

Nexuiz

Downloads



Intern

- [Nexuiz 2.5.2](#)
- [Nexuiz 2.5.2 als Torrent](#)
- [Nexuizstats 0.35](#)

Extern

- [Nexuiz 2.5.2](#)
- [Nexuiz Mappack](#)
- [Mehr Maps](#)
- [CTF-Maps](#)
- [Nexuizstats 0.35](#)

Installation

Eigentlich ganz einfach: Runter laden und entpacken. Für Linux gibt's das ganze über die Paketverwaltung. Das Mappack sollte entweder nach Nexuiz\data (Windows) oder nach ~/.nexuiz/data (Linux) entpackt werden. Einen Pferdefuß könnte es aber geben, je nach dem, wie aktuell die installierte Version ist. Insbesondere über die Paketverwaltung kann es schon sein, dass da nichts mehr aktualisiert wurde, was die Master-Server angeht. An diese kann ein Server-Betreiber seinen Server melden. Clients fragen diesen ab, um Internet-Server zu finden. Blöderweise ist die standardmäßige Adresse dafür inzwischen nicht mehr richtig, da der Server down ist. Das lässt sich aber sehr leicht beheben, indem man bei Client und/oder Server die folgenden Zeilen in die Konfig

einträgt (auf dem Client ist das die Datei `config.cfg`, im selben Verzeichnis, in dem auch das Mappack lande sollte):

```
sv_master1 dpmaster.deathmask.net
sv_master2 91.121.161.160
sv_master3 dpmaster.vecxis.com
sv_master4 ghdigital.com:27950
```

Infos zum Spiel

Nexuiz ist ein Multiplayer-Shooter ganz im Stile von Unreal Tournament, mit dem Unterschied, dass es vollkommen Open Source ist und auf verschiedenen Plattformen verfügbar ist. Es können bis zu 32 Spieler pro Map spielen. Leider ist der Wechsel zwischen Modi nicht so angenehm einfach möglich, wie bei [UT2004](#). Es ist dafür dann immer ein Neustart des Servers notwendig. Es gibt aber Mods dafür, um dieses Problem zu umgehen. Wer sich das Spiel einfach mal ansehen möchte, findet im Netz auch entsprechende [Videos](#).

Spielmodi

Es bietet viele Spielmodi:

- DM: Deathmatch (aka Free for all), also jeder gegen jeden, Punkte geben nur die Frags.
- TDM: Teamdeathmatch, wie DM, nur dass es zwei Teams gibt und Teamkills Punkteabzug gibt.
- LMS: Last man Standing, sehr ähnlich dem DM, nur mit dem Unterschied, dass gefragte aus dem Spiel bleiben, bis nur noch einer übrig ist, der die Runde gewinne.
- Arena: Immer zwei Spieler spielen gegeneinander (vorzugsweise auf 1on1-Maps), sobald einer der beiden Spieler gefragt wurde, wird er durch einen anderen ersetzt, der in der Warteschlange ist.
- CTF: Capture the Flag, Ziel ist, die Flagge des Gegners zu entwenden und in die eigene Basis zu bringen, bevor er seinerseits die eigene Flagge stiehlt. Die Flagge kann nur abgeliefert werden, wenn die eigene Flagge anwesend ist.
- DOM: Domination: Es müssen sog. Dominationpoints eine Zeit lang gleichzeitig gehalten werden.
- Runematch: Punkte für das Einsammeln von Symbolen
- Keyhunt: Punktgewinn, wenn ein Team alle Schlüssel auf einer Karte hält
- ONS: Onslaught, Punktgewinn beim Einnehmen von Kontrollpunkten und anschließendem Zerstören des gegnerischen Generators
- Race: Rennen, wobei man möglichst schnell von A nach B kommen muss
- Race CTS: Ähnlich Race, nur dabei auf einer Art Track ablaufend, ähnlich z. B. Track Mania.
- Nexball: Ähnlich UTs Bombingrun eine Art Fußball

Punkte

Für alle DM-Modi gilt: Selbstmord zieht einen Punkt fürs Maplimit ab, sodass man auch mehr Frags haben kann, als nach dem Maplimit gilt. Man hat aber dann entsprechend zusätzliche Selbstmorde auf dem Scoreboard. Insbesondere bei CTF gibt es diverse Punktere Regelungen, die auch von Server zu Server unterschiedlich sein können. Hier mal eine List der Punkte, wie sie beim CTF als Default

vergeben werden:

Ereignis	Punkte
gegnerische Flagge aus deren Basis entwenden	-3 (ja, Punkteabzug!)
eine früh fallen gelassene Flagge einsammeln	2
eine spät fallen gelassene Flagge einsammeln	2
eine Flagge erfolgreich in die eigene Basis entwenden	28
Flaggenträger fraggen	2
Flagge fallen lassen	-2
Flagge wegen Selbstmord fallen lassen	-2
Flagge wird vom Gegner zurück geholt	-1
eigene Flagge zurück holen	5
Flagge wird automatisch zurück geholt(?)	10
Flagge wird vom Angreifer zurückgeholt	6

Die Punkte können durch entsprechende [cvars](#) auf dem Server überschrieben sein, sodass andere Regeln gelten können. Dieselben Punkte werden auch dem Team-Konto angerechnet. Je nach Modus kann man CTF auch so ablaufen lassen, dass nicht einfach das Team mit den meisten Captures gewinnt, sondern das mit den meisten Punkten. CTF ist daher eigentlich viel anspruchsvoller gedacht, als es oft gespielt wird. Denn z. B. bringt ein erfolgloser Versuch die Flagge zu entwenden (pickup (-3) kill (-2) drop (-2) return (-1)) -8 Punkte. Erfolgreich (pickup (-3) capture (28) gibt immerhin +25. Punktemäßig macht der genaue Ablauf also einen großen Unterschied. Wobei tendenziell die Captures natürlich letztlich sehr viel wiegen. Wenn dazu aber sehr viele Versuche notwendig waren, kann es durchaus sein, dass das gegnerische Team trotzdem gewinnt, weil es am Ende mehr Punkte hat.

Netzwerk/Internetserver

Der Server der mobilen Infanterie hält zwei Nexuiz-Server bereit. Wer mehr auf Capture the Flag steht, sollte sich unter `mobile-infanterie.epicgamer.org:26002` umsehen. Für Deathmatch hier: `mobile-infanterie.epicgamer.org:26000`. Siehe auch die [Status-Seite](#). Wenn ein Server *nicht* auf dem Standard-Port läuft und auch *nicht* an die Master-Server bekanntgegeben wurde, ist er so erst mal nicht zu sehen. Dann muss man entweder mit [HLSW](#) arbeiten, oder den Server in den eigenen Favoriten anlegen.

Server betreiben

Prinzipiell reicht zum Server erstellen der Port aus, den man gekonft hat, +1 dazu. Standard ist Port 26000 (UDP), also 26000-26001 UDP. Zum Mitspielen muss man i. A. nichts extra machen. Um einen Server aufzusetzen, entweder einfach das Nexuiz-Archiv runter laden (Windows) oder aus den Paketquellen installieren (Linux). Es sollte dann im Verzeichnis server (im *.zip-File für Windows) die Datei `server.cfg` zu finden sein. Unter Ubuntu scheint man sich netterweise dieses File gleich komplett gespart zu haben. (Hinweise zu anderen Distris gerne gesehen...) Falls es mit der benutzten Distri nicht auf der eigenen Platte zu finden ist, einfach das File aus dem Windows-Archiv nehmen. Geht genau so gut.

Das ist das Sample-File! Wenn nach Möglichkeit nicht daran rum fummeln, sondern immer eine Arbeitskopie machen. Diese sollte dann unter `~/nexusiz/data/` (Linux) oder `data\server.cfg` im

Nexuiz-Ordner (Windows) liegen. Es enthält schon sehr viele (auskommentierte) Einträge, die man bei Bedarf einfach durch löschen der // aktivieren kann. Einige Dinge, die man auf jeden Fall anpassen sollte, um den Server ein wenig zu personalisieren (Details siehe auch [wichtige \(RCON-\)Befehle](#)):

- `sv_public`: Soll der Server bekanntgegeben werden?
- `hostname "Name des Servers"`: Das wird angezeigt, wenn man in Serverlisten nach dem Server sucht und wenn man ihm beitrifft
- `sv_motd "Text1\nUnd nächste Zeile"`: Message of the day: Wird beim Beitreten kurz angezeigt.
- `maxplayers` : Wie viele Spieler maximal?
- `sv_eventlog 1`: Falls man Logs der eigentlichen Matches haben will. Diese landen dann entweder in `~/ .nexuiz/data` (Linux) oder `nexuiz\data\` (Windows).
- `gametype dm`: Wenn man dm spielen will, sonst anderen Modus eintragen
- `minplayers`: Falls man mindestens x Spieler haben will, die von Bots aufgefüllt werden, ist bei ggf. von Vorteil, wenn man wenigstens 2 einträgt, denn sonst wird es ein recht langweiliges Spiel für einen Spieler alleine.
- `g_maplist_check_waypoints 1`: Stellt sicher, dass nur Maps zur Auswahl gestellt werden, auf denen die Bots auch Waypoints haben. Bei den mit gelieferten Maps nicht dringend nötig, bei Drittanbieter-Maps evtl. schon. Ohne das kann es sein, dass die Bots dann auf der Map sehr wenig rum laufen, sodass es recht langweilig wird.
- `sv_autoscreenshot 1`: Wenn man will, dass wie bei [CS](#), am Ende einer Map ein Screenshot gemacht wird (mit den Ergebnissen).
- `g_balance_teams 1`: Wenn man will, dass der Server beim Beitreten automatisch Teams ausgleicht
- `g_balance_teams_force 1`: Dasselbe, nur auch während der Map

Ggf. auch den Abschnitt unter [Mapdownloads](#) lesen!

Und dann den Server starten: `./nexuiz -dedicated +serverconfig server.cfg` (Linux) bzw. `nexuiz -dedicated +serverconfig server.cfg` (Windows).

Damit wird ein Dedicated Server gestartet, der mit STRG + C jederzeit beendet werden kann. Er lädt automatisch die eben erstellte Konfigurations-Datei. Man kann theoretisch auch Server-Einstellungen in dieser Konsolen-Zeile mit übergeben, mit einem + davor. Angaben in der Konfig gehen aber vor. Man kann auf diese Weise natürlich mehrere Server starten, die aber auf unterschiedlichen Ports arbeiten müssen. Unter Windows kann man das dann als Service anlegen, oder unter Linux als Service via Init-Skript (im V-Init-System) laufen lassen. Letzteres ist dann aber etwas komplizierter, da man für starten, stoppen, neu starten und ggf. die Status-Anzeige, verschiedene Fallunterscheidungen einarbeiten sollte. Auch bietet es sich (sowohl unter Windows als auch Linux) an, den Dienst unter einem eigenen (eingeschränkten) Benutzer laufen zu lassen. Dieser muss sich auch nicht anmelden können. Es reicht, wenn er Dienste laufen lassen kann. Im Fall der Fälle macht man es so einem Angreifer schwerer, über den Nexuiz-Server ins System einzudringen, da er dann recht wenige Rechte hat, selbst wenn er es schafft, Nexuiz zu kompromittieren.

Maps

Maps sind gewissermaßen ein Sorgenkind. Es gibt gewisse Kompatibilitäten zu Quake 3, die allerdings den Haken haben, dass entsprechende Maps nicht zwangsläufig für Nexuiz bug-frei sein müssen, nur weil sie das für Quake 3 waren. Das führt dann zu allerlei komischen Verhalten, von seltsamen Spawn-Points bis zu Stellen auf der Map, an denen man ohne Hindernis hängen bleibt und sich nicht

mehr weg bewegen kann. Deshalb sollte man Maps vor der Benutzung auf einem Server ausgiebig testen.

Nicht alle Maps sind für alle Spielmodi geeignet. Es geht z. B. nicht, eine Map für CTF zu verwenden, wenn der CTF-Modus nicht unterstützt wird. Nexuiz analysiert neue Maps beim ersten Starten/Laden automatisch darauf hin und legt die Ergebnisse in `~/.nexuiz/data/data/maps/` ab, bzw. unter Windows im Ordner `nexuiz\data\data\maps`. Das sind dann reine Text-Files, in denen u. A. auch steht, für welche Modi sie geeignet sind.

Der Ablageort für Maps ist im Verzeichnis `data` der Nexuiz-Installation, wenn man die Map selbst spielbar ablegen will, insbesondere für alle Benutzer des PCs. Das gilt auch für Server. Der Server sucht in diesem Ordner nach `*.pk3`-Dateien. Auf dem Client liegen sie, wenn sie automatisch heruntergeladen wurden, auch im Ordner `~/.nexuiz/data/dlcache` (Linux), bzw. dem Pedant unter Windows.

Dass einem Maps fehlen, merkt man daran, dass er sich darüber beschwert, dass ihm ein `*.bsp`-File fehlt und alle Map-Texturen schwarz bleiben. In diesem Fall hat der automatische Download offensichtlich nicht funktioniert, bzw. ist deaktiviert. Dann muss man sich die fehlende Datei anderweitig besorgen.

Mapdownloads

Sollte beim Beitreten auf einen Server Meldungen angezeigt werden, dass eine `*.bsp`-Datei nicht geladen werden konnte, weil nicht vorhanden, wurde diese nicht herunter geladen auf den Client. In diesem Fall muss man den Mapdownload serverseitig einstellen. Leider ist dazu ein FTP- oder Webserver notwendig, auf dem die entsprechenden Files liegen. Dazu muss dann in der Konfig des Servers ein Abschnitt stehen, der definiert, wo genau die Maps zu finden sind:

```
sv_allowdownloads 1 //Sind Mapdownloads grundsätzlich möglich?  
sv_allowdownloads_archive 1 //Sind pk3-Downloads möglich?  
sv_allowdownloads_inarchive 1 //Sind Downloads von innerhalb PK3s möglich?  
sv_curl_defaulturl <url> //fallback download URL
```

Das hat zwar den Nachteil, dass man auf eine externe Datenquelle angewiesen ist, aber den Vorteil, dass Mapdownloads entsprechend schnell sind, bzw. überhaupt möglich sind. Mappacks sind daher auch nicht unbedingt notwendig, außer wir reden hier von sehr niedrigen Upload-Bandbreiten unterhalb der 100 kB/sek.. Die Maps können dazu aber auch einfach auf *irgend einem* Webspace liegen, das muss nicht auf demselben Rechner sein, auf dem der Nexuiz-Server läuft.

RCON einrichten

Nachdem man da dran verzweifeln kann, wenn man nicht genau weiß, was man tut, hier eine kurze Beschreibung, was man server- und clientseitig setzen muss, damit man in der Lage ist, via RCON Befehle an den Server abzusetzen. Zunächst muss auf beiden Seiten `rcon_secure 0` in der Config stehen. Andere Modi wären prinzipiell möglich, verlangen aber komplett synchrone Uhren. Nachdem das alles andere als gewährleistet ist, besser ohne. Wenn man das nicht setzt, wird der Server keine Fehlermeldungen raus werfen, wenn man versucht, sich zu authentifizieren. Aber er wird nicht auf Befehle reagieren, da sie gar nicht bei ihm ankommen. Der Befehl *muss vor dem Passwort stehen*, da bei seiner Ausführung das Passwort zurück gesetzt wird.

Auf beiden Seiten muss auch das gleiche Passwort gesetzt sein: `rcon Password` ist die einfache Variante und sollte immer gehen. Mit Leerzeichen dazwischen ist es aber notwendig, dass man das Passwort in `"` setzt, also z. B.: `rcon_password "Password mit Leerzeichen"`.

Die Clientseitigen Befehle kann man auch manuell eingeben, wenn sie nicht in der Konfig stehen. Wer also auf Nummer Sicher gehen will, lässt das raus. Man kann das aber auch auf Tastendruck machen, indem man sich beides auf eine Taste 'BIND'et:

```
alias getmein "rcon_secure 0; rcon_password Blahfasel"  
bind HOME getmein
```

Mit diesen zwei Teilen wird auf die Taste POS1 bzw. HOME die Befehlssequenz `rcon_secure 0` `rcon_password Blahfasel` gelegt. Danach kann man, nach Druck auf die POS1-Taste, fröhlich mittels `rcon <Befehl>` den Server steuern. Hier die für den Anfang wichtigsten Befehle:

wichtige (RCON-)Befehle

- `status`, `status 2`, `status 3`: Zeigt an, wer gerade alles auf dem Server ist und unterschiedliche Infos dazu. Alle geben einem die Spielernummer aus.
- `maps`: Listet alle maps auf
- `kick # <nummer>`: Kickt den Spieler mit der Nummer `<nummer>` vom Server.
- `restart`: Startet die Map neu
- `changelevel <mapname>`: Map-Wechsel ohne Client-Drops
- `loadconfig`: Config neu laden und neu starten
- `map <mapname>`: Wie `changelevel`, nur mit Clients kicken
- `net_stats`: Netzwerkstatistiken
- `lsmaps`: Liest die Liste der Maps neu ein. Gut, wenn man im laufenden Betrieb neue Maps hinzufügen will. Auf dem Client, ohne `rcon` zeigt es die Liste der auf dem Server verfügbaren Maps für den aktuellen Spieltyp an. Also z. B. bei CTF nur die CTF-fähigen Maps.
- `set ...`: Um eine Server-Variable zu verändern, z. B.
 - `maxplayers`: Maximale Anzahl Spieler, 0 bedeutet unbegrenzt
 - `timelimit <dauer>`: Maps enden spätestens nach `<dauer>` Minuten.
 - `minplayers`: Minimale Anzahl an Spielern, die aufgefüllt wird mit Bots
 - `bot_join_empty`: Spielen Bots auch auf leeren Servern?
 - `fraglimit`: Name ist Programm
 - `fraglimit_override`: Frag limit overriding the mapinfo specified one (use 0 to play without limit, and -1 to use the mapinfo's limit)
 - `g_ctf_capture_limit`: CTF Capture Limit, überschreibt die Angabe in der Map (0=kein Limit, -1 für Limit aus der Mapinfo)
 - `fly <1/0>`: Gibt es einen Flugmodus?
 - `g_footsteps <1/0>`: Mit oder ohne Schritt-Geräusche?
 - `g_grappling_hook <1/0>`: Mit Grappler, oder ohne?
 - `g_jetpack <1/0>`: Mit Jetpack?
 - `g_laserguided_missile <1/0>`: Mit Ziel-Laser für Raketenwerfer?
 - `g_maplist "<mapnamen mit leerzeichen getrennt>"`: Liste der erlaubten Maps, wenn ohne Angabe, sind alle erlaubt, die möglich sind
 - `g_maplist_shuffle <1/0>`: Zufällige Reihenfolge der Maps?
 - `g_maplist_votable <anzahl>`: Wie viele Maps stehen am Ende einer Runde zur Auswahl?
 - `g_maxplayers <anzahl>`: Maximale Anzahl an Spielern auf dem Server, 0 bedeutet

- unbegrenzt
- `g_minstagib <1/0>`: Ist Instagib (Nur one-hit-kills) aktiviert?
- `g_use_ammunition <1/0>`: Brauchen Waffen Mun?
- `g_weapon_stay <0-2>`: Bleiben Waffen nach dem Aufsammeln bestehen? 0=nein, 1=mindestens Munition dafür, 2=wenn schon vorhanden, wird nichts aufgesammelt
- `pausable <1/0>`: Server pausierbar
- `sv_public <-2 - 1>`: Wie wird der Server gelistet? -2: keine Verbindung möglich, -1: Keine Abfragen möglich, 0: Abfragen möglich, 1: wird Bekanntgegeben
- `g_ctf <1/0>`: Ist es ein CTF-Server?
- `g_dm <1/0>`: Ist es ein DM-Server?
- `g_domination <1/0>`: Ist es ein Domination-Server?
- `g_lms <1/0>`: Last man Standing: Jeder hat eine Anzahl an Leben. Sind die weg, ist man raus, bis der letzte übrig ist
- `g_nexball <1/0>`: Ist es ein Nexball-Server?
- `g_onslaught <1/0>`: Ist es ein Onslaught-Server?
- `g_runematch <1/0>`: Ist es ein Runematch-Server?
- `g_tdm <1/0>`: Ist es ein Team-DM-Server?
- `g_tdm_teams <2-4>`: Wie viele Teams gibt es?
- `pause`: Server pausieren
- `condump <filename>`: Speichert alle bisherigen Konsolen-Ausgaben in der angegebenen Datei im Nexuiz-User-Verzeichnis (`~/ .nexuiz/data/` bzw. `nexuiz\data\` im Eigene Dateien Ordner)

Siehe auch [Nexuiz cvars](#) und [Nexuiz commands](#)

Stats

Von Haus aus bringt Nexuiz keine Werkzeuge zur Statsanalyse mit. Wohl aber lassen sich mit den entsprechenden Server-Einstellungen sogenannten Eventlogs anlegen, die schön fleißig jeden Frag, Unfall, Selbstmord, Flaggenklau, usw... aufzeichnen. Das kann man natürlich auch zusammenfassen und ein Ranking daraus erstellen. Nexuizstats ist ein Programm von Martin Wohlaue, dass in [Freebasic](#) geschrieben ist und sich daher auf Windows und Linux, als Serverbetriebssystemen, gleichermaßen kompilieren lässt. Mit diversen Kommandozeilenparametern lässt sich Einfluss auf Ausgabeformat und andere Verhaltensweisen nehmen. Eine detaillierte Dokumentation liegt dem Programm bei, ebenso der Quellcode, da das Programm unter der [GPLv3](#) steht.

Scoring

Insbesondere für CTF erhält man für diverse Aktionen Punkte. Im Archiv `data20091001.pk3` im Server-Verzeichnis `data` ist auch die Datei `ctfscoring-div0.cfg` enthalten, die regelt, wie die Default-Werte dafür aussehen. Standard-Mäßig sieht das dann so aus: `<spoiler>`

```
// Scoring from the democratic poll at
http://www.alientrap.org/forum/viewtopic.php?p=49529#49529
// based on ctfscoring-z, but adjusted to be less harsh

set g_ctf_flagscore_pickup_base -3
set g_ctf_flagscore_pickup_dropped_early 2
```

```

set g_ctf_flagscore_pickup_dropped_late      2
set g_ctf_flagscore_capture                  28
set g_ctf_flagscore_kill                     2
set g_ctf_flagpenalty_drop                   2
set g_ctf_flagpenalty_suicidedrop            2
set g_ctf_flagpenalty_returned                1
set g_ctf_flagscore_return                   5
set g_ctf_flagscore_return_rogue             10
set g_ctf_flagscore_return_by_killer          6
set g_ctf_flagscore_return_rogue_by_killer   10

// succeeded capture (pickup capture)          25 (0 for enemy)
// failed capture (pickup kill drop return)    -6 (8 for enemy)
// failed (shot into void) (pickup kill drop) -5 (3 for enemy)
// capture retry (kill drop pickup)           0 (3 for enemy)
// suicide, then retake (suicidedrop pickup)   0 (0 for enemy)

set g_ctf_personalscore_pickup_base          -3
set g_ctf_personalscore_pickup_dropped_early  2
set g_ctf_personalscore_pickup_dropped_late   2
set g_ctf_personalscore_capture               28
set g_ctf_personalscore_kill                  2
set g_ctf_personalpenalty_drop                2
set g_ctf_personalpenalty_suicidedrop         2
set g_ctf_personalpenalty_returned            1
set g_ctf_personalscore_return                5
set g_ctf_personalscore_return_rogue          10
set g_ctf_personalscore_return_by_killer       6
set g_ctf_personalscore_return_rogue_by_killer 10

```

</spoiler> Die entsprechenden Variablen sind in den [Nexuiz cvars](#) zu finden und lassen sich in einer entsprechenden Server-Konfig mit eigenen Werten überschreiben.

Weblinks

- <https://icculus.org/twilight/darkplaces/readme.html#NetworkingTips>, etwas ältere Liste mit Infos zur Engine

[Zurück zur Games-Datenbank](#)

From:

<https://www.mobile-infanterie.de/wiki/> - **mwohlauer.d-n-s.name** / **www.mobile-infanterie.de**

Permanent link:

<https://www.mobile-infanterie.de/wiki/doku.php?id=games:nexuiz&rev=1471148680>

Last update: **2016-08-14-04-24**

