

Overview

In dieser Datei wird der Entwicklungsprozess des TeleportStone Plugin festgehalten. So werden die einzelnen Versionen genauer beschrieben und einzelne Gedankenverläufe gezeigt.

Table of Contents

- [Overview](#)
 - [Table of Contents](#)
- [Fragen](#)
 - [Wie viele Zeichen kann ein Schild haben?](#)
 - [Liefert die getWorld\(\) Methode die selben Welten, wie im Server Ordner?](#)
- [Version 0.1 \(Signs only\) \[DONE\]](#)
 - [Commands](#)
 - [TeleportStone erstellen](#)
 - [TeleportStone entdecken](#)
 - [Teleportieren zwischen TeleportStones](#)
- [Version 0.2 \(Signs only\) \[DONE\]](#)
 - [Commands](#)
 - [Start Up Validation](#)
 - [Constraints](#)
 - [Database](#)
 - [Pretty Format Outputs](#)
 - [Refactor, Modularity](#)
- [Version 0.3 \(Signs only\) \[DONE\]](#)
 - [Behandlungen von Datenbankänderungen, Konsistenz mit der Spielwelt](#)
 - [Behandlung mehrerer gleicher Nutzernamen](#)
 - [Fehlendes Discover und Destroy Event](#)
 - [Ingame Anleitung bei erster Entdeckung eines TeleportStone's](#)
- [Version 0.4 Structure \[UNTESTED\]](#)
 - [TeleportStone \(TpSt\)](#)
 - [Create, Build and Validate](#)
 - [Änderungen der Datenbank](#)
 - [SignChangeEvent](#)
 - [Constraints](#)
- [Version 0.5 Clean Up, Rollback and Auto Safe \[UNTESTED\]](#)
 - [Event calling Commands](#)
 - [Auto Safe](#)
 - [Rollback](#)
 - [Undo-Befehl](#)
- [Version 0.6 Better Commands \[UNTESTED\]](#)
 - [Auto Completion](#)
- [Version 0.7 Unterschiedliche Typen von TeleportStones \(Teil 1\) \[UNTESTED\]](#)
 - [Formen und Materialien](#)
 - [Scopes](#)

- Standard Scopes
- Weitere Merkmale
 - Teleportreichweite
 - Weltgebunden
- Baupläne (BluePrints)
 - Umgestaltung TeleportStone
- Scope Validierer
- Bauplan Validierer
- Schlüsselwörter verbieten
- Mitgelieferte Lösungen
 - Mitgelieferte Scopes
 - Mitgelieferte Pakete
- Custom Baupläne und Scopes
 - Test Custom Baupläne und Scopes
- Version 0.8 GUI Menu
 - Pageable Inventory
 - Favorit TeleportStones
 - Alphabetische Sortierung
 - Weitere Knöpfe
 - Änderungen der User Datenbank
 - User Validator
 - Befehle
- Version 0.9 Konfiguration
 - Namensänderungen von Dateien
 - Plugin Konfiguration
 - Default Packages überarbeiten
 - Mitgelieferte Pakete **Überblick**
 - Event Log in extra Datei speichern?
- Version 0.10 Protect TeleportStone / Permissions
 - Blöcke des TeleportStones schützen
 - Änderungen an TeleportStone
 - TeleportStone Validator
 - Platzierungsanfragen für TeleportStones
 - Anfrage
 - Bearbeiten von Anfragen
 - Permissions
 - Befehle
- Version 0.11 Unterschiedliche Typen von TeleportStones (Teil 2)
 - Ingame Konfiguration
 - Gruppierung
 - Ausgewählte TeleportStones Ziele
 - Ausgewählte Welten
 - Visual Effects
- Version 0.12 Thread Safty
- Version 0.13 Performance
- Version 1.0 Finish

- Mehr Ideen und Features
 - High Priority
 - Info in GUI
 - Befehle überarbeiten und zusammenfassen
 - Exclusive Scope Angaben
 - Spielernamenänderung
 - Teleport Position festlegen
 - TeleportStone aktivieren und deaktivieren
 - Sonstiges
 - Teleportationskosten
 - Teleporteffekte
 - Beacon Beam Effect
 - Nutzen und Kompatibilität mit anderen Plugins
 - Low Priority
 - Startup Validation Failure Handler
 - Log Detail
 - Minimaler Radius für TeleportCommand
 - TeleportStone mit Schildern konfigurieren
 - Discarded
 - Behandlung mehrerer gleicher Namen
 - Undo Befehl
- Ideen
 - More Discover

Fragen

Wie viele Zeichen kann ein Schild haben?

Abhängig von den Zeichen. Das Symbol "I" kann 22 mal in Folge auf eine Zeile platziert werden. Ein Schild kann maximal 4 Zeilen haben.

Liefert die `getWorld()` Methode die selben Welten, wie im Server Ordner?

- check if get world does really work (name im spigot folder / name in teleportstone config)

Version 0.1 (Signs only) [DONE]

Diese Version des TeleportStone Plugin erstellt erst einmal eine Basis. Bis jetzt ist ein "TeleportStone" nur ein Schild, d.h. er hat noch nicht seine finale Struktur.

Commands

- `/tpst help` - Gibt TeleportStone Hilfe aus.
- `/tpst signs` - Zeigt alle existierenden TeleportStones an.
- `/tpst players` - Zeigt alle für dieses Plugin registrierten Spieler an.
- `/tpst discovered` - Zeigt alle entdeckten TeleportStones an.

- /tpst nearest - Zeigt den am wenigsten entferntesten TeleportStone an.
- /tpst distance <signname> - Gibt Entfernung zu gegebenen TeleportStone an.
- /tpst tp <signname> - Teleportiert zu gegebenen TeleportStone, wenn man in der Nähe eines entdeckten TeleportStone's ist.
- /tpst events - Zeigt kritische Events (Create, Delete, Teleport, Info, Error) in verschiedenen Farben an.
- /tpst version - Gibt Informationen über TeleportStone Plugin zurück. Version, etc.

TeleportStone erstellen

- Schild setzen und beschriften
 - Erste Zeile: TeleportStone
 - Weitere Zeilen: Name des TeleportStone's
- **Intern** Prüfung, ob die Beschriftung des TeleportStone's dem speziellen Muster entspricht.
- **Intern** Wenn ja, Schild in Datenbank eintragen. (ID, Name, Welt, Koordinaten)
- **Intern** Rückmeldung ausgeben.

TeleportStone entdecken

- TeleportStone anklicken. (linke oder rechte Maustaste)
- **Intern** Testen, ob angeklickter Block ein TeleportStone ist.
- **Intern** Wenn ja, TeleportStone der persönlichen Liste entdeckter Schilder des Spielers hinzufügen.
- **Intern** Rückmeldung ausgeben.

Teleportieren zwischen TeleportStones

- In den Chat Command eingeben: /tpst tp <nameTeleportStone>
- **Intern** Spieler muss registriert sein.
- **Intern** Spieler muss in der Nähe eines entdeckten TeleportStone's sein.
- **Intern** Spieler muss den Ziel TeleportStone entdeckt haben.
- **Intern** Teleport starten.
- **Intern** Rückmeldung ausgeben.

Version 0.2 (Signs only) [DONE]

Commands

- /tpst delete <name> - Entfernt den angegebenen TeleportStone.
- /tpst destroy <name> - Entfernt und zerstört den angegebenen TeleportStone.
- /tpst unregister <username> - Entfernt den angegebenen Nutzer und seine entdeckten TeleportStone's aus dem Register.
- /tpst undiscover <username> <nameTeleportStone> - Entfernt den angegebenen TeleportStone aus der Liste entdeckter TeleportStones des angegebenen Nutzers.
- /tpst manualdiscover <username> <nameTeleportStone> - Fügt den angegebenen TeleportStone der Liste entdeckter TeleportStones des angegebenen Nutzers hinzu.

Start Up Validation

Nach jedem Start sollte überprüft werden, ob die gespeicherten Daten mit dem Server übereinstimmen. Das heißt, wenn beispielsweise ein TeleportStone entfernt wurde, ohne dass diese Änderung vom TeleportStone Plugin bemerkt wurde (z.B. nicht aktiviert ist), dann soll ein Fehlerzustand behandelt werden. Eine Lösung wäre den nicht mehr vorhandenen TeleportStone zu entfernen.

Constrains

Der Name des TeleportStones hat eine minimale Länge von 3 Zeichen. Der Name darf nur die zweite und dritte Zeile eines Schildes belegen.

Database

- Nutze Player UUID anstatt Namen als ID.

Pretty Format Outputs

Die Lesbarkeit einiger Ausgaben wahr unansehnlich. Einige Befehle werden jetzt schöner ausgegeben.

- signs: Minimale Länge der Namenspalte. Zu kurze Namen werden mit WhiteSpace aufgefüllt.
- discovered: Selbes.
- Entfernungen in Befehlen wie nearest oder distance werden nun mit nur noch 2 Nachkommastellen angegeben.

Refactor, Modularity

Wiederholende Aufgaben auslagern. Extra Klassen für Löschvorgänge.

Version 0.3 (Signs only) [DONE]

Version 0.3 beschäftigt sich hauptsächlich mit dem CleanUp der letzten beiden Versionen.

Behandlungen von Datenbankänderungen, Konsistenz mit der Spielwelt

Es existieren im Moment zwei Speicherorte. Einmal die Minecraftwelt selbst und die Konfigurationsdatei. Dadurch entsteht eine Problematik der Konsistenz.

Beispiel: Ein Spieler gibt den Befehl "destroy" ein, um einen TeleportStone zu zerstören. Der Befehl sucht die Datenbank auf, löscht dort mit CRUD Methoden den Eintrag des TeleportStone's. Auch wird er aus den entdeckten TeleportStones aller User gelöscht. Als nächstes sucht er die Koordinaten des TeleportStone's in der Welt auf und löscht den Block an der gegebenen Position. Der Befehl "delete" macht das Selbe nur ohne die Zerstörung. Die Validierung des Storage nutzt ein ähnliches Verfahren wie der "delete" Befehl, falls Einträge nicht konsistent sind.

Das Problem mit diesen Prozessen ist, dass sie an mehreren Stellen im Programm vorkommen können. Jede dieser Stellen muss die selbe Logik implementieren. Das hat Probleme und Risiken zu folge:

- Wiederholender Code
- Mehrmals gleicher Rollback Code (spätere Versionen)
- Fehleranfälligkeit

- Einheitlichkeit nicht gewährleistet
- Aspekte könnten nicht beachtet werden.

Befehle greifen nicht mehr direkt auf Datenbank zu. Oder KEIN direkter Zugriff auf Datenbank allgemein. Eine Schicht zwischen Daten und Nutzercode wird eingeführt. (Package: dataaccess)

Manipulations Methoden (delete, add, etc.) in eigenes package verlagern, verringert doppelten Code.

Behandlung mehrerer gleicher Nutzernamen

- Kein Nutzernamen sollte doppelt vorkommen. (Sonst kann die Fehlerlosigkeit der Suche nach einem Spieler in der Datenbank nicht gewährleistet werden.)
- ~~ODER spezielle Behandlung / Prüfung bei einigen Anfragen.~~ (Vielleicht in späteren Versionen)

Fehlendes Discover und Destroy Event

Wenn ein Spieler einen TeleportStone entdeckt, sollte auch ein Event ausgelöst werden. Selbes gilt für das Zerstören von TeleportStones.

Ingame Anleitung bei erster Entdeckung eines TeleportStone's

Wenn ein Spieler das erste mal einen TeleportStone entdeckt, sollte ihm eine Erklärung ausgegeben werden. Beispielsweise könnte der Spieler gar nicht wissen, was TeleportStones sind.

Version 0.4 Structure [UNTESTED]

TeleportStone (TpSt)

Zuerst muss geklärt werden, was ein TpS eigentlich ist. Ein TpS soll ähnlich zu MemoryStones aussehen. Das heißt, es gibt ein Fundament mit einem Maß von 3x3. Dieses besteht aus Stein (CleanStone). Auf die Mitte des Fundaments wird dann ein drei Blöcke hoher Turm aus Obsidian platziert. Somit hat die gesamte Struktur eine Höhe von vier Blöcken. Zuletzt wird an den mittleren Block des Obsidianturms ein Schild platziert. Dieses Schild muss einem bestimmten Muster folgen und definiert den Namen des TpSt.

Baumaterialien:

- 9 Blöcke Stein (Stone (CleanStone))
- 3 Blöcke Obsidian
- 1 Schild

Create, Build and Validate

Beim Erstellen des (nun richtigen) TeleportStone's muss nun auf eine valide Struktur hinter dem Schild geachtet werden. Vorher musste nur das Schild selbst geprüft werden. Dies geschah einfach in der TeleportStoneCreator-Klasse. Die Aspekte der Erstellung und Validierung wurden nun ausgelagert in die Klassen TeleportStoneBuilder und TeleportStoneValidator.

Der TeleportStoneValidator testet einen TeleportStone (oder Teile dessen) auf verschiedene Aspekte.

- **MISSING_SIGN**, Kein Schild vorhanden.

- **INVALID_SIGN_MATERIAL**, Schild besteht aus einem nicht zugelassenen Material.
- **UNKNOWN_WORLD**, Angegebene Welt ist unbekannt.
- **INCORRECTLY_MARKED**, Das Schild ist nicht richtig beschriftet.
- **MISSING_NAME**, Kein Name im TeleportStone POJO vorhanden.
- **INVALID_NAME_LENGTH**, Die Länge des angegebenen Namen ist nicht valide.
- **INVALID_NAME**, Der angegebene Name ist nicht valide.
- **MISSPLACED_SIGN**, Ein Schild muss an die Seite eines Blocks der Struktur platziert werden.
- **MISSING_ATTACHED_BLOCK**, Der Block, an welchem das Schild platziert sein sollte, konnte nicht gefunden werden.
- **INCONSISTENT_NAME**, Der auf dem Schild spezifizierte Name und der Name im TeleportStone POJO unterscheiden sich.
- **INCONSISTENT_WORLD**, Blöcke des TeleportStone's sind nicht in der selben Welt.
- **STRUCTURE_BLOCK_NOT_FOUND_IN_WORLD**, Ein Block der Struktur konnte nicht in der Welt gefunden werden.
- **INVALID_STRUCTURE**, Die gespeicherten Blöcke der Struktur stimmen nicht mit den Blöcken der Welt oder dem Bauplan überein.
- **INVALID_AMOUNT_BLOCKS**, Die Anzahl der gespeicherten Blöcke stimmen nicht mit dem Bauplan überein.
- **WRONG_MATERIAL**, Beim Bau des TeleportStone's wurde falsches Material genutzt.

Die ID wird nicht validiert. Hier muss das Datenbanksystem einen konsistenten Zustand sicherstellen.

Der TeleportStoneBuilder kann (sollte) genutzt werden, um neue TeleportStone-Objekte zu erstellen. Es ist empfohlen seine Bau-Und-Validierungsmethode zu nutzen.

Änderungen der Datenbank

Die TeleportStone-Pojo muss entsprechend erweitert werden. Neben dem Schild, Namen und ID werden nun auch die Blöcke der Struktur abgespeichert.

- New POJO SimpleBlock (x, y, z, Welt, Material)
- New POJO SimpleSign (erweitert SimpleBlock um Zeichenkettenarray (4 Zeilen eines Schildes))
- Update TeleportStone POJO
 - Ein TeleportStone besteht nun aus der ID (ganze Zahl, automatisch vergeben), dem Namen (Zeichenkette), der Name der Welt (Zeichenkette), einem SimpleSign (spiegelt das Schild des TeleportStones wieder) und SimpleBlocks (spiegeln die Struktur des TeleportStone's wieder).
- StartUp-Validierung nutzt nun TeleportStoneValidator, um TeleportStone zu validieren. D.h. die Daten, das Schild und die Blöcke der TeleportStones werden überprüft. (Siehe weiter oben.)

SignChangeEvent

Das SignChangeEvent wird immer ausgelöst wenn ein Schild geändert wird. Neben dem Test für die TeleportStone-Erstellung müssen auch die Testfälle für das Entfernen und Ändern eines Schildes geprüft werden.

- Wenn das Schild vorher kein TeleportStone Schild war und nun Änderung mit TeleportStone Keyword auftritt -> CREATE
- Wenn das Schild vorher ein TeleportStone Schild war und Änderung mit TeleportStone Keyword auftritt -> UPDATE

- Wenn das Schild vorher ein TeleportStone Schild war und Änderung ohne TeleportStone Keyword auftritt -> DELETE
- Sonst, nichts machen

Beachte Fallbacks, wenn Updates, Deletes, etc. fehlschlagen.

Constrains

Nur ein TeleportStone Schild sollte an einem TeleportStone sein. D.h. nicht mehrere TeleportStones können die selbe Struktur teilen.

Um diesen Fall zu beachten, muss der Validator erweitert werden, um einen neuen Testfall.

- **STRUCTURE_ALREADY_IN_USE**

Version 0.5 Clean Up, Rollback and Auto Safe [UNTESTED]

- Rechtschreibung
 - Multiply => multiple
- Bei einigen wichtigen Befehlen wird nicht geloggt, wer sie auslöst.
 - Info zu Destroy Event hinzugefügt.
 - Info zu Delete Event hinzugefügt.
- Einige Formulierungen in Log Nachrichten sind unpassend und werden überarbeitet.
- Prüfen, ob in allen Error Zuständen ein passendes TeleportStoneEvent mit Error geworfen wird.

Event calling Commands

Sollten Commands nur Events auslösen? No.

Auto Safe

Um das Risiko von Datenverlusten bei Serverabstürzen und den Schaden zu minimieren, sollten die Daten automatisch nach bestimmten Zeiträumen oder Aktionen abgespeichert werden.

Bestimmte Aktionen:

- Erstellen eines TeleportStone
- Löschen eines TeleportStones
- ~~Zerstören eines TeleportStones~~ (Nutzt sowieso Löschmethode)

Bestimmte Zeiträume:

- Alle 3 Stunden (Standard, in Konfiguration änderbar)

Rollback

Wenn Aktionen wie das Zerstören von TeleportStones fehlschlägt, sollten alle Änderungen die durch die Aktion bis zum Zeitpunkt ausgeübt wurden, rückgängig gemacht werden. Beispielsweise besteht die Zerstörung eines TeleportStones aus zwei Phasen. Dem Löschen aus der Datenbank und der Zerstörung der Blöcke in der Welt. Falls Phase 1 erfolgreich abschließt aber ein Fehler in Phase 2 auftritt, wurde der TeleportStone trotzdem aus der Datenbank gelöscht. Hier sollte nun diese Änderung rückgängig gemacht werden.

- Rollback wenn löschen fehlschlägt.
- Zerstörte Blöcke werden nicht wieder hergestellt. Das scheint als könnte man das ausnutzen.

~~Undo Befehl~~

Für einen Undo Befehl müssen verschiedene Aspekte beachtet werden.

- Ein Undo Befehl kann nur für eine Aktion ausgeübt werden, für welche auch eine Gegenaktion vorhanden ist. D.h. für CREATE muss ein DELETE Befehl existieren. Für den Distance Befehl existiert nicht wirklich eine sinnvolle Gegenaktion. (Vielleicht den Chat des Spielers löschen?)
- Die letzte Aktion oder eine Liste letzter Aktionen muss geführt werden.
- Ist der Undo Befehl spielerabhängig oder wird er genutzt, um alle Änderungen rückgängig zu machen? Falls der Undo Befehl spielerabhängig ist, muss pro Spieler eine Liste letzter Aktionen geführt werden.

Da die Funktionalität und Verwendung dieses Befehls im unteren Bereich liegt und der Arbeitsaufwand im Vergleich hoch ist, wird diese Idee vermerkt für spätere optionale Versionen.

Version 0.6 Better Commands [UNTESTED]

- Den Befehl "signs" umbenennen zu "list" ~~oder "ls"~~
- Hilfebefehl verbessert.
- Bugfix: Leerzeichen von Namen wurden im "delete", "manualdiscover" und "destroy" Befehl nicht beachtet.

Auto Completion

Command auto complete halt. Generell werden nun alle SubCommands angezeigt, wenn /tpst geschrieben wurde. Die SubCommands implementieren nun auch für individuelle Argumente mögliche Werte die angezeigt werden:

- **tp** Zeigt alle entdeckten TeleportStones als Option an.
- **delete** Zeigt alle TeleportStones an.
- **destroy** Zeigt alle TeleportStones an.
- **manualdiscover** Zeigt zuerst alle registrierten Spieler und alle offline/online Spieler an und dann alle TeleportStones die vom angegebenen Spieler noch nicht entdeckt wurden.
- **undiscover** Zeigt zuerst alle registrierten Spieler an und die entdeckten TeleportStones des Spielers.
- **unregister** Zeigt alle registrierten Nutzer an.

Version 0.7 Unterschiedliche Typen von TeleportStones (Teil 1) [UNTESTED]

Formen und Materialien

Die Struktur des TeleportStone's könnte auch aus anderen Blöcken bestehen. Im Moment besteht eine TeleportStone aus einem Schild und den 12 Blöcken der Struktur, wie im Abschnitt [TeleportStone \(TpSt\)](#) beschrieben. TeleportStones sollen sich nun von einem festen Aussehen loslösen.

TeleportStones können nun ganz andere Formen haben und aus anderen Materialien bestehen. Dieses wird umgesetzt, indem jedem Typ eines TeleportStones Offsets angegeben werden. Als Ausgangspunkt dient der Block an dem das Schild befestigt ist.

OffsetBlock Ist ein Block der die relative Position zu dem Block, an welchem das Schild befestigt ist, beinhaltet. Auch kann eine Liste an möglichen Materialien für den Block angegeben werden.

- Relative Koordinate X
- Relative Koordinate Y
- Relative Koordinate Z
- Liste an möglichen Materialien für den Block

Der Block mit der relativen Position (0, 0, 0) beschreibt den Block, an dem das Schild befestigt ist.

Beispiel:

- OffsetBlocks:
 - OffsetBlock1 (0, 0, 0, (STONE))
 - OffsetBlock2 (0, 1, 0, (STONE))
 - OffsetBlock3 (0, 2, 0, (GOLD_BLOCK, DIAMOND_BLOCK))
- Nun befindet sich in der Spielwelt an Position (10, 10, 10) ein Schild, welches an dem Block (10, 10, 11) befestigt ist.
- Wenn man nun die Positionen der OffsetBlocks auf die Spielwelt überträgt, erhält man:
 - OffsetBlock1 → Block(10, 10, 11)
 - OffsetBlock2 → Block(10, 11, 11)
 - OffsetBlock3 → Block(10, 12, 11)
- Nun müssen nur noch die Materialien der Blöcke der Spielwelt an die Materialien der OffsetBlocks angepasst werden und man erhält eine valide Struktur.

Es muss beachtet werden, dass das Schild an einem Block platziert ist und nicht frei steht.

Nun kann man für jeden Typ eines TeleportStones eine individuelle Liste an OffsetBlocks erstellen. Die Idee mit Offsets wurde schon für den TeleportStone aus Abschnitt [TeleportStone \(TpSt\)](#) genutzt, wird aber nun auf eine allgemeine, konfigurierbare Ebene angehoben.

Scopes

Scopes beschreiben die Erreichbarkeiten von TeleportStones. Einem TeleportStone wird 1 Scope zugeordnet. Für dieses Scope ist konfiguriert, zu welchen anderen Scopes von diesem TeleportStone teleportiert werden kann.

Beispiel

- Es existieren zwei Scopes: **ScopeA**, **ScopeB**
 - **ScopeA** kann zu TeleportStones mit **ScopeA** und **ScopeB** teleportieren.

- **ScopeB** kann nur zu anderen TeleportStone auch mit **ScopeB** teleportieren.
- Es existieren drei Typen von TeleportStones: **T1, T2, T3**
 - Diese können alle unterschiedliche Strukturen und Materialien verwenden. Etc.
- Jedem TeleportStone-Typen wird nun **exakt 1** Scope zugewiesen.
 - **T1** → **ScopeA**
 - **T2** → **ScopeB**
 - **T3** → **ScopeB**
- So können TeleportStones von Typ **T2** und **T3** untereinander teleportieren, aber nicht zu **T1** und TeleportStones vom Typ **T1** können zu allen anderen Typen teleportieren.

Für ein weiteres Beispiel können die mitgelieferten TeleportStones betrachtet werden. Siehe Abschnitt [Mitgelieferte Lösungen](#).

Standard Scopes

Die Standard Scopes bieten generelle Lösungen für TeleportStones an.

- **All** Ist erreichbar von jedem Scope und kann zu jedem Scope teleportieren.
- **Reachable** Ist erreichbar von jedem Scope.

Weitere Merkmale

Neben Aussehen und Scopes können TeleportStones noch anders konfiguriert werden.

Teleportreichweite

Ein weiteres Merkmal eines TeleportStone's könnte seine Teleportationsreichweite sein. Dabei wird ein Blockradius angegeben und nur andere TeleportStones im Radius sind erreichbar.

Wichtig zu beachten ist hierbei, das nicht überprüft werden kann, ob sich ein TeleportStone im Radius eines anderen TeleportStone befindet, wenn diese nicht in der selben Welt platziert sind.

Weltgebunden

Wenn ein TeleportStone an die Welt, in der er platziert wurde, gebunden werden soll, kann die Option gesetzt werden. Ist diese Option gesetzt, kann dieser nur zu anderen TeleportStones in der selben Welt teleportieren.

Baupläne (BluePrints)

Um die oben beschriebenen Erweiterungen umzusetzen, werden Baupläne eingeführt. Wie ein TeleportStone aussieht und was seine Eigenschaften sind, wird nun nicht mehr fest im Programmcode festgehalten, sondern aus externen Dateien ausgelesen. Der Inhalt der Dateien muss dabei einem gewissen Format folgen, dem Bauplanformat.

Ein Bauplan hat folgende Einträge:

- **Bauplan ID** Ist eine eindeutige Bezeichnung für einen Bauplan. Die ID kann dabei ein Name, Text oder einfach eine Nummer sein.
- **Scope Name** Ist eine Referenz auf einen Namen eines Scopes.
- **BlockOffsets** Eine Liste an BlockOffsets, die die Struktur eines TeleportStones bilden.

- **Schildmaterialien** Eine Liste an möglichen Materialien aus denen das Schild des TeleportStones bestehen kann.
- **istWeltgebunden** Ein boolescher Wert (True/False), der angibt ob ein TeleportStone weltgebunden ist.
- **Reichweite** Ein numerischer Wert der die Teleportreichweite beschreibt.

Umgestaltung TeleportStone

Ein gebauter aufgestellter TeleportStone (nicht Bauplan) besteht im Moment aus folgenden Werten:

- ID
- Name
- Welt
- Schild (SimpleSign)
- Blöcke der Struktur (SimpleBlocks)

Durch die Einführung von Bauplänen muss die Definition eines TeleportStones etwas angepasst werden. Ein TeleportStone wird um folgenden Eintrag erweitert:

- **Bauplan ID** Für jeden aufgestellten TeleportStone wird nun auch die Bauplan ID abgespeichert.
 - Beim Erstellen und Laden eines TeleportStones wird dieser immer validiert. Darunter zählt eine valide Struktur und valide Werte für die Eigenschaften. In einem Bauplan werden diese Aspekte festgehalten werden.

Scope Validierer

Genau wie bei TeleportStones können auch in Scopes falsche Angaben gemacht werden. Zum Beispiel wird der Name nicht angegeben oder ein anderer Scope mit dem selben Namen existiert bereits, etc.

Auf diese Fehler wird nun geprüft. Der Scope Validierer testet verschiedene Aspekte:

- **MISSING_SCOPE** Scope ist NULL.
- **MISSING_NAME** Scope hat keinen Namen.
- **EMPTY_NAME** Scope hat leeren Namen.
- **RESERVED_NAME** Scope hat einen reservierten Namen, wie **All**.
- **NAME_ALREADY_EXISTS** Ein anderes Scope hat bereits den Namen.
- **MISSING_REACHABLE_SCOPES** Reachable Scopes ist NULL.
 - Dabei ist anzumerken, dass ein Scope keine erreichbaren Scopes braucht, aber zumindest muss eine leere Liste angegeben werden.
- **UNKNOWN_REACHABLE_SCOPE** Ein Scope hat in der Liste der erreichbaren Scopes unbekannte Scopes vermerkt.

Bauplan Validierer

Genau wie bei TeleportStones können auch in Bauplänen falsche Angaben gemacht werden. Zum Beispiel wird die ID nicht angegeben oder ein OffsetBlock hat keine Materialien, etc.

Auf diese Fehler wird nun geprüft. Der Bauplan Validierer testet verschiedene Aspekte:

- **MISSING_BLUEPRINT** Bauplan ist NULL
- **MISSING_ID** Bauplan hat keine ID.

- **EMPTY_ID** Bauplan ID ist leer.
- **ID_ALREADY_EXISTS** Bauplan ID ist schon vergeben.
- **MISSING_SCOPE** Bauplan hat kein Scope.
- **EMPTY_SCOPE** Bauplan hat leeres Scope.
- **UNKNOWN_SCOPE** Angegebenes Scope ist unbekannt.
- **NOT_SET_VARIABLE_WORLDBOUND** Der Wert, ob ein TeleportStone weltgebunden ist, ist nicht angegeben.
- **RANGE_SPECIFIED_IN_NOT_WORLDBOUND** Ein TeleportStone kann nicht gleichzeitig **nicht weltgebunden** sein **und** eine **Reichweite haben**.
- **INVALID_RANGE** Die angegebene Reichweite ist nicht zulässig. Beispielsweise ein negativer Wert.
- **MISSING_OFFSET_BLOCKS** OffsetBlocks ist NULL oder leer.
- **INVALID_OFFSET_BLOCK** Ein angegebener OffsetBlock hat invalide Einträge.
- **MISSING_SIGN_MATERIALS** Möglichen Materialien ist NULL.
- **EMPTY_SIGN_MATERIALS** Keine möglichen Materialien für das Schild angegeben
- **INVALID_SIGN_MATERIAL** Angegebene mögliche Materialien sind invalide. Beispielsweise wurde ein unbekanntes Material angegeben oder ein Tippfehler happend.

Schlüsselwörter verbieten

Einige Namen und Einträge haben Standardwerte bzw. reservierte Werte. Beispielsweise All oder Reachable. Diese Wörter sind als Angaben verboten und sollten auch durch Validierer abgefangen werden.

Mitgelieferte Lösungen

Baupläne und Scopes bieten die Möglichkeit das TeleportStonePlugin ganz eigen zu gestalten. Aber auch werden einige Lösungen mitgeliefert, welche eine sinnvolle Basis darstellen. Dabei kann man aus mehreren Möglichkeiten (Paketen) auswählen.

Natürlich können die mitgelieferten Baupläne und Scopes auch deaktiviert werden.

Mitgelieferte Scopes

Scope Name	Possible Destinations
Local	Local, Regional, Global, Regional+, Global+
Regional	Regional, Global, Regional+, Global+
Global	Global, Global+
Regional+	Local, Regional, Global, Regional+, Global+
Global+	Regional, Global, Regional+, Global+

Mitgelieferte Pakete

- **One Stone To Rule Them All**
 - Es gibt 1 Sorte an Standard TeleportStones. Ein Standard TeleportStone kann zu jedem TeleportStone teleportieren.
 - 1 Bauplan
- **Simple**

- Es gibt 2 Sorten an Standard TeleportStones. Nur **Global** und **Regional**.
 - Bauplan **Global** hat Scope Global
 - Bauplan **Regional** hat Scope Regional, ist weltgebunden
- 2 Baupläne
- **Advanced**
 - Es mehrere Sorten an Standard TeleportStones. Sie haben Erreichbarkeiten und Reichweiten konfiguriert.
 - Bauplan **Global** hat Scope Global
 - Bauplan **Regional** hat Scope Regional, ist weltgebunden
 - Bauplan **Global+** hat Scope Global+
 - Bauplan **Regional+** hat Scope Regional+, ist weltgebunden
 - Bauplan **Local500** hat Scope Local, ist weltgebunden, hat Reichweite 500
 - Bauplan **Local1200** hat Scope Local, ist weltgebunden, hat Reichweite 1200
 - Bauplan **Local3000** hat Scope Local, ist weltgebunden, hat Reichweite 3000
 - Bauplan **Local6700** hat Scope Local, ist weltgebunden, hat Reichweite 6700
 - 8 Baupläne
- **Disable** Deaktiviert alle Standard TeleportStones. Es werden nur Custom TeleportStones geladen.

Für das Aussehen und Materialien der einzelnen Baupläne müssen die Dateien aufgesucht werden.

Custom Baupläne und Scopes

Hinweis Um Erreichbarkeiten von TeleportStones leicht unterscheiden zu können, sollen TeleportStones sich anhand dieser auch visuell unterscheiden. So könnten andere Materialien und Strukturen die Reichweite erhöhen bzw. verringern.

Baupläne und Scopes können genutzt werden, um eigene Ideen umzusetzen. Im Plugin Folder des TeleportStone Plugin existieren zwei Ordner, **Custom Blueprints** und **Custom Scopes**.

In diese Ordner müssen, jeweils eigen angefertigte Baupläne und Scopes gelegt werden, sodass diese geladen werden.

Test Custom Baupläne und Scopes

- **/tpst blueprints** Gibt alle geladenen Baupläne aus.
- **/tpst scopes** Gibt alle geladenen Scopes aus.

Version 0.8 GUI Menu

Ein grafische Oberfläche die umständliche Nutzung von Befehlen reduziert, ist schon seit Anfang dieses Projektes ein erstrebenswertes Feature. In einem Mod für Vanilla Minecraft, welche in den Referenzen für dieses Projekt vermerkt ist, wurde ein GUI speziell angefertigt, welches das Teleportieren durch ein paar einfache Klicks abwickeln lässt. Solch eine Oberfläche ist in SpigotMC aber nicht umsetzbar. (Soweit wie ich gesehen habe.) In SpigotMC sind GUI Menüs durch Inventare und ummodellerte Items gelöst.

So wird hier nun auch eine Lösung mit Inventar und Items angeboten.

Pageable Inventory

Wenn ein Spieler einen entdeckten TeleportStone anklickt, wird ihm eine Oberfläche (Inventar) geöffnet, in welchem alle TeleportStones gelistet sind, die er von diesem erreichen kann. Dabei wird beachtet, welche TeleportStone er entdeckt hat.

Das Inventar hat eine Platzanzahl von 27 Feldern und eine Dimension von 9 Breite und 3 Höhe. In den Zeilen 1 und 2 werden die erreichbaren TeleportStones als Türen dargestellt. Somit passen in ein Inventar 18 erreichbare TeleportStones.

Nun stellt sich die Frage: Was passiert wenn der Spieler mehr als 18 TeleportStones von einem TeleportStone aus erreichen kann? Die Antwort: **Pagination**. Falls mehr als 18 TeleportStones erreichbar sind, werden diese über mehrere Inventarseiten verteilt. Ist eine Inventarseite voll (d.h. 18 TeleportStones), kann auf die nächste Seite gewechselt werden und so weiter. Die "Knöpfe" um Seiten zu wechseln, werden in die dritte freie Zeile des Inventares platziert. Der Knopf um auf die vorherige Seite zu wechseln, befindet sich im ersten Platz der dritten Zeile und der Knopf um auf die nächste Seite zu wechseln, befindet sich im letzten Platz der dritten Zeile.

Favorit TeleportStones

Wenn viele TeleportStones auf dem Server existieren und ein Spieler viele davon entdeckt hat, kann die Seitenanzahl sehr hoch werden. Um zu vermeiden, dass ein Spieler immer alle Seiten durchsuchen muss, kann er TeleportStones zu seinen Favoriten hinzufügen. Favoriten werden immer auf den ersten Seiten angezeigt.

Dieses Feature setzt eine Änderung der User Datenbank voraus. Siehe weiter unten.

Auch müssen neue Befehle für das Hinzufügen und Entfernen von Favoriten erstellt werden. Siehe weiter unten.

Auch müssen neue Befehle für das Hinzufügen und Entfernen von Homes erstellt werden. Siehe weiter unten.

Alphabetische Sortierung

Die TeleportStones in den Seiten sollten alphabetisch sortiert werden, da sonst Suchen sehr umständlich werden würden. Jeder TeleportStone müsste betrachtet werden.

Weitere Knöpfe

Die dritte Zeile bietet noch 7 freie Plätze für andere Knöpfe.

- Zum Beispiel ein Knopf zum löschen des TeleportStones.
- Bsp. Knopf um zu Favoriten hinzuzufügen. (Painting and Itemframe)
- Bsp. Knopf um als Home festzulegen. (CAMPFIRE and SOULFIRE)

TODO

Änderungen der User Datenbank

Ein User besteht im Moment aus der **UUID**, dem **Username (Ingame name)** und einer **Liste an IDs der entdeckten TeleportStones**.

Diese Einträge werden nun erweitert. Neben IDs der entdeckten TeleportStone IDs werden nun auch die Namen der TeleportStones abgespeichert. Dies kann zum einen genutzt werden, um die Datenbankeinträge zu validieren und zum anderen beschleunigt diese extra Angabe einige Suchvorgänge. So müssen die IDs nicht erst mit allen TeleportStones der Datenbank verglichen werden, wenn alle Namen der entdeckten TeleportStones ausgegeben werden soll.

- IDs und Namen von TeleportStones speichern.

Um Favoriten zu realisieren wird eine extra Liste angelegt. Wird nun ein entdeckter TeleportStone ein Favorit, wird er aus der normalen Liste entdeckter TeleportStones entfernt und in die Favoritenliste eingefügt. Dies hat zur Folge, dass wenn alle entdeckten TeleportStones ausgegeben werden sollen, müssen die Einträge aus beiden Listen zusammen zurück gegeben werden.

- Favoritenliste. TeleportStones in der Favoritenliste werden aus der normalen Liste entdeckter TeleportStones entfernt.

Neben Favoriten soll auch **genau 1** TeleportStone als Home bestimmt werden können. Dieser HomeStone muss dann aus der Favoritenliste und normalen Liste entdeckter TeleportStones entfernt werden.

- HomeStone.

User Validator

Nutzer und die Listen der Nutzer werden nun validiert bei jedem Start.

Befehle

- **/tpst favorite <nameTeleportStone>** Entfernt TeleportStone von Favoriten bzw. fügt TeleportStone zu Favoriten hinzu.
- **/tpst home <nameTeleportStone>** Bestimmt einen TeleportStone als Home bzw. entfernt einen TeleportStone als Home.

Falls ein TeleportStone ein Favorit ist und dann als Home bestimmt wird, wird er aus den Favoriten entfernt. Und vise versa.

Version 0.9 Konfiguration

Eine Konfiguration wurde eigentlich schon in Version 0.7 eingeführt. In diesem Abschnitt werden schon gemachte und weitere Änderungen beschrieben.

Namensänderungen von Dateien

- Neue Datei **storageTeleportStone**: Speichert alle aufgestellten TeleportStones
- Neue Datei **storageUser**: Speichert alle Nutzer
- Entfernte Datei **custom.yml**: Alte Datei in der alles gespeichert wurde.
- Neue Datei **config.yml**: Konfigurationsdatei, die hier genauer beschrieben wird.

Plugin Konfiguration

- Enable Plugin
 - Values: true/false
 - Default: true
- Keyword
 - Values: String
 - Default: TeleportStone
- Minimal TeleportStone name length:
 - Values: Integer
 - Default: 3
- Teleport Protect Method
 - Values: Enumeration
 - Default: INTERN
 - Other possible values: WORLD_GUARD
- Auto Safe Intervall:
 - Values: Float
 - Default: 3.0
- Enable tab completion:
 - Values: true/false
 - Default: true
- Default TeleportStones:
 - Values: Enumeration
 - Default: NORMAL
 - Other possible values: ONE_STONE_TO_RULE_THEM_ALL, SIMPLE, ADVANCED, DISABLE

Default Packages überarbeiten

Die vorgestellten mitgelieferten Pakete in Version 0.7, werden hier etwas überarbeitet. Es existieren weiterhin die Pakete "One Stone To Rule Them All", "Simple" und "Advanced". Diese Pakete nutzen alle die gleichen Scopes. Dies wird nun geändert. Jedes Paket lädt separat auch eigene Scopes. Auch wird ein weiteres Paket "normal" eingeführt welches leicht zu verstehen sein sollte, aber auch eine gewisse Komplexität mitbringt.

Mitgelieferte Pakete **Überblick**

- **One Stone To Rule Them All**
 - Es gibt 1 Sorte an Standard TeleportStones. Ein Standard TeleportStone kann zu jedem TeleportStone teleportieren. (Er gehört dem Standard Scope All an.)
 - 1 Bauplan
 - 0 Scopes

Scope Name Possible Destinations

- **Simple**
 - Es gibt 2 Sorten an Standard TeleportStones. Nur **Global** und **Regional**.
 - Bauplan **Global** hat Scope Global
 - Bauplan **Regional** hat Scope Regional, ist weltgebunden
 - 2 Baupläne
 - 2 Scopes

Scope Name	Possible Destinations
Regional	Regional, Global
Global	Global, Regional

- **Normal**

- Es mehrere Sorten an Standard TeleportStones. Sie haben Erreichbarkeiten und Reichweiten konfiguriert.
 - Bauplan **Global** hat Scope Global
 - Bauplan **Regional** hat Scope Regional, ist weltgebunden
 - Bauplan **Local500** hat Scope Local, ist weltgebunden, hat Reichweite 500
 - Bauplan **Local1200** hat Scope Local, ist weltgebunden, hat Reichweite 1200
 - Bauplan **Local3000** hat Scope Local, ist weltgebunden, hat Reichweite 3000
 - Bauplan **Local6700** hat Scope Local, ist weltgebunden, hat Reichweite 6700
- 6 Baupläne
- 3 Scopes

Scope Name	Possible Destinations
Local	Local, Regional, Global
Regional	Regional, Local, Global
Global	Global, Local, Regional

- **Advanced**

- Es mehrere Sorten an Standard TeleportStones. Sie haben Erreichbarkeiten und Reichweiten konfiguriert.
 - Bauplan **Global** hat Scope Global
 - Bauplan **Regional** hat Scope Regional, ist weltgebunden
 - Bauplan **Global+** hat Scope Global+
 - Bauplan **Regional+** hat Scope Regional+, ist weltgebunden
 - Bauplan **Local500** hat Scope Local, ist weltgebunden, hat Reichweite 500
 - Bauplan **Local1200** hat Scope Local, ist weltgebunden, hat Reichweite 1200
 - Bauplan **Local3000** hat Scope Local, ist weltgebunden, hat Reichweite 3000
 - Bauplan **Local6700** hat Scope Local, ist weltgebunden, hat Reichweite 6700
- 8 Baupläne
- 5 Scopes

Scope Name	Possible Destinations
Local	Local, Regional, Global, Regional+, Global+
Regional	Regional, Global, Regional+, Global+
Global	Global, Global+
Regional+	Local, Regional, Global, Regional+, Global+
Global+	Regional, Global, Regional+, Global+

- **Disable** Deaktiviert alle Standard TeleportStones. Es werden nur Custom TeleportStones geladen.

Für das Aussehen und Materialien der einzelnen Baupläne müssen die Dateien aufgesucht werden.

Anmerkung: In jedem Paket sind die Standard Scopes **All** und **Reachable** enthalten.

Event Log in extra Datei speichern?

Nein, da sie sowie im Log stehen.

Version 0.10 Protect TeleportStone / Permissions

Im Moment können TeleportStones einfach zerstört werden. Auch können Explosionen ihn zerstören. Enderman nicht vergessen.

Jeder Block eines TeleportStone muss geschützt werden. Möchte ein unberechtigter Spieler einen TeleportStone zerstören bzw. einen Block von diesem abbauen, muss dieses Event unterbrochen werden. Andererseits könnte ein berechtigter Spieler einen Block eines TeleportStones abbauen, muss der TeleportStone automatisch entfernt werden.

Blöcke des TeleportStones schützen

- Schutz vor dem Abbau der Blöcke durch Spieler
- Schutz vor Explosionen
- Schutz vor Enderman

Die Implementierung eines Schutzverfahren stellt eine Performanceherausforderung dar. Nicht jedes mal wenn ein Block abgebaut wird, will man alle Blöcke von allen TeleportStones auf Übereinstimmung prüfen.

Eine **erste Lösungsidee** nutzt Meta-Data. Diese Idee funktioniert, die Implementation erweist sich aber doch etwas anders als gedacht. Meta Data wird nach jedem Neustart des Servers entfernt, sie ist nicht persistent. Deswegen muss nach jedem Neustart die Meta Data wieder in die Blöcke der TeleportStones geschrieben werden.

Meta Data

- ID des TeleportStone's
- Name des TeleportStone's
- Erbauer des TeleportStone's
- Eigentümer des TeleportStone's
- Erstelldatum des TeleportStone's

Diese Einträge erlauben schnelle Tests, ob der Eigentümer bzw. Erbauer den TeleportStone abreißen möchte.

Änderungen an TeleportStone

Ein TeleportStone muss um folgende Eigenschaften erweitert werden:

- **Builder** beschreibt den Spieler, der den TeleportStone gebaut hat. Dabei wird der Spieler, der das Schild setzt als Builder anerkannt.

- **Owner** Beschreibt den Spieler, dem der TeleportStone gehört. **Builder** und **Owner** müssen nicht übereinstimmen. Falls ein Spieler keine Berechtigung hat einen TeleportStone zu errichten, kann er eine Anfrage stellen. Ein berechtigter Spieler kann die Anfrage dann akzeptieren oder ablehnen. Akzeptiert er die Anfrage wird er als Owner eingetragen. **Für Anfragen siehe weiter unten.**
- **Erstellungsdatum** Datum an dem der TeleportStone aufgestellt bzw. die Anfrage akzeptiert wurde.

TeleportStone Validator

Tests auf Ersteller und Eigentümer müssen nach der Nutzerdatenbankinitialisierung erfolgen.

Platzierungsanfragen für TeleportStones

Permissions können Spielern verbieten TeleportStones aufzustellen. Ein Spieler könnte aber einer Permission-Gruppe angehören, die ihm erlaubt Anfragen für TeleportStones zu stellen.

Eine Anfrage wird automatisch erstellt, wenn ein Spieler, der keine Erlaubnis für das Erstellen von TeleportStones, aber eine Erlaubnis für Anfragen hat, probiert einen TeleportStone zu erstellen. Platziert solch ein Spieler das Schild an einer validen Struktur, wird der TeleportStone automatisch als Anfrage abgespeichert.

Dabei kann ein Spieler nur eine bestimmte Anzahl gleichzeitiger Anfragen haben. Als Standardwert wird 3 gewählt.

Anfrage

Eine Anfrage besteht aus:

- Datum
- Spieler der Anfrage stellt
- ID des TeleportStone
- Name des TeleportStone
- TeleportStone

Ein angefragter TeleportStone wird in einer extra Datei requests.yml gespeichert. In diesem sind alle Felder bis auf "Owner" ausgefüllt. Als Datum wird vorübergehend das Anfragedatum festgelegt.

Angefragte TeleportStones werden auch geschützt.

Bearbeiten von Anfragen

Befehle

- **/tpst request accept [requestTeleportStoneName]**
 - Akzeptiert nächste TeleportStone-Anfrage bzw. akzeptiert TeleportStone-Anfrage mit angegebenem Namen.
- **/tpst request deny [requestTeleportStoneName]**
 - Verweigert nächste TeleportStone-Anfrage bzw. verweigert TeleportStone-Anfrage mit angegebenem Namen.
- **/tpst request show [requestTeleportStoneName]**
 - Zeigt eine Übersicht über nächste TeleportStone-Anfrage bzw. zeigt eine Übersicht über TeleportStone-Anfrage mit angegebenem Namen.

- Übersicht:
 - Datum
 - Anfragesteller
 - Name des TeleportStones
 - Koordinaten
 - Welt
 - Bauplan

Permissions

- Commands
- Create
- Destroy
- Nutzen

Befehle

Übersicht über seine Rechte ausgeben lassen. Kann man TeleportStone erstellen und so.

Version 0.11 Unterschiedliche Typen von TeleportStones (Teil 2)

Ingame Konfiguration

Um einzelnen TeleportStones unabhängig von Typen, etc. individuelle Einstellungen zu verleihen, werden im folgenden noch weitere Optionen angeboten.

Diese Einstellungen können in den Konfigurationen für TeleportStone aktiviert bzw. deaktiviert werden.

Gruppierung

TeleportStones können mit einem Befehl Gruppen hinzugefügt werden. Ein TeleportStone kann mehreren Gruppen angehören. Gehört ein TeleportStone einer Gruppe an, kann er nur zu anderen TeleportStones der Gruppe teleportieren. Er ist auch nicht mehr erreichbar von anderen TeleportStones, die nicht der Gruppe angehören.

Soll ein TeleportStone, welcher in einer Gruppe ist, trotzdem noch von anderen TeleportStones, die nicht in der Gruppe sind, erreichbar sein, muss er der Standardgruppe **Reachable** hinzugefügt werden.

Alle anderen Restriktionen des TeleportStones gelten weiterhin.

Das ein TeleportStone einer Gruppe angehört, wird auf der vierten Zeile eines Schildes vermerkt. Gruppen werden durch das Zeichen # kenntlich gemacht. **Beispielschild:**

- Erste Zeile: TeleportStone
- Zweite Zeile: Teststein
- Dritte Zeile:
- Vierte Zeile: #Teststeine

Hierbei ist "Teststeine" der Name der Gruppe.

Falls ein TeleportStone mehreren Gruppen angehört, welche nicht auf den Platz der vierten Zeile des Schildes passen, wird **#Multiple Groups** geschrieben. Auch muss die Länge von Gruppennamen begrenzt werden.

Falls ein TeleportStone der StandardGruppe angehört, wird ein zweites # an den Anfang der vierten Zeile platziert.

Befehl

- /tpst addGroup <nameTeleportStone> ; <nameGroup>
 - Man beachte das Semikolon ;
 - Der Befehl erstellt automatisch die Gruppe falls diese noch nicht existiert.

Ausgewählte TeleportStones Ziele

Ähnlich zu Gruppierungen bietet diese Optionen eine weitere Möglichkeit TeleportStones zu beschränken. Ist für einen TeleportStone-Typen die Option **Selected TeleportStones** aktiviert, können per Befehl einem platzierten TeleportStone andere TeleportStones als mögliche Ziele hinzugefügt werden. Dabei müssen die hinzugefügten TeleportStones alle anderen Restriktionen (Scope, Radius, etc.) erfüllen.

Werden einem TeleportStone andere TeleportStones als mögliche Ziele hinzugefügt, entfallen alle anderen normalerweise erreichbaren TeleportStones.

Befehl

- /tpst addDestination <nameTeleportStone> ; <nameTeleportStone>
 - Man beachte das Semikolon ;

Ausgewählte Welten

Auch können die Welten beschränkt werden, die von einem platzierten TeleportStone erreichbar sind. Dazu

Befehl

- /tpst addDestination <nameTeleportStone> ; <nameWorld>
 - Man beachte das Semikolon ;
- /tpst currentWorld
 - Dieser Befehl gibt den Namen der Welt aus, in der man sich grade befindet.

Visual Effects

Vielleicht ist es ganz schön, wenn einige visuelle Effekte beim Teleportieren zu sehen sind. Beispielsweise könnte ein Blitz in den TeleportStone einschlagen zu dem man teleportiert.

- **Blitz beim Porten** Wie schon gesagt könnte ein Blitz beim Teleportieren in den TeleportStone einschlagen. Dieser soll lautlos sein und keinen Schaden verursachen (oder vielleicht doch wenn Leute auf dem TeleportStone rumspielen, wo sie nicht spielen sollten 🤪).
- Umsetzbar. `player.getLocation().getWorld().strikeLightning(player.getLocation()).setSilent(true);`

- **Beacon Beam** Ein anderer visueller Effekt wäre der Strahl eines Beacon's. Dieser Idee wird aber erst mal nicht weiter nachgegangen, da erste Suchen keine Lösungen für die Umsetzung anboten. (Idee vermerkt in späteren Abschnitten.)
- **Farbe und Effekte der Schildbeschriftung** Um deutlich zu machen, dass es sich bei einem Schild nicht um ein normales Schild handelt, sollte die Beschriftung gefärbt werden. (Vielleicht gibt es auch einige Schrifteffekte.)

In der Struktur bzw. im Bauplan muss der Block, von dem die Effekte ausgehen, mit angegeben werden.

Auch sollte jeder TeleportStone individuell konfigurierbar sein, ob visuelle Effekte angewendet werden sollen.

Version 0.12 Thread Safty

I don't wanna.

Factory Pattern

Maybe good opportunity to learn junit

Version 0.13 Performance

- Better data structures
- Better Search Methods
- Real Database

Version 1.0 Finish

Code documentation

Mehr Ideen und Features

High Priority

Info in GUI

In der GUI noch die Typen, Scopes, Koordinaten, Welt eines TeleportStones in die Meta schreiben.

Auf den letzten Seiten (mit Eisentüren) die nicht erreichbaren TeleportStones schreiben und warum diese nicht erreichbar sind.

In der unteren Zeile ein Info Item, welches Info über den aktuellen TeleportStone als Lore hat.

Befehle überarbeiten und zusammenfassen

Mit der Zeit sammeln sich viele Befehle an. Viele von diesen braucht man selten. Manche Befehle zusammenlegen unter dem Überbefehl "maintenance". Befehle sollten vielleicht auch nicht angezeigt werden,

wenn man für sie keine Berechtigungen hat.

Exclusive Scope Angaben

Falls sehr viele Scopes existieren, möchte man sicherlich diese nicht in allen Bauplänen immer mit angeben. Vielleicht kann man Scope exklusiv angeben. Beispielsweise mit einem - vor dem Scope Namen. Dann werden alle Scopes **außer** dem **-Scope** den erreichbaren Scopes hinzugefügt.

Spielernamenänderung

Beim join eines Spielers prüfen, ob seine UUID immer noch mit seinem Namen übereinstimmt.

Teleport Position festlegen

Im Blueprint kann auch eine relative TeleportPosition festgelegt werden.

TeleportStone aktivieren und deaktivieren

Bei einem aktivierten TeleportStone könnte ein LightEffect oder so sein.

Sonstiges

Teleportationskosten

Auch können Teleportationskosten konfiguriert werden.

Teleporteffekte

- spezielle Buffs beim Nutzen
- Invincibility 5 Sekunden lang nach Teleport

Beacon Beam Effect

Nutzen und Kompatibilität mit anderen PlugIns

TpS automatisch mit WorldGuard sichern, wenn TpS erstellt wird. oder eigene Implementierung. (make configurable)

Low Priority

Startup Validation Failure Handler

Select what should happen with storage entries that fail to validate. (For example an incorrect ID or a block is missing.) Possible Values:

- PURGE (removes entry from storage and files)
- MOVE_BIN (move faulty entry to trash bin in plugin directory)
- TRY_FIX

onFailedStartUpValidation: PURGE

- TRY_FIX

- Falls keyword nicht übereinstimmt, dieses einfach durch das aktuelle ersetzen

Log Detail

Select how much should be logged. Possible Values:

- ALL
- LOW
- HIGH

logDetail: ALL

Minimaler Radius für TeleportCommand

Vielleicht konfigurierbar machen.

TeleportStone mit Schildern konfigurieren

- Add additional info / config by placing a sign on a TeleportStone (Sign will be read, and then destroyed)

Discarded

Behandlung mehrerer gleicher Namen

Undo Befehl

Ideen

More Discover

TeleportStones automatisch in der Welt generieren. Gibt ein TeleportStone Finder Item. So kann niemand einfach neben sein Haus ein TeleportStone setzen. Städte bilden sich um TeleportStone. TeleportStone kann auch mit einer bestimmten Wahrscheinlichkeit unterirdisch generiert werden. So muss ein nicer Runtergang gebaut werden.

Abstände abhängig von Biomen, etc.

Der Entdecker des TeleportStone darf ihn benennen.