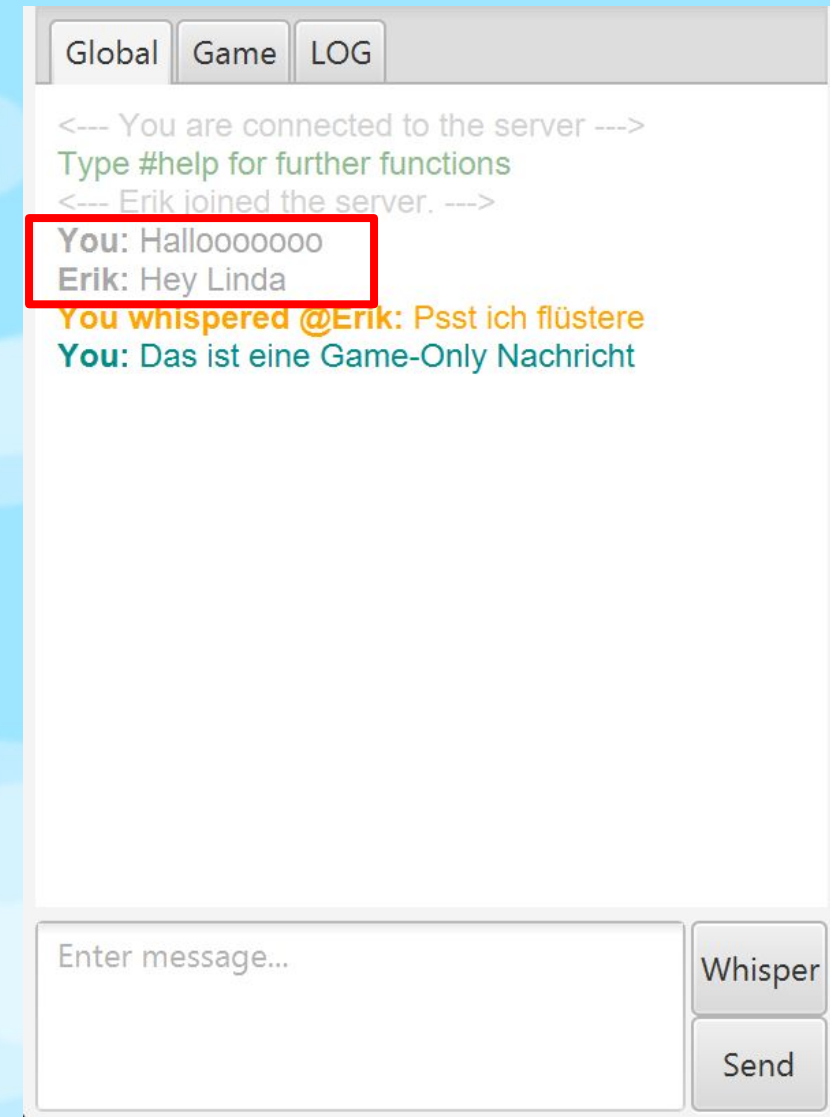
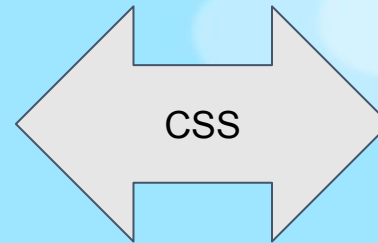
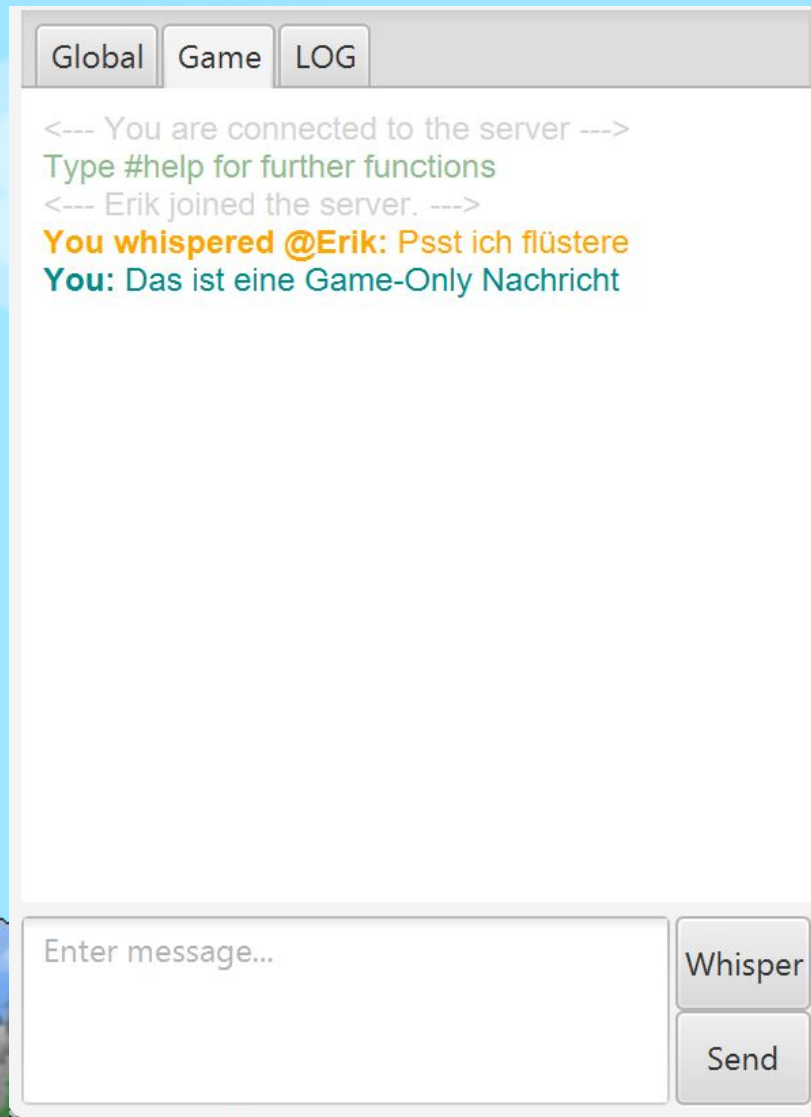
# GUI - Chat & Game Log

- Verschiedene Funktionen:
  - Game Log
  - Flüsternachrichten
  - root
- Besondere:
  - Webview für Farben und JavaScript für automatisches Scrollen

Abbildung:  
Game Log







Abbildungen zeigen Unterschiede, welche mittels CSS Realisiert wurden



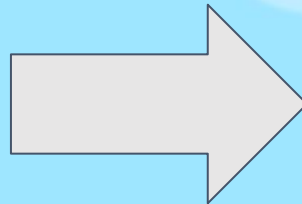
GlobalGameLOG

<--- You are connected to the server --->  
Type #help for further functions  
Following command are available:  
#help: to show this message  
#root <password>: enter the correct password to get the root state  
#shutdown: shutdown the server (only with root state)  
**You:** Um Root zu bekommen:

#root 1234

Whisper

Send



Clients

Kick

Clear Selection

ID	Name	Game ID
0	Burak	-1

GlobalGameLOG

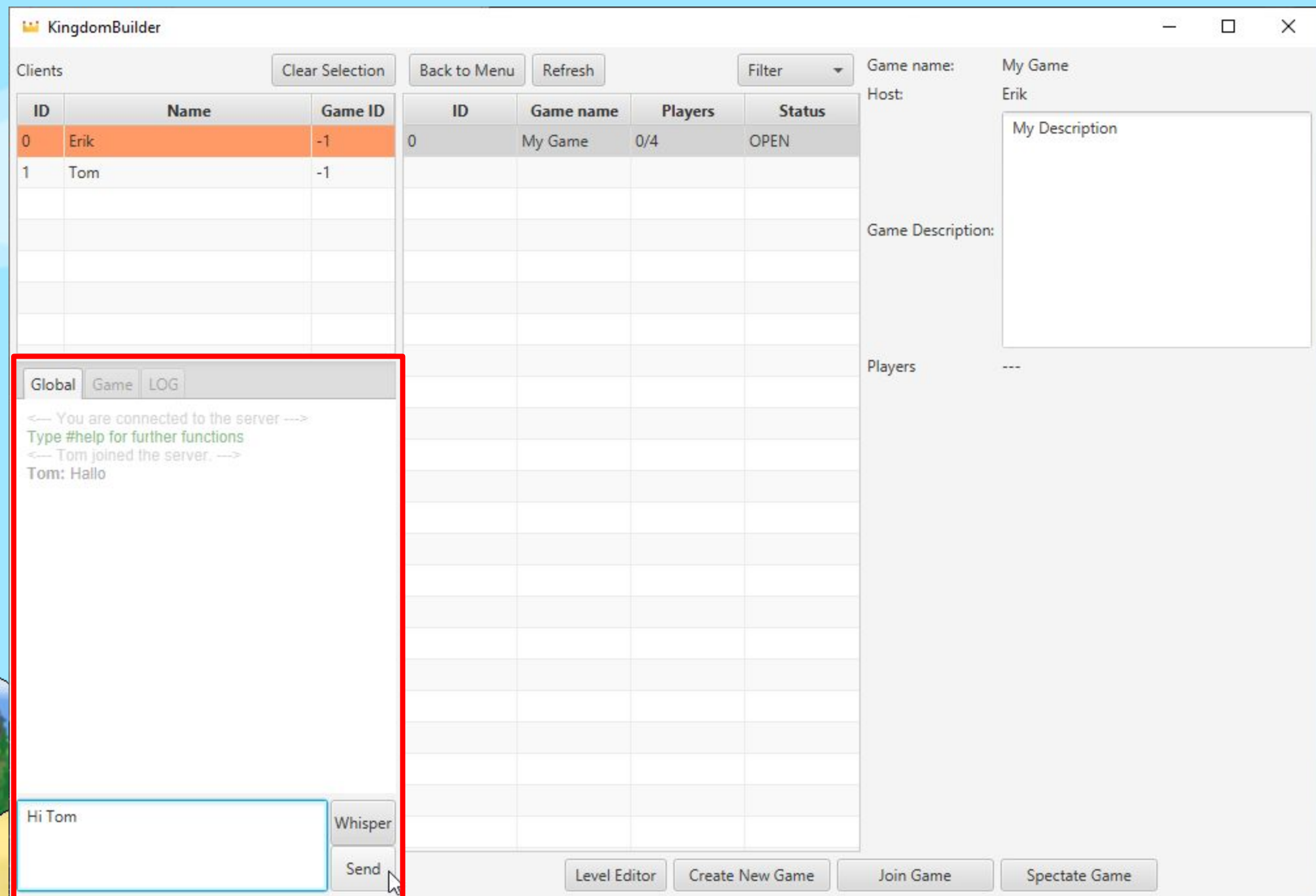
<--- You are connected to the server --->  
Type #help for further functions  
Following command are available:  
#help: to show this message  
#root <password>: enter the correct password to get the root state  
#shutdown: shutdown the server (only with root state)  
**You:** Um Root zu bekommen:  
**You are (g)root!**

Enter message...

Whisper

Send

Abbildung:  
Root Zugang  
mittels Chat



[illegible]

A screenshot of a chat application interface. At the top, there are three tabs: 'Global', 'Game', and 'LOG'. The 'Global' tab is selected. Below the tabs, the chat history shows: '<--- You are connected to the server ---->', 'Type #help for further functions' (in green), '<--- Tom joined the server. ---->', and 'Tom: Hallo'. At the bottom, there is a text input field containing 'Hi Tom' and two buttons: 'Whisper' and 'Send'.





# GUI - Texturen



Besondere:

- alle Texturen selbst gezeichnet

Abbildung: Menü  
Hintergrund

# GUI - Texturen

- Felder des Spielbretts
- verwenden UV Maps
- PLYLoader - Methode
  - vom .ply zu Triangle Mesh

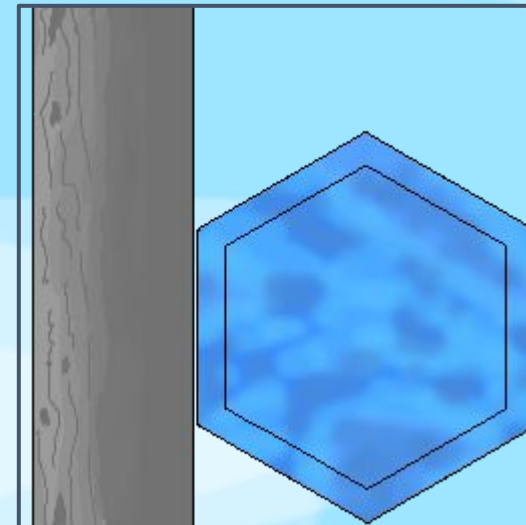


Abbildung:  
Beispiele für  
UV Maps





# GUI - Texturen

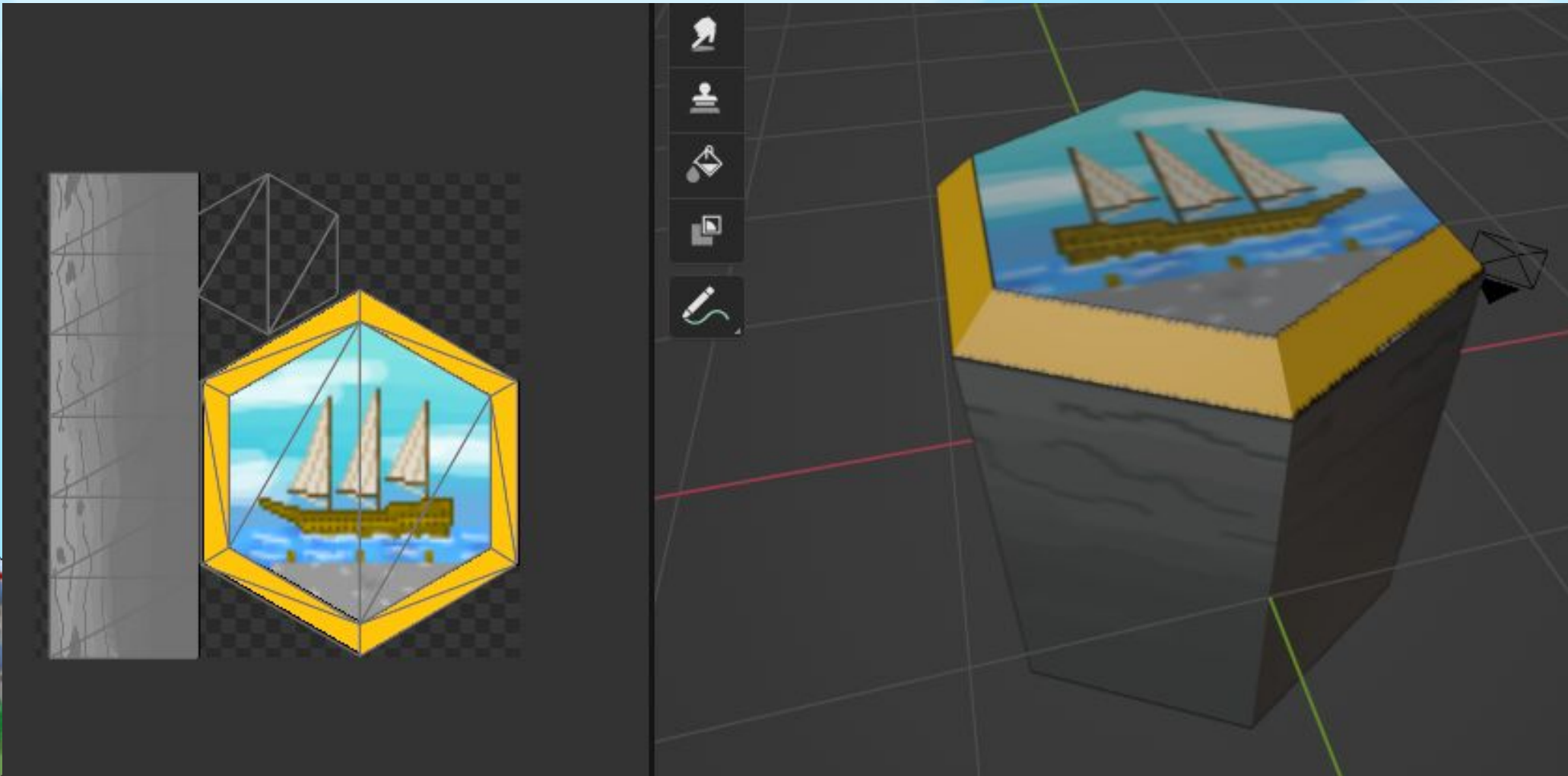


Abbildung: Von 2D in 3D (angezeigt via Blender)

Clients

Clear Selection

Back to Menu

Refresh

Filter

Game name:

name

Host:

name

ID	Name	Game ID
0	Erik	-1

Global

Game

LOG

<--- You are connected to the server --->  
Type #help for further functions

Kein Inhalt in Tabelle

Enter message...

Whisper

Send

Level Editor

Create New Game

Join Game

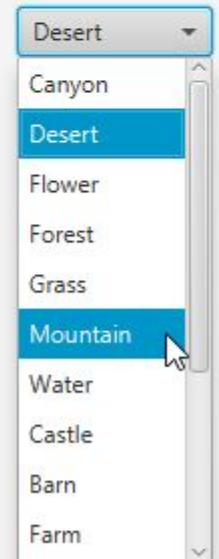
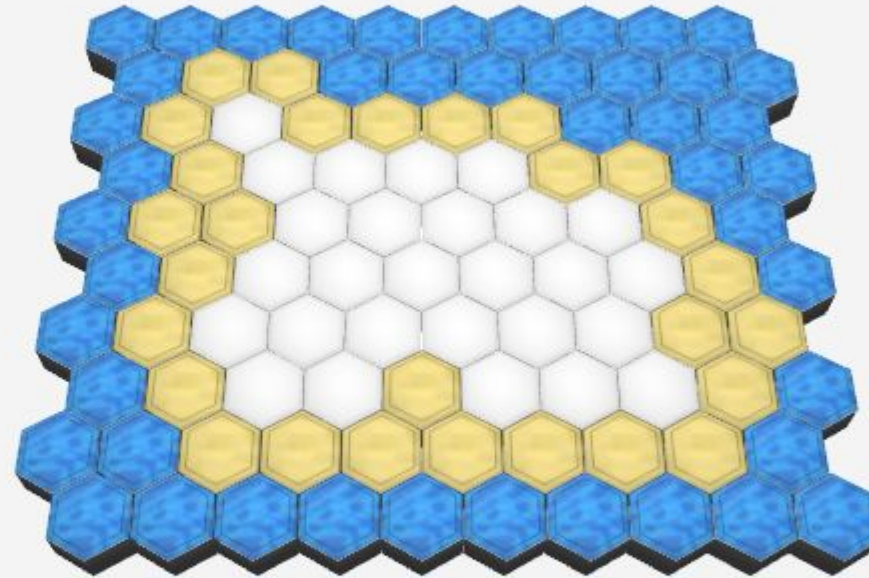
Spectate Game



# GUI - Level Editor

- Besonderes Merkmal:
  - Zeichenfunktion mittels Linksklick (haltend)
- Client überprüft, ob der Quadrant zulässig ist

Abbildung: Level Editor in Aktion



KingdomBuilder

Clients

Clear Selection

ID	Name	Game ID
0	Erik	-1
1	Tom	-1

Global

Game

LOG

<— You are connected to the server —>

Type #help for further functions

<— Tom joined the server. —>

Tom: Hallo

You whispered @Tom: Hi Tom

<— Easy AI joined the server. —>

<— Normal AI joined the server. —>

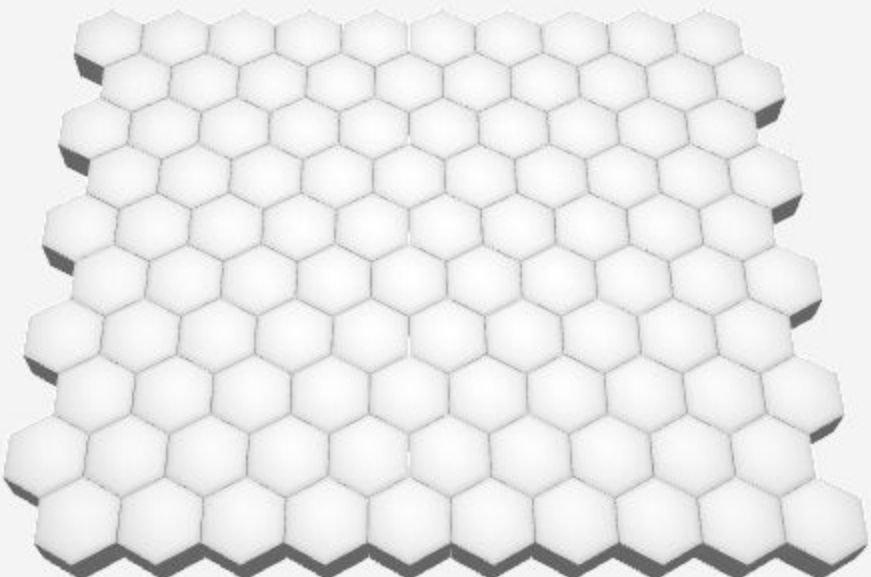
<— Easy AI left the server. —>

<— Normal AI left the server. —>

Enter message...

Whisper

Send



Water

Upload

Cancel

Clients

Clear Selection

ID	Name	Game ID
0	Erik	-1
1	Tom	-1

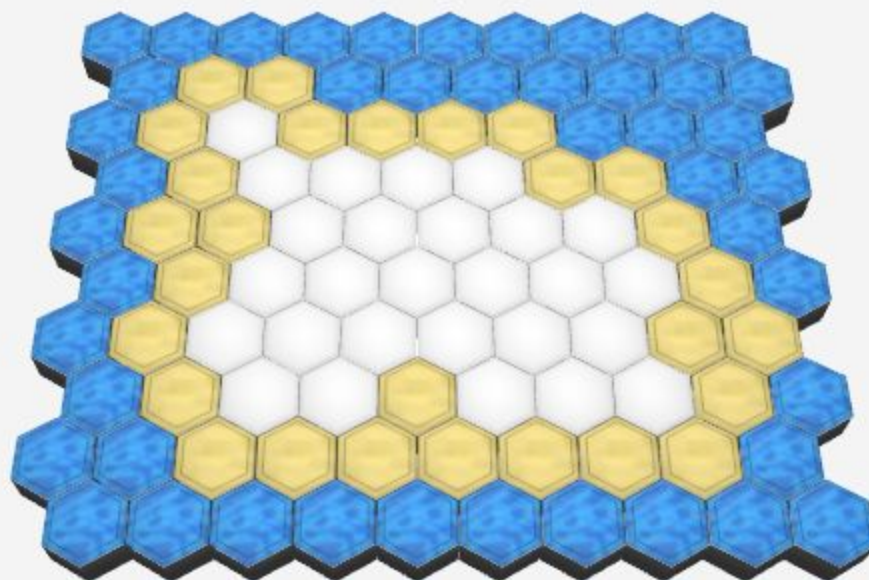
Global Game LOG

← You are connected to the server →  
Type #help for further functions  
← Tom joined the server. →  
Tom: Hallo  
You whispered @Tom: Hi Tom  
← Easy AI joined the server. →  
← Normal AI joined the server. →  
← Easy AI left the server. →  
← Normal AI left the server. →

Enter message...

Whisper

Send



Desert

Canyon

Desert

Flower

Forest

Grass

Mountain

Water

Castle

Barn

Farm



Clients

Clear Selection

ID	Name	Game ID
0	Erik	-1
1	Tom	-1

Global

Game

LOG

&lt;— You are connected to the server —&gt;

Type #help for further functions

&lt;— Tom joined the server. —&gt;

Tom: Hallo

**You whispered @Tom: Hi Tom**

&lt;— Easy AI joined the server. —&gt;

&lt;— Normal AI joined the server. —&gt;

&lt;— Easy AI left the server. —&gt;

&lt;— Normal AI left the server. —&gt;

Enter message...

Whisper

Send



Flower

Upload

Cancel



Clients

Clear Selection

Back to Menu

## Game Settings

ID	Name	Game ID
0	Erik	-1

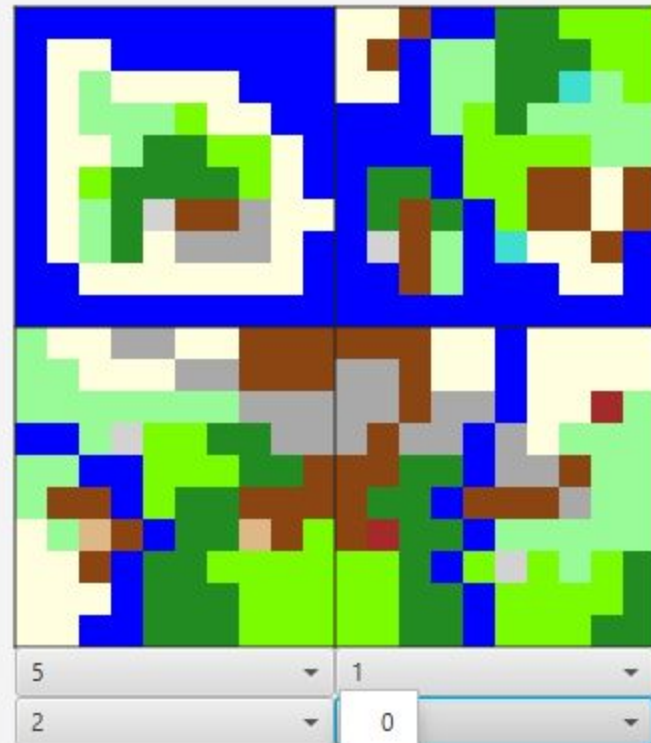
Global Game LOG

<--- You are connected to the server >--->  
Type #help for further functions

Enter message...

Whisper

Send



Legend:

- Oasis
- Grass
- Paddock
- Mountain
- Barn
- Castle
- Farm
- Water
- Harbor
- Flower
- Oracle
- Forest
- Canyon
- Desert
- Tower
- Tavern

Game name

Game description

Players that can join:

0

☐ Enable Timelimit for game (in min)

5

☐ Enable Turn Limit for Game

50

Create Game

# GUI - 3D Spielbrett

- Schatten/Wolken
  - besitzt States: STOPPED, FADE\_IN, PLAYING, FADE\_OUT
  - garantiert flüssiges Ein- / Ausblenden
  - animierte Bewegung der “Wolken” / Pixel
  - Schatten zum besseren Hervorheben der möglichen Züge
- Animationen
  - Bewegen von Settlements wird Visualisiert





Abbildung:  
Spielbrett vor  
Spielbeginn





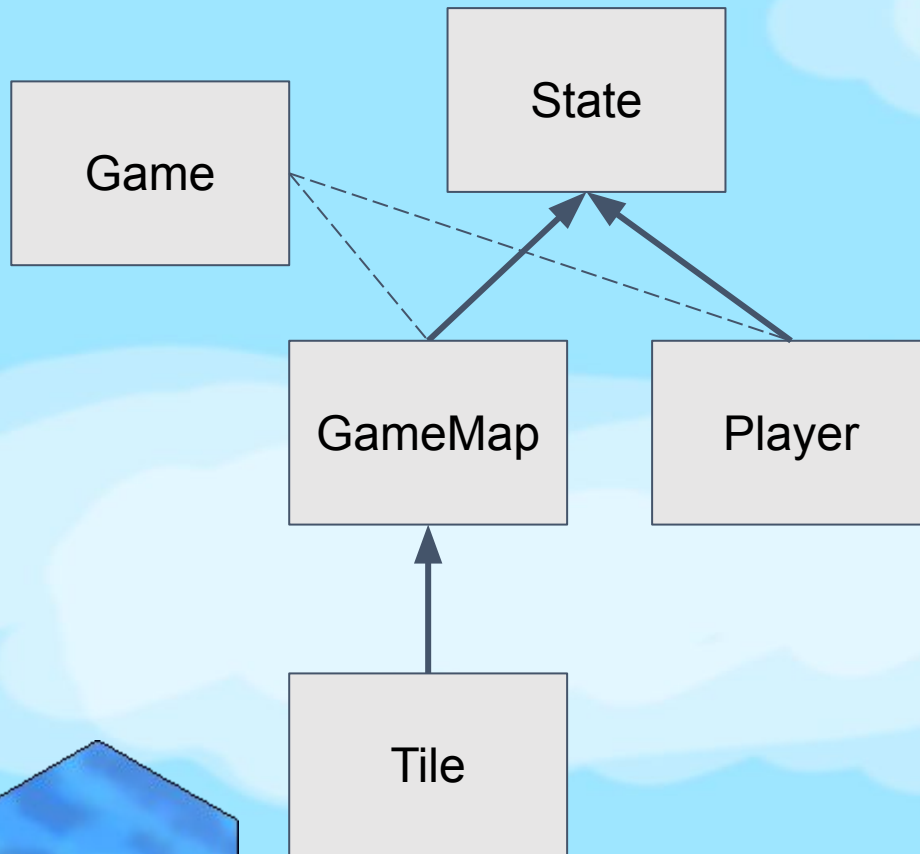
Abbildung:  
Spielbrett mitten  
im Spiel



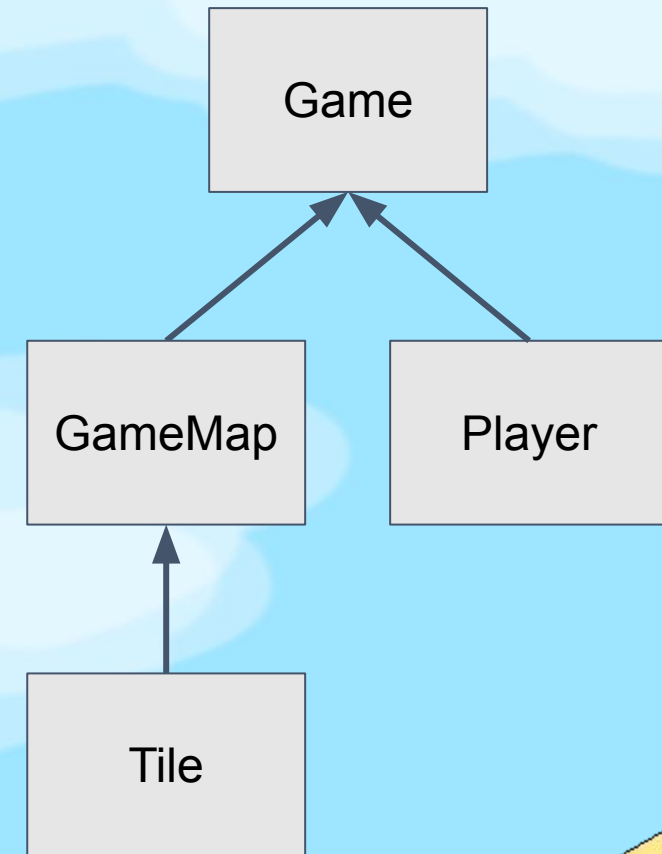


# Gamelogic

Redux

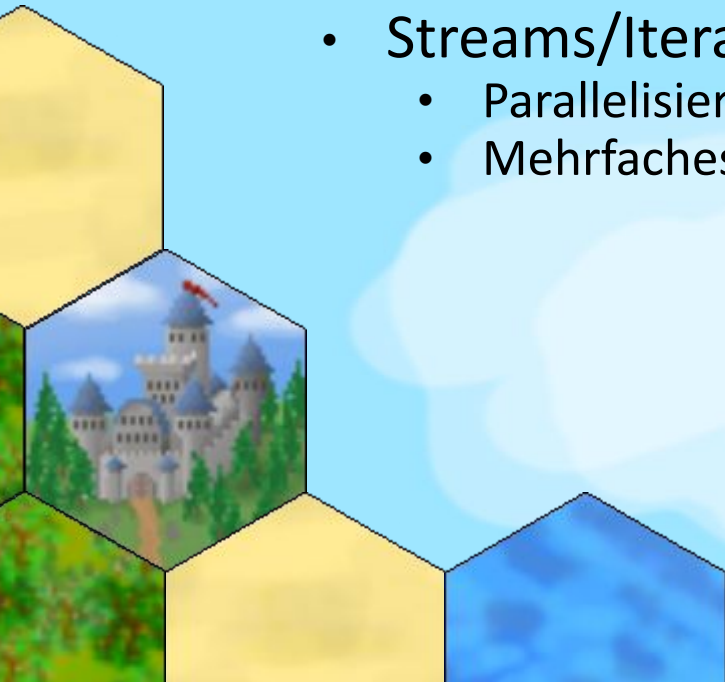
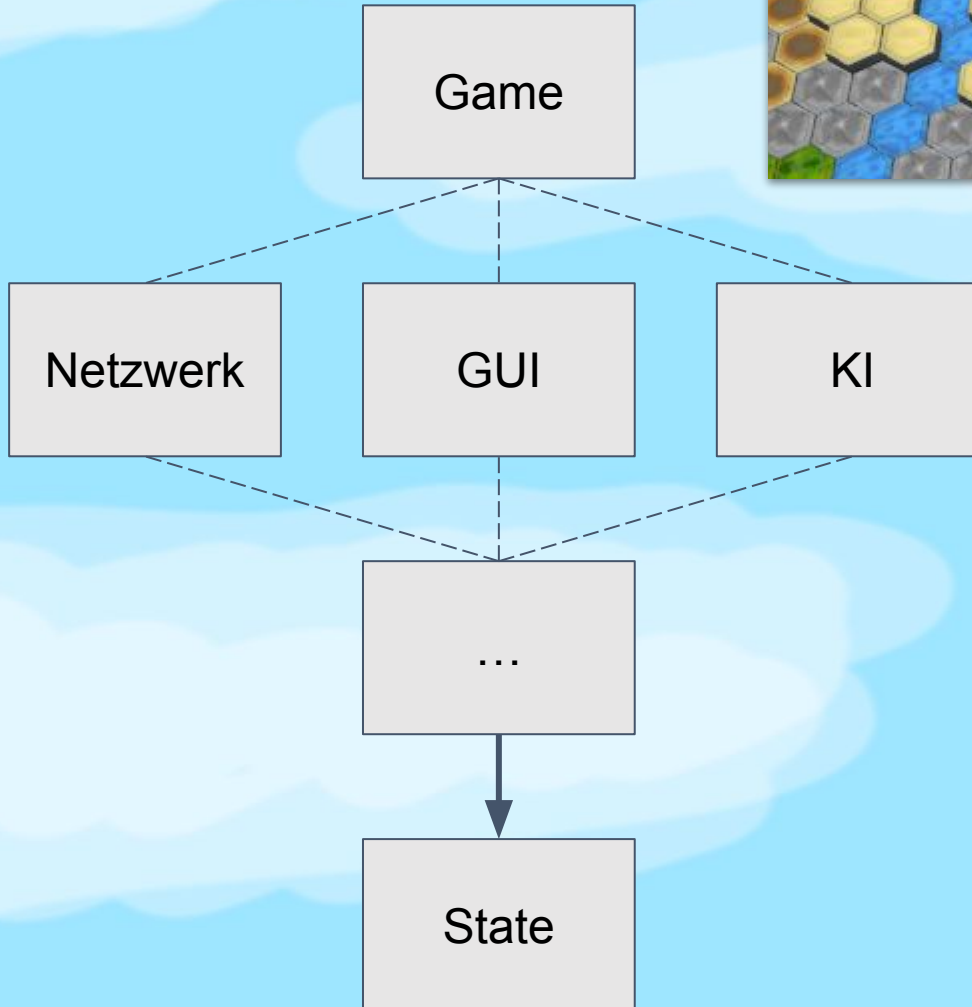


Objektorientiert



# Gameologic - Anbindung

- Züge
  - Client
  - Server
- Game
  - Statische Methoden
  - Streams/Iteratoren
    - Parallelisierbar
    - Mehrfaches Filtern



# Künstliche Intelligenz

- Verschiedene Schwierigkeitsstufen
  - Easy, Normal, Hard und Expert
- Zwei verschiedene Implementierungen
  - Normal und Hard
  - Expert
- Gewinnstrategie mittels Greedy-Algorithmus





# KI - Erster und wichtigster Zug





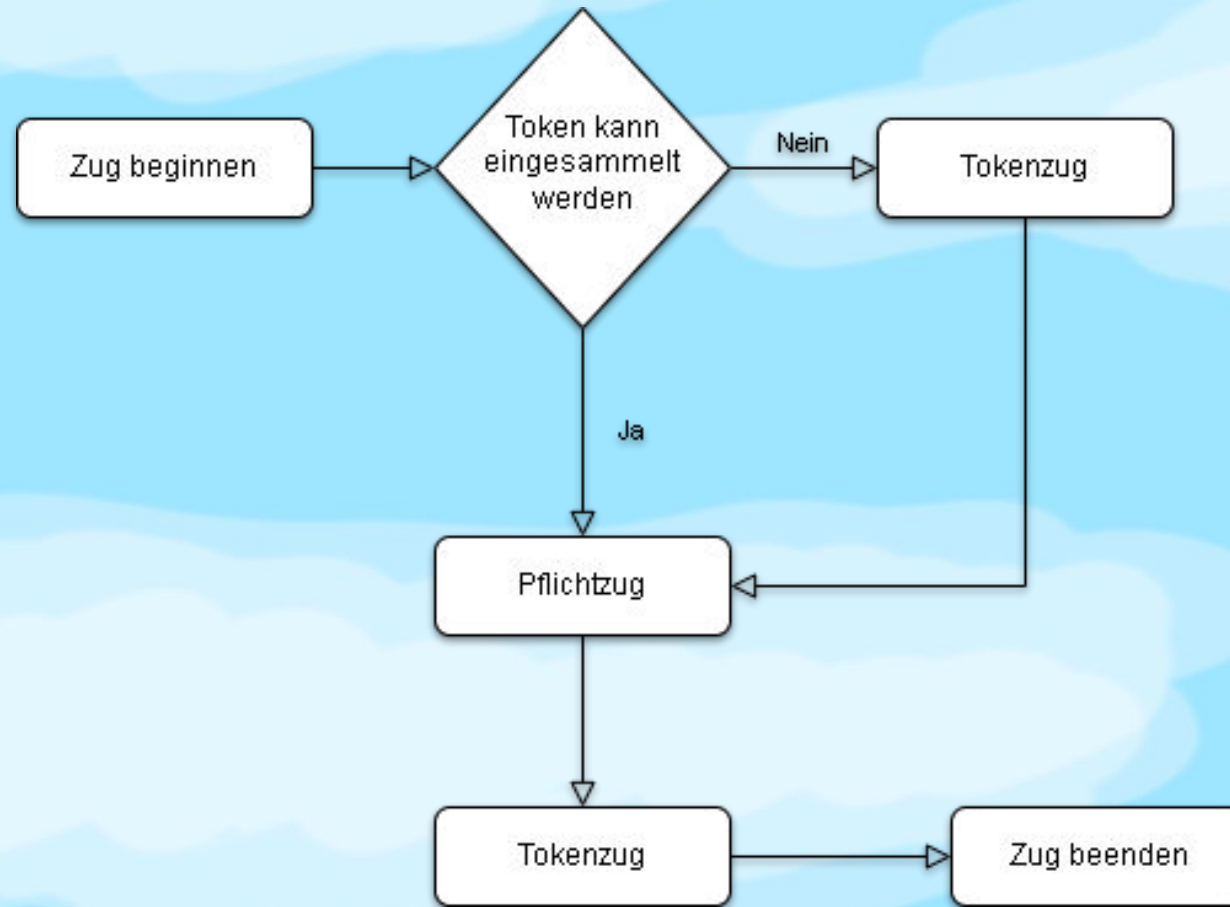
# KI - Priorisierung der Tokens

nach Win Conditions:

- Lords:
  - suche nach Token die platzieren
- Knight und Explorer:
  - Tavern
  - Tower
- Anchorite:
  - Paddock

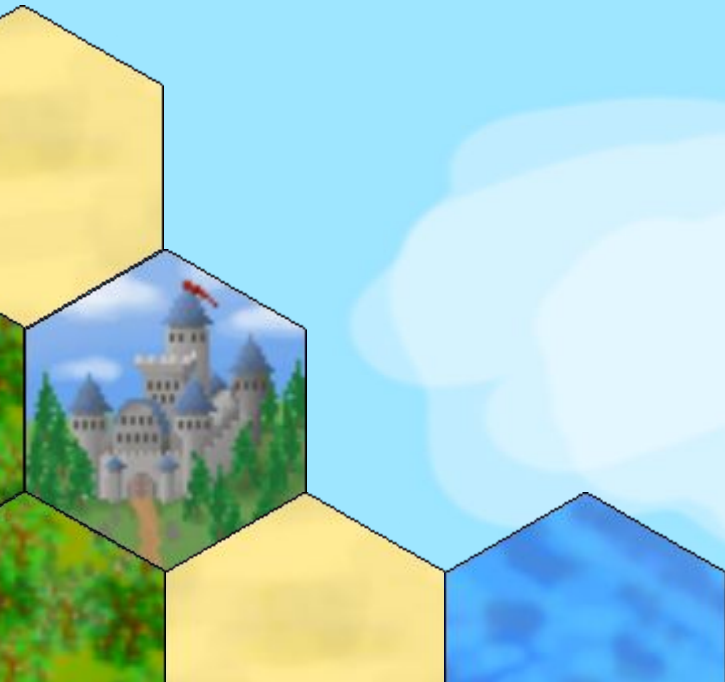


# KI- Allgemeiner Ablauf eines Zuges



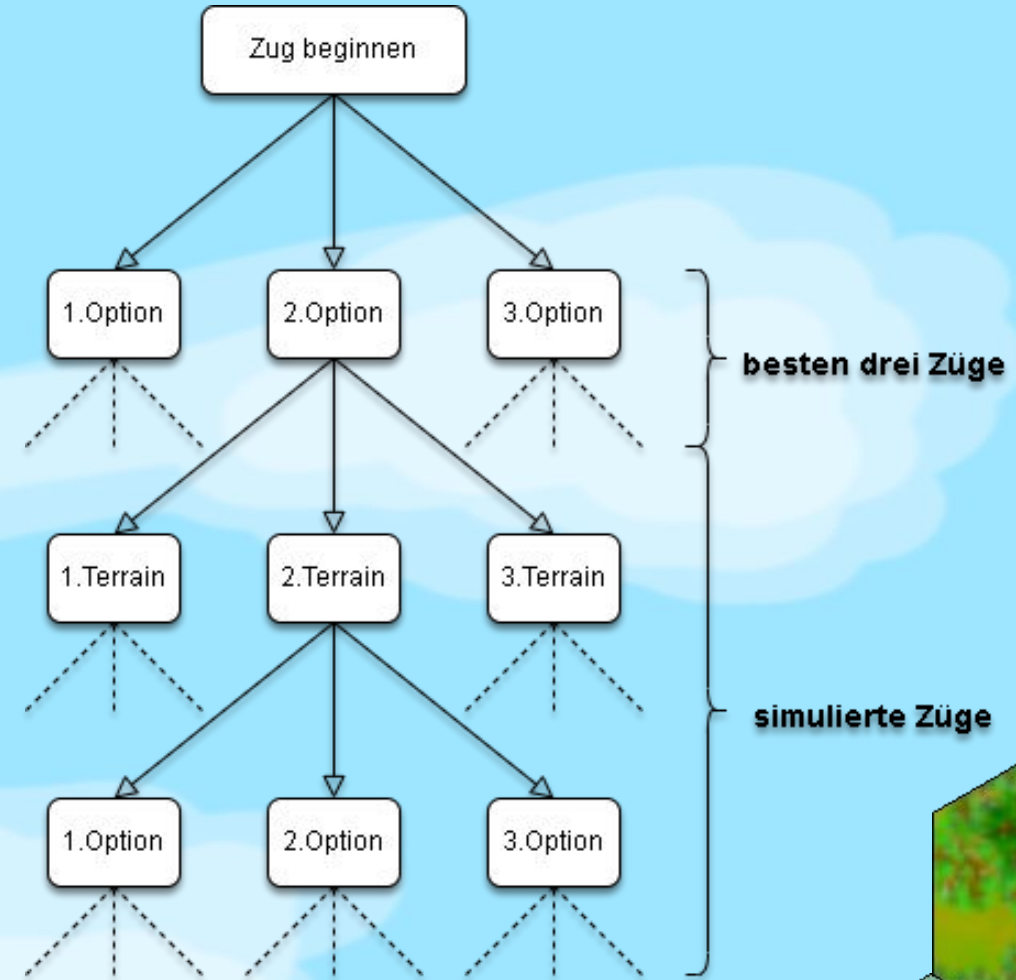
# KI- Normal und Hard

- betrachtet nur das nächste Settlement
- berechnet jedes mal den aktuellen Stand
- schaut nicht weiter voraus



# KI- Expert

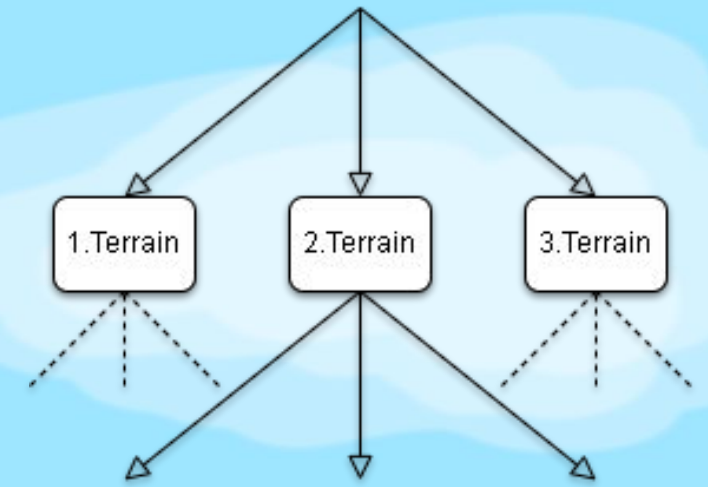
- betrachtet alle möglichen Kombinationen von Platzierungen in jedem Zug
- simuliert noch den darauffolgenden Zug
- wählt am Ende den besten Zweig aus, indem die Summe berechnet wird
- spielt anschließend den ersten Zug





# KI- Expert

- Wahl der Terrain-Karte des nächsten Zugs:
  - Karten werden beim Spiel gezählt
  - Lediglich 3 mit der höchsten Wahrscheinlichkeit, gezogen zu werden, werden betrachtet
- Grund: Effizienz
  - Züge der Mitspieler werden auch nicht beachtet



already played

Forest:	2 / 5
Desert:	4 / 5
Flower:	1 / 5
Gras:	2 / 5
Canyon:	5 / 5



# KI- Hard vs Expert

- Hard:
  - Gewinnt öfters bei Win Conditions, die auf kurze Sicht Punkte erbringen
- Expert:
  - Gewinnt häufiger bei Win Conditions, die auf lange Sicht Punkte erbringen

Im Allgemeinen kommt es darauf an, welche Terrain Karten gezogen werden

EASY		NORMAL		HARD		EXPERT
	29		49		44	60
	23		36		43	34
	28		47		44	47
	23		62		50	56
	12		14		29	38
	30		58		47	73
	38		55		57	42
	21		41		35	33
	22		52		57	54
	31		38		47	43
	27		39		37	46
	23		50		49	57
	17		42		22	48
	20		35		35	63
	20		41		53	33
	28		44		47	55
	27		27		49	52
	24		35		70	59
	17		29		56	34
	15		48		51	41

# Demonstration

- Kleinigkeiten zum Vorzeigen (geht aus Bildern nicht hervor)
  - Zoomen/Pannen der Kamera
  - Animation von move tokens
  - Animation von Schatten/Wolken
  - Hover Effekte (Win Conditions, Tokens, Special Places)

