

ARTHUR'S NETWORK

Hail the Grail

Handbuch & Dokumentation

Simon Wallny, Patrick Zumsteg, Phillip Oldenburg, Martin Jehle, Oliver Engist

Inhalt

Inhaltsverzeichnis

Funktionalität und Spielanleitung.....	3
Das Spiel.....	3
Das Programm starten.....	4
Ein Spiel erstellen.....	6
Regeln und Ablauf.....	6
Das Spiel beenden.....	7
Beispielsession.....	7
Verwendete Libraries.....	8
Credits.....	8
Programmstruktur.....	9
Qualitätsmanagement.....	9

Funktionalität und Spielanleitung

Das Spiel

Hail the Grail ist ein rundenbasiertes Spiel nach dem Prinzip „Capture the Flag“. Die Spieler werden in zwei gleich grosse Teams unterteilt, die im Wettkampf gegeneinander den heiligen Gral suchen. Jedes Team besitzt eine Basis mit einem Abgabepunkt für den Gral, und eine Spawn-Plattform, die in der Nähe der gegnerischen Basis zu finden ist. Ziel des Spieles ist es, den Gral in der eigenen Basis abzuliefern. Jeder Spieler kontrolliert einen Gralsritter. Diese Ritter haben zwei wesentliche Ressourcen, die ihren Handlungsspielraum bestimmen:

- Lebenspunkte: Jeder Gralsritter besitzt einen Lebens-Pool, der 100 Lebenspunkte fassen kann. Wenn der Ritter zu Beginn des Spieles oder nach einem Respawn in der Welt platziert wird, wird dessen Lebenspool aufgefüllt. Sinken die Lebenspunkte eines Ritters auf 0, wird er bewusstlos und wird in der nächsten Runde an seinem Spawnpunkt wiederbelebt.

- Ausdauer: Der Ritter besitzt auch einen Pool an Ausdauerpunkten, der maximal 10 Punkte fasst. Jede Aktion, die der Ritter ausführen kann, also Bewegung, Angriff oder das Platzieren von Fallen, kosten den Ritter eine vorbestimmte Anzahl an Ausdauerpunkten. Dieser Ausdauerpool wird zu Beginn jeder Runde wieder aufgefüllt.

Jeder Gralsritter kann in seinem Zug beliebig viele der folgenden Aktionen ausführen, solange er die Ausdauer dafür besitzt:

- Bewegen: Der Ritter kann sich auf ein Feld bewegen, das links, rechts, oben oder unten an seine aktuelle Position angrenzt. Diese Aktion kostet einen Aktionspunkt.

- Angreifen: Wenn der eigene Ritter direkt neben einem Ritter aus dem gegnerischen Team steht, kann er diesen angreifen, indem er versucht, sich auf das Feld des gegnerischen Ritters zu bewegen. Diese Aktion kostet den eigenen Ritter 2 Ausdauerpunkte und senkt die Lebenspunkte des gegnerischen Ritters um 25.

- Eine Falle platzieren: Ein Ritter kann auf dem Feld, auf dem er sich gerade befindet, eine von acht verschiedenen Fallen platzieren. Diese kosten je nach Fallentyp mehr oder weniger Ausdauerpunkte. Fallen werden nach dem Platzieren unsichtbar, sowohl für die Gegner als auch die Ritter selbst. Es kann zudem nur eine Falle pro Feld platziert werden; falls eine neue Falle platziert wird, wird die alte ersetzt.

Wenn der Ritter sich nun auf ein Feld bewegt, auf dem eine Falle des gegnerischen Teams liegt, löst er diese aus und erhält je nach Typ der Falle Schaden oder einen Statuseffekt. Die Falle wird dabei zerstört. Die Effekte der Fallen sind folgende:

- **Reveal**: Die Position des Ritters wird für 5 Runden an alle anderen Spieler übermittelt. Das Platzieren dieser Falle kostet einen Ausdauerpunkt.

- **Blind**: Der Ritter sieht während drei Runden nur noch die an ihn angrenzenden Felder und den Gralsträger, sofern dieser existiert. Das Platzieren dieser Falle kostet 3 Ausdauerpunkte.

- **Slow**: Während drei Runden kostet die Bewegung den Ritter zwei anstelle von einem Ausdauerpunkt. Das Platzieren dieser Falle kostet 3 Ausdauerpunkte.

- **Stun**: Der Zug wird sofort beendet, und der Ritter ist im nächsten Zug handlungsunfähig. Das Platzieren dieser Falle kostet 5 Ausdauerpunkte.

- **Poison**: Der Ritter wird vergiftet und erleidet während 5 Runden am Ende jeder Runde 10 Punkte Schaden. Das Platzieren dieser Falle kostet 4 Ausdauerpunkte.

- **Prone**: Der Ritter erhält während 4 Runden durch Angriffe oder Dispel-Fallen doppelten Schaden. Giftschaden wird von diesem Effekt nicht beeinflusst. Das Platzieren dieser Falle kostet 5 Ausdauerpunkte.

- **Dispel**: Der Ritter erhält 40 Schadenspunkte. Diese Falle zieht dem Ritter nur Lebenspunkte ab, sie appliziert keinen Statuseffekt. Das Platzieren dieser Falle kostet 4 Ausdauerpunkte.

- **Purge**: Der Ritter erleidet sofort kritischen Schaden und stirbt. Das Platzieren dieser Falle kostet 10 Ausdauerpunkte.

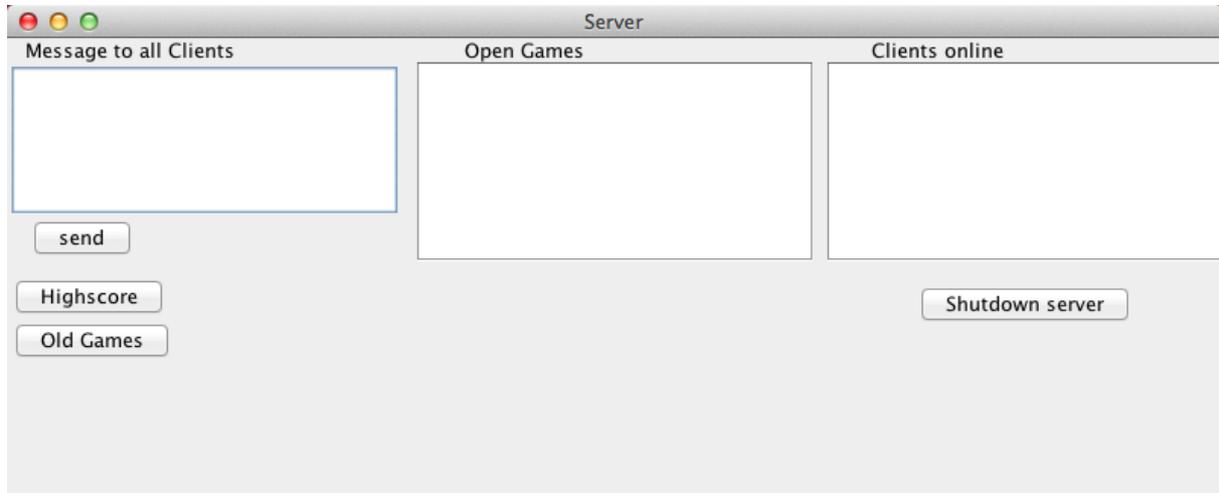
Die Effekte und Kosten dieser Fallen können ausserdem in-game via Tooltips eingesehen werden.

Das Programm starten

Das Spiel wird über ein Jar gestartet. Der Server kann über die Kommandozeile durch den Befehl

```
java -jar HailTheGrail.jar server [port]
```

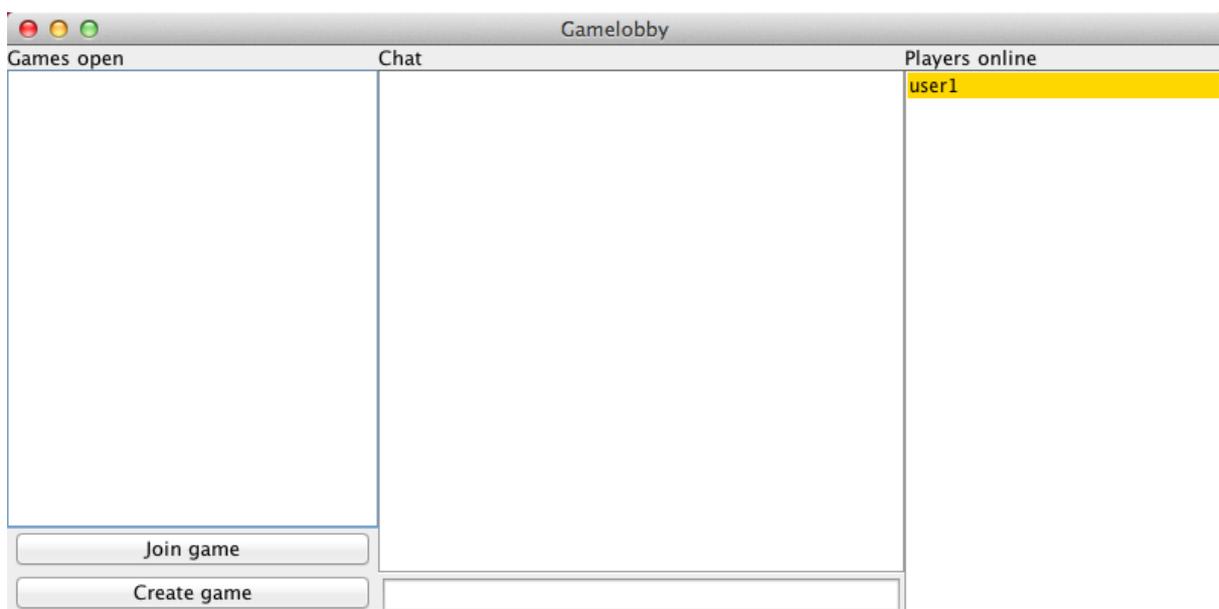
gestartet werden. Die Eingabe eines Ports ist dabei optional, im Unterlassungsfall wird ein Fenster geöffnet, in dem der Port eingegeben werden kann. „server“ kann ausserdem durch host oder s ersetzt werden. Bei erfolgreichem erstellen eines Servers erscheint ein GUI, mit dem grundlegende Funktionen des Servers genutzt werden können:



Auf ähnliche Weise wird auch der Client gestartet:

```
java -jar HailTheGrail.jar client [ip]:[port]
```

Die IP ist hierbei die IP-Adresse des Rechners, auf dem der Server läuft. Eingabe von IP und Port sind auch hier optional; falls allerdings eine IP-Adresse angegeben wird, muss auch ein Port angegeben werden und umgekehrt. Hier kann „client“ auch durch ein c ersetzt werden. Nach erfolgreichem Einloggen wird der Spieler in der Lobby platziert und es erscheint folgendes GUI:



In der Lobby können die Spieler chatten, Spiele erstellen oder einem laufenden Spiel beitreten. Per default schreiben die Spieler in der Lobby im globalen

Chatraum. Folgende Chat-Kürzel können vor eine Nachricht gesetzt werden, um in anderen Chaträumen zu schreiben:

- /t Beispieltext

Hiermit wird „Beispieltext“ im Teamchat des eigenen Teams geschrieben, der nur für Teammitglieder sichtbar ist.

- /w [Spielername] Beispieltext

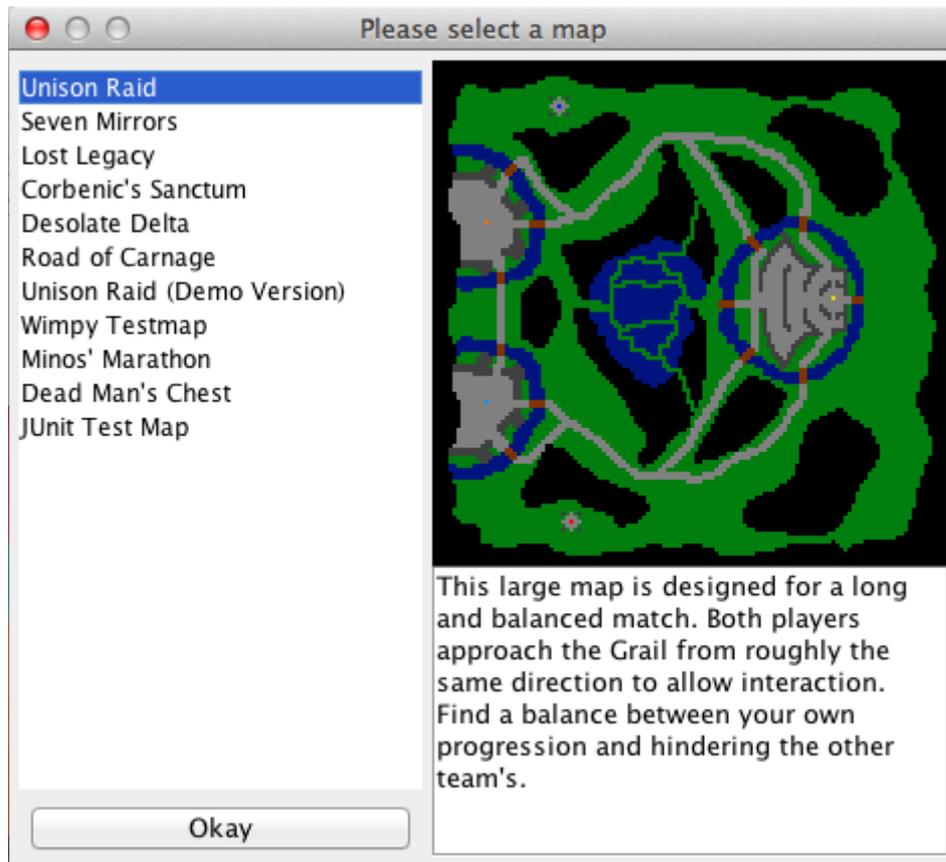
Sendet die Nachricht „Beispieltext“ an den Spieler mit dem angegebenen Spielernamen.

- /g Beispieltext

Sendet die Nachricht „Beispieltext“ im Gamechat. Diesen sehen nur Spieler, die sich momentan im selben Spiel wie der Schreibende befinden.

Ein Spiel erstellen

Beim Erstellen eines Spiels muss zuerst ein frei wählbarer Spielname angegeben werden, unter der Bedingung, dass noch kein Spiel mit diesem Namen existiert. Danach erscheint ein Auswahlmeneu, in dem aus verschiedenen Karten gewählt werden kann:



Nachdem ein Spiel erstellt wurde, sehen diese andere Spieler in der Liste der offenen Spiele und können diesem über die Schaltfläche „Join Game“ beitreten. Jeder Spieler, der diesem Spiel beigetreten ist, kann dieses starten, sofern beide Teams ausgeglichen sind.

Regeln und Ablauf

Hail the Grail ist als rundenbasiertes Spiel implementiert. Zu jedem Zeitpunkt ist nur ein Spieler am Zug. Die Dauer eines solchen Spielzuges ist beliebig; der Zug wird jedoch automatisch abgebrochen, falls die Lebens- oder Ausdauerpunkte des Ritters auf 0 sinken. Der Spieler hat zudem die Möglichkeit, seinen Zug über die „End Turn“-Schaltfläche sofort zu beenden.

Grundsätzlich ist jede Aktion durchführbar, solange die Aktionspunkte dafür zur Verfügung stehen. Es ist ausserdem nicht möglich, sich über unpassierbare Feldtypen zu bewegen. Ungültige Züge lässt der Server nicht zu; der Zug wird nicht ausgeführt, und es werden auch keinerlei Kosten verrechnet.

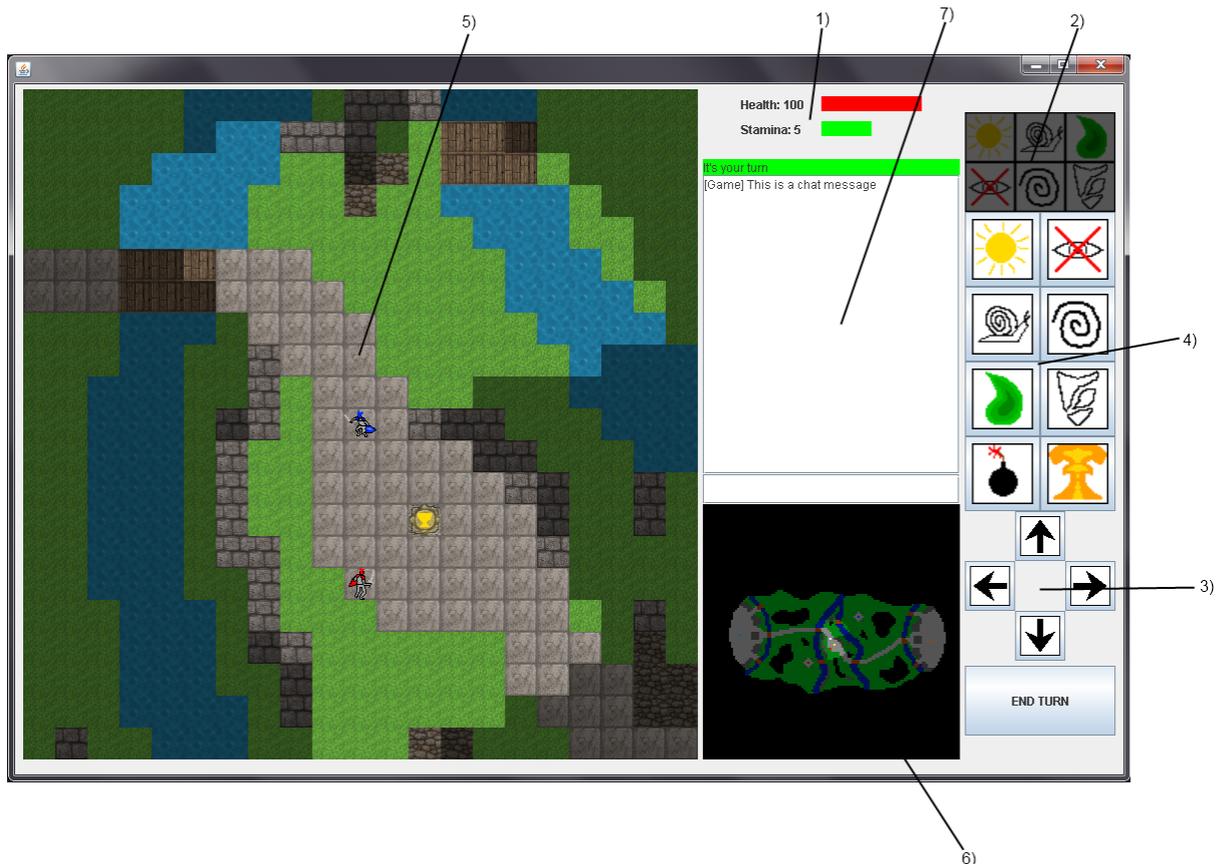
Das Spiel beenden

Das Spiel endet immer mit dem Sieg eines Teams. Dieser kann erreicht werden, indem einer der Ritter den Gral am Abgabepunkt seines Teams abliefern.

Ausserdem verliert, falls einer der Spieler das Spiel verlässt, automatisch das Team dieses Spielers. Wenn das Spiel entschieden wurde, können die Spieler den die Game-Oberfläche schliessen, um in die Lobby zurückzukehren.

Falls einer der Spieler die Verbindung verliert, wird ihm eine gewisse Zeit zum reconnecten geben. Wenn der Spieler nach dieser Zeit nicht zurückkehrt, wird das Spiel beenden und das Team dieses Spielers verliert.

Beispielsession



- 1) Hier werden die verbleibenden Lebens- und Ausdauerpunkte des Ritters angezeigt
- 2) Wenn der Ritter von einem Status effekt betroffen ist, leuchtet das entsprechende Symbol auf. Ein Tooltip gibt ausserdem die verbleibende Dauer jedes Status effekts an.
- 3) Mithilfe dieser Schaltflächen kann der Ritter nach oben, unten, links und rechts bewegt werden. Bewegungs-Befehle können auch über die Pfeiltasten erteilt werden.
- 4) Mit diesen Schaltflächen können die verschiedenen Fallen platziert werden. Kosten und Effekt werden dabei in einem Tooltip angezeigt. Die Fallen können auch mit den Tasten 1-8 platziert werden; dabei steht 1 für Reveal, 2 für Blind, 3 für Slow usw.
- 5) Hier wird das Spielfeld angezeigt, die Kamera ist dabei immer auf den eigenen Ritter zentriert. Die Farbe der Rüstung gibt die Teamzugehörigkeit eines Ritters wieder. Felder, die der eigene oder verbündete Ritter nicht sehen können, werden abgedunkelt dargestellt. Folgende Feldtypen haben besondere Effekte:

	Geröll; blockiert Sicht, unpassierbar		Wasser; unpassierbar
	Spawnpunkt des Grals		Mauer; blockiert Sicht,

			unpassierbar
	Spawnpunkte der Teams in entsprechender Farbe		Abgabepunkte für den Gral in jeweiliger Teamfarbe

6) Die Minimap. Hier wird eine Übersicht über die ganze Karte angezeigt. Der eigene Ritter wird weiss angezeigt, Verbündete cyan und Gegner rot, sofern diese zu sehen sind. Der Gralsträger bzw. der Gral, wenn er am Boden liegt, wird gelb angezeigt. Spawnpunkte eines Teams werden dunkelblau bzw. dunkelrot angezeigt, Abgabepunkte hellblau bzw. cyan.

7) Das Chatfenster. Auf der obersten Zeile wird immer angezeigt, ob man gerade am Zug ist. Darunter werden die Chat-Nachrichten und der Chatraum, in dem sie gesendet wurden, angezeigt. Die untere Zeile dient der Eingabe. Es gelten hier die gleichen Chat-Präfixe wie in der Lobby; /t für den Teamchat, /w Spielernamen für die whisper-Funktion. Der Spieler schreibt hier jedoch per default im Game- und nicht im globalen Chat.

Verwendete Libraries

Hail the Grail verwendet nur die Standard-Libraries, die in jedem Java7 Runtime Environment enthalten sind.

Credits

Sounds: -<http://www.freesfx.co.uk>

Animationen: -<http://forum.chaos-project.com/index.php?topic=8831.0>
-game_guy, YoYo Games

-<http://mrbubblewand.wordpress.com/download/>

Tileset: -Conquest ([Minecraft](#) Texture Pack by [Stefan Krein](#))

-<http://www.minecraftforum.net/topic/1452033-174-32x32-conquest-wip-weekly-updates/>

Programmstruktur

Spiellogik: Zentral für die Spiellogik ist die Map-Klasse; diese kennt das Spielfeld, die Fallen, die Ritter, behandelt Bewegungs-, Angriffs- und Fallen-Kommandos der Clients, und benachrichtigt die Spieler, wenn in ihrer Nähe etwas passiert, wie zum Beispiel zwei kämpfende Ritter oder das Auslösen einer Falle.

Das Spielfeld ist mittels mehrerer Arrays realisiert; es gibt zum einen die Karte selbst, auf der die Feldtypen an ihren jeweiligen Koordinaten gespeichert sind, und zum anderen ein Array für die Fallen. Auch in diesem wird an ihren entsprechenden Koordinaten der Typ einer platzierten Falle gespeichert.

Die wichtigen Daten eines jeden Avatars im Spiel wird serverseitig in Instanzen der Knight-Klasse gespeichert und bei Bedarf an die Clients weitergegeben. Das beinhaltet Informationen wie Leben, Ausdauer, die Position des Ritters, Blickrichtung usw. Jegliche unzulässige Manipulation dieser Daten von Seiten des Clients ist ausgeschlossen, da die Regelüberprüfung komplett serverseitig abläuft.

Für das Auslösen einer Falle wird eine statische Klasse verwendet, die als eine Art Schiedsrichter dient; wenn ein Ritter auf ein Feld tritt, wo das Fallen-Array der Map eine Falle enthält, übergibt diese Fallen-Info-Klasse den entsprechenden Statureffekt direkt an den Ritter.

Zu Beginn und am Ende eines Spielzuges werden bei dem Ritter, der gerade an der Reihe ist, Statureffekte und andere limitierende Faktoren, wie zum Beispiel, ob man gerade den Gral trägt, appliziert.

Die Sicht eines Ritters wird sowohl client- als auch serverseitig berechnet. Welche Felder sichtbar sind und welche nicht, berechnet das Spiel anhand des Bresenham-Algorithmus. Anhand dieses Algorithmus' wird entschieden, ob ein Feld aus der Sicht des Ritters durch ein Hindernis verdeckt wird.

Der Server ermittelt, welche Ritter der Spieler sehen kann, und ob er den Gral sieht, da diese Informationen für die Fairness des Spiels entscheidend sind. Der Client berechnet selbst, welche einzelnen Felder effektiv sichtbar sind und welche sich im „fog of war“ befinden. Dies ist unbedenklich, da diese Information eigentlich rein kosmetisch ist.

Client-Server Kommunikation

Client und Server kommunizieren auf einem einfachen textbasierten Protokoll. Zu übermittelte Parameter werden durch ein Trennzeichen getrennt. Damit Parameter auch diese Trennzeichen enthalten dürfen werden die Parameter mit UTF8 kodiert.

Die Übertragungsschicht wird durch Schnittstellen abstrahiert. Diese enthalten Methoden wie z.B. „login(name)“ um die Kommunikation zu vereinfachen.

Außerdem werden empfangene Nachrichten, welche an die Spiellogik gelangen müssen via Java-Interface übermittelt.

Qualitätsmanagement

JUnit-Test

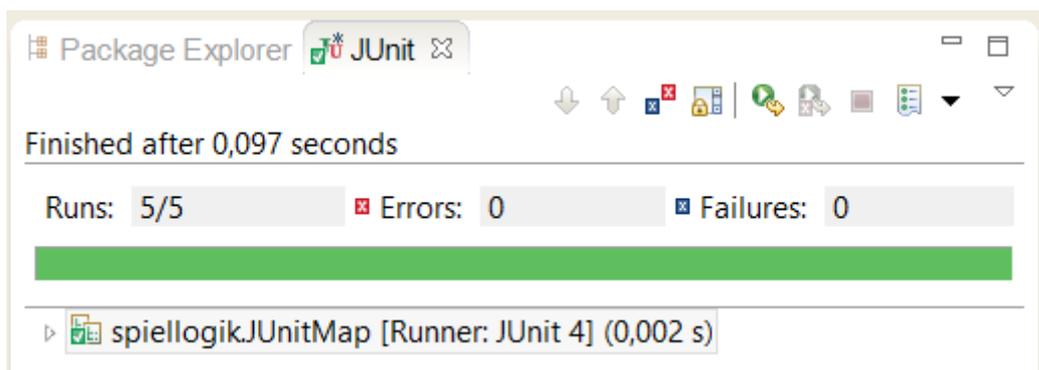
Für den Test haben wir eine eigene Test-Map erstellt

Insgesamt werden 5 Tests durchgeführt:

- In Wände / Wasser laufen
- Aktionspunktkosten von Bewegungen/Schlägen
- Feindliche Spieler töten
- Den Gral aufnehmen
- Das Spiel beenden durch Aufnahme des Grals und Abgabe beim Abgabepunkt.



Screenshot des Tests:



Die Initialisierung des Tests ist etwas umständlich. Da unsere Spiellogik stark mit der Client-Server Kommunikation verbunden ist. Deshalb muss bei der Initialisierung ein Server und 2 Clients, sowie Knights erstellt werden.

Weitere Qualitätsmaßnahmen:

Der wichtigste Teil unserer QM waren die regelmäßigen Treffen auf denen wir Testläufe durchführten. Bei diesen Testläufen sind uns immer wieder Unschönheiten sowie auftretende Exceptions aufgefallen welche wir dann untereinander aufteilen und beheben konnten.