

Unit XXIV Assignment III

By Nathan Windisch

PIV: Implementation

The following are pieces of code from individual classes. I shall comment lines that I believe are important.

Main.java

```
package net.survivalz.chill; // sets the package name so that the compiler knows what
to do

import net.survivalz.chill.events.*; // imports various APIs from bukkit and other
classes
import org.bukkit.ChatColor;
import org.bukkit.entity.HumanEntity;
import org.bukkit.entity.Player;
import org.bukkit.plugin.PluginManager;
import org.bukkit.plugin.java.JavaPlugin;

import java.util.ArrayList; // imports APIs from Java itself
import java.util.UUID;

public class ChillMain extends JavaPlugin { // the main class, where the magic
happens, extends JavaPlugin so that the server knows to run it
    public static ArrayList<UUID> frozen = new ArrayList<>(); // creating the list
that will be populated with frozen player's UUIDs
    public final static String prefix = (ChatColor.GOLD + "[" + ChatColor.YELLOW +
"Colossal" + ChatColor.GOLD + "]" + ChatColor.DARK_AQUA + " "); // setting up the
colours and prefixes so that I can call them easily later
    public final static ChatColor pri = ChatColor.DARK_AQUA;
    public final static ChatColor sec = ChatColor.GREEN;
    public final static String TeamSpeakStr = "TS.ColossalMC.net";

    public static void blank(Player player) { // a function that clears the player's
chat, will be called a lot
        for (int i = 0; i < 1; ++i) {
            player.sendMessage("");
        }
    }

    private onBlockBreak onBlockBreakEvent = new onBlockBreak(this); // creates new
variables for all of the events that are used and sets them to that event
    private onBlockPlace onBlockPlaceEvent = new onBlockPlace(this);
    private onCommandPreprocess onCommandPreprocessEvent = new
onCommandPreprocess(this);
    private onEnderpearlThrow onEnderpearlThrow = new onEnderpearlThrow(this);
    private onFrozenGiveDamage onFrozenGiveDamageEvent = new
onFrozenGiveDamage(this);
    private onPlayerDropItem onPlayerDropItemEvent = new onPlayerDropItem(this);
    private onInventoryCloseEvent onInventoryCloseEvent = new
onInventoryCloseEvent(this);
    private onPlayerInventoryClickItem onPlayerInventoryClickItem = new
onPlayerInventoryClickItem(this);
    private onPlayerMove onPlayerMoveEvent = new onPlayerMove(this);
    private onPlayerPickupItem onPlayerPickupItem = new onPlayerPickupItem(this);
    private onPlayerQuit onPlayerQuitEvent = new onPlayerQuit(this);
    private onPlayerTakeDamage onPlayerTakeDamageEvent = new
onPlayerTakeDamage(this);
```

```

private onPlayerTeleport onPlayerTeleport = new onPlayerTeleport(this);

public void onEnable() { // required in all plugins
    PluginManager pluginManager = getServer().getPluginManager(); // starts to
register tthe events
    pluginManager.registerEvents(onBlockBreakEvent, this); // registers all
events used
    pluginManager.registerEvents(onBlockPlaceEvent, this);
    pluginManager.registerEvents(onCommandPreprocessEvent, this);
    pluginManager.registerEvents(onFrozenGiveDamageEvent, this);
    pluginManager.registerEvents(onInventoryCloseEvent, this);
    pluginManager.registerEvents(onPlayerDropItemEvent, this);
    pluginManager.registerEvents(onPlayerInventoryClickItem, this);
    pluginManager.registerEvents(onPlayerMoveEvent, this);
    pluginManager.registerEvents(onPlayerPickupItem, this);
    pluginManager.registerEvents(onPlayerQuitEvent, this);
    pluginManager.registerEvents(onPlayerTakeDamageEvent, this);

    getCommand("freeze").setExecutor(new onCommand(this)); // registers all
commands used
    getCommand("ss").setExecutor(new onCommand(this));
    getCommand("frozen").setExecutor(new onCommand(this));
    getCommand("thaw").setExecutor(new onCommand(this));
    getCommand("unfreeze").setExecutor(new onCommand(this));
    getCommand("sscheck").setExecutor(new onCommand(this));
    getCommand("checkss").setExecutor(new onCommand(this));
    getCommand("checkfreeze").setExecutor(new onCommand(this));
    getCommand("freezecheck").setExecutor(new onCommand(this));
    getCommand("panic").setExecutor(new onCommand(this));
    getCommand("chill").setExecutor(new onCommand(this));
    getCommand("freezehelp").setExecutor(new onCommand(this));
    getCommand("helpfreeze").setExecutor(new onCommand(this));
    getCommand("sshhelp").setExecutor(new onCommand(this));
    getCommand("helpss").setExecutor(new onCommand(this));
}

public void onDisable() { } // required by default. as we are not saving anything
when the plugin disables, nothing is needed here
}

```

MenuItems.java

```
package net.survivalz.chill;

import org.bukkit.ChatColor;
...

import java.util.ArrayList;

public class MenuItems {
    public static ItemStack close() { // creates a new item that I can call later
        ItemStack item = new ItemStack(Material.STAINED_GLASS_PANE, 0, (short) 14);
        // sets the item to be a stained glass pane with the metadata of 14, meaning that the
        // colours is set to red
        ItemMeta itemMeta = item.getItemMeta(); // starting to set the itemmeta
        itemMeta.setDisplayName(ChatColor.RED + "Close Menu"); // sets the item's
name
        ArrayList<String> lore = new ArrayList<>(); // sets the item's lore, unneeded
in this context
        lore.add("");
        itemMeta.setLore(lore); // adds lore to the item
        item.setAmount(1); // sets the amount of items in the stack to be 1
        item.setItemMeta(itemMeta); // applies the itemmeta to the item

        return item; // returns the final item
    }

    public static ItemStack info() {
        ItemStack item = new ItemStack(Material.PAPER);
        ...

        return item;
    }

    public static ItemStack admit() {
        ItemStack item = new ItemStack(Material.STAINED_GLASS_PANE, 0, (short) 5);
        ...

        return item;
    }
}
```

onCommand.java

```
package net.survivalz.chill;

import net.survivalz.chill.events.FrozenMenu;
import org.bukkit.Bukkit;
...

public class onCommand implements CommandExecutor { // sets this class to be the
place where all the commands can be found
    public ChillMain plugin; // these next few lines link back to the main plugin,
for referencing
    public onCommand (ChillMain plugin) {
        this.plugin = plugin;
    }

    public boolean onCommand(CommandSender sender, Command cmd, String commandlabel,
String[] args) { // starting to create commands
        String prefix = plugin.prefix; // setting up local variables of the prefix
and colours
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;
        if (commandlabel.equalsIgnoreCase("freeze") ||
commandlabel.equalsIgnoreCase("ss")) { // if the command is "/freeze" and if they
have the permission "chill.freeze", as seen below
            if (sender.hasPermission("chill.freeze")) {
                if (args.length == 0) { // if they dont have any arguments, display
help text
                    sender.sendMessage(prefix + "Please specify a " + sec + "player"
+ pri + ".");
                    return true;
                }
                if (args[0].equalsIgnoreCase("*") || args[0].equalsIgnoreCase("all"))
{ // if they try to freeze all and have the perm ,go through the security steps
                    if (!sender.hasPermission("chill.freeze.all")) {
                        sender.sendMessage(prefix + "You do " + sec + "not " + pri +
"have permission to perform this " + sec + "command" + pri + ".");
                        return true;
                    } else {
                        if (args.length == 1) {
                            sender.sendMessage(prefix + "This is a " + sec + "very "
+ pri + "serious command.");
                            sender.sendMessage(prefix + "This freezes " + sec + "all
" + pri + "online players.");
                            sender.sendMessage(prefix + "If you are okay with this,
type " + sec + "/freeze all confirm");
                        } else if (args.length < 1) {
                            if (args[1].equalsIgnoreCase("confirm")) {
                                for (Player player : Bukkit.getOnlinePlayers()) {
                                    if (plugin.frozen.contains(player)) {
                                        plugin.frozen.add(player.getUniqueId());
                                        sender.sendMessage(prefix + "Froze " + sec +
player.getName() + pri + ".");
                                    }
                                }
                                return true;
                            } else {
                                sender.sendMessage(prefix + "This is a " + sec +
"very " + pri + "serious command.");
                                sender.sendMessage(prefix + "This freezes " + sec +
"all " + pri + "online players.");
                                sender.sendMessage(prefix + "If you are okay with
this, type " + sec + "/freeze all confirm");
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

        return true;
    }
}
return true;
}
}

Entity targetEntity = Bukkit.getServer().getPlayer(args[0]); //
creates a temporary entity placeholder that will be what the player's first arg is
if (targetEntity instanceof Player) { // checks if the entity
selected is really a player, just in case
    Player target = (Player) targetEntity; sets the entity to be a
player
    if (target == null ||
!Bukkit.getOnlinePlayers().contains(target)) { if the player isnt real, throw error
        sender.sendMessage(prefix + sec + target.getName() + pri + "
does not " + sec + "exist" + pri + ". " + sec + "(" + args[0] + ")");
        return true;
    }
    if (target.isOp() || target.hasPermission("chill.override")) { //
if the target has override perms, don't freeze them for security measures
        sender.sendMessage(prefix + pri + "You cannot freeze " + sec
+ target.getName() + pri + ".");
        for (Player all : Bukkit.getOnlinePlayers()) {
            if (all.hasPermission("chill.notify")) {
                all.sendMessage(prefix + sec + sender.getName() + pri
+ " tried to freeze " + sec + target.getName() + pri + " but they had the permission
" + sec + "chill.override" + pri + ".");
            }
        }
        return true;
    } else {
        if (plugin.frozen.contains(target.getUniqueId())) { // if
selected player is already frozen, unfreeze and notify all staff
            plugin.frozen.remove(target.getUniqueId());
            for (Player all : Bukkit.getOnlinePlayers()) {
                if (all.hasPermission("chill.notify")) {
                    all.sendMessage(prefix + sec + sender.getName() +
pri + " just unfroze " + sec + target.getName() + pri + ". ");
                }
            }
            target.removePotionEffect(PotionEffectType.BLINDNESS);
            sender.sendMessage(prefix + sec + target.getName() + pri
+ " has been " + sec + "unfrozen" + pri + "!");
            target.sendMessage(prefix + "You have been " + sec +
"unfrozen" + pri + "!");
            target.closeInventory();
            return true;
        }
        plugin.frozen.add(target.getUniqueId()); // else add them to
the frozen list and notify staff
        sender.sendMessage(prefix + sec + target.getName() + pri + "
has been " + sec + "frozen" + pri + "!");
        for (Player all : Bukkit.getOnlinePlayers()) {
            if (all.hasPermission("chill.notify")) {
                all.sendMessage(prefix + sec + sender.getName() + pri
+ " just froze " + sec + target.getName() + pri + ". ");
            }
        }
        target.addPotionEffect(new
PotionEffect(PotionEffectType.BLINDNESS, Integer.MAX_VALUE, 9));
        plugin.blank(target);
        target.sendMessage(prefix + "You have been " + sec + "frozen"
+ pri + "!");
        target.sendMessage(Strings.joinTS1);
    }
}

```

```

        target.sendMessage(Strings.joinTS2);
        return true;
    }
}
} else { // tell the player if they dont have perms
    sender.sendMessage(prefix + "You do " + sec + "not " + pri + "have
permission to perform this " + sec + "command" + pri + ".");
    return true;
}
}

    if (commandlabel.equalsIgnoreCase("thaw") ||
commandlabel.equalsIgnoreCase("unfreeze")) { // unfreezes player selected, same logic
as above
    if (sender.hasPermission("chill.freeze")) {
        if (args.length == 0) {
            sender.sendMessage(prefix + "Please specify a " + sec + "player"
+ pri + ".");
            return true;
        } else {
            Entity targetEntity = Bukkit.getServer().getPlayer(args[0]);
            ...
        }
    } else {
        sender.sendMessage(prefix + "You do " + sec + "not " + pri + "have
permission to perform this " + sec + "command" + pri + ".");
        return true;
    }
}

    if (commandlabel.equalsIgnoreCase("panic")) { // lets famous people freeze
themselves as they are more likely to be targeted. cannot unfreeze, like the opposite
of /thaw
    Player player = (Player) sender;
    if (!sender.hasPermission("chill.panic")) {
        sender.sendMessage(prefix + "You do " + sec + "not " + pri + "have
permission to perform this " + sec + "command" + pri + ".");
        return true;
    } else {
        if (!plugin.frozen.contains(player.getUniqueId())) {
            plugin.frozen.add(player.getUniqueId());
            ...
        } else {
            sender.sendMessage(prefix + "You are already in " + sec + "/panic
mode" + pri + ".");
            return true;
        }
    }
}

    if (commandlabel.equalsIgnoreCase("frozen")) { // checks if the player is
frozen, same logic as /freeze
    if (!sender.hasPermission("chill.frozen")) {
        sender.sendMessage(prefix + "You do " + sec + "not " + pri + "have
permission to perform this " + sec + "command" + pri + ".");
        return true;
    } else {
        if (args.length == 0) {
            sender.sendMessage(prefix + "Please specify a " + sec + "player"
+ pri + ".");
            return true;
        } else if (args.length == 1) {
            Player target = Bukkit.getServer().getPlayer(args[0]);
            if (target == null) {

```

```

        sender.sendMessage(prefix + sec + args[0] + pri + " does not
" + sec + "exist " + pri + "or is not " + sec + "online" + pri + ".");
        return true;
    } else {
        if (plugin.frozen.contains(target.getUniqueId())) {
            sender.sendMessage(prefix + sec + target.getName() + pri
+ " is currently " + sec + "frozen" + pri + ".");
            return true;
        } else {
            sender.sendMessage(prefix + sec + target.getName() + pri
+ " is " + sec + "not " + pri + "currently " + sec + "frozen" + pri + ".");
            return true;
        }
    }
} else {
    sender.sendMessage(prefix + "Too little or few " + sec +
"arguments" + pri + ".");
    return true;
}
}
if (commandlabel.equalsIgnoreCase("chill")) { //outputs plugin information
    sender.sendMessage("");
    sender.sendMessage(prefix + sec + "/freeze <playerName> " + pri + "-
Freezes/Unfreezes a player, used as a toggle. " + sec + "chill.freeze");
    sender.sendMessage(prefix + sec + "/frozen <playerName> " + pri + "-
Checks if a player is frozen. " + sec + "chill.frozen");
    sender.sendMessage(prefix + sec + "/panic " + pri + "- Places you into
panic mode, whereby you cannot be unfrozen. " + sec + "chill.panic");
    sender.sendMessage("");
    sender.sendMessage(prefix + "Created by " + sec + "Natfan" + pri + ". " +
sec + "http://natfan.github.io/");
    sender.sendMessage(prefix + "For the " + sec + "Colossal Network" + pri +
". " + sec + "http://colossalmc.net/");
    sender.sendMessage("");
    return true;
}
return true;
}
}

```

Events

FrozenMenu.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.MenuItems;
import org.bukkit.Bukkit;
...

public class FrozenMenu {
    public static Inventory inv = Bukkit.createInventory(null, 8, ChatColor.GOLD +
"Frozen Menu"); // creates the menu, sets it to be 1 row (9 spaces wide), sets the
name of the menu and sets the owner to be null so that anyone can access

    static {
        inv.setItem(2, MenuItems.admit()); // sets the specific slots to be specific
items
        inv.setItem(4, MenuItems.info());
        inv.setItem(6, MenuItems.close());
    }
}
```


onBlockBreak.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
...
import org.bukkit.ChatColor;
...

public class onBlockBreak implements Listener { // tells the server that this listens
for events
    public ChillMain plugin;
    public onBlockBreak (ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler // required for listeners
    public void onBlockBreakEvent(BlockBreakEvent e) { // activates whenever a block
breaks
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;

        Player p = e.getPlayer(); // sets a local var to be the player that broke the
block
        if (plugin.frozen.contains(p.getUniqueId())) { // if they're frozen...
            e.setCancelled(true); // cancel the event
            plugin.blank(p);
            p.sendMessage(prefix + "You are currently " + sec + "frozen " + pri +
"and cannot " + sec + "break " + pri + "blocks!");
            p.sendMessage(Strings.joinTS1);
            p.sendMessage(Strings.joinTS2);
        }
    }
}
```

onBlockPlace.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
...
import org.bukkit.ChatColor;
...

public class onBlockPlace implements Listener {
    public ChillMain plugin;
    public onBlockPlace (ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler
    public void onBlockPlaceEvent(BlockPlaceEvent e) { // checks when a block is
placed
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;

        Player p = e.getPlayer();
        if (plugin.frozen.contains(p.getUniqueId())) { // cancels the event if player
is frozen
            e.setCancelled(true);
            plugin.blank(p);
            p.sendMessage(prefix + "You are currently " + sec + "frozen " + pri +
"and cannot " + sec + "place " + pri + "blocks!");
            p.sendMessage(Strings.joinTS1);
            p.sendMessage(Strings.joinTS2);
        }
    }
}
```

onCommandPreprocess.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
...
import org.bukkit.ChatColor;
...

public class onCommandPreprocess implements Listener {
    public ChillMain plugin;
    public onCommandPreprocess(ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler
    public void onCommandPreprocess(PlayerCommandPreprocessEvent e) { // fires
when a command is sent
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;

        Player p = e.getPlayer();
        if (plugin.frozen.contains(p.getUniqueId())) { // cancels the event if player
is frozen
            if (!e.getMessage().startsWith("/thaw " + p.getName()) ||
!e.getMessage().startsWith("/unfreeze " + p.getName())) {
                e.setCancelled(true);
                plugin.blank(p);
                p.sendMessage(prefix + "You are currently " + sec + "frozen " + pri +
"and cannot " + sec + "type " + pri + "commands!");
                p.sendMessage(Strings.joinTS1);
                p.sendMessage(Strings.joinTS2);
            }
        }
    }
}
```

onEnderpearlThrow

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
import org.bukkit.ChatColor;
...

public class onEnderpearlThrow implements Listener {
    public ChillMain plugin;

    public onEnderpearlThrow(ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler
    public void onEnderpearlThrowEvent(ProjectileLaunchEvent e) { // fires when any
projectile is thrown
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;

        Entity entity = e.getEntity();
        if (entity instanceof EnderPearl) { // checks if the projectile is an
enderpearl, an item which lets the player teleport
            Player player = (Player) ((EnderPearl) entity).getShooter(); // get the
person who threw the enderpearl
            if (plugin.frozen.contains(player.getUniqueId())) {
                player.sendMessage(prefix + sec + "You " + pri + "are currently " +
sec + "frozen " + pri + "and cannot " + sec + "use " + pri + "Ender Pearls!");
                e.setCancelled(true); // cancels if the player is frozen
            }
        }
    }
}
```

onFrozenGiveDamage.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
import org.bukkit.ChatColor;
...

public class onFrozenGiveDamage implements Listener {
    public ChillMain plugin;

    public onFrozenGiveDamage(ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler
    public void onFrozenGiveDamageEvent(EntityDamageByEntityEvent e) { // is fired
when an entity damages another
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;

        Entity damager = e.getDamager();
        if (damager instanceof Player) { // checks if the person who dealt the damage
was a player
            Player damagerPlayer = (Player) damager;
            if (plugin.frozen.contains(damagerPlayer.getUniqueId())) { // if the
player is frozen, dont let them deal damage
                damagerPlayer.sendMessage(prefix + sec + "You " + pri + "are
currently " + sec + "frozen " + pri + "and cannot " + sec + "give " + pri +
"damage!");
                e.setCancelled(true);
            }
        }
        if (damager instanceof Projectile) { // checks if the damager was a
projectile (arrow, egg, snowball etc)
            Projectile projectile = (Projectile) damager;
            if (projectile.getShooter() instanceof Player) {
                Player damagerShooter = (Player) projectile.getShooter();
                if (plugin.frozen.contains(damagerShooter.getUniqueId())) { // checks
if the player that fired the shot was frozen
                    damagerShooter.sendMessage(prefix + sec + "You " + pri + "are
currently " + sec + "frozen " + pri + "and cannot " + sec + "give " + pri +
"damage!");
                    e.setCancelled(true); // cancels the event
                }
            }
        }
        if (damager instanceof Potion) { // does the same for potions
            Projectile projectile = (Projectile) damager;
            ...
        }
    }
}
```

onInventoryClose.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
import org.bukkit.Bukkit;
...

public class onInventoryCloseEvent implements Listener {
    public ChillMain plugin;
    public onInventoryCloseEvent(ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler
    public void onInventoryClose(InventoryCloseEvent e) { // is fired whenever an
inventory is closed
        Player p = (Player) e.getPlayer();
        BukkitScheduler scheduler = Bukkit.getServer().getScheduler();
        scheduler.scheduleSyncRepeatingTask(plugin, () -> {
            if (plugin.frozen.contains(p.getUniqueId())) { // checks if the player
that closed the inventory was frozen every 3 seconds
                if (e.getInventory().getName().equals(FrozenMenu.inv.getName())) {
                    p.closeInventory(); // closes their current inventory and opens
up the menu
                    p.openInventory(FrozenMenu.inv);
                }
            }
        }, 60L, 60L);
    }
}
```

onPlayerDropItem.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
...
import org.bukkit.ChatColor;
...

public class onPlayerDropItem implements Listener {
    public ChillMain plugin;
    public onPlayerDropItem (ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler
    public void onPlayerDropItemEvent(PlayerDropItemEvent e) { // fires whenever a
player drops and item
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;

        Player p = e.getPlayer();
        if (plugin.frozen.contains(p.getUniqueId())) {
            e.setCancelled(true); // cancels the event if the player is frozen so
that they cannot give their stuff to an unfrozen player
            plugin.blank(p);
            p.sendMessage(prefix + "You are currently " + sec + "frozen " + pri +
"and cannot " + sec + "drop " + pri + "items!");
            p.sendMessage(Strings.joinTS1);
            p.sendMessage(Strings.joinTS2);
        }
    }
}
```

onPlayerInventoryClickItem.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
...
import org.bukkit.Bukkit;
...

public class onPlayerInventoryClickItem implements Listener {
    public ChillMain plugin;
    public onPlayerInventoryClickItem(ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler
    public void onPlayerClickItemEvent(InventoryClickEvent e) { // fires whenever a
player clicks in an inventory
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;

        HumanEntity humanEntity = e.getWhoClicked();
        Inventory inventory = e.getInventory();
        Player p = (Player) humanEntity;
        ItemStack clicked = e.getCurrentItem();
        InventoryType.SlotType slotType = e.getSlotType();

        if (plugin.frozen.contains(p.getUniqueId())) {
            if (inventory.getType() != null) {
                if (p.getOpenInventory().getTopInventory().getName() !=
FrozenMenu.inv.getName()) {
                    e.setCancelled(true); // stop players from interactive with
inventories that arent the menu
                    plugin.blank(p);
                    p.sendMessage(prefix + "You are currently " + sec + "frozen " +
pri + "and cannot " + sec + "move " + pri + "items!");
                    p.sendMessage(Strings.joinTS1);
                    p.sendMessage(Strings.joinTS2);
                } else {
                    if (slotType.equals(InventoryType.SlotType.CONTAINER)) {
                        if (clicked.isSimilar(MenuItems.admit())) { // tells staff to
ban the player if they admit
                            p.sendMessage("YOU CLICKED ADMIT");
                            e.setCancelled(true);
                            Bukkit.broadcastMessage(p.getName() + "admitted to
cheating, please issue /ban " + p.getName() + " Modified Client (Admitted)");
                        }
                        if (clicked.isSimilar(MenuItems.info())) {
                            p.sendMessage("YOU CLICKED INFO");
                            e.setCancelled(true);
                        }
                        if (clicked.isSimilar(MenuItems.close())) {
                            p.sendMessage("YOU CLICKED CLOSE");
                            e.setCancelled(true);
                            if (p.hasPermission("chill.override")) { // only lets
them close the inventory if they have the perm "chill.override"
                                p.closeInventory();
                            } else {
                                p.sendMessage("You cannot close this menu.");
                            }
                        }
                    }
                }
            }
        }
    }
}
```


onPlayerMove.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
...
import org.bukkit.ChatColor;
...

public class onPlayerMove implements Listener {
    public ChillMain plugin;
    public onPlayerMove (ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler
    public void onPlayerMoveEvent(PlayerMoveEvent e) { // fires whenever a player
moves
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;

        Player p = e.getPlayer();
        if (plugin.frozen.contains(p.getUniqueId())) {
            Location location = p.getLocation(); // creates a new location and sets
it to the player's previous position
            location.setY(p.getWorld().getHighestBlockYAt(p.getLocation())); //
places the on the highest block so that they are not in midair
            location.setPitch(e.getTo().getPitch());
            location.setYaw(e.getTo().getYaw());
            p.teleport(location); // sets their location to their old location
            p.setVelocity(new Vector().zero()); // sets their velocity to 0
            plugin.blank(p);
            p.sendMessage(prefix + "You are currently " + sec + "frozen " + pri +
"and cannot " + sec + "move" + pri + "!");
            p.sendMessage(Strings.joinTS1);
            p.sendMessage(Strings.joinTS2);
        }
    }
}
```

onPlayerPickupItem.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
...
import org.bukkit.ChatColor;
...

public class onPlayerPickupItem implements Listener {
    public ChillMain plugin;
    public onPlayerPickupItem(ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler
    public void onPlayerPickupItemEvent(PlayerPickupItemEvent e) { // fires whenever
a player picks up an item
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;

        Player p = e.getPlayer();
        if (plugin.frozen.contains(p.getUniqueId())) {
            e.setCancelled(true); // prevents them from picking up items
            plugin.blank(p);
            p.sendMessage(prefix + "You are currently " + sec + "frozen " + pri +
"and cannot " + sec + "pick up " + pri + "items!");
            p.sendMessage(Strings.joinTS1);
            p.sendMessage(Strings.joinTS2);
        }
    }
}
```

onPlayerQuit.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
...
import org.bukkit.Bukkit;
...

public class onPlayerQuit implements Listener {
    public ChillMain plugin;
    public onPlayerQuit(ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler
    public void onPlayerQuitEvent(PlayerQuitEvent e) { // fires when a player leaves
the server
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;
        Player p = e.getPlayer();
        if (plugin.frozen.contains(p.getUniqueId())) {
            Bukkit.broadcast(prefix + sec + p.getName() + pri + " has " + sec +
"disconnected " + pri + "while " + sec + "frozen" + pri + ".", "chill.notify"); //
notifies all staff that a player disconnected while frozen
        }
    }
}
```

onPlayerTakeDamage.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
...
import org.bukkit.ChatColor;
...

public class onPlayerTakeDamage implements Listener {
    public ChillMain plugin;
    public onPlayerTakeDamage (ChillMain plugin) {
        plugin = plugin;
    }

    @EventHandler
    public void onPlayerTakeDamageEvent(EntityDamageEvent e) { // fires whenever an
entity takes damage
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;

        if (e.getEntity() instanceof Player) {
            Player p = (Player) e.getEntity();
            if (plugin.frozen.contains(p.getUniqueId())) {
                e.setCancelled(true); // prevents frozen players from taking damage
                plugin.blank(p);
                p.sendMessage(prefix + "You are currently " + sec + "frozen " + pri +
"and cannot " + sec + "take " + pri + "damage!");
                e.getCause();
                p.sendMessage(Strings.joinTS1);
                p.sendMessage(Strings.joinTS2);
            }
        }
    }
}
```

onPlayerTeleport.java

```
package net.survivalz.chill.events;

import net.survivalz.chill.ChillMain;
import org.bukkit.ChatColor;
...

public class onPlayerTeleport implements Listener {
    public ChillMain plugin;

    public onPlayerTeleport(ChillMain plugin) {
        this.plugin = plugin;
    }

    @EventHandler
    public void onPlayerTeleportEvent(PlayerTeleportEvent e) { // fires whenever a
player telteports
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;

        Player player = e.getPlayer();
        if (plugin.frozen.contains(player.getUniqueId())) {
            if (!(e.getCause() == PlayerTeleportEvent.TeleportCause.PLUGIN) || !
(e.getCause() == PlayerTeleportEvent.TeleportCause.COMMAND)) { // prevents them from
teleporting via enderpearls but allows teleportation via plugins or commands
                player.sendMessage(prefix + sec + "You " + pri + "are currently " +
sec + "frozen " + pri + "and cannot " + sec + "use " + pri + "Ender Pearls!");
                e.setCancelled(true);
            }
        }
    }
}
```

Program Screenshots

The following images are proof that my program works.

Before Being Frozen: Player's Perspective



After Being Frozen: Player's Perspective



After Being Frozen: Administrator's Perspective



Documentation

This program is a plugin for piece of server software called **Spigot** that is used to make the generation of add-ons to the video game **Minecraft**. The plugin in question is called **Chill**, a tool which allows Administrators to freeze possible cheaters to prevent them from causing any more damage to legitimate players. After this point they can a voice chat service such as Teamspeak3 or Discord so that they can communicate with the staff member that froze them, possibly share their screen and, after rigorous checks, may be unfrozen. If it turns out that the player was cheating, they will be punished accordingly. The input of the program is when the Administrator freezes a player, the output is when the user tries to perform an action that is prohibited while they are frozen, along with notifying staff members when a person who should not be frozen is attempted to be frozen, and the process is the blocking of these actions. There are a few different tools that I shall use within this program, and I shall list them and describe why they are important.

PV: Testing the Event Driven Program

In the following segment I shall create a table which has a list of all of the tests, their expected outcomes and an image to show for it, which can be seen in the table below, title with the corresponding number that is in the table.

Test	Expected Result	Image Number
<code>/freeze</code> once	Will freeze the player.	1
<code>/freeze</code> twice	Will unfreeze the player.	2
<code>/thaw</code>	Will unfreeze the player if they are frozen, otherwise do nothing.	3
<code>/frozen</code>	Will output if the player is frozen or not.	4

Image #1:



Image #2:



Image #3:





Image #4:



MIII: Testing with Corrective Action

In the following segment I shall create a table which has a list of all of the tests as shown before, with a few new ones as well. I shall say what result they should have, what result they actually had and the fix that was created, if required

Test	Expected Result	Actual Result	Fix
<code>/freeze</code> once	Will freeze the player.	Freezes the player.	None.
<code>/freeze</code> twice	Will unfreeze the player.	Unfreezes the player.	None.
<code>/thaw</code>	Will unfreeze the player if they are frozen, otherwise do nothing.	Unfreezes the player if needed, else outputs error.	None.
<code>/frozen</code>	Will output if the player is frozen or not.	Outputs if the player is frozen if they are, else outputs that they are not.	None.
Player moving when thawed.	Players can move.	Players could move.	None.
Player moving when frozen.	Players cannot move.	Players could not move.	None.
Player placing blocks when thawed.	Players can place blocks.	Players could place blocks.	None.
Player placing blocks when frozen.	Players cannot place blocks.	Players could not place blocks.	None.
Player breaking blocks when thawed.	Players can break blocks.	Players could break blocks.	None.
Player breaking blocks when frozen.	Players cannot break blocks.	Players could not break blocks.	None.
Player picking up items when thawed.	Players can pick up items.	Players could pick up items.	None.
Player picking up items when frozen.	Players cannot pick up items.	Players could not pick up items.	None.
Player dropping items when thawed.	Players can drop items.	Players could drop items.	None.

Player dropping items when frozen.	Players cannot drop items.	Players could not drop items.	None.
Player teleporting when thawed.	Players can teleport.	Players could teleport.	None.
Player teleporting when frozen.	Players cannot teleport.	Players could not teleport.	None.
Player executing commands when thawed.	Players can execute commands.	Players could execute commands.	None.
Player executing commands when frozen.	Players cannot execute commands.	Players could not execute commands.	None.
Player throw enderpearls when thawed.	Players can throw enderpearls.	Players could throw enderpearls.	None.
Player throw enderpearls when frozen.	Players cannot throw enderpearls.	Players could not throw enderpearls.	None.
<code>/freeze</code> permissions.	Players need the right permission to execute <code>/freeze</code> .	Players could only execute <code>/freeze</code> if they had the correct permissions.	None.
<code>/freeze</code> override permissions.	Players who have the permission <code>chill.override</code> cannot get <code>/freeze</code> .	Players could <code>/freeze</code> those who did not have the permission.	None.
<code>/thaw</code> permissions.	Players need the right permission to execute <code>/thaw</code> .	Players could only execute <code>/thaw</code> if they had the correct permissions.	None.
<code>/frozen</code> permissions.	Players need the right permission to execute <code>/frozen</code> .	Players could only execute <code>/frozen</code> if they had the correct permissions.	None.

DII: Evaluation

Evaluation Form

The following is an evaluation form that was filled out by one of my friends and users of the plugin, Brendan Taylor.

Does the plugin freeze player specified, unless they have the specific permission?

Yes

Does the plugin keep the player frozen even if they log out and log back in again, or change their username?

Yes, it does both.

Are there any bugs with the program?

One: If the player with the username "All" is connected then they cannot get frozen.

Are there any changes that you would request in version 2.0 if this program?

The ability to change the colours and text of the messages.

Did you enjoy this plugin? Did it satisfy all requirements?

Yes and yes. This was worth my time to download and install, it works almost flawlessly.

Changes based on Evaluation

The following is the `onCommand.java` class that I changed to fix the bug that was reported. I shall only show the code that was changed, along with the header of the program and annotations of what was changed.

```
package net.survivalz.chill;

import net.survivalz.chill.events.FrozenMenu;
import org.bukkit.Bukkit;
...

public class onCommand implements CommandExecutor {
    public ChillMain plugin;
    public onCommand (ChillMain plugin) {
        this.plugin = plugin;
    }

    public boolean onCommand(CommandSender sender, Command cmd, String commandlabel,
String[] args) { // starting to create commands
        String prefix = plugin.prefix;
        ChatColor pri = plugin.pri;
        ChatColor sec = plugin.sec;
        if (commandlabel.equalsIgnoreCase("freeze") ||
commandlabel.equalsIgnoreCase("ss")) {
            if (sender.hasPermission("chill.freeze")) {
                if (args.length == 0) {
                    sender.sendMessage(prefix + "Please specify a " + sec + "player"
+ pri + ".");
                    return true;
                }
                if (args[0].equalsIgnoreCase("*")) { // I removed the "all" argument
from the command, and as usernames cannot contain symbols like "*", everything is
safe.
                    if (!sender.hasPermission("chill.freeze.all")) {
                        sender.sendMessage(prefix + "You do " + sec + "not " + pri +
"have permission to perform this " + sec + "command" + pri + ".");
                        return true;
                    } else {
                        sender.sendMessage(prefix + "This is in " + sec +
```

```

"development" + pri + ".");
        if (args.length == 1) {
            sender.sendMessage(prefix + "This is a " + sec + "very "
+ pri + "serious command.");
            sender.sendMessage(prefix + "This freezes " + sec + "all
" + pri + "online players.");
            sender.sendMessage(prefix + "If you are okay with this,
type " + sec + "/freeze all confirm");
        } else if (args.length < 1) {
            if (args[1].equalsIgnoreCase("confirm")) {
                for (Player player : Bukkit.getOnlinePlayers()) {
                    if (plugin.frozen.contains(player)) {
                        plugin.frozen.add(player.getUniqueId());
                        sender.sendMessage(prefix + "Froze " + sec +
player.getName() + pri + ".");
                    }
                }
                return true;
            } else {
                sender.sendMessage(prefix + "This is a " + sec +
"very " + pri + "serious command.");
                sender.sendMessage(prefix + "This freezes " + sec +
"all " + pri + "online players.");
                sender.sendMessage(prefix + "If you are okay with
this, type " + sec + "/freeze all confirm");
                return true;
            }
        }
        return true;
    }
}

```



As stated in the annotations, this is better as it allows the Administrator to have complete control over their server so that they can freeze anyone that they want, resulting in a cleaner community as no regular user can get around the `/freeze` command.

