Virtual Enaction

"This file will tell you how to make an advance use of the virtualenaction project"

http://team.inria.fr/mnemosyne

fpouget

18 september 2013
# Table des matières

1 Basic utilisation of the project ........................................ 2
   1.1 Basic presentation of Minecraft .................................. 2
   1.2 Basic presentation of Virtualenaction ............................ 2

2 The virtualenaction server's command .................................. 4
   2.1 The basics about the server ......................................... 4
   2.2 The simple commands ............................................... 5
      2.2.1 veavailable .................................................. 5
      2.2.2 vebots ....................................................... 6
      2.2.3 vefreeze ..................................................... 6
      2.2.4 veunfreeze .................................................. 6
      2.2.5 veloadconfig ................................................. 6
      2.2.6 vepause ...................................................... 6
      2.2.7 verelease ..................................................... 6
      2.2.8 verelease ..................................................... 7
      2.2.9 vesetspawn ................................................... 7
      2.2.10 verespawn .................................................. 7
      2.2.11 verunning ................................................... 7
      2.2.12 vesleeping ................................................. 7
      2.2.13 vewaiting ................................................... 7
   2.3 The world saving related commands ................................ 7
      2.3.1 vesave ....................................................... 7
      2.3.2 veload ...................................................... 8
      2.3.3 veclean ..................................................... 8
      2.3.4 veworlds ................................................... 8
   2.4 The commands which can only be run by a player .................. 8
      2.4.1 veconnect .................................................. 8
      2.4.2 vedisconnect ............................................... 8
   2.5 The commands vetcp ................................................. 8

3 The configurations files .................................................. 10
   3.1 Configuring the bot ................................................ 10
      3.1.1 Exhaustion or energy related variables ...................... 10
      3.1.2 Gauges related variables ................................... 11
      3.1.3 other variables ............................................. 11
   3.2 The VirtualenactionPlugin config.yaml ............................ 12
      3.2.1 The first part : connection ................................ 12
      3.2.2 The second part : damage multiplier ......................... 12
      3.2.3 The third part : teleportation ................................ 13
   3.3 The VirtualenactionPlugin foods.yaml .............................. 13
   3.4 The TactilePlugin TactileValues.yml .............................. 14
   3.5 The gauges configuration configGauges.yaml ....................... 14
Chapitre 1

Basic utilisation of the project

1.1 Basic presentation of Minecraft

Minecraft is a game made by the studio mojang.
It is a game about construction and survival, alone or with other players. Our project use Minecraft to provide a believable survival environment for an IA.
Our project use a custom server named Bukkit.
The version of Minecraft we use is 1.4.5. Our project is only compatible with this version, updating Minecraft and the project will require a lot of work due to the fact that several methods change between version and the fact that the code is obfuscated.
To know more about modding the game or the server, or make plugin for the server you will have to search the internet for tutorials. The community around Minecraft is very large and you should find what you want.

However here are some links to usefull website about Minecraft :
The Mojang official website : https://mojang.com/
The Minecraft official website : https://minecraft.net/
How to make mods ( one of many tutorials) : http://www.minecrafthost.net/topic/96862-creating-mods-modding-tutorials-21511/
MCP downloading website : http://mcp.ocean-labs.de/index.php/MCP_Releases
FORGE downloading website : http://www.minecraftforge.net/forum/
Bukkit Documentation for the server’s plugin : http://jd.bukkit.org/rb/doxygen/
Bukkit’s Wiki, contain tutorials about installing and coding : http://wiki.bukkit.org/Main_Page
The Bukkit main website, Presentation of available plugins and help forums : http://bukkit.org/
MCPC website (MineCraftPortCentral) : http://www.mcportcentral.co.za/

1.2 Basic presentation of Virtualenaction

The Virtualenaction project is about creating a natural and flexible environment that allow the integration of a bot simulating the human brain ; and owning some internal characteristics ( health, hunger, thirst ,etc ...) that influence its decisions and behavior. This project is part of a bigger project called mnemosyne wich is under the authority of Bordeaux Inria.

For a more detailed presentation of our project you can check our prezi presentation.

Here are some links that can be usefull to you about our project :
The project’s website : http://virtualenaction.gforge.inria.fr/
The mnemosyne website page : https://team.inria.fr/mnemosyne/
The prezi: http://prezi.com/ino7kazuszo7/virtualenaction-project/
Chapitre 2

The virtualenaction server’s command

This chapter will explain what the commands we made available in our plugin do and how to use them.

2.1 The basics about the server

Once you launched the server it should look like this:

![Figure 2.1 – A server picture](image)

We will only give explanation about our commands, the one that begin with 've'. The other commands are Minecraft commands, and if you have problems with them you should search help by yourself.

The only command that is not from Minecraft or virtualenaction is tl, wich come from a plugin and allow to lock the time in the server to a certain moment of the day.

Here is the link to this plugin: [http://dev.bukkit.org/bukkit-plugins/timelock/](http://dev.bukkit.org/bukkit-plugins/timelock/)
You can hit 'help' to see all the commands available to you and a description. Also yan can do 'help aCommand' to see the description of a command and the usage:

2.2 The simple commands

2.2.1 veavailable

This command show all the players that are ready to host (but not the one that are connected to a bot). All you have to do is enter 'veavailable' in the server and you will a list of player printed in the terminal. Just for this virtualenaction command we give a picture, not for the others at it is basically the same.

Description: List players ready to host bot.
Usage: /veavailable
2.2.2 vebots

This command is really similar to the previous one. It give a list of player that are currently controled by a bot.

Description : List currently active bots.
Usage : /vebots

2.2.3 vefreeze

This command will enable the freeze of the world ( nothing will move as long as the bots are all computing, but will when they need to act) if the freeze was not already on. It is usefull for the bot that have an IA that take a long time to think. Also allow a player or the server to do vepause.

Description : pause while bots are inactive.
Usage : /vefreeze

2.2.4 veunfreeze

This command disable the freeze. All the component of the world will act normally, even if all the bots are thinking ( they can take domage when thinking). You can’t do vepause if the freeze is not enabled.

Description : cancel freeze.
Usage : /veunfreeze

2.2.5 veloadconfig

This will reload the configuration file of a bot. This file is located here :
release/minecraft/minecraft_server.v145/plugins/VirtualEnactionPlugin/yourLogin/config.yaml
If you want more informations about this configuration file go to our website and check the page about configuration files.

Description : load a bot config.
Usage : /veloadconfig <player>

2.2.6 vepause

This command will pause everything if freeze is enabled. So even if the bot want to act they can’t. Only a player in creative mode can move if pause is enabled.

Description : cancel event all cancellable event.
Usage : /vepause

2.2.7 verelease

This command cancel pause. Making every player able to move when the bots are doing something.

Description : cancel pause.
Usage : /verelease
2.2.8 verelease

This command delete all files that the plugin created. In particular all the files concerning the players.

Description : erase all datas created and stored by the plugin.
Usage : /vereset

2.2.9 vesetspawn

This command define for a bot a custom place to respawn.

Description : Set the custom respawn point for the bot.
Usage : /vesetspawn <player>

2.2.10 verespawn

This command make a bot respawn to the location you previously save with vesetspawn. If no setspawn was made, it will make the bot spawn to his default location ( not the one of the player he is controlling ).

Description : respawn a bot with values in its config file.
Usage : /verespawn <player>

2.2.11 verunning

This command give you a list of all the bot currently running ( all the player controlled by a bot).

Description : List currently running bots.
Usage : /verunning

2.2.12 vesleeping

This command give you a list of all bot that are sleeping ( the IA is thinking and not doing anything ).

Description : List currently sleeping bots.
Usage : /vesleeping

2.2.13 vewaiting

This command give a list of all bot that are waiting for a player to be available for control . The server will connect them to a player if both the player and the bot have the same login and if the player is not already controlled by a bot.

Description : List of connected bot waiting for player to host them.
Usage : /vewaiting

2.3 The world saving related commands

2.3.1 vesave

This command save the a world and the position of the bot inside it. So it can be loaded to retrieve a bot environment.

Description : Save a Bot relative world.
Usage : /vesave <bot> <src\_name> <dst\_name>
2.3.2 veload

This command load a world you previously saved. The bot will be teleport to where he was when you saved the world.

Description : load a Bot relative world.
Usage : /veload <bot> <src\_name> <dst\_name>

2.3.3 veclean

This command delete all the world you previously saved.

Description : Erase unused worlds and bot saves.
Usage : /veclean

2.3.4 veworlds

This command give all the differents world that are loaded by the server.

Description : List worlds loaded by the bots and all bukkit worlds.
Usage : /veworlds

2.4 The commands wich can only be run by a player

These commands can only be called by a player. Check the option menu of Minecraft if you don’t know how to execute command as a player.

2.4.1 veconnect

This command make the player open a socket to send pictures to the server. It also tell the server that this player is ready to host a bot.

By default this command is automatically sent by the player ( with our modded Minecraft ).

Description : Prepare Minecraft Client to host a bot.
Usage : /veconnect

2.4.2 vedisconnect

This command make the server know that the player do not want to host a bot. It will remove the player of the list of player ready to host a bot if a bot was not already controlling it, and it will also remove it as well as deconnecting him from the bot if a bot was controlling it.

Description : stop hosting a bot or remove yourself from the host list.
Usage : /vedisconnect

2.5 The commands vetp

This command is special as it is dependent to a configuration file.

The configuration file is located here :

release/minecraft/minecraft_server_v145/plugins/VirtualEnactionPlugin/config.yaml
This command teleport a player into a previously registered location. Each location have a name in the configuration file.

For example you can do `vetp mnemosyne bigradius`. This will teleport the player to the location with the coordinates written in the configuration file.

To add location all you have to do is add lines in the configuration file like the one already existing.

For example if you want that `vetp mnemosyne onehundred` teleport the player mnemosyne to the position (100,100,100) you will have to add in the file:

```
onehundred:
  z: 100
  y: 100
  x: 100
```

Description : tp player to a specific room.
Usage : `/vetp <player> <room>`
Chapitre 3

The configurations files

This chapter will explain how to change the different configuration files of our project so you can personalize your utilization.

3.1 Configuring the bot

In this section the configuration file of the bot will be explained.

This file is located here:
release/minecraft/minecraft_server_v145/plugins/VirtualEnactionPlugin/yourBot/config.yaml

3.1.1 Exhaustion or energy related variables

Here we will describe the variables that define the exhaustion for each bot action. The value that are here are the default values.

- The exhaustion that is added each time the bot attacks: attackExhaustion : 0.02,
- The exhaustion that is added each time the bot climbs Blocks: climbExhaustion : 0.08
- The exhaustion that is added each time the bot attacks: defaultExhaustion : 4.0E-5
- The exhaustion that is added each time the bot drops a stack of item or blocks: dropExhaustion : 0.005
- The exhaustion that is added each time the bot moves his head: headExhaustion : 2.0E-5,
- The exhaustion that is added each time the bot ingests an item (drink or eat): ingestExhaustion : 0.0025
- The exhaustion that is added each time the bot shifts item in his inventory: inventoryShiftExhaustion : 0.0025,
- The exhaustion that is added each time the bot jumps: jumpExhaustion : 0.02
- The exhaustion that is added each time the bot rotates his body: rotateExhaustion : 2.0E-5
- The exhaustion that is added each time the bot switches item in his 'hand' inventory: switchExhaustion : 0.0025,
- The exhaustion that is added each time the bot is throwing a stack of item or Blocks: \text{throwExhaustion} : 0.01, \\
- The exhaustion that is added each time the bot touches something: \text{touchExhaustion} : 2.0E-5, \\
- The exhaustion that is added each time the bot moves (translate): \text{translateExhaustion} : 4.0E-4, \\

- The coefficient the energy replenish when the player rests (in the exponent formula \text{a}\times x^k \text{ it is } x, where \text{k} is the number of time you spend resting, it means the more you wait the more you gain energy): \text{energyRegainCoefficient} : 0.001, \\
- The factor the energy replenish when the player rests (in the exponent formula \text{a}\times x^k \text{ it is } a): \text{energyRegainFactor} : 1.1

3.1.2 Gauges related variables

Those values are used on bot reset, for instance if you hit the respawn button in the GUI then the bot's gauges values are set to this ones.

- The energy the bot have when he connects fo the first time (1 is the max): \text{energy} : 1.0, \\
- The food the bot have when he connects fo the first time (1 is the max): \text{food} : 1.0 \\
- The health the bot have when he connects fo the first time (1 is the max): \text{health} : 1.0, \\
- The oxygen the bot have when he connects fo the first time (1 is the max): \text{oxygen} : 1.0, \\
- The water the bot have when he connects fo the first time (1 is the max): \text{water} : 1.0

3.1.3 other variables

- True if the bot must receive the images of what the player sees, else false: \text{forwardScreen} : true, \\
- The IP adress of the machine the bot is in, initially null: \text{inetAddress} : null, \\
- The login of the bot: \text{login} : mnemosyne, \\
- The range (in Blocks) the bot receives data about blocks: \text{nearByRange} : 4.0, \\
- A multiplier of how many damages the bot do to creatures or blocks: \text{strength} : 1.0, \\
- The angle in which the bot receives data about blocks: \text{viewAngle} : 100.0, \\
- The range in which the player receives data about block, limited by the viewAngle: \text{viewRange} : 3.0.0, \\
- The name of the world the bot will respawn in: \text{world} : world, \\
- The x respawn position of the bot: \text{x} : -250.0, \\
- The y respawn position of the bot: \text{y} : 1.0,
- The z respawn position of the bot: z: 275.0
- The pitch of the bot (define orientation with yaw at respawn): pitch: 0.0,
- The yaw of the bot (define orientation with pitch at respawn): yaw: 0.0,
- The time since the last respawn of the bot: time: 0,
- The number of time the bot did respawn: respawns: 1,

3.2 The VirtualEnactionPlugin config.yaml

This section explains the configuration file of the VirtualEnaction Plugin.

This file is located here:
release/minecraft/minecraft_server_v145/plugins/VirtualEnactionPlugin/config.yaml

This file is formed of different parts. The first one allows you to change the behavior of the plugin concerning its connection with the bot and the Minecraft client.

The second part allows to configure the damage multiplier of some items in Minecraft (for the bot only).

The third part is about the teleportation to some location.

3.2.1 The first part: connection
- The port the bot MUST connect on the server: Bots port: 25561
- The port the server must await a connection from the player so he can send pictures (DO NOT change it unless you change the Minecraft client): Players port: 25562
- The time the server waits for a response from the bot when it checks if it is alive: timeout: 2000
- True if the server must freeze the world by default (equivalent of launching vefreeze); else false: allowFreeze: true
- Between two freeze, the minimum time that the server must wait in milliseconds (so the world had to evolve): freezeTimeRange: 1000
- The time in milliseconds it takes for a block to return to undamaged state after the last hit by a bot: blockHealDelay: 10000
- The number of steps it takes to return to undamaged block after the last hit (1 means the bot will return to full health immediately after blockHealDelay): blockHealSteps: 1

3.2.2 The second part: damage multiplier

Each object which is in this part makes damages to blocks or entities when the bot attack.

You can change the multiplier for each object (in comparison with bare hands).
By default the ironpickaxe does two time more damage:
ironpickaxe : 2.0
If you want it to do 4 time more damages all you have to do is change the line to:
ironpickaxe : 4.0

You cannot add object here, if you want to make other objects to have a damage multiplier you will have
to edit the code.

3.2.3 The third part: teleportation

This part is linked with the vetp command which teleport a player into a previously registered location.
Each location have a name in the configuration file.

For example you can do vetp mnemosyne bigradius. This will teleport the player to the location with
the coordinates written in the configuration file:

bigradius:
  z: 275
  y: 2
  x: -250

To add location all you have to do is add lines in the configuration file like the one already existing.

For example if you want that vetp mnemosyne onehundred teleport the player mnemosyne to the position (100,100,100) you will have to add in the file:

onehundred:
  z: 100
  y: 100
  x: 100

3.3 The VirtualEnactionPlugin foods.yaml

This section explains the foods.yaml configuration file of the VirtualEnaction Plugin.

This file is located here:
release/minecraft/minecraft_server.v145/plugins/VirtualEnactionPlugin/foods.yaml

In this file is a list of item that the player can ingest (eat or drink). Each one have a value associated
to each gauge. You can change this value so the player will earn points in this particular gauge when he
consumes the item.

For example by default the bread gives 5 points (on a maximum of 20) in the food gauge but nothing else:

bread:
  feed: 5
  water: 0
  energy: 0
  heal: 0

If you also want the bot to earn 2 water when he consumes it, you have to change it that way:
3.4 The TactilePlugin TactileValues.yml

This section explains the TactileValues.yml configuration file of the Tactile Plugin.

This file is located here:
release/minecraft/minecraft_server_v145/plugins/TactilePlugin/TactileValues.yaml

This file is composed of two part:
The first one is about the hardness of blocks and the second about their roughness.

Each block in our project have a roughness and an hardness, they are properties we added to the Minecraft blocks. They both work the same way except they mean something different.

Only the first part will be described as the second is almost identical.

The first part is a list of association of numbers, the one on the left is the block id and the one on the right is the hardness the block have.

There is a default value defined at the beginning that is the hardness value for all the blocks that are not in the list. You can change the hardness value of a block or add new blocks with a custom hardness.

For example:

1 : 0.8

mean the the block with the ID 1 (Stone) have an hardness of 0.8 (1 is the max).

3.5 The gauges configuration configGauges.yaml

This section explain the configGauges.yaml configuration file of the server.

This file is located here:
release/minecraft/minecraft_server_v145/configGauges.yaml

/*
 * boolean saying if we should print debug
 */

- public static boolean debug = false;

/*
 *The max Value of the differents gauges : they are huge for a need of continuity
 */

- public static int maxWaterLevelG = Integer.MAX_VALUE;
- public static int maxEnergyLevelG = Integer.MAX_VALUE;
- public static int maxFoodLevelG = Integer.MAX_VALUE;
public static int maxHealthLevel = Integer.MAX_VALUE;

/*
 * float saying by how much we must divide the damages the player is receiving
 * put one to change nothing, do not put 0 or negative value
 */
public static float damageDivider = 1;

/*
 * float saying by how much we multiply the damages the player receives
 * The initial value is chosen to be coherent with an not modded minecraft
 */
public static float damageMultiplier = ((float)Integer.MAX_VALUE)/20.0F;

/*
 * boolean saying if the decline of the gauges should be exponential or linear
 */
public static boolean decExpFood = true;
public static boolean decExpWater = true;
public static boolean decExpEnergy = true;
public static boolean decExpHealth = true;

/*
 * The step value for the health exponential decrease
 * It is recommended to let this value (20)
 */
public static int pasHealth = 20;

/*
 * The dividing value for the others gauges in the exponential decrease
 * Each time points in the gauge must be lost, the gauge will be divided by the value here
 * example : newValue = oldValue/foodDivider;
 */
public static float foodDivider = 1.1F;
public static float waterDivider = 1.1F;
public static float energyDivider = 1.1F;

/*
 * The speed with which the gauges (except health) will decrease
 * The smaller the value, the higher the decline speed
 */
public static float vitDecFood = 1.0F;
public static float vitDecWater = 0.5F;
public static float vitDecEnergy = 0.5F;

/*
 * The speed at which the player earn water when he is wet (in water or under rain)
 * The smaller the value, the higher the increase speed
 */
public static int vitIncWater = 5;

/**
 * The speed at which the player earn energy when he is sneaking
 * The smaller the value, the higher the increase speed
 * only for players, the bot have to use the rest method
 */

public static int vitIncEnergy= 1;