Außerdem

- Was zum singen
- GCT Tech
- Gottklotzen (am See)
- Haus der Woche

Inhalt

Seite

Greenville Pick-Ups 2

Nur noch kurz die Welt Laden 2

Haus der Woche 3

Frost Gard 4

Gottklotzen (am See) 6

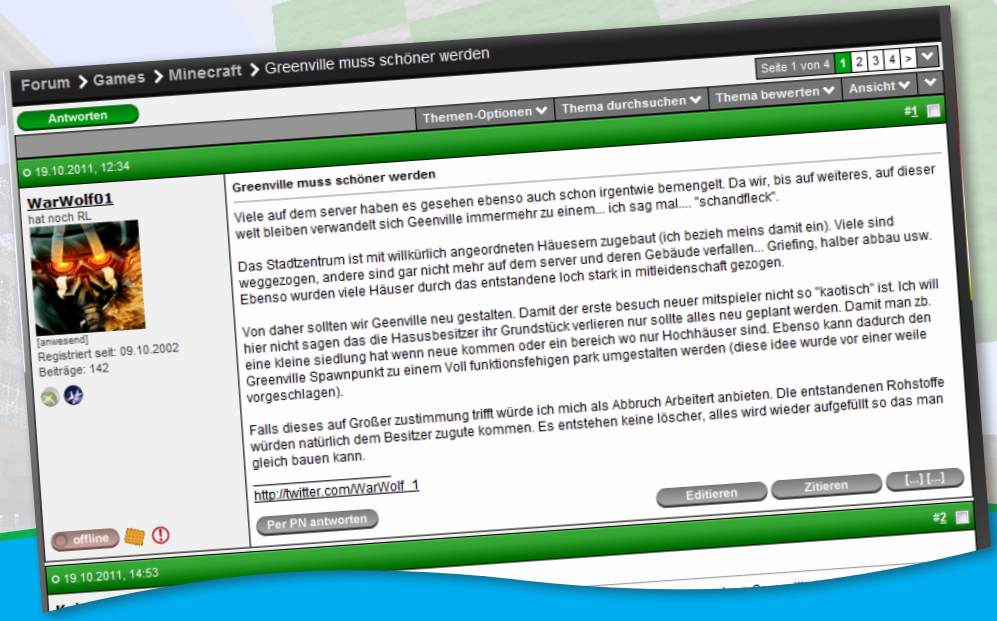
Grafik 1990,
Systemanforderung 2012 7

Das Team 9

Greenville Pick-Ups

Greenville muss schöner werden

Unter dem Motto hat WarWolf01 aka ScarScrow eine Gesprächrunde im Forum eröffnet und alle dazu eingeladen, sich an dem Dialog, zu einer möglichen Verschönerung der Lage von Greenville, zu beteiligen.



Nur noch kurz die Welt laden – Eine Minecraft Parodie auf Tim Bedzko's „Nur noch kurz die Welt retten"

Ich würde dir ja gerne helfen, das ist für dich ja viel zu viel,
lass mich kurz mal was versuchen.
Da draußen gibt's so viel zu sehn, die Situation ist zum untergehen.
Und vielleicht mach ich irgendwas verkehrt.
Ich weiß, es ist dir ernst, du kannst mich hier grad nicht entbehren, doch Minecraft hält mich einfach von dir fern.

Refrain
Muss nur noch kurz die Welt laden, danach komm ich zu dir.
Noch 148 Chunks laden, wer weiß, was mir dann noch passiert, denn ich bau hier so viel.
Muss nur noch kurz die Welt laden und gleich danach bin ich wieder bei dir.

Irgendwie bin ich spät dran, fang schon mal mit den ersten Wänden an.
Ich stoß' dann später dazu.
Du fragst, wieso, weshalb, warum, ich sag, wer sowas fragt, ist dumm.
Denn du scheinst wohl nicht zu wissen, was ich bau.
`n ganz beklemmender Ping, ich bin hier nicht mehr der King.
Genug gesagt, genug Information.

Refrain...

Die Zeit läuft mir davon, zu warten, wäre eine Schande für die ganzen Minecraftmitspieler.
Ich muss jetzt los, sonst gibt's die große Katastrophe.
Merkst du nicht, dass wir echt Nerds sind.

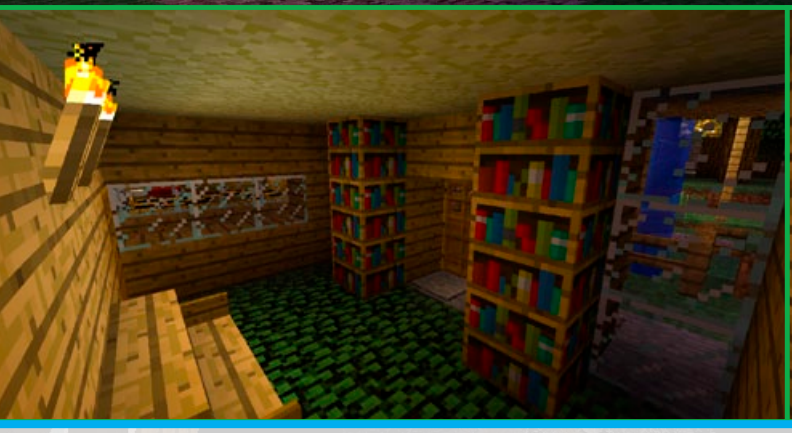
Refrain
Ich muss jetzt echt die Welt laden, danach komm ich zu dir.
Noch 148 Chunks laden, wer weiß, was mir dann noch passiert, denn ich bau hier so viel.

Muss nur noch kurz die Welt laden, danach komm ich zu dir.
Noch 148.713 Chuncks laden, wer weiß, was mir dann noch passiert, denn ich bau hier so viel.

Muss nur noch kurz die Welt laden und gleich danach bin ich wieder bei dir.

(Ma)/(Mr)

Haus der Woche



Nun ist es endlich soweit. Wir dürfen euch das nächste Haus der Woche präsentieren. Nachdem wir euch letzte Wochen leider kein Haus der Woche vorgestellt haben, freuen wir uns um so mehr euch diese Woche eine echte Überraschung vorstellen zu können.
Das Haus, welches sich nun mindestens eine Woche lang Haus der Woche nennen darf wurde gebaut von **Japanese.** Wie einige Bewohner von Greenville eventuell schon wissen, weilt dieser leider nicht mehr unter uns, sondern wurde von der Whitelist entfernt. Dies ändert aber nichts an der Tatsache, dass das Haus uns gefällt.
Das Gebäude steht südlich des Spawns, erreichbar ist es vom Spawn aus in unter einer Minute. Das Gebäude erinnert durch die schmale und künstlerische Bauweise genauso wie der Name des Erbauers an den Tech-Staat Japan. Von außen sehen wir zwei Wasserfälle von einem gemütlichen Balkon fließen, über dem Balkon können wir noch einen kleinen Turm sehen der das Haus ziemlich spitz erscheinen lässt. Auf einen Garten haben wir uns leider vergeblich gefreut, weil den werden wir nicht vorfinden können.
Nachdem wir durch die Doppeltür in das schicke Haus eingetreten sind, sahen wir uns um und erblickten eine Bank neben der eine Leiter in die erste Etage führt, auf der anderen Seite des Raumes führt eine Tür in den zweiten Raum des Erdgeschosses. Der zweite Raum scheint ein Lager Raum zu sein, zwei Wände des Raumes bestehen aus besten Wooden Planks, die anderen zwei Wände sind mit einer Cleanstonewand verziert.
Die zweite Etage bietet den schon eben erwähnten Balkon mit zwei Wasserfällen und einen Hintereingang. In der ersten Etage ist auch eine mittige Leiter die auf einen kleinen, überdachten Aussichtsturm führt. Der Ausblick von dem Turm ist wirklich spektakulär. Dadurch das das Gebäude mitten im Herzen Greenvilles liegt hat man sogar den Blick über die berühmte Skyline Greenvilles.
Das Haus hat sogar einen ziemlich großen Keller der aber leider ohne große Inneneinrichtung auskommen muss.
Zusammenfassend kann man sagen das es ein ziemlich schnuckeliges Haus ist das es eigentlich verdient hätte das dort jemand leben würde. Grade die schmale Bauart und die zwei Wasserfälle gefallen uns sehr gut. (Gi)

Ihr würdet Euer Haus auch gerne der restlichen Welt präsentieren? Oder ihr würdet gerne ein bestimmtes Haus hierfür vorschlagen, welches ihr bisher nur innerhalb der Videos beim vorbei laufen sehen konntet? Dann wendet euch an unsere Redaktion!

Frost Gard

„Frost Gard“, im Vergleich zum Namen handelt es sich, bei diesem abgeschiedenen Dorf, ganz und gar nicht um ein kaltes, steriles Vorstädtchen. In einer wunderschön gelegenen Stelle, der große Hotels und riesige Redstone-Schaltungen fern sind, liegt „Frost Gard“. Ein kuschliges Dorf, geschützt durch eine Holzmauer, mit Wachtürmen und schön verzierten Gebäuden.

Die Idee entstand daraus, dass **Cpt_Badass, TheMeierz** und ein Freund von **Crysis_Nerd,** welcher unter dem Account von **Sebi707** (Timelapsvideo Omashu), neu auf den Server gekommen sind. Sie kennen sich aus dem mythischem „Real Life“, welche nur wenige je zu Gesicht bekommen konnten. Aus dem gescheiterten Versuch einen sogenannten Mobspawner zu bauen, ist dann das Dorf entstanden. Zusammen mit Crysis_Nerd und Stricker 1993 begannen sie Mitte Oktober mit dem Bau.

Eines fehlte ihnen jedoch, der passende Name für ihr Projekt. Leichter gesagt als gedacht, der Name wollte ihnen nicht einfallen, also gab ihnen **Bit01** den Vorschlag von „Frost Gard“, welcher direkt übernommen wurde und in das Grundbuch der Stadt eingemeißelt wurde.
Alle Häuser werden seitdem im gleichen Stil aufgebaut. Eine leichte mittelalterliche Anlehnung, mit Ranken an den Steinwänden, zusammen mit Rohholz verarbeitet.
Die Dorfmauer aus Holzplanken soll Frost Gard lästige Mobs fern halten, sowie innerhalb der Mauern alles systematisch ausgeleuchtet wurde.
Es stehen bereits einige Wohnhäuser der Erbauer, sowie ein Rathaus und ein Haupt-

bahnhof welcher an das Schienensystem von Greenville angeschlossen ist. Außerdem gibt es einen prunkvollen Brunnen, aus feinstem Gold, aus den tiefsten Minen im Umland, deren jeder Zwerg stolz wäre.

Für Frost Gard geplante Bauten sind nützliche, öffentliche Gebäude. Ähnlich wie das Japanische Badehaus, sowie weitere Häuser um das Gelände innerhalb der Mauer bestmöglich auszunutzen. Alles soll sich weiterhin im gleichen Stil, schön weiterentwickeln.

Frost Gard erreicht ihr am besten, indem ihr den Anschlussbahnhof im Seriengebiet betretet und dort Richtung Bit´s berühmter Wolle-Farm fahrt. (Sb)

Koordinaten
X: -195
Z: -1300

Anbindung Bahn
Abfahrt von Bahnhof „Serien“ aus

Gottklotzen (am See)

Ihr wusstet mit Sicherheit das Greenville nicht die einigste Ansiedlung auf dem Server ist, es gibt mittlerweile einige andere Städtchen und Dörfer. Einige sind bekannt andere nicht. Wir möchten euch heute ein unbekanntes Dorf vorstellen. Das Dorf trägt den schönen Namen **Gottklotzen (am See),** es wurde von dem Spieler Volmat entdeckt. Mittlerweile leben schon sieben Spieler in Gottklotzen, namentlich der Entdecker **Volmat** persönlich, **Bobbobbsen, Darochris, Jukka1996, SchimmelBoi, Wamperator und Gigantisch.** Natürlich haben sich die eben erwähnten Spieler nicht nur einfach ein Haus gebaut, sondern es ist mittlerweile ein richtiges Dorf mit einer Kapelle, einem Wachturm, einer Mühle, einem kleinem Bahnhof, mehreren Feldern, einem kleinem Hafen und einer großen Mine entstanden.
Damit auch ihr euch einen Eindruck des Dorfes machen könnt, haben wir extra für euch unsere besten Fotografen in die Siedlung geschickt, sie durften einige schöne Fotos der Kapelle, dem Hafen, der Mühle und einigen Feldern machen.

Hier sehen wir die kleine Kapelle in der schon so einige kleine Feste und Opfergaben der Uhreinwohner umher gingen. Die Kapelle wurde schon vor einiger Zeit von den zwei ganz besonders gläubigen Dorfbewohnern SchimmelBoi und Gigantisch gebaut.

Auf dem Bild sehen wir die Zukunft und die Vergangenheit von Gottklotzen – Die Landwirtschaft. Nach Aussagen der Dorfbewohner ist es eine lange Tradition das sich das Dorf nur von ihren eigenen Vorräten ernährt, so mussten auch mit neuen Dorfbewohnern, neue Felder in das Dorf. Das die Landwirtschaft in Gottklotzen (am See bringt auch noch den Vorteil mit, dass alle Dorfbewohner einer gesicherten Arbeit nachgehen können.

Fisch ist in Gottklotzen eine wahre Delikatesse gewesen, doch durch den neulich erbauten Hafen auf dem Bild wurde Fisch nun ein Alltagsprodukt, sagten die Alteingesessenen. Der Hafen wurde auf einem engen Fluss gebaut, wenn man lang genug auf ihm fährt kommt man auf das fischreiche Meer.
Gottklotzen besitzt im Prinzip alles was das kleine Dorf gebrauchen könnte, von der Landwirtschaft über einer Mine bis hin zu einem Hafen oder einer Kirche. Wie an uns heran gebracht wurde sind die Arbeiten in Gottklotzen auch noch nicht abgeschlossen, sondern es sollen noch einige nützliche Gebäude entstehen. Wir dürfen gespannt sein! (Gi)

Grafik 1990, Systemanforderung 2012

GCT TECH

Es gibt eine Frage, die Freunde mir immer wieder stellen und die mich selber zum Grübeln bringt: Warum „laggt“ Minecraft? „Laggen“ ist bekanntlich das Phänomen, was auftritt, wenn der PC nicht genügend Leistung für ein Spiel hat. Jeder der sich Minecraft anguckt, sieht die furchterregenden 16x16 Texturen und merkt dann, dass das Spiel trotz dieser schlechten Grafik nicht flüssig läuft. Verglichen mit heutigen Spielen und der Rechenpower der heutigen Computer ist die Verwunderung zu verstehen.
Ich möchte in diesem Artikel diese Frage klären, so gut es geht. Ich versuche alles so zu formulieren, dass jeder es versteht, ohne ein Spezialist in dem Gebiet zu sein. Viele Fachbegriffe werden außerdem in den kleinen Boxen am Rand erklärt. Trotzdem werde ich weit ins Detail gehen und viel Hintergrundwissen vermitteln, um möglichst viele offene Fragen zu klären. Wer keine Lust hat, alles zu lesen, kann gerne auch nur die kurzen Abschnitte unter „Fazit“ lesen.

Grundlegend kann man das Minecraft-Performance Problem in zwei Bereiche einteilen. Zum einen die grafisches Ausgabe, wofür selbstverständlich die Grafikkarte zuständig ist, und zum anderen den Berechnungsteil, der maßgeblich die CPU und den RAM belastet. Nachdem ich beide Probleme geklärt habe, gebe ich noch Tipps, wie man es selber schaffen kann, dass Minecraft flüssiger läuft.

1. Die Grafikausgabe
Man erkennt Laggs daran, dass die Framerate zu niedrig ist. Dazu ein bisschen Hintergrundwissen: Um den Eindruck eines bewegten Bildes zu haben, braucht das menschliche Auge ca. 25-30 Bilder pro Sekunde. Diese Bilder heißen in der Computergrafik „Frames“ und die Framerate oder FPS (frames per second) gibt an, wie viele Bilder pro Sekunde ausgegeben werden. Man braucht also 30 FPS, damit das Spiel flüssig läuft bzw. nicht „laggt“.
Um nun diese 30 FPS zu erreichen, muss jedes Spiel 30 mal pro Sekunde ein neues Bild berechnen, was recht anspruchsvoll ist. Aber genau dafür gibt es die Grafikkarte. Diese Komponente des Computers ist darauf spezialisiert Grafik zu berechnen und ist demnach auch schneller.
Da stellt sich nur die Frage, warum Minecraft diese 30 FPS nicht erreicht, obwohl es komplexere Spiele mit Leichtigkeit schaffen.
Und nun die leider banale Erklärung für Minecraft: Mit der Programmiersprache Java bekommt man keine vernünftige Grafikausgabe hin. Jedenfalls nicht auf die Art, wie Notch es macht. Das Problem ist, dass Java eine Hochsprache ist, die nicht direkt ausgeführt wird, sondern in einer Laufzeitumgebung läuft. Die Laufzeitumgebung übersetzt die Java-Anweisungen während der Ausführung in die Maschinensprache. Somit hat Java wenig Bezug zu der Hardware. Durch diese Übersetzung geht irre viel Leistung verloren. Die Grafikausgabe leidet besonders darunter, weil es in dem Bereich sehr wichtig ist, direkt mit der Hardware zu kommunizieren.

Dazu kommt noch, dass Minecraft grafisch anspruchsvoller ist, als man vermuten würde. Die Texturen sind zwar sehr schlecht aufgelöst, trotzdem besitzt das Spiel recht viele Polygonen, die die Grafikkarte entsprechend fordern.

Fazit:
Ein wesentlicher Faktor für die schlechte Leistung bei der Grafikausgabe ist Java. Dadurch, dass die Hochsprache sehr hardware-fern ist, geht viel Leistung verloren. Zudem ist die Programmierung von Minecraft sicherlich nicht perfekt, weil Notch´s Team nicht jede Prozedur richtig optimiert hat.
Zuletzt muss man berücksichtigen, dass Minecraft´s Grafik anspruchsvoller ist, als man vermutet. Die Texturen sind zwar schlecht, aber trotzdem müssen eine Menge Polygonen berechnet werden.

2. Berechnungen im Kern von Minecraft
Grafik ist das eine, aber auch der Prozessor wird ordentlich in die Mangel genommen. Hohe Auslastung der CPU und des RAM‘s sind bei Minecraft Standard.
Aber die Grafikkarte war doch für die Laggs verantwortlich? Ja auch. Aber jeder Berechnungsschritt kann das ganze Programm verlangsamen und somit für Laggs sorgen. Ein Spiel läuft in einer Endlosschleife ab und führt immer die gleichen Schritte durch: Auswerten der Eingaben, Simulieren der Welt, Berechnung der Ausgabe. Egal in welchem Schritt eine Verzögerung auftritt, der ganze Prozess wird verlangsamt. Was wir hier nun behandeln ist der Schritt „Simulation der Welt“.
Bevor wir uns die Berechnungsprozedur näher anschauen, gucken wir zunächst aus technischer Sicht auf die Minecraftwelt, um die RAM Auslastung nachvollziehen zu können.
Sie ist 127 Blöcke hoch und praktisch unendlich weit, soviel weiß fast jeder. Und manche wundern sich auch: Warum so eine dumme Zahl wie 127?
Antwort: Eigentlich 128 Blöcke, nämlich von 0 bis 127. Aber warum? Einfache theoretische Informatik: Im Computer wird alle in Bits gespeichert, die nur den Zustand 1 oder 0 annehmen können (also zwei Möglichkeiten). Hat man nun 7 Bits kommt man auf 2^7= 128 Möglichkeiten.
Zweite Frage: Ist sie wirklich unendlich weit? Antwort: Auch nicht. Erstens würde der Speicherplatz auf der Festplatte irgendwann voll werden, würden wir eine riesige Karte generieren, hinzu kommt aber ein stärker limitierender Faktor: Die Speicherung der Koordinaten. Aber darauf werde ich eventuell mal in einem späteren Artikel drauf eingehen. Wichtig ist hier nur, dass die Karte doch mal endet, nämlich bei x/y=43Mrd und merkwürdige Formationen ab x/y=12Millionen aufweist. Die Karte von Minecraft ist aufgeteilt in

Chunks, die jeweils eine Fläche von 16x16 Blöcken speichern und somit einen Raum von 16x16x128=32768 Blöcken. Weil 16 Blöcke nicht besonders weit sind, lädt Minecraft immer 9*9=81 Chunks. Ist ein Chunk, der geladen werden soll, noch nicht vorhanden, wird er generiert. Wir können hier schon mal festhalten, dass theoretisch immer 81x32768 = 2.654.208 Blöcke geladen sind. Die „Luftblöcke“ werden natürlich nicht wirklich gespeichert, weil es ja im Prinzip „nichts“ ist. Der Einfachheit halber gehen wir von einer Anzahl von 2.000.000 Blöcken aus. Pro Block werden verschiedene Daten gespeichert, die letztendlich den RAM belegen. Welche Daten pro Block gespeichert werden, darüber kann man nur spekulieren. Typ des Blocks, Position, etc. brauchen ihren Speicherplatz. Ungenaue Rechnungen vermuten, dass ca. 50 Byte pro Block gespeichert werden. Das würde heißen: 2.000.000x50 = 100.000.000 Byte ~ 100 MB. Dazu kommen diverse andere Daten, wie Position des Spielers, aller Mobs, aller Items die einfach so herumliegen und weitere spielinterne Daten. Außerdem werden zur Sicherheit alle Daten, die für die Grafikkarte relevant sind, zusätzlich auch im RAM gespeichert. Dies merkt man daran, dass Minecraft deutlich mehr RAM benötigt, wenn man ein HD Texturenpack installiert.
Eine Sache wirkt sich ebenfalls auf den RAM-verbrauch aus: Nicht Minecraft sondern Java verwaltet den Speicher. Wenn gewisse Daten nicht mehr gebraucht werden, weil ein Chunk nicht mehr benötigt wird, gibt Java den Speicher nicht direkt frei, sondern behält alle Daten, bis eine Speichersäuberung durchläuft. So kann sich recht viel „Müll“ ansammeln, der den RAM verstopft. Nach all den Rechnungen kommt man zwar noch nicht ganz auf den tatsächlichen RAM-verbrauch von Minecraft, aber schon sehr nahe.

Wenden wir uns nun der horrenden CPU Auslastung zu, die beim Minecraft-Spielen entsteht. Zwei wesentliche Faktoren belasten die CPU, nämlich die Generierung der Welt und das Simulieren der Welt. Die Generierung ist ja sozusagen das Kernstück von Minecraft, das was Minecraft ausmacht. Wenn man in diversen Programmierforen guckt, hört man hin und wieder, dass diese Zufallsgenerierung das einzige ist, was an Minecraft bewundert wird. Denn der Algorithmus für diese Aufgabe ist mit Sicherheit recht komplex und braucht auch seine Zeit. Für das Resultat ist die dafür benötigte Leistung recht annehmbar, denn ein Chunk zu generieren dauert nur wenige Sekunden, manchmal auch unter einer Sekunde. Trotzdem benötigt es eine Menge Leistung, was man im Singleplayer recht gut merkt: Das Spiel ruckelt immer, wenn man in neue Gebiete läuft.
Nach der Generierung muss man die Welt aber auch am Leben erhalten, indem man sie simuliert. In Minecraft heißt das, dass Wasser und Lava fließt, der Tag wechselt und somit Chunks neu beleuchtet werden müssen, Kreaturen herumlaufen und spawnen, Gras und Bäume wachsen, Feuer sich ausbreitet und jeder Schneesturm den Ozean zum Gefrieren bringt. Außerdem rein physikalische Sachen wie das Herunterfallen von Sand und Gravel und die Kollisionsabfrage. Ohne Kollisionsabfrage würde der Spieler und alle Tiere durch den Boden fallen. Dies sind alles Aufgaben, die sich einfach anhören, aber bei weitem nicht so einfach für den PC sind. Um eine Kollision abzufragen, müsste man die Position des Spielers mit der Position jedes Blocks vergleichen. Aber hier hat sogar Notch eine Optimierung eingebaut, indem nur alle Blöcke im selben Chunk verglichen werden.

Alle diese Prozesse benötigen eine Menge Leistung. Also kann man damit die hohe CPU Auslastung erklären? Jein. Die letztendliche Performance hängt maßgeblich vom Entwickler ab. Während in Crysis hoch-optimierte Algorithmen verwendet werden, sitzen an Minecraft nur ein paar ehemalige Hobbyentwickler. Und außerdem spielt auch hier Java wieder eine kleine Rolle, da durch Java die Performance immer noch ein bisschen verschlechtert wird.

Fazit:
Der hohe RAM-Verbrauch liegt im Wesentlichen an zwei Punkten: Die große Welt mit vielen Blöcken, die im RAM gespeichert werden müssen, und die Speicherverwaltung von Java, bei der Speicher nicht direkt freigegeben wird.
Die hohe CPU-Auslastung liegt an den komplexen Algorithmen zu Generierung und Simulation der Welt. Diese Algorithmen sind grundsätzlich schwer zu berechnen und sind im Gegensatz zu anderen Spielen auch nur wenig optimiert. Und im letzten Schritt verlangsamt Java alles noch ein wenig.

3. Kann ich die Performance von Minecraft selber verbessern?
Ja, allerdings nur bedingt. Ich habe hier ein paar Tipps zusammengestellt, mit denen man Minecraft beschleunigen kann.
Als Erstes kommt gleich das Allerbeste: Einen Mod direkt für Minecraft, der bessere FPS-Werte verspricht und „OptiFine“ genannt wird. Ich habe ihn sofort ausprobiert und muss sagen, dass ich recht zufrieden war. Die FPS-Werte schnellten nicht gerade in Richtung Himmel, aber ich habe, sobald alle Chunks geladen wurden, mit einem 64 Pixel Texturenpack im Regen stabile 45 FPS. Vorher lag meine Framerate bei ca. 25-30 mit diversen Frame-Einbrüchen. Und der Mod bietet sogar noch mehr als nur Optimierungen: Er ermöglicht es HD-Texturen ohne weiteren Mod zu laden und fügt eine Reihe neuer Grafikeinstellungen hinzu. Dazu gehören praktische Sachen, wie Wolken entfernen und auch echte grafische Aufwertungen, wie das Verschwischen von Texturen. Hört sich eher an, als wäre es eine Abwertung, aber durch das Verwischen von Texturen in der Ferne setzen keine Flimmereffekte mehr ein und das Bild sieht schöner aus. Ist aber letztendlich Geschmackssache.

Hier noch weitere Tipps, die ohne Mod funktionieren:
Software updaten! D.h. Immer die neuste Javaversion installieren, aber achtet darauf, dass ihr in 64bit Betriebssystemen immer die 64bit Version nehmt! Außerdem kann

man die von Notch genutzte Bibliothek aktualisieren. Wie das geht findet ihr alles unter http://www.minecraftwiki.net/wiki/Tutorials/Update_LWJGL
Zuletzt kann man noch die Priorität des Java Prozesses hochsetzen. Das erreicht man ganz einfach, indem man den Taskmanager mit [STRG], [SHIFT] und [ESC] aufruft. Dort sucht man im Reiter „Prozesse“ nach „javaw.exe“. Rechtsklick auf den Prozess und unter „Priorität festlegen“ die passende Einstellung auswählen. Achtung dabei: Bitte stellt nicht „Echtzeit“ ein, da das böse enden kann. Alles andere könnt ihr jedoch ohne Bedenken ausprobieren. Leider speichert Windows die Einstellung nicht. Um es dauerhaft einzustellen müsst ihr euch eines Batchscripts bedienen oder ein extra Programm runterladen. Beides ist leicht über Google zu finden.

Als Abschluss kann ich sagen: Durch all die Optimierungen habe ich meine FPS von ca. 25-30 (Standardtexturen) auf durchschnittlich 60 (64x Texturenpack) gebracht. Das meiste hat der Mod „OptiFine“ gemacht. Daher kann ich ihn euch nur empfehlen.

Abschließendes Fazit:
Man meint, Minecraft würde auf einem Gameboy laufen, weil dir Grafik so schlecht ist. Man irrt sich aber gewaltig.
Drei Hauptgründe führen dazu, dass Minecraft so langsam ist, wie wir es kennen. Zum einen sind die Algorithmen, die im Spiel ausgeführt werden, teilweise sehr anspruchsvoll, und die Speicherung der Welt benötigt eine Menge RAM. Zum anderen ist die Grundlage „Java“ eine schlechte Wahl für ein Spiel. Besonders die Grafikausgabe leidet sehr unter Performanceschwächen. Auch der Speichermanager von Java führt oft dazu, dass viel zu viel RAM belegt wird.
Und zuletzt sollte man eins nicht vergessen: Minecraft ist in der Beta Phase und wird von ein paar Hobbyprogrammierern entwickelt. Diese haben sicherlich nicht das Know-How, um das Beste aus Java herauszuholen und ihre Algorithmen perfekt zu optimieren.

Um nicht komplett den Laggs ausgesetzt zu sein, gibt es Mods und Tipps für Minecraft, die die Leistung insgesamt verbessern.

Mit diesem Artikel habe ich versucht allen Lesern Hintergrundwissen zu Minecraft zu vermitteln, Grundlagen der Informatik zu erklären und vor allem das Spielen angenehmer zu machen. Ich hoffe das ist mir gelungen.

Weblinks:

Tutorial um LWJGL zu aktualisieren:
http://www.minecraftwiki.net/wiki/Tutorials/Update_LWJGL

Minecraft-Mod „OptiFine“:
http://www.minecraftforum.net/topic/249637-181-optifine-hd-c1-fps-boost-hd-textures/
(Cn)

Das Team

Crysis_nerd (Crysis_nerd, Cn)
Gigantisch (Abenragala, Gi)
GigaVfb (Giga90Freak, Gf)
guerkschen (guerkschen, Gu)
Maniraiser (Maniraiser, Mr)
Matze029 (Matze029, Ma)
Offpunch (Offpunch, Op)
SchimmelBoi (SchimmelBoi, Sb)
SniPer]cAt (Werner4Giga, Sc)
Spwankuecken (SpawnKuecken, Sk)
Volmat (Volmat3, Vo)
Wamperator (Wampe, Wa)

[Minecraft] ([Forum], [Kürzel])

Erklärungsbox

CPU

„Central Processing Unit“ ist der Hauptprozessor eures PC´s. Er berechnet eigentlich alles, was berechnet werden muss. Nur sehr spezielle Aufgaben wie z.b. Das Berechnen eines Bildes werden an spezielle Komponenten, wie die Grafikkarte abgegeben.

RAM

„Random Access Memory“ ist der Arbeitsspeicher eures PCs. Alle Daten, die laufende Programme benötigen, werden dort gespeichert. Beim Herunterfahren des PC´s werden alle Daten im RAM gelöscht. Zurzeit ist reicht die Standardgröße des Speichers von 1 GB zu 8 GB.

Polygon

Ein Polygon ist in der 3D-Grafik ein Dreieck im Raum. Jede noch so komplexe 3D Szene in PC-Spielen setzt sich im Prinzip aus Dreiecken zusammen. Die Anzahl der Dreiecke bestimmt die Komplexität der Szene und die Rechendauer.

GCT TECH