



Freiheit für Onlinespieler

Das Projekt GGZ Gaming Zone

Intro

GGZ Gaming Zone: Internationales Projekt zur Integration und Verbreitung von freien Onlinespielen.

Veröffentlichung der Version 0.0.13:
live von den Chemnitzer Linux-Tagen!

(Versionsnummernjunkies: 0.0.13 == 6 Jahre Arbeit)

Spieler sind anspruchslos

- * Werbebanner
- * Verletzung der Privatsphäre
- * Willkürliche Änderungen am Regelwerk
- * Eingeschränkte Spieleauswahl
- * Spiele-Clients für nur eine einzige Plattform

Muss das sein? Nein!

GGZ: desktopübergreifend spielen

Betriebssysteme: Linux, BSD, Solaris, (Windows)

Oberflächen: Qt/KDE, Gtk+/GNOME, SDL (Vollbildmodus)

* Integration in den Desktop ist wichtig

Java-Applet

* Kartenspiele, demnächst mehr

* zum Anlocken von Windows-Nutzern

GGZ Community: Webportal

Statistischer Überblick

"Size does matter!"

~ 30 Spiele, jeweils mit Client und Server

GGZCards, GGZBoard, GGZ-Java: Multi-Spiele-Clients

7 externe Spiele

Freeciv, Freelords, Widelands, T.E.G., Kamikaze, Tuxman, CGoBan
(teilweise noch nicht funktional)

160 000 Zeilen Quelltext

16 Pakete zu jeder Veröffentlichung

Wie funktioniert GGZ?

Klassisch:

- * Spieler startet Core-Client (Chat-Anwendung mit Spielestarter)
- * Auswahl verschiedener Räume: Lobby, Schach, Reversi, Geekgame, ...
- * Neues Spiel starten
oder existierendem Spiel beitreten
oder einfach nur zuschauen :)
- * Eintrag in Statistiktabelle, Neuberechnung der Wertungen

Alternative Zugänge:

- * Spiele-Plugins für Kopete und Gaim
- * Embedded GGZ: Spiel verbindet sich von sich aus zum GGZ-Server

Merkmale von GGZ-Spielen

- * KI-Spieler (anonym oder personalisiert)
- * Sitzreservierungen, Austausch zwischen Spielern und Bots
- * Wiederaufnahme abgebrochener Spiele
- * spiel-interner Chat
- * Zuschauer
- * Bewertungen: Highscore-Tabelle, Gewinne/Verluste, ELO-Werte etc.
- * Spielerinformationen: IP-Adresse, Foto/Avatar-Bild, vollständiger Name
- * Nachvollziehen eines Spielverlaufes anhand von Grafiken von Spielständen

Architektur: Die Entwickler-Sicht

Programmiersprachen: C, C++, Python, Java, (Ruby)

Bibliotheken:

- * libggz (für geplagte C-Programmierer)
- * ggzcore (Chat-Clients mit Spieleintegration)
- * ggzmod (Spieleclients)
- * ggzmod (Spieleserver)

- * weitere spezielle für Schach, Tic-Tac-Toe, ...

Netzwerkcode: teilweise Autogenerierung aus XML-Beschreibung

Schlagwörter: IPv6, TLS, Datenbankabstraktion, Skriptsprachen, Plugins

Dokumentation: Entwicklerleitfaden, jede Menge HTML-Dokus, Protokollreferenzen

Visuelle Impressionen

Ein paar Bilder folgen nun...

Aktuelle Themen

Arimaa-Implementierung

- * Schach mit Tierfiguren
- * 10000 US\$ für starke KI?
- * gleichzeitig Probleme mit Patenten, Lizenzen, Nutzungsrecht, ...

Skat für GGZCards

- * irgendwer aus Altenburg hier?

GGZBoard-Implementierung für KDE mit Python

- * evtl. sogar KDE4
- * gleich 6 neue Spiele für den KDE-Desktop!

Hilf mit!

- * dedizierter Server, anyone?
- * Grafiken, Hintergrundbilder, Icons
- * Übersetzungen und Spielberichte für GGZ Community
- * Spieleautoren: GGZ-Integration in eure Spiele!

<http://www.ggzgamingzone.org/> - Projektseite

<http://www.ggzcommunity.org/> - einfach nur losspielen...