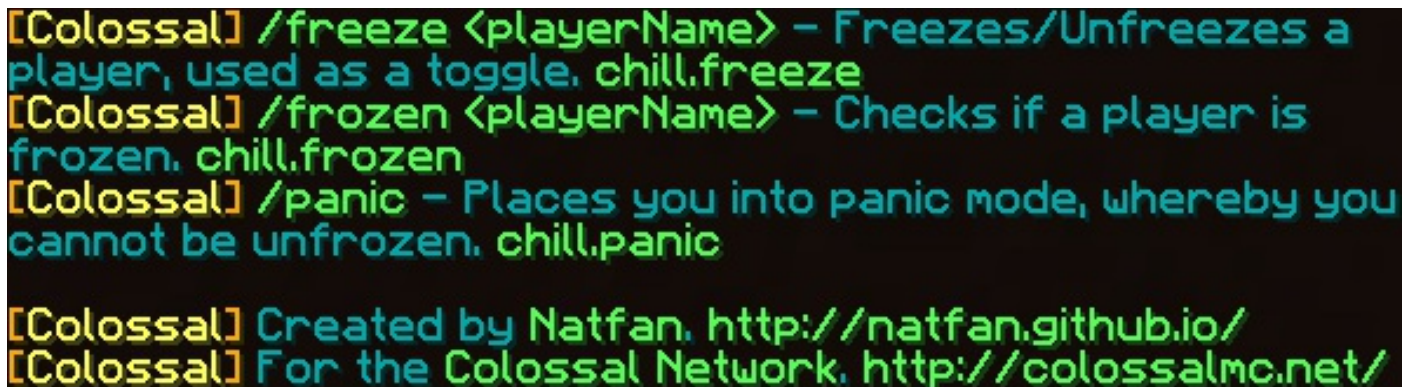# Unit XIV Assignment IV

By Nathan Windisch

PVI: Onscreen Help

For this segment I will be listing on screen help that will be displayed when the user enters a command wrong, or if they type in the command without the arguments.

The Main Help Page

The following page will be sent when a user executes the command `/chill`:

This is what the code for it looks like:

```
sender.sendMessage("/freeze <playerName> - Freezes/Unfreezes a player, used as a
toggle. chill.freeze");
sender.sendMessage("/frozen <playerName> - Checks if a player is frozen.
chill.frozen");
sender.sendMessage("/panic - Places you into panic mode, whereby you cannot be
unfrozen. chill.panic");
```

The Command Reference

Freeze

The following is what will be sent if the user execute too many or too few arguments after the '/freeze' command:
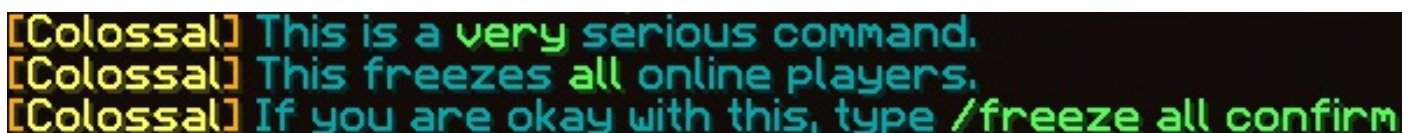
And this is what the code for it looks like:

```
sender.sendMessage("Please specify a player.");
```

Freeze All

The following is what will be sent if the user execute too many or too few arguments after the '/freeze all' command:

And this is what the code for it looks like:

```
sender.sendMessage("This is a very serious command.");
sender.sendMessage("This freezes all online players.");
sender.sendMessage("If you are okay with this, type /freeze all confirm");
```

Thaw

The following is what will be sent if the user execute too many or too few arguments after the '/thaw' command:

And this is what the code for it looks like:

```
sender.sendMessage("Please specify a player.");
```

Frozen

The following is what will be sent if the user executes too many or too few arguments after the `/frozen` command:

And this is what the code for it looks like:

```
sender.sendMessage("Please specify a player.");
```

MI: Technical Documentation

Overview

This program was developed for staff members on a game server that I own and administrate. The plugin allows those with specific permissions to execute commands that will let them stop suspected cheaters from interacting with the world, thus allowing legitimate players to enjoy the game without interruption. Once a player has been frozen, they will not be able to move, attack other players or interact with the environment, among other things. A menu will appear on their screen informing them as to why they have been frozen and the IP address to the voice server that they need to connect to so that they can talk to an Administrator. Once they have joined the voice server, they may need to download a piece of software that will allow the Administrator to scan their game files to ensure that they are not cheating. If they confess to cheating, then their punishment may be reduced. However, if they do not confess and it turns out that they were cheating, their ban will be permanent.

Prerequisites

There are a few prerequisites for being able to run this program, these are as follows:

- `Java 8`, needed to be able to run the server software.
- Preferably a VPS or Dedicated server to run the Minecraft server code on for easier setup.

- `Ubuntu` or `CentOS`, both in headless mode, are recommended as they are easy to set up and are lightweight.

- A `Minecraft` server running the software `Bukkit` or a variation thereof such as:

- `Spigot`
- `Paper`
- `Sponge`
  - Please note that the `Pore` plugin must be installed to be able to run `Bukkit` plugins.

- `Wolf in a Bukkit`
- `Trident`
- `ForgeBukkit`, or it's competitor `BukkitForge`

- Access to modify file permissions if running a Linux distribution, to allow `run.sh` to be executed.
- Internet access to be able to verify the clients connection with Mojang authentication servers.
- Internet access to be able to allow users to connect to the server outside of a LAN environment.

User Guide

Freezing

If you believe that a player is cheating beyond reasonable doubt, and you have recorded evidence of doing actions which either do not seem legitimate or set of large amounts/high numbers of AntiCheat violations, then you may freeze a player. You are also allowed to freeze a player if they are obviously cheating and are in the action of preventing other legitimate players from having an enjoyable experience, such as killing them to take their stuff. You are not allowed to freeze a player as a joke, even if they explicitly request it. Freezing is a serious action and will be broadcast to all online staff members. To prevent abuse, you cannot freeze other staff members as they have the `chill.override` permission. If you attempt to freeze a staff member then the action will be denied and all online staff will be notified of this. If you abuse the features of freezing you may be demoted or even banned from the server. The command to freeze a player is `/freeze <playerName>`
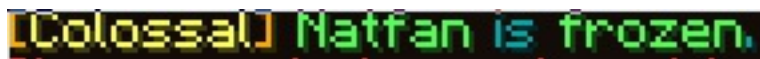
Freezing All Players

Only members of the Leadership and Ownership Team have the ability to freeze all players on the server, due to the fact that there is no `/thaw all` command. This is because there is currently no way to distinguish between a player that has been frozen in a single command, and players that have been frozen in a mass command, due to the fact that they are both added to the `frozen` ArrayList with the same method. Freezing all players without a good reason and adequate warning is a very serious offence, and can get the command removed from your repertoire. The command to issue this is `/freeze all`. You will then be prompted to confirm it, at which point you will have to have to type in `/freeze all confirm`.

Thawing

If you believe that the player that you froze was not cheating, you can either type `/freeze <playerName>` to toggle the frozen status of the player, or you can type `/thaw <playerName>` to unfreeze them directly. It is preferred that you use the latter command due to the fact that you could accidentally freeze an incorrect player. Please only thaw players that you personally have frozen, to prevent interference with other Administrators punishments issues.

Frozen

To check if a player is frozen, simply issue `/frozen` command to check if the player is frozen or not. The command output will either show if the player IS frozen, like so:

And this is what the output will be if the player is NOT frozen:

Finally, this is the output when the player is offline:

This is a very simple command and cannot really be abused. Please note that all information gained as a staff member should not be revealed to the player base unless a consensus from the Ownership Team is reached. While we try to be a transparent organisation, some inner workings of the Administrative side of things need to be kept secret.

Development Guide

Naming Schema

Please note that the following capitalisation is used to mean different things:

- Words like "Player" with a capital at the start are a custom data type.
- Words like "player" are a commonly used variable name.
- Words like "string" are primitive data types.

Variables

Players

Player

The `player` variable is assigned to the user that is the main target of the event within regular events that are prefixed with the `@EventHandler` tag. If the `player` variable is used in the CommandExecutor code, then it is a Player instantiation of the user that sent the command.

Target

The `target` variable is assigned to the user that is the main target of a CommandExecutor event, meaning that they are normally found in the arguments of a command. An example of this would be `/freeze <playerName>`, where the first argument (`args[0]`) is the target for the command

Sender

The `sender` variable is assigned to the Entity that executed the command. The `sender` is **always** the CommandSender type, and will be used before converted into the `Player` type. If a command can be run by both a player in-game and a user accessing the CLI, then the `sender` will not be assigned to a new `Player` type, as they will not need to be manipulated as such. An example of this is the `/frozen <playerName>` command, which merely outputs text to the user meaning that a non-player can still gain the output. If, on the other hand, the user needs to be manipulated then they will be passed to a `Player` type. An example of this is the `/panic` command, which requires the user to be placed in the `frozen` ArrayList, added effects to them and sent a message. The only thing that can be done to a Console in this instance is the messaging. So, the code checks if the sender is a `Player` and if it is, the code is executed. If it is not, an error is thrown.

Entities

An Entity is any dynamic, moving object within the game. This encompasses but is not limited to players, animals and monsters, along with projectiles and boats. These are need to be intercepted when preventing actions, and are converted into `Player` when used in specific events such as EntityDamageByEntityEvent and the like.

ChatColor

The ChatColor variable type is used for setting the colour of a message that is being sent to a user. The variable type is rarely used in actual lines of code, but master variables are used to enable changing of the colours of the messages on the fly. Examples of this are:

```
ChatColor pri = ChatColor.DARK_AQUA;
```

```
ChatColor sec = ChatColor.GREEN;
```

ItemStack

ItemStacks are all items that can be stored within an Inventory. These will be used when showing the menu to the player.

ItemMeta

ItemMeta is a subset of an ItemStack that is used for customizing the Item. It can change the name of the Item, as well as adding lore to the item.

Strings

Strings are used primarily in this code to store data such as commonly used text. Examples of this include the prefix text, which is `[COLOSSAL]`. That being said, Strings as a datatype are used a lot more frequently, such as when the user is sent a message.

ArrayLists

There is only really one usage of the ArrayList variable, and that is the `frozen` list. This is due to the fact that it can store large numbers of Unique Identifiers that can be accessed easily. Frozen lists can also be added to and removed with only one line of code, as follows:

```
arrayList.add(variable);
```

And can be removed thus:

```
frozen.remove(variable)
```

UUID

UUID, or Unique Identifier, is a Java method that can be used by Bukkit to check what the user's Identification is. In practical terms, this means that the user can be frozen on the server, log out, change their name, log back in and will still be frozen as their name is added to the ArrayList rather than their username.