# erykdocs Documentation

*Release 0.0.1*

**Eryk**

**Apr 16, 2019**

# Contents

**1 Farmingmod** **3**

1.1 Guides . . . . . . . . . . 3

1.2 References . . . . . . . . . . 15

**2 Craftingmod** **17**

2.1 Guides . . . . . . . . . . 17

2.2 References . . . . . . . . . . 42

**3 Fishingmod** **45**

3.1 Guides . . . . . . . . . . 45

3.2 References . . . . . . . . . . 46

This documentation page, includes references and guides for my scriptfodder addons. This is for users, which want to search for code references or guides to add new content and customize the bought addon.

# CHAPTER 1

# Farmingmod

Welcome to the wiki site for farmingmod.

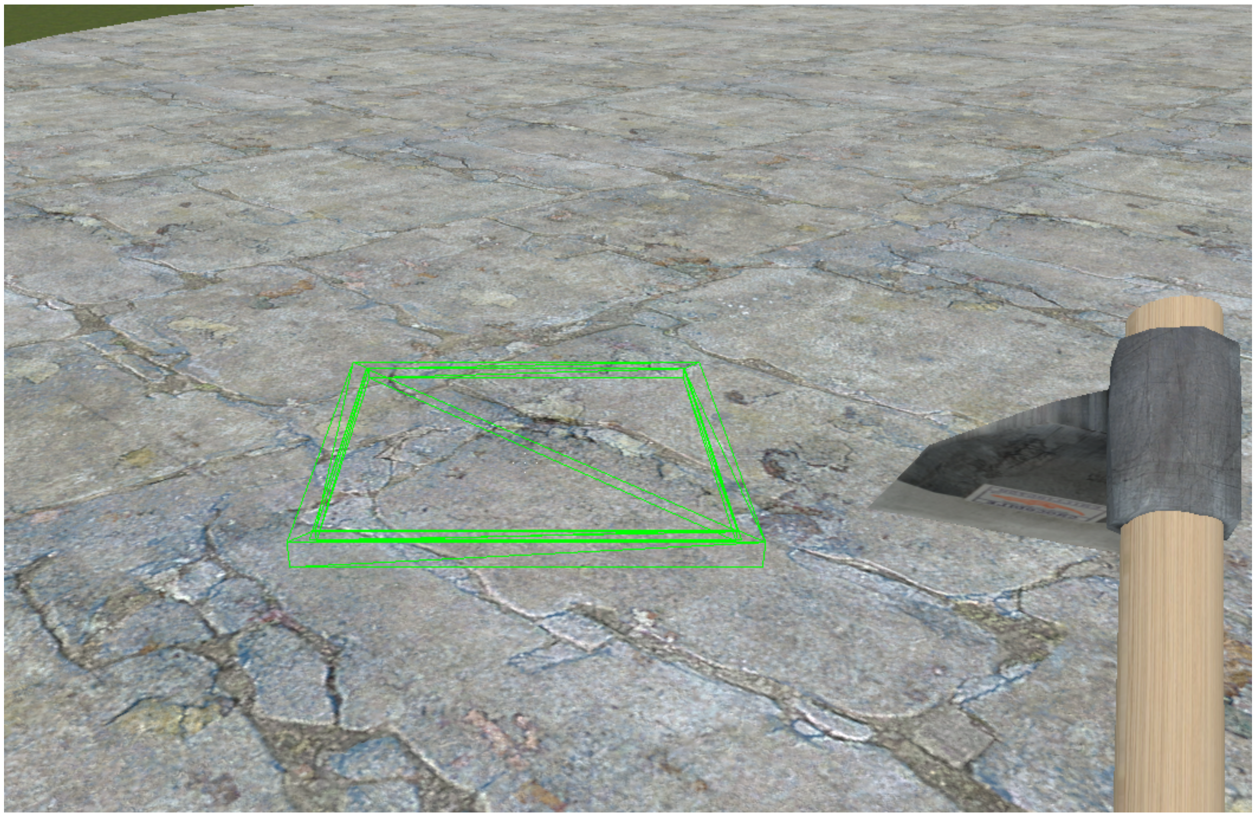
## 1.1 Guides

### 1.1.1 Getting started

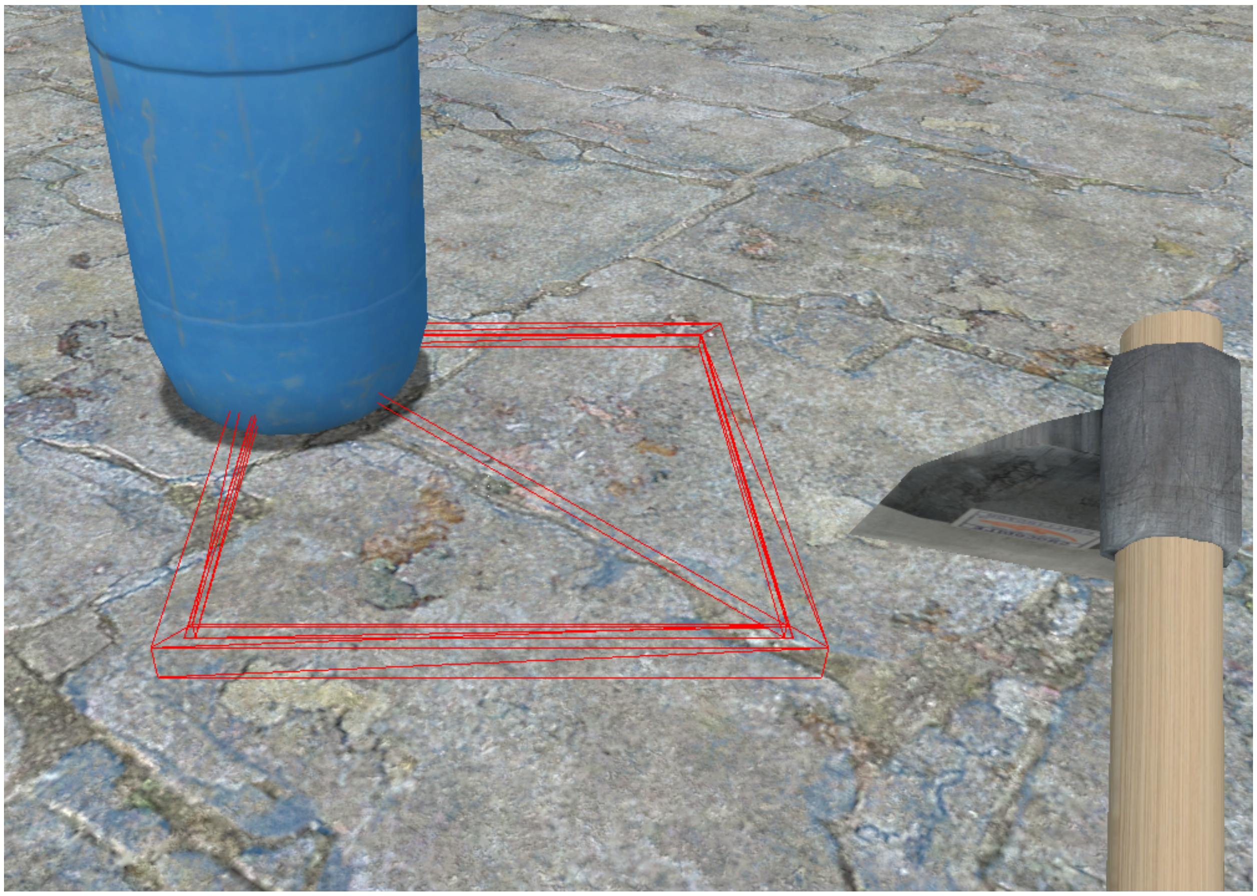
#### Weapon Hoe & Watercan

The weapon hoe has following controls: Right click: Delete crop plot Left click: Create crop plot ALT or ALT + Gr opens up the farmers inventory

There is only one type of crop plot. The crop plot can be build by the hoe_weapon, which normally can be achieved with the farmer job. (There is a option that can disable the farmer class)



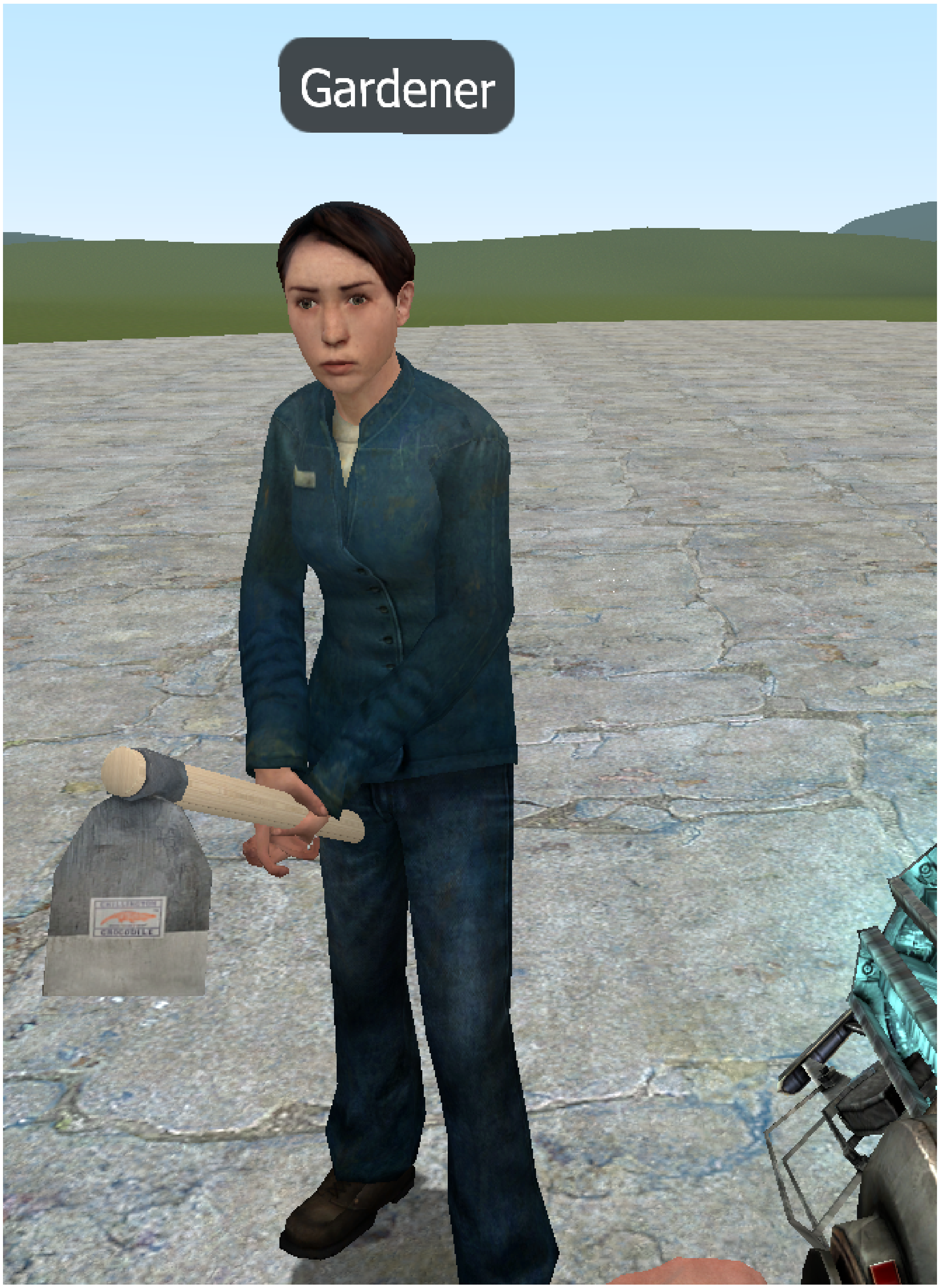
You can't build on these following places: Ceilings or any surface that doesn't meet the right angles A maximum range of 150 Prop or any kind of entity (This means that if there are any invisible entities on the map, as player spawn points then you can't build) The ghost prop will appear red if the cropplot is unable to be build.



The weapon watercan has the following controls: Left click: Water trace plant

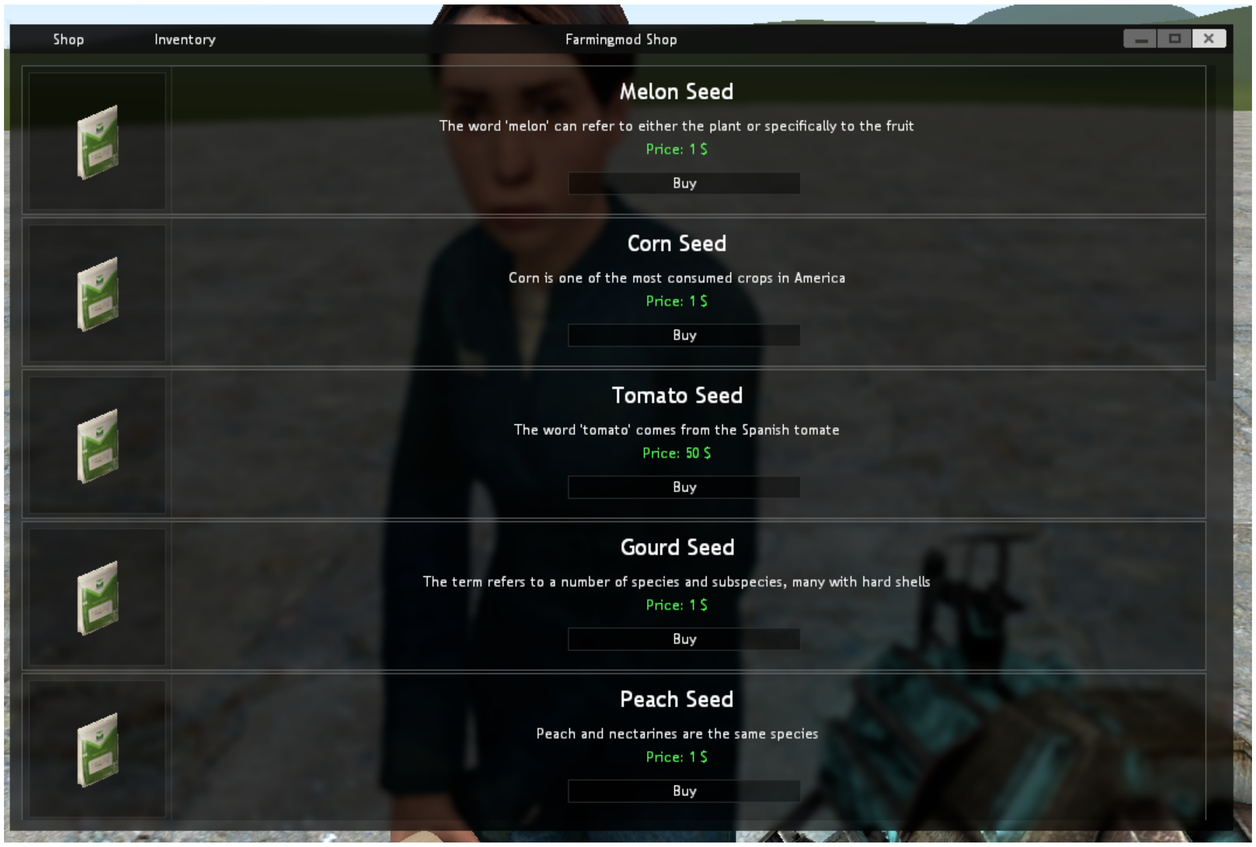
### Shop

This NPC can be spawned by the admin on the server. After the NPC is spawned in the wanted position it can be saved by console commands. The gardener allows you to sell your outcome and to buy seeds. In the gardener you can only buy seeds without any quality. They are generic.


Gardener

By pressing E you can interact with the NPC

The gardener has two submenus. There is the shop submenu and your own inventory submenu. As you see there are plenty of seeds to buy. There is the name of the crop, a little bit of information and a price. You can press on the buy button and there will come options on which quantity you want to buy.



Melon Seed

The word 'melon' can refer to either the plant or specifically to the fruit

Price: 1 $

Corn Seed

Corn is one of the most consumed crops in America

Price: 1 $

Tomato Seed

The word 'tomato' comes from the Spanish tomate

Price: 50 $

Gourd Seed

The term refers to a number of species and subspecies, many with hard shells

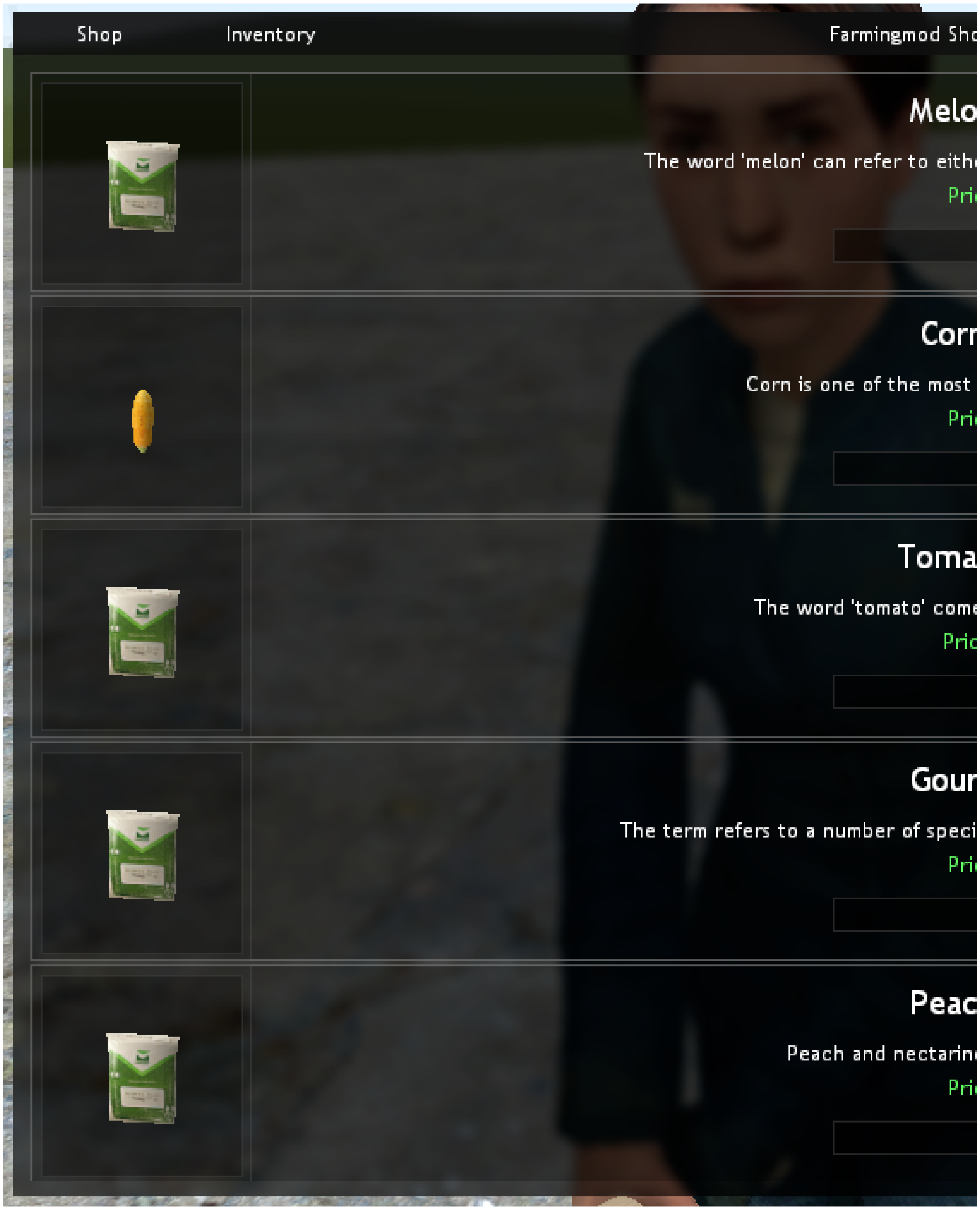
Price: 1 $

Peach Seed

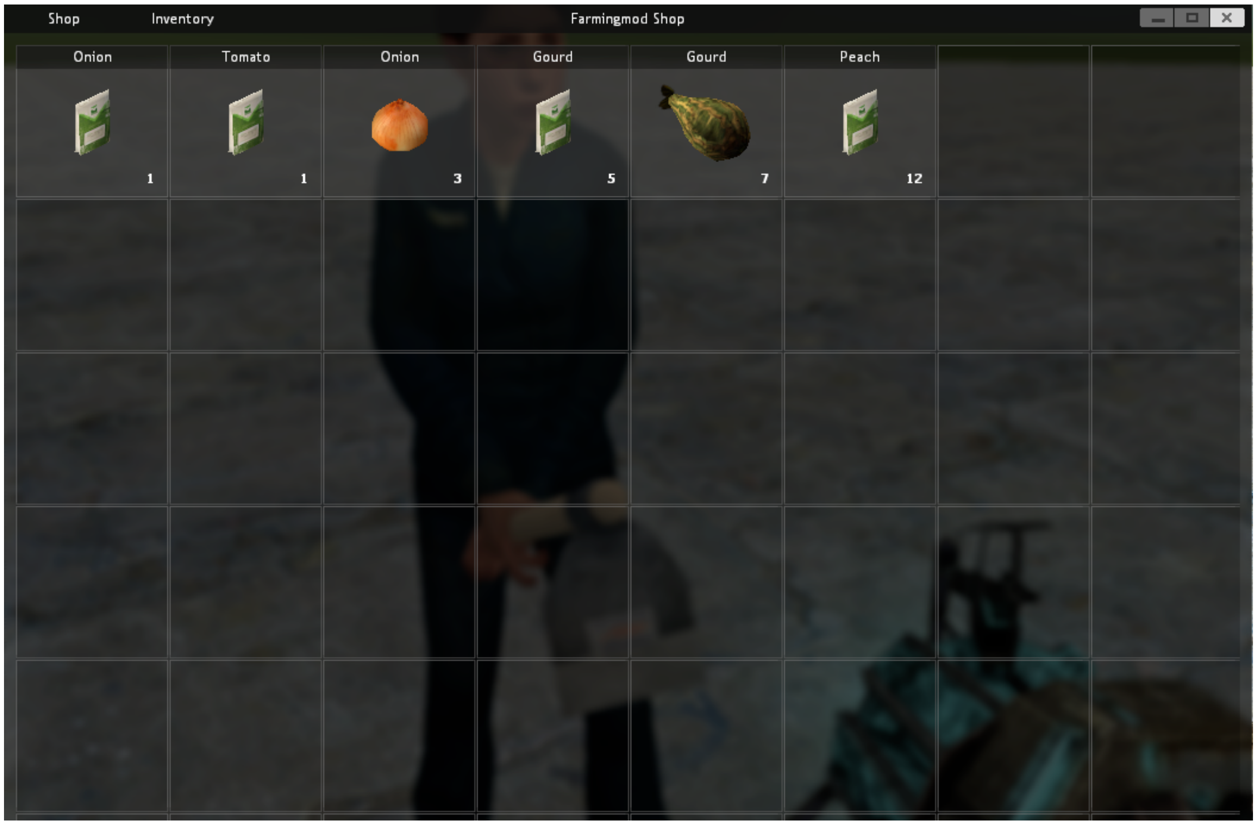
Peach and nectarines are the same species

Price: 1 $

There is a little neat feature. If you put the mouse over the seed packet then the appereance of the outcome will appear.

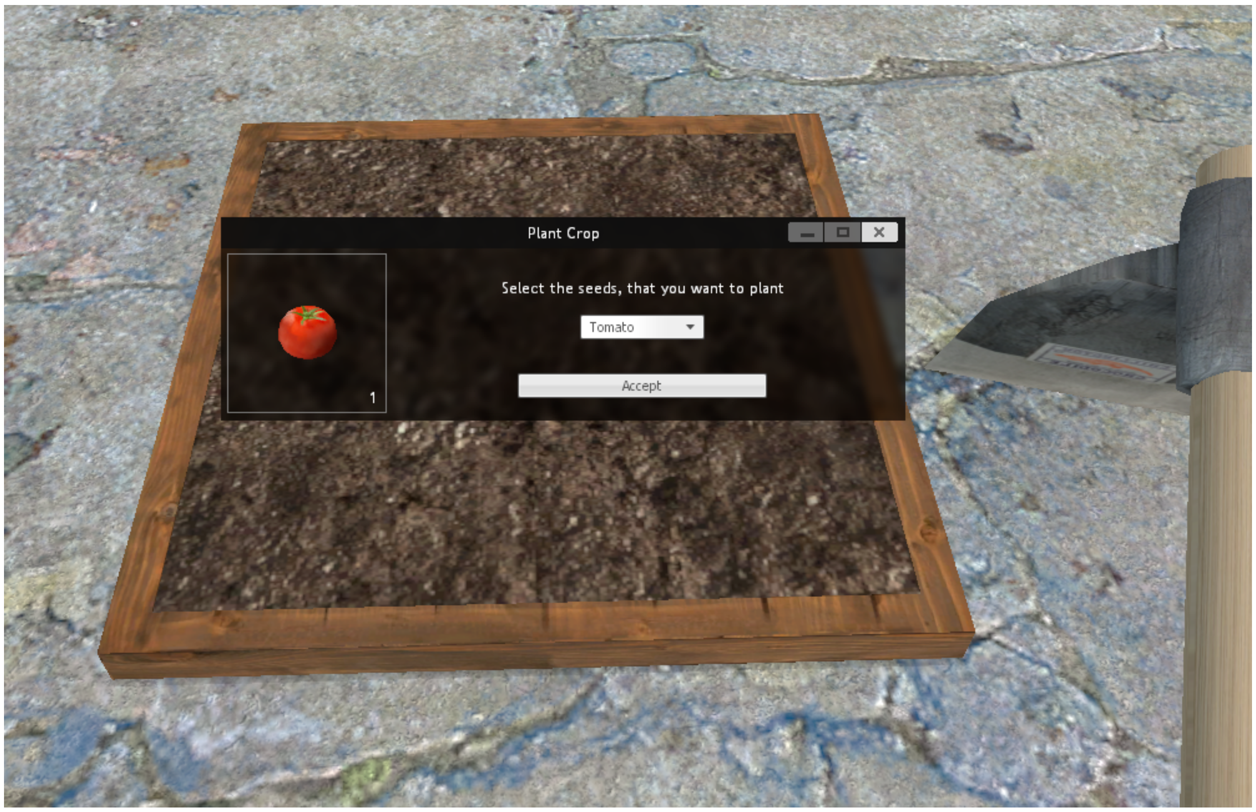


In the inventory submenu you can see your seeds and your harvested crops. By clicking on the icons you will have different options: Sell, Remove and drop. Remember that you can’t sell seeds.



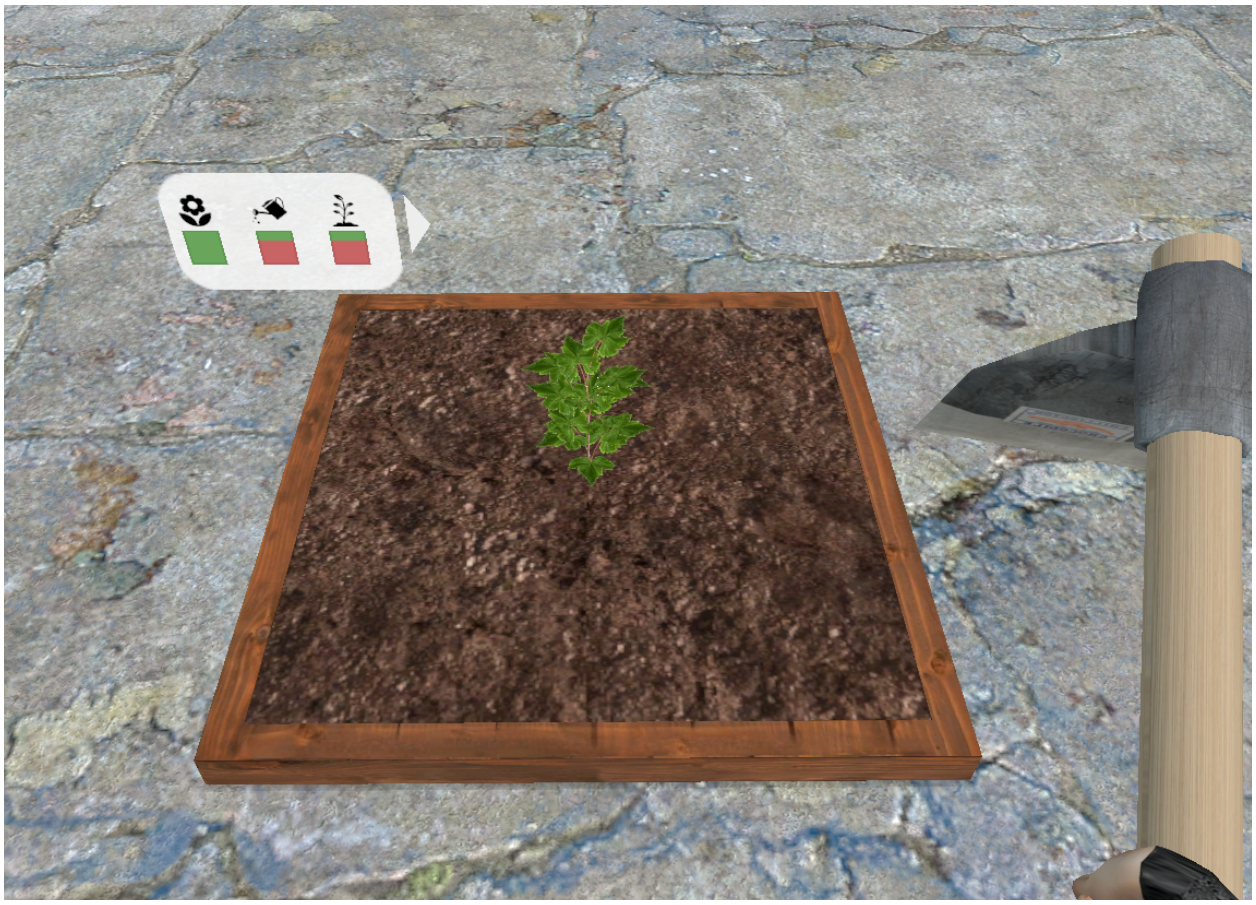
### Sowing & Planting

To sow seeds you have to have a crop plot and bought seeds from the gardener. When you have these requirements you can go to your crop plot and press E. This menu will come up. It will allow you to choose between the seeds that you have on your inventory. When you have chosen the one you want, then accept. Shortly the plant will start to grow.

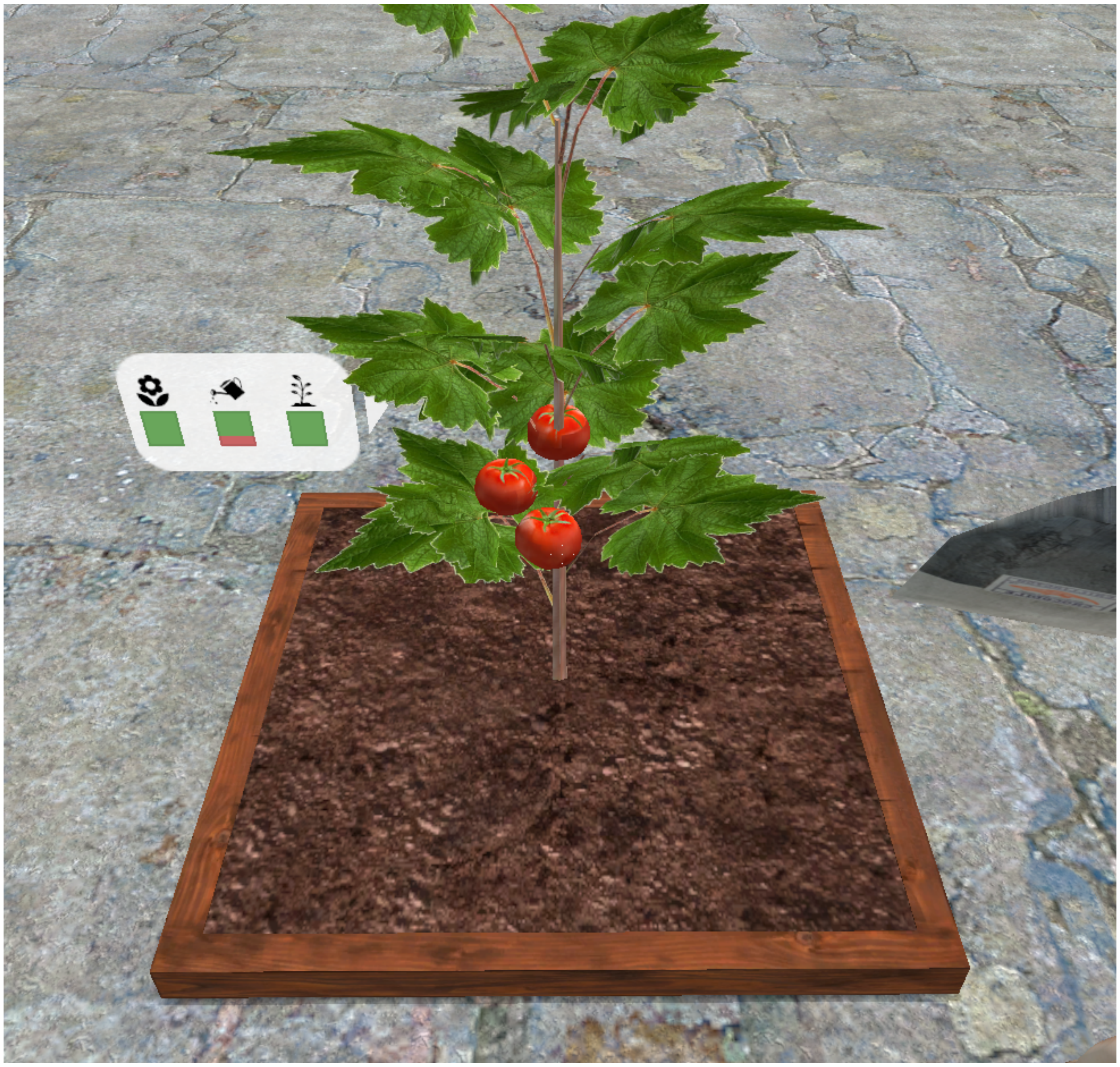


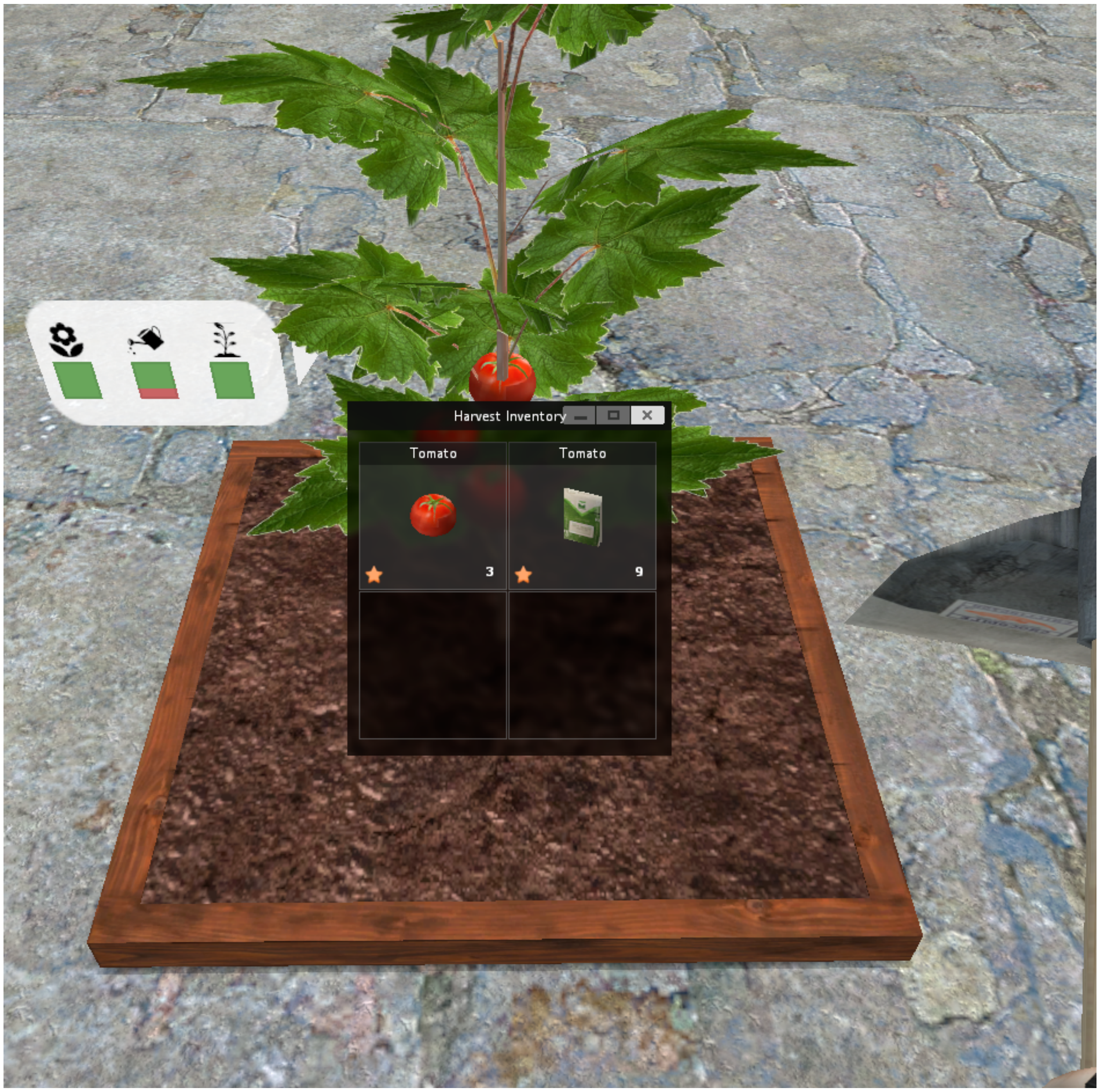
In the options you will be able to see which quality the seed have because there will be stars in the name depending on quality.

While the plant grows there will be information about the plant. The flower tells about the health of the plant. The watercan tells about the humidity of the soil. The plant icon shows the growth progres. When the growth progress bar is done, then the plant will start producing offspring. Remember to keep your plant watered or it will slowly die.



When the plant looks like this, you will be ready to harvest its offspring. You just have to press E on the cropplot and a menu will appear.



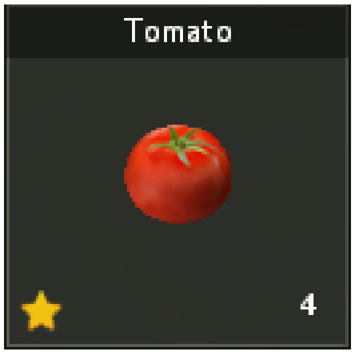


When this menu appear you can by clicking on the icons take what you want from the offspring.

At the moment there aren't any different with the growing of them, but there will in the future be some differences on how to grow each. Like some might require more water than others and etc.

The quality system is based on that there is a little chance of getting a mutation on your crop, which leads to a better offspring. This will encourage players, to keep farming and get better seeds.

Gold (2x on sell) Silver (1.5x on sell) Bronze (1.3x on sell) None (No effects on sell)



Mutations can occur on crops. Mutation gives the crop a better rating in quality. By default there is only 2% chance of getting a mutation.

## 1.2 References

### 1.2.1 addItems

```
FARMINGMOD:AddItems(table tbl,num amount,player ply)
```

With this code you can add items to player inventory. The table has to be an item from crops.lua table

CHAPTER 2

# Craftingmod

Welcome to the wiki site for craftingmod.

## 2.1 Guides

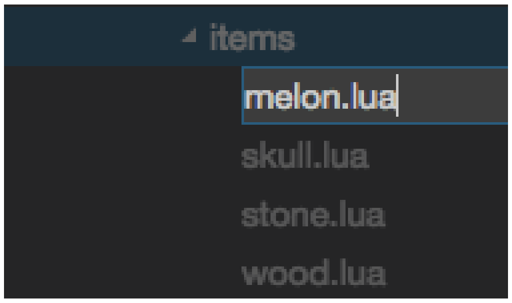
### 2.1.1 Adding Content

#### Items

Its easy to add new items to craftingmod. By adding files to craftingmod/lua/autorun/craftingmod/sh/items/* you automatically add new items to craftingmod. In the mentioned folder, you have to create a lua file. The lua file can be named whatever you like as long as it is compatible with your severs OS.(Keep it in lowered text and dont use spaces) In this guide I will call my lua. file for melon.lua. In this guide, I will show an example on how to add an item and my example item will be a melon.

Find the folder: **craftingmod/lua/autorun/craftingmod/sh/items/*.**

Inside the folder create an empty lua file. In this example i will call mine **melon.lua**, because im adding a melon as an item.



Add some code to melon.lua to give craftingmod some information about the item.

```lua
local ITEM = {}

ITEM.NAME = "Melon"
ITEM.INFORMATION = "A melon that is eatable"
ITEM.MODEL = "models/props_junk/watermelon01.mdl"
ITEM.SKIN = 0
ITEM.WEIGHT = 5

ITEM.SELL = 100
ITEM.BUY = 1

CRAFTINGMOD.ITEMS:Register(ITEM) -- This register the item (DO NOT REMOVE THIS)
```

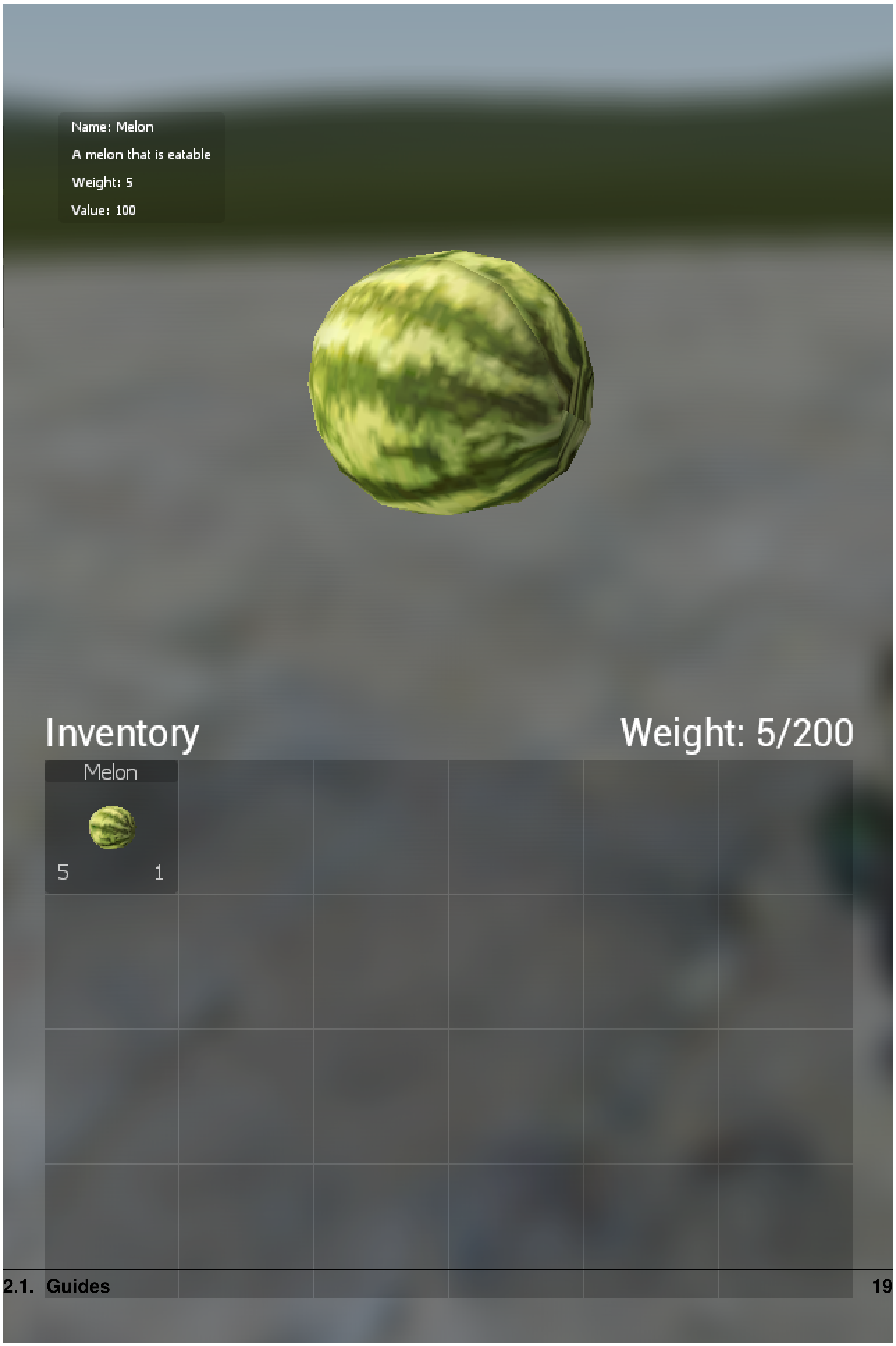
Now you have configured a new item to craftingmod. There will now be an item named melon in craftingmod. The item can be further used now to to create new crafting recipes (See recipes guide).

In craftingmod I have added a new feature for the items, which is the USE function. This allow server owners to further customize each item. With this function you can add eatable items and items that gives you ammo, makes you high or kills you. Its up to you what happens to the player!

```lua
local ITEM = {}

ITEM.NAME = "Melon"
ITEM.INFORMATION = "A melon that is eatable"
ITEM.MODEL = "models/props_junk/watermelon01.mdl"
ITEM.SKIN = 0
ITEM.WEIGHT = 5

ITEM.SELL = 100
ITEM.BUY = 1

-- ITEM.USE is optional (It is not required on the item)
ITEM.USE = function(self, ply) -- An use option will now appear on the inventory, which lets you use the item.
        ply:SetHealth(ply:Health() + 10) -- This will add 10+ health to the player.
end

CRAFTINGMOD.ITEMS:Register(ITEM)
```

This function can be assigned to different items. It can be used if you want to make eatable items or an item that adds ammo to the player and so on. It does only require a simple lua knowlegde and remenber: **ITEM.USE is serverside!**

Congrulations you have now added a melon item to craftingmod which can give the player 10+ in health! Wasn't this easy? I cant make gmod write lua files, so im sorry that you have todo a litle bit of work.


Name: Melon
A melon that is eatable
Weight: 5
Value: 100
Inventory
Weight: 5/200
Melon
5
1

Here there is a code snipped to add 10+ in health and hunger with a limitation.

```lua
ITEM.USE = function(self, ply)
        local
        if (DarkRP) then
                for k, v in pairs(DarkRP.disabledDefaults["modules"]) do
                        if(k == "hungermod" and v == false) then
                                ply:setSelfDarkRPVar("Energy", math.
↪Clamp((ply:getDarkRPVar("Energy") or 100) + 10, 0, 100))
                        end
                end
        end
        if (ply:Health() >= 100) then
                ply:SetHealth(100)
        else
                ply:SetHealth(ply:Health() + 10)
        end
end
```

This gives the player 200 pistol ammo.

```lua
ITEM.USE = function(self, ply)
        ply:GiveAmmo( 200, "Pistol", true )
end
```

Feel free to write your own USE functions!

### Weapons

Adding Weapon is similar to adding items. Though it does have some essential differences that you have to learn about.

Find the folder: **craftingmod/lua/autorun/craftingmod/sh/weapons/*.**

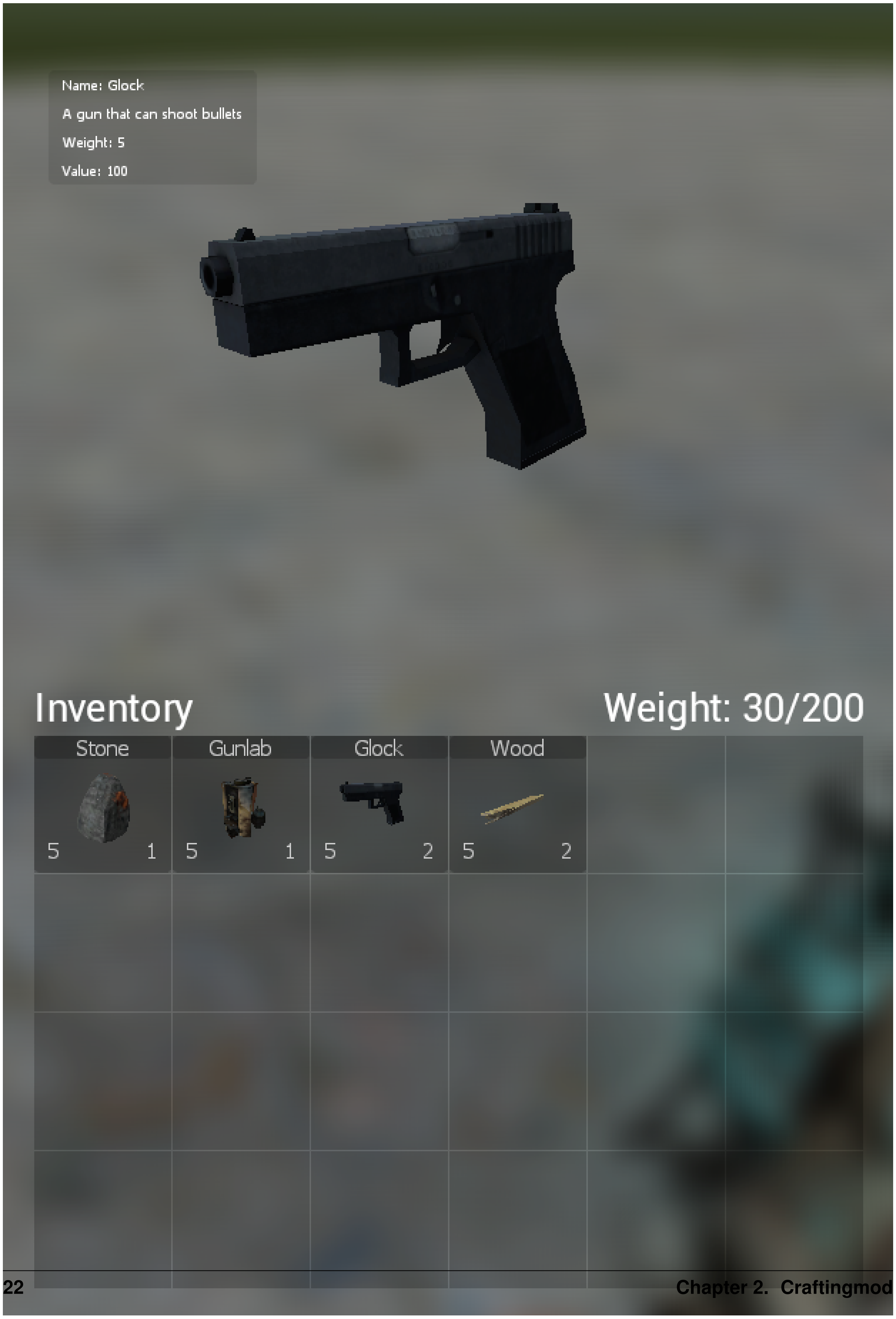
Inside the folder create an empty lua file. In this example i will call mine **glock.lua**, because im adding a Glock as a weapon.

Add some code to glock.lua to give craftingmod some information about the weapon.

```lua
local WEAPON = {}

WEAPON.NAME = "Glock"
WEAPON.INFORMATION = "A gun that can shoot bullets"
WEAPON.MODEL = "models/weapons/w_pist_glock18.mdl"
WEAPON.SKIN = 0
WEAPON.SWEP = "weapon_glock2"

WEAPON.SELL = 100
WEAPON.BUY = 100

WEAPON.WEIGHT = 5

CRAFTINGMOD.WEAPONS:Register(WEAPON)
```

Now you have configured a new weapon to craftingmod. There will now be a weapon named glock in craftingmod. The item can be further used now to to create new crafting recipes (See recipes guide). Note that this will allow you to pick the weapon Glock up because you have informed craftingmod about that the weapon weapon_gloc2 from DarkRP is in craftingmod.

Congrulations you have now added a glock weapon to craftingmod! - I’m sorry that you have todo a litle bit of work to add weapons, but garrysmod don’t allow me to generate lua files.


Name: Glock
A gun that can shoot bullets
Weight: 5
Value: 100
Inventory
Weight: 30/200
Stone
5
1
Gunlab
5
1
Glock
5
2
Wood
5
2

If you want the weapon to be taken out of the inventory when you equip the weapon from the inventory, then you can add a configuration line.

```
WEAPON.SINGLE = true
```

This line is usefull if you configure grenades or molotove cocktails or stuff like that.

### Entities

Its easy to add new entities to craftingmod. By adding files to craftingmod/lua/autorun/craftingmod/sh/entities/* you automatically add new entities to craftingmod. In the mentioned folder, you have to create a lua file. The lua file can be named whatever you like as long as it is compatible with your severs OS.(Keep it in lowered text and dont use spaces) In this guide I will call my lua. file for gunlab.lua. I will show an example on how to add an the gunlab entity from DarkRP.

Find the folder: **craftingmod/lua/autorun/craftingmod/sh/entities/*.**

Inside the folder create an empty lua file. In this example i will call mine **gunlab.lua**, because im adding the gunlab entity.

Add some code to gunlab.lua to give craftingmod some information about the entity. (Yes it uses mostly the same code as for adding items. I made this folder to keep items and entities separated. Also in entities there is ITEM.ENTITY = “the entity”)

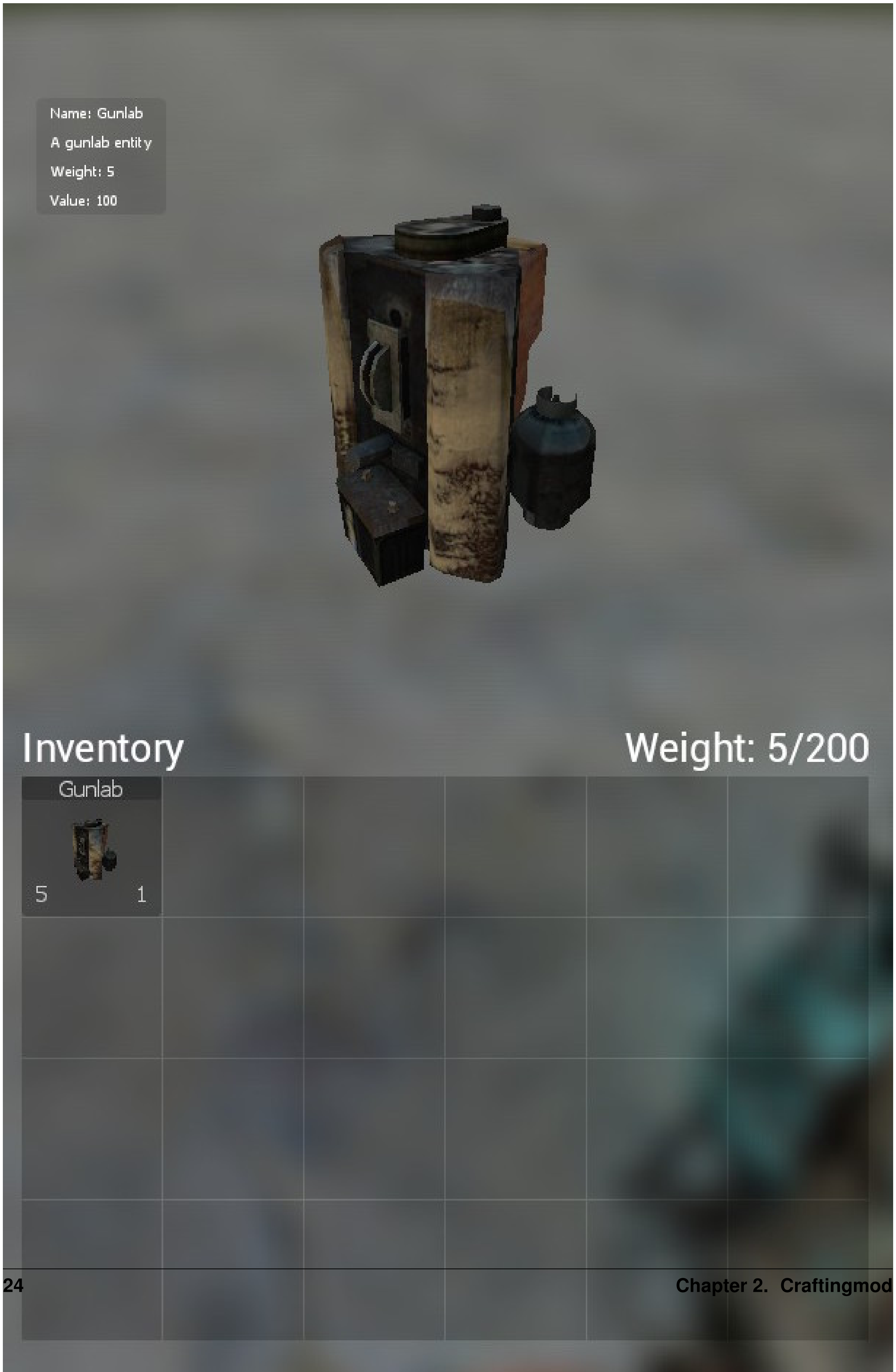
```
local ITEM = {}

ITEM.NAME = "Gunlab"
ITEM.INFORMATION = "A gunlab entity"
ITEM.MODEL = "models/props_c17/TrapPropeller_Engine.mdl"
ITEM.SKIN = 0
ITEM.ENTITY = "gunlab"

ITEM.SELL = 100
ITEM.BUY = 100

ITEM.WEIGHT = 5

ITEM.USE = nil

CRAFTINGMOD.ITEMS:Register(ITEM)
```

Now you have configured a new entity to craftingmod. The gunlab entity is now configured into craftingmod. The entity can be further placed by the inventory and used as all other entities. (Note that there is a maximum for placed props/entities. This can be configured in sh_config.lua)

Congrulations you have now added the gunlab to craftingmod which can be placed down. Wasn’t this easy? I cant make gmod write lua files, so im sorry that you have todo a litle bit of work.

Remenber that you can add all kinds of entities! Feel free to add whatever you want.

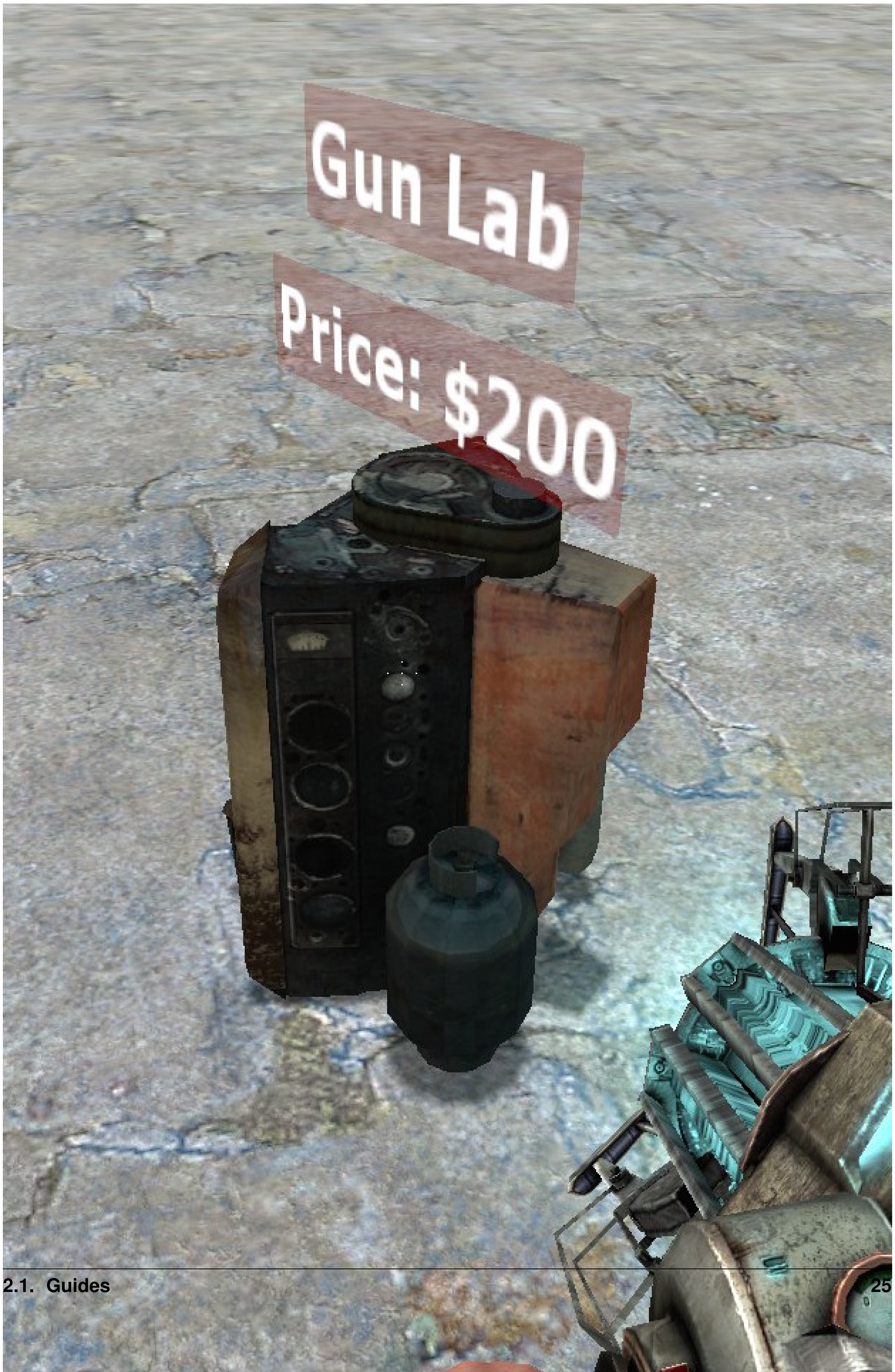
**Remenber that you can pickup placed entities / props by looking at it and typing /pickup**


Name: Gunlab

A gunlab entity

Weight: 5

Value: 100


Gun Lab
Price: $200

You can add entity restrictions in sh/sh_config in the RESTRICTION list, by simply adding the entity class name.

You can even add place entity restrictions based on usergroup and specific amount. If you haven't set ITEM.RESTRICTION on a item configuration then the default place config will rule.

```
ITEM.RESTRICTION = {
    {1, "superadmin"},
}
```

Or you can restrict the item by job name

```
ITEM.RESTRICTION = {
    {1, "Citizen"},
}
```

### Vehicles

Its easy to add vehicles. Just follow the small steps and you will be ready.

Find the folder: **craftingmod/lua/autorun/craftingmod/sh/entities/*.**

Inside the folder create an empty lua file. In this example i will call mine **jeep.lua**, because im adding the jeep vehicle entity.

Add some code to jeep.lua to give craftingmod some information about the vehicle entity. The code reminds much of the generic entity item configuration.

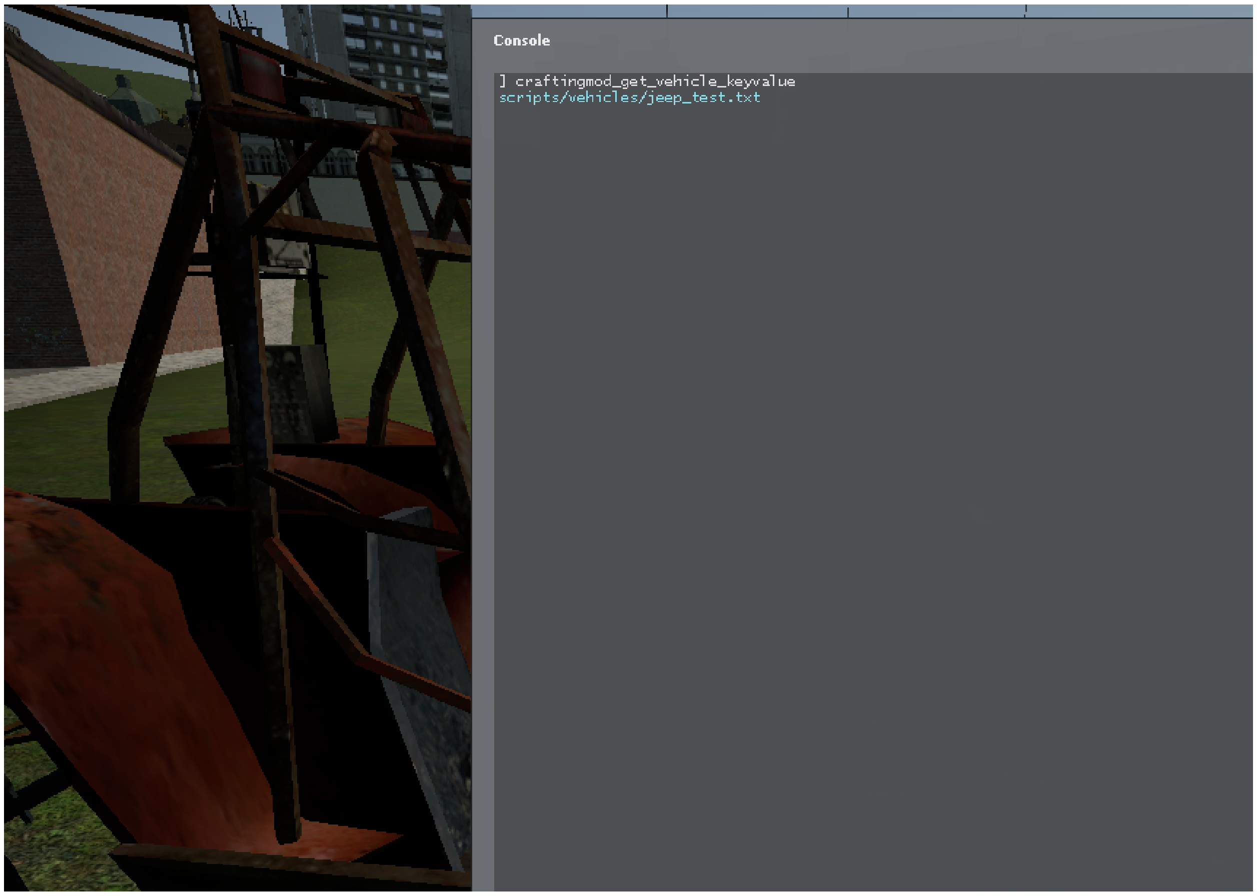
```
    local ITEM = {}

    ITEM.NAME = "Jeep"
    ITEM.INFORMATION = "A simple oldschool jeep"
    ITEM.MODEL = "models/buggy.mdl"
    ITEM.SKIN = 0
    ITEM.ENTITY = "prop_vehicle_jeep"
    ITEM.VEHICLE = "scripts/vehicles/jeep_test.txt"

    ITEM.SELL = 150
    ITEM.BUY = 300

    ITEM.WEIGHT = 20

    ITEM.USE = nil

    CRAFTINGMOD.ITEMS:Register(ITEM)
```

Congratulations, you have now configured a vehicle.

**Where do i find the ITEM.VEHICLE script path?**

No worries. I have created a command in craftingmod, that prints out the vehicle script path in console.

```
    craftingmod_get_vehicle_keyvalue
```



```
] craftingmod_get_vehicle_keyvalue
scripts/vehicles/jeep_test.txt
```

### Levels

In craftingmod you can add your own levels. They can be used for level requirements in recipes and they can control the resource gathering speed.

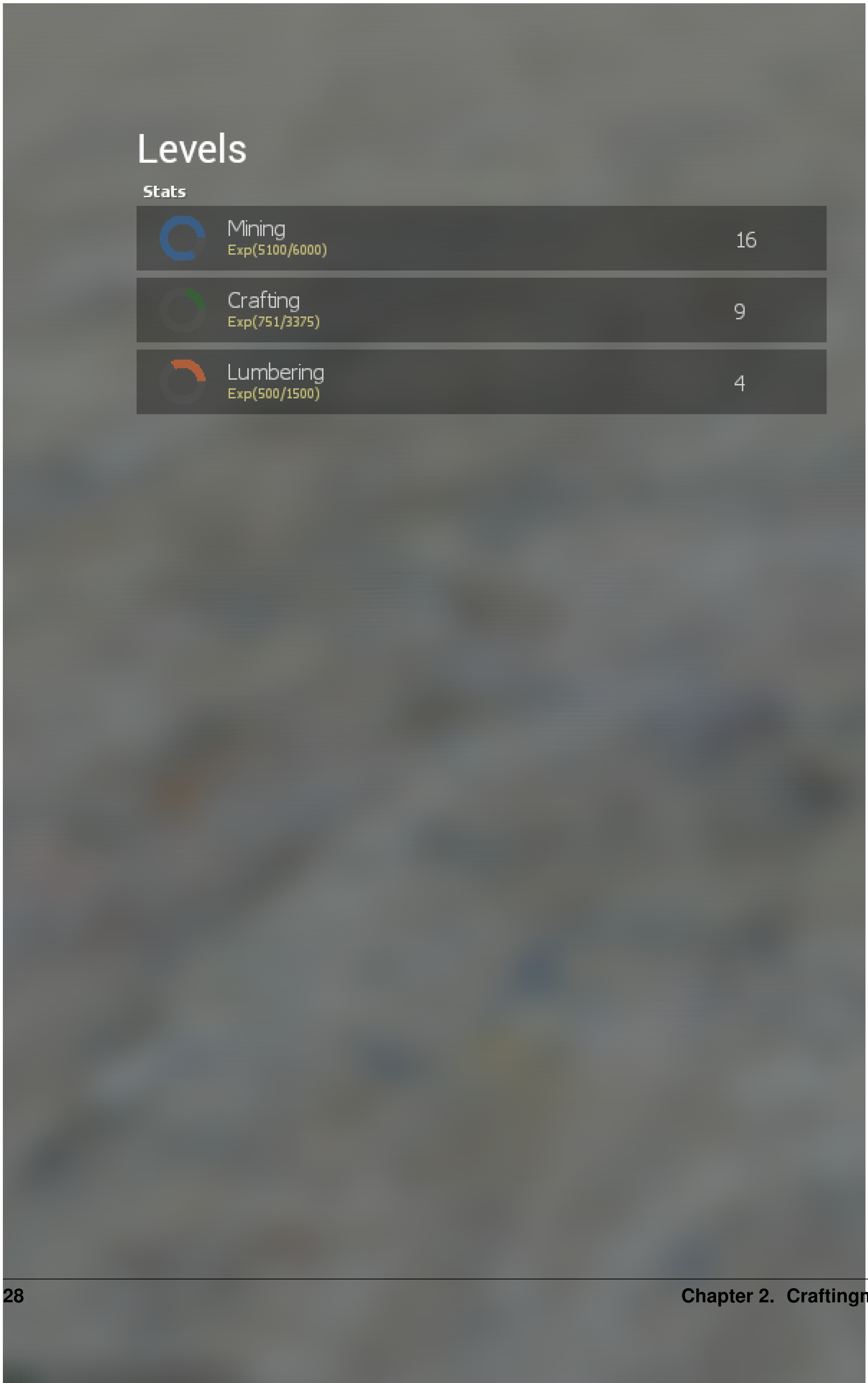
Find the folder: **craftingmod/lua/autorun/craftingmod/sh/levels/*.**

Inside the folder create an empty lua file. In this example i will call mine **lumbering.lua**, because im adding a lumbering level.

Lets add some code to lumbering.lua to give craftingmod some information about the level.

```
local LEVEL = {}

LEVEL.NAME = "Lumbering"
LEVEL.INFORMATION = "Determines the speed of the hatchet"
LEVEL.COLOR = Color(0,255,0,255)

LEVEL.OnLevelUp = function(ply)
    --Empty function for anything special when the player levels up
end

CRAFTINGMOD.LEVELS:Register(LEVEL)
```

Congrulations! You have now added a new level to craftingmod. With this you will now be able to create a craftingmod weapon that gather resources in a speed thats based on the level. This will be further discussed in the resourcespots section.



| | | |
|---|---|---|
| Mining | Exp(5100/6000) | 16 |
| Crafting | Exp(751/3375) | 9 |
| Lumbering | Exp(500/1500) | 4 |

### Resource Spots

Resource spots are now configurable in craftingmod 2.0! With this you can create your own resourcespots for players to harvest.

Find the folder: **craftingmod/lua/autorun/craftingmod/sh/resources/*.**

Inside the folder create an empty lua file. In this example i will call mine **rock.lua**, because im adding a rock resource spot.

Now you have to add some code to the empty lua file, to configure a new resourcespot. In this guide im configuring a rock resourcespot.

The resource have 3 main variables that are the NAME of the resourcespot, the MODEl of the resourcespot and the SKIN of the resourcespot.

The two others tells it, which weapon from craftingmod/lua/weapons/* from craftingmod that can harvest this resourcespot and which item it outputs. Remenber that i showed how to add a melon? Alright in craftingmod you just have to tell the ITEM.NAME to the resourcespot, so that it can find the item in the craftingmod data.

This code will create a Rock, that when harvested by the craftingmod_pickaxe it gives you a melon.

```
local RESOURCE = {}

RESOURCE.NAME = "Rock"
RESOURCE.MODEL = "models/eryk/craftingmod/rock01.mdl"
RESOURCE.SKIN = 0

RESOURCE.ITEM = "Stone"
RESOURCE.WEAPON = { "craftingmod_pickaxe", "craftingmod_pickaxe_copper", "craftingmod_
↪pickaxe_iron", "craftingmod_pickaxe_gold", "craftingmod_pickaxe_diamond" }
RESOURCE.LEVEL = {100, "Mining"}

RESOURCE.OUTPUT = 3

CRAFTINGMOD.RESOURCES:Register(RESOURCE)
```

Congrulations! You have now added a new resource spot to the stool resourcespot under the category craftingmod. You can make single resourcespots or create radius generated resourcespots. Remenber that after having created them you have to save them by the command CRAFTINGMOD_SaveEntities, so that they are created at server restart.

With this you can make all kinds of resource spots that people can harvest and all this can be configured easily by you!

I will for this last thing tell you how to create new weapons for RESOURCE.WEAPON. Remenber that normal weapons and weapon items are not the same thing!

> **craftingmod/lua/weapons/*.**

Here you can see folders of different craftingmod_* weapons. To create a new one, just copy and paste one of the folders and rename the folder to the new weapon that you're adding. It could be in this case: craftingmod_<insert new weapon name>

Now go inside the new folder and open the shared.lua file. In this file you just have to change the following lines to your own liking.

```
SWEP.PrintName                              = "The printname"
SWEP.Author                                 = "Your name here"
SWEP.Purpose                                = "What does this weapon do?"

SWEP.ViewModel                              = Model( "my viewmodel" )
SWEP.WorldModel                             = Model( "my worldmodel" )
```

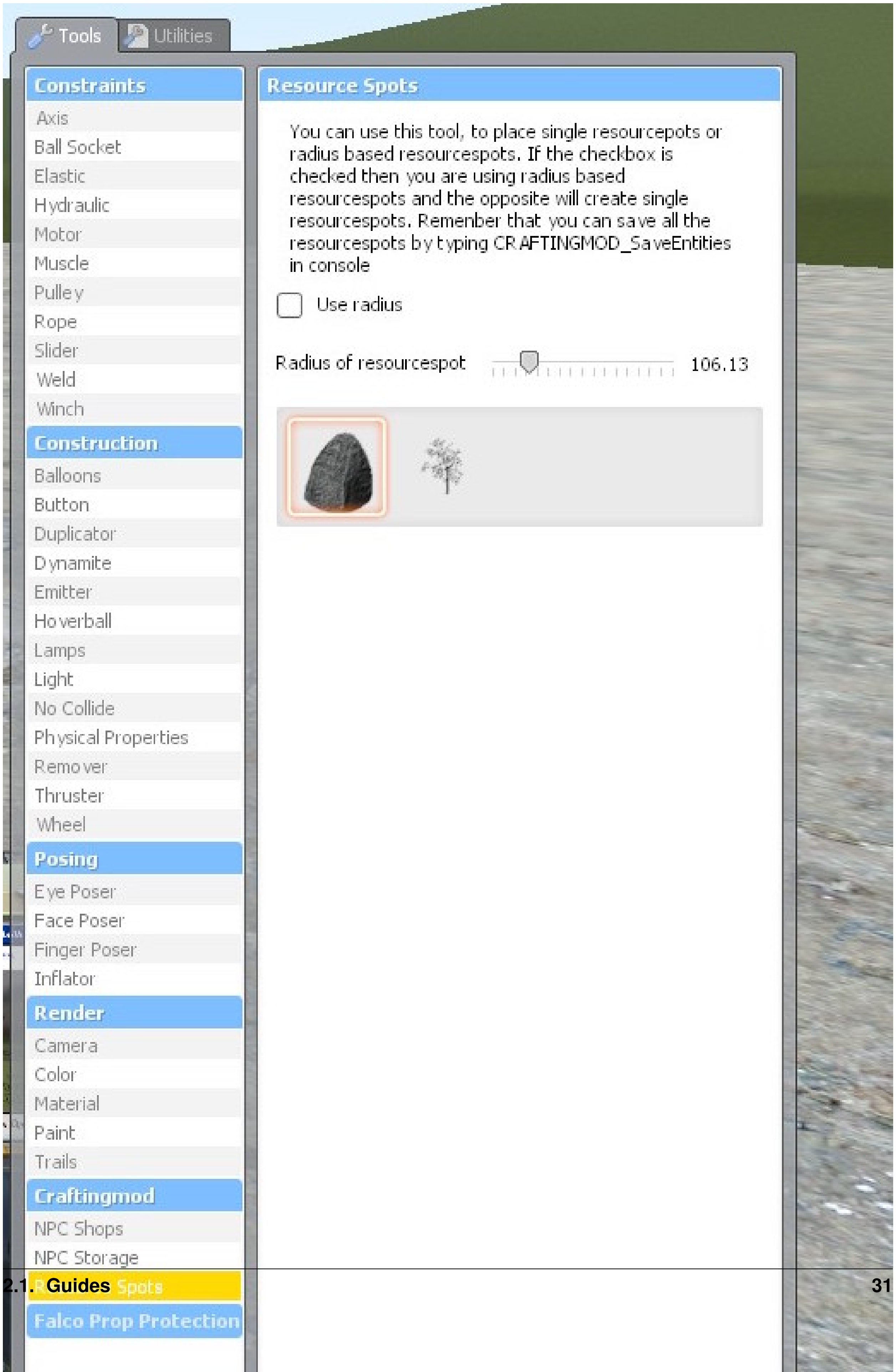
(continues on next page)

(continued from previous page)

```

SWEP.Level = "Mining" -- What level will control the gathering speed

SWEP.SoundList = { -- Add some sounds to it
    Sound( "<mypath>.wav" ),
Sound( "<mypath>.wav" ),
}
```

After this you can use the new craftingmod_<insert new weapon name> as a weapon in RESOURCE.WEAPON because they run off an base file. How simple can this be?

**Have fun with your new resourcespots!**



Resource Spots

You can use this tool, to place single resourcepots or radius based resourcespots. If the checkbox is checked then you are using radius based resourcespots and the opposite will create single resourcespots. Remenber that you can save all the resourcespots by typing CRAFTINGMOD_SaveEntities in console

☐ Use radius

Radius of resourcespot 106.13

### Workbenches

In craftingmod you can add new workbenches. Workbenches are entities that can hold recipes. In this guide I will show how they are connected to recipes and how they work.

Find the folder: **craftingmod/lua/autorun/craftingmod/sh/workbenches/*.**

Inside the folder create an empty lua file. In this example i will call mine **carpentry.lua**, because im adding a Carpentry workbench.

Lets now add some code to it, to make craftingmod add this new workbench.

Most of the configuration are common to other files. You just have to fill out the different configurations and the names says pretty much what they mean. Remenber that the name of the workbench is important on the recipes.

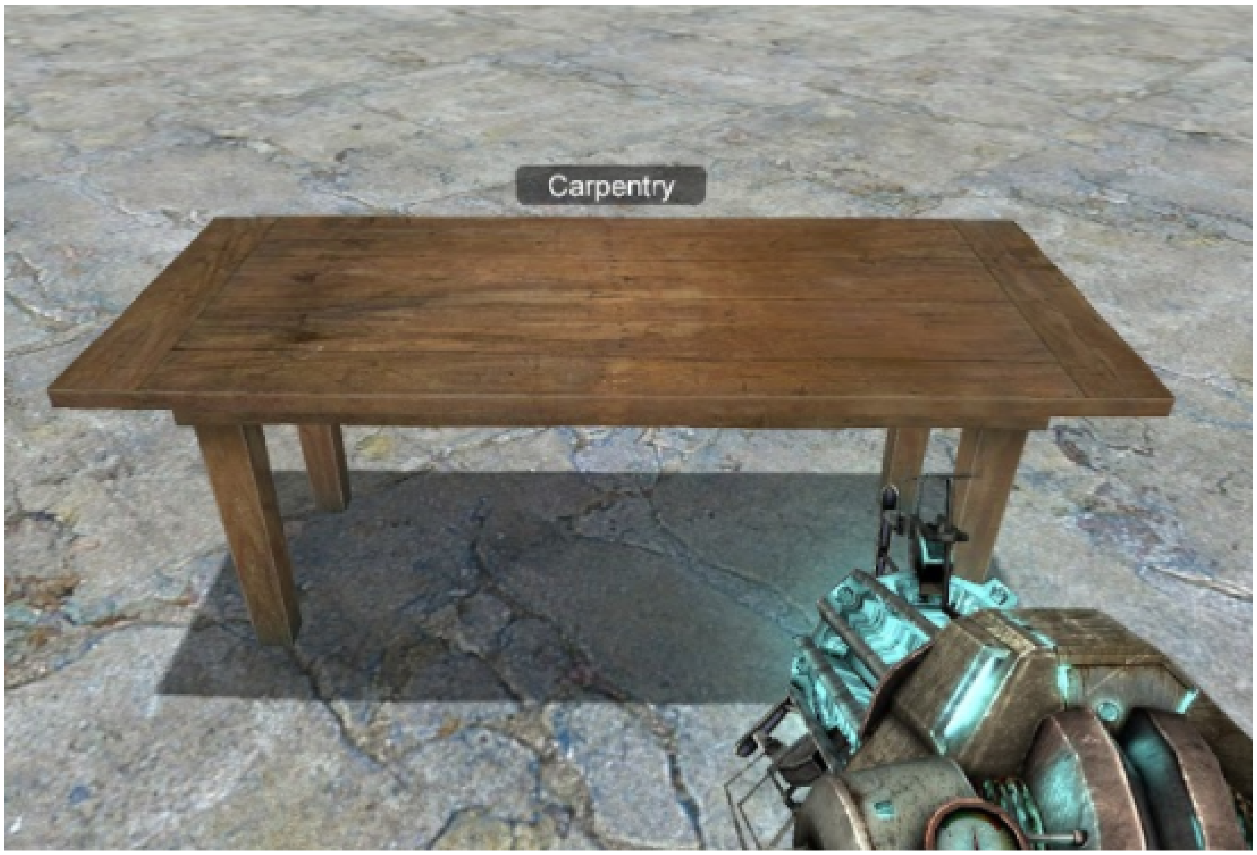
```lua
local WORKBENCH = {}

WORKBENCH.NAME = "Carpentry"
WORKBENCH.INFORMATION = "A carpentry station"
WORKBENCH.MODEL = "models/props/cs_militia/wood_table.mdl"
WORKBENCH.SKIN = 0

WORKBENCH.ENTITY = "craftingmod_workbench" -- DONT CHANGE THIS!

WORKBENCH.SELL = 100
WORKBENCH.BUY = 100

WORKBENCH.WEIGHT = 5

CRAFTINGMOD.WORKBENCHES:Register(WORKBENCH)
CRAFTINGMOD.ITEMS:Register(WORKBENCH)
```

The workbench is now configured and in craftingmod all the items are normally defined by their name. For example if we have an recipe that has this configuration **RECIPE.WORKBENCH** then we should write the exact name if we want the recipe to be connected to the carpentry workbench. It would look like this:

**A SNIP OF RECIPE CODE**

```lua
RECIPE.WORKBENCH = "Carpentry"
```

Congrulations you have now added a new workbench! This can be further used for holding recipes!



### Recipes

In craftingmod you can easily add new recipes to either a workbench or the player menu. This guide will show you how to add recipes to one of them.

Find the folder: **craftingmod/lua/autorun/craftingmod/sh/recipes/*.**

Inside the folder create an empty lua file. In this example i will call mine **coal.lua**, because im adding a coal item recipe.

Lets add some code to the empty coal.lua file.

```lua
local RECIPE = {}

RECIPE.NAME = "Coal"
RECIPE.CATEGORY = "Fuel"
RECIPE.MODEL = "models/eryk/craftingmod/small_rock.mdl"
RECIPE.SKIN = 1

RECIPE.ITEM = "Coal"
RECIPE.WORKBENCH = "Furnace"

RECIPE.LEVEL = {
    {5, "Crafting"},
}

RECIPE.RESOURCES = {
    { 10, "Wood"},
```

(continues on next page)

(continued from previous page)

```
}

RECIPE.OnCrafting = function(amount, ply)
    local steamID = ply:SteamID()
    CRAFTINGMOD.User[steamID].Levels:AddExperience("Crafting", math.ceil(math.
↪random(50, 200)) * amount, ply)
end

CRAFTINGMOD.RECIPES:Register(RECIPE)
```

The RECIPE.ITEM is to tell it which item you want out of this recipe. The item must be an item that has been configured. You just have to put the exact name. The item can be an entity, workbench, weapon, item.. as long as it is configured.

The RECIPE.WORBENCH is to tell the recipe which workbench where you want this recipe to be in. You have just to put the name of the workbech. Remenber that the workbench has to be configured. There is a guide on how todo it.

**If you want it on the player menu, then remove the RECIPE.WORKBENCH line**

The RECIPE.LEVEL tells the recipe any level requirement. {level_requirement, "Name of the Level"}.

**If you dont want any level requirement then remove the RECIPE.LEVEL line**

The RECIPE.RESOURCES tells the recipe which items it requires. You can add more resources like this:

```
RECIPE.RESOURCES = {
    { amount, "Name of item/weapon/entity/workbench"},
    { amount, "Name of item/weapon/entity/workbench"},
    { amount, "Name of item/weapon/entity/workbench"},
}
```

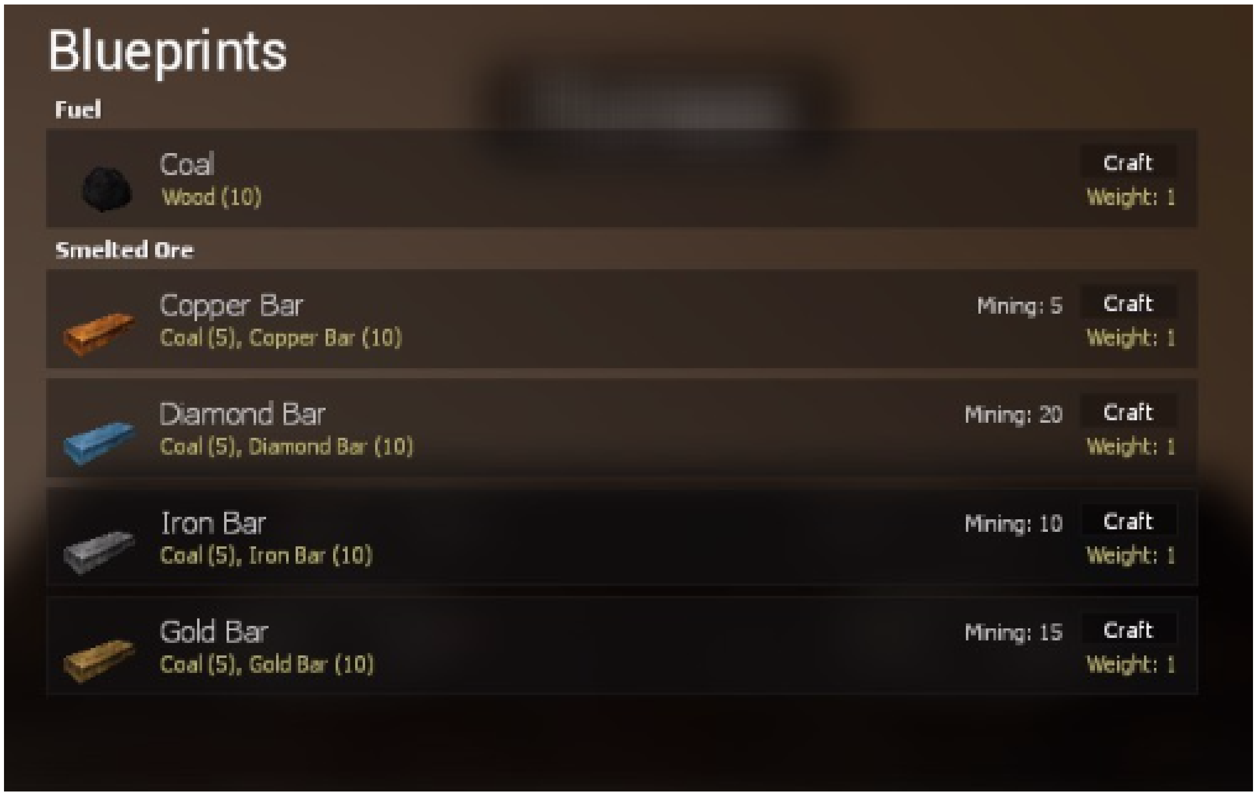
And so on..

You can add either no levelrequirement by setting the RECIPE.LEVEL = nil or you can add multiple level requirements like this:

```
RECIPE.LEVEL = {
    {5, "Crafting"},
}
```

Nearly all items/weapons/entity/workbenches are found by craftingmod by their name. So you have to watch out carefully for the name that you type. It have to be the exact same.

Congrulations! You have now added a new recipe to the Furnace workbench! (Remenber this recipe is already pre-configured in craftingmod 2.0)



Blueprints

Fuel

Coal
Wood (10)
Craft
Weight: 1

Smelted Ore

Copper Bar
Coal (5), Copper Bar (10)
Mining: 5
Craft
Weight: 1

Diamond Bar
Coal (5), Diamond Bar (10)
Mining: 20
Craft
Weight: 1

Iron Bar
Coal (5), Iron Bar (10)
Mining: 10
Craft
Weight: 1

Gold Bar
Coal (5), Gold Bar (10)
Mining: 15
Craft
Weight: 1

If you want to change the delay time of the recipe you can add this configuration.

```
RECIPE.DELAY = <insert time in seconds>
```

If you want to add restriction to jobs then you can add this configuration.

```
RECIPE.JOBS = { <job name>, <job_name> }
```

### Base Addons Items

In this guide you will learn the concept of saving an entities data into the inventory. With this method, you can for example add different upgrades of a money printer or any other addon that use variables and one entity type. In our case we have an “ingredient” class entity that, has a SetIngredient(tbl) function and a GetIngredient() function. The SetIngredient(tbl) gives the ingredient entity class informations about if it is a tomato or anything else. The GetIngredient() return the type of the ingredient. These functions are on the entity class.

Find the folder: **craftingmod/lua/autorun/craftingmod/sh/addons/<create addon name folder>/*. Inside the folder create an empty lua file. In this example i will call mine **ingredient.lua**, because im adding a configuration for all ingredient types recipe.

Let me show you the configuration of this entity item. We start by configuring the same way as we normally do with an entity. The exeption is that, the model wont really matter because this is a base configuration and the model will change based on SetIngredient(tbl). Another thing is, that we will tell craftingmod that we are configuring an item from another addon (ITEM=ADDON).

```
local ITEM = {}
ITEM.NAME = "Ingredient"
ITEM.INFORMATION = "An ingredient"
ITEM.MODEL = "models/error.mdl"
```

(continues on next page)

(continued from previous page)

```
ITEM.SKIN = 0
ITEM.ENTITY = "cook_ingredient"

ITEM.SELL = 20
ITEM.BUY = 50

ITEM.WEIGHT = 2

ITEM.ADDON = true
```

Covering the new functions available for the item configuration. SaveData allows us to add a custom item to the inventory and loadData allows us to load variables back on the entity with its old configurations. Remember that the "ingredient_cook" entity has the GetIngredient and SetIngredient functions coded into.

When any item is configured there is added a function to ITEM. A ITEM.SetData() and ITEM.GetData() to be able to add tables to the ITEM table.

```
ITEM.SaveData = function( self, ent ) -- When we pick an "cook_ingredient" entity␣
↪this is invoked
    self:SetData( "Ingredient", ent:GetIngredient() ) -- The line saves a "Ingredient
↪" table on this ITEM table with the returned values from ent:GetIngredient().

    self.NAME = ent:GetIngredient() -- We change the NAME based on the␣
↪ent:GetIngredient() value
    self.MODEL = ent:GetModel() -- We change the model of this ITEM table from␣
↪information about the ent
end

ITEM.LoadData = function( self, ent ) -- When we place an "cook_ingredient" item this␣
↪is invoked.
    ent:SetIngredient( self:GetData( "Ingredient" ) ) -- We use the SetIngredient and␣
↪self:GetData("Ingredient") to get back the old values
end
```

**LETS MAKE A EASIER EXAMPLE [DONT USE THIS]**

```
ITEM.SaveData = function( self, ent ) s
    self:SetData( "TheModelOfTheEntity", ent:GetModel()) -- We save the model of the␣
↪entity
    self:SetData( "TheNameOfTheEntity", ent:GetName()) -- We save the model name

    self.NAME = ent:GetName() -- We change the name here to give the item its unique␣
↪name
    self.MODEL = ent:GetModel() -- -- We change the model here to give the item its␣
↪unique model
end

ITEM.LoadData = function( self, ent )
    print(self:GetData("TheModelOfTheEntity"))
    print(self:Getdata("TheNameOfTheEntity"))
end
```

This will output the name and model showing that we can save variables from the entity into a custom item.

Congrulations! We have now configured a base entity that changes based on the spawned "cook_ingredient". So if the "cook_ingredient" is a tomato with a tomato model then it will change the ITEM.NAME to tomato, the ITEM.MODEL to tomate and it will load up the data into the entity if we place the tomato from the inventory.

### 2.1.2 Inventory

#### Introduction

The inventory has the following options:

- Use
- Equip & UnEquip
- Place
- Move
- Buy
- Sell
- Drop
- Remove

The following options appear on different occasions.

The **Use** option appear when the selected item has an ITEM.USE assigned.

The **Equip & UnEquip** option appear when the selected item is a weapon.

The **Place** option appear when the selected item is either a weapon or a an entity. This allows you to place the entity or weapon as the "real" item and not as an craftingmod_item. You can take the entity or weapon back by typing /pickup on the chatbox.

The **Move** option appear when you are using the storage to save items or take them back.

The **Buy** option appear on the shop (Its actually not an option from the inventory)

The **Sell** option appears when you're inside a shop, so that you can sell your items.

The **Drop** appear on all items and can be used to drop the item as a craftingmod_item. A dropped item can be picked back up by pressing E.

The **Remove** option appear on all items and can be used to completely remove the item. It just dissapears from the inventory.

### 2.1.3 Shops

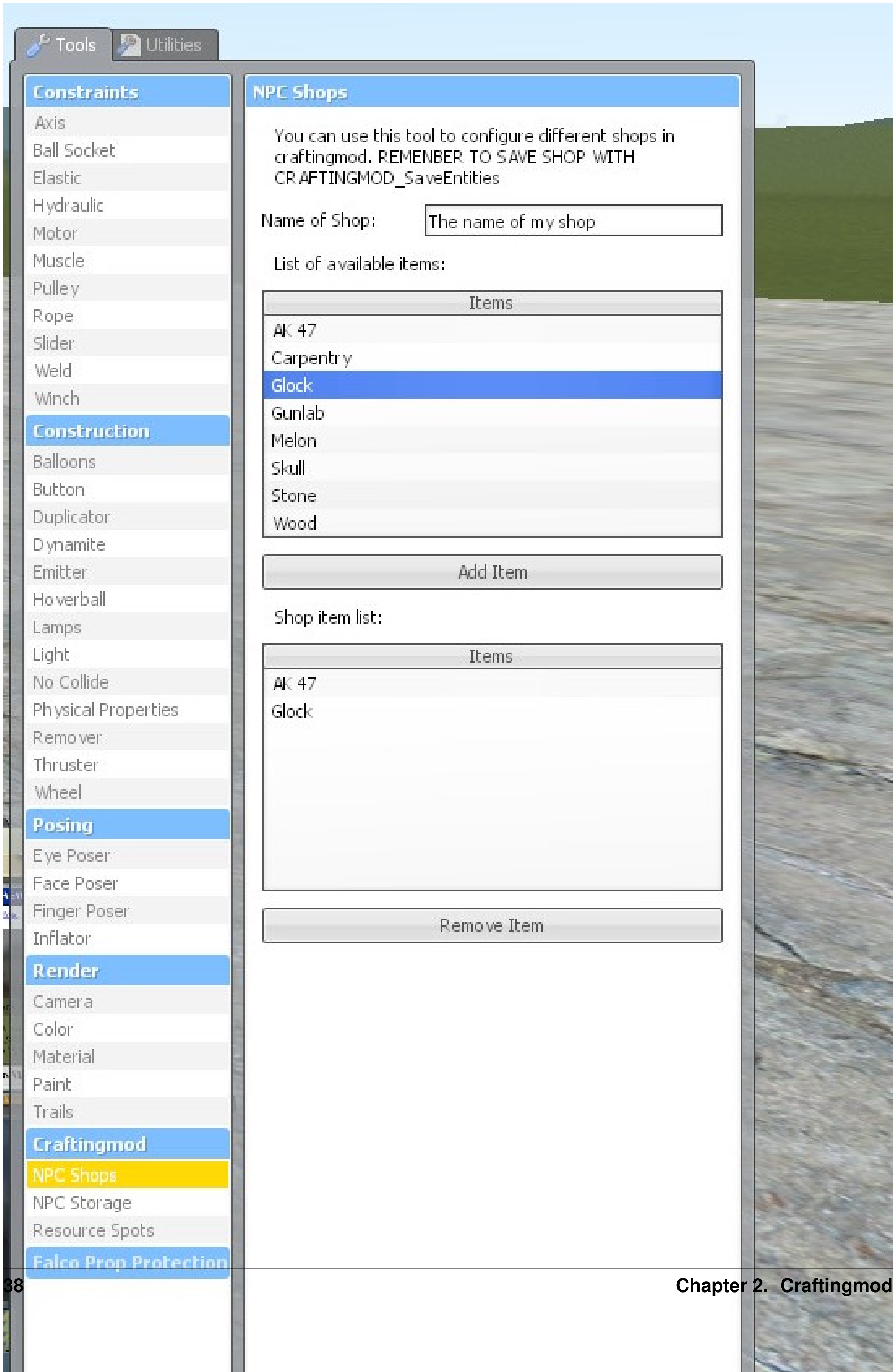
#### Creating custom shops

In craftingmod there is now a stool, that allows you to create custom shops, with the items that you prefer around the map.

In craftingmod you can now name your shop. This can be done at the stool in the Qmenu in the craftingmod category. Go inside the stool and then name the shop in the textentry. (See the next picture)

In the stool there are two listbox. You can move the items from the first one to the other. This way it allows you to fill out a custom list with the wanted items. All items are added automatically from the sh_* files.

As you see in this image, the I have named my shop "The name of the shop" and I have moved the items (Ak47 and Glock). The items can be removed by the remove button and added with the add button by selecting the different items.



You can use this tool to configure different shops in craftingmod. REMENBER TO SAVE SHOP WITH CRAFTINGMOD_SaveEntities

Name of Shop: The name of my shop

List of available items:

| Items |
|---|
| AK 47 |
| Carpentry |
| Glock |
| Gunlab |
| Melon |
| Skull |
| Stone |
| Wood |

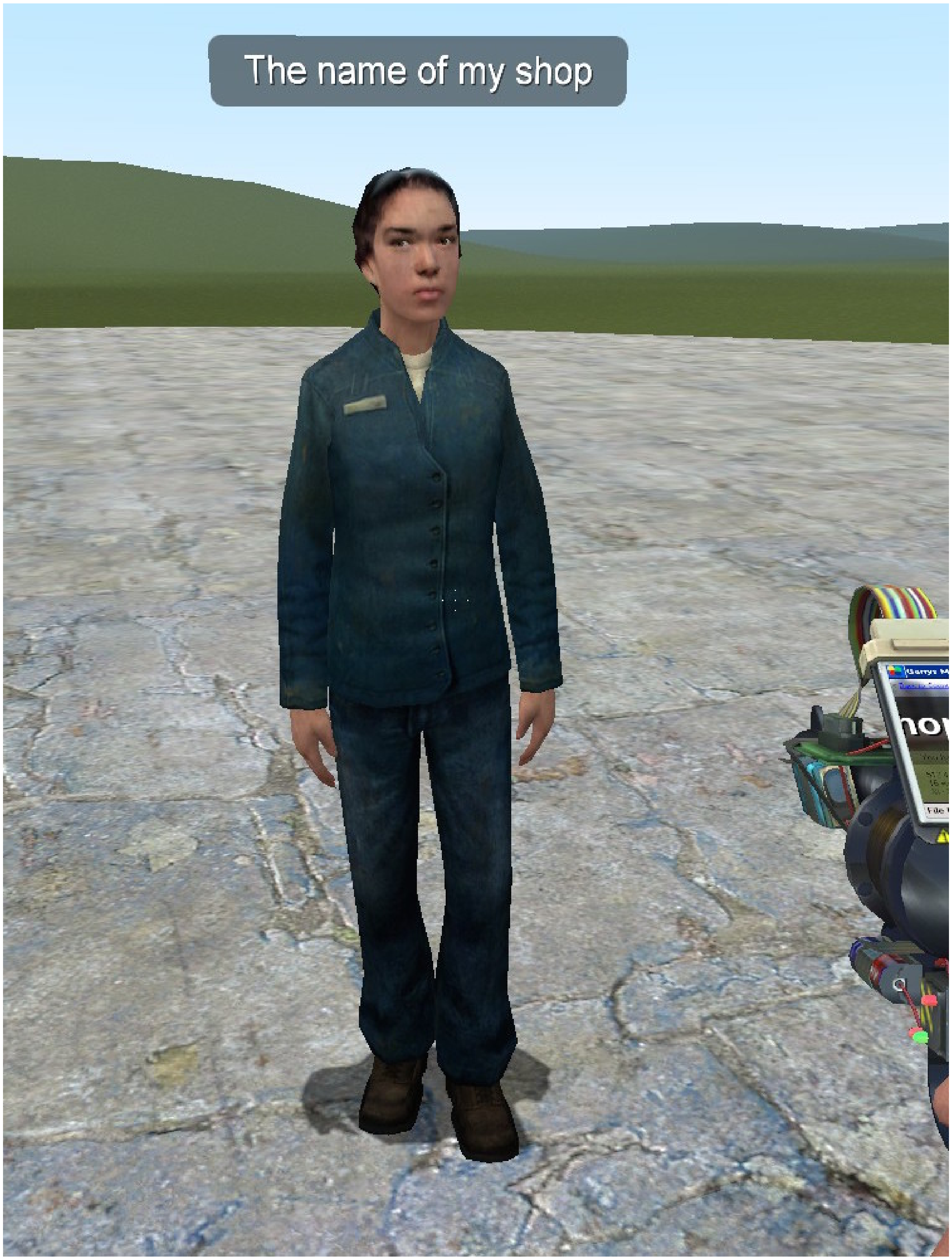
Shop item list:

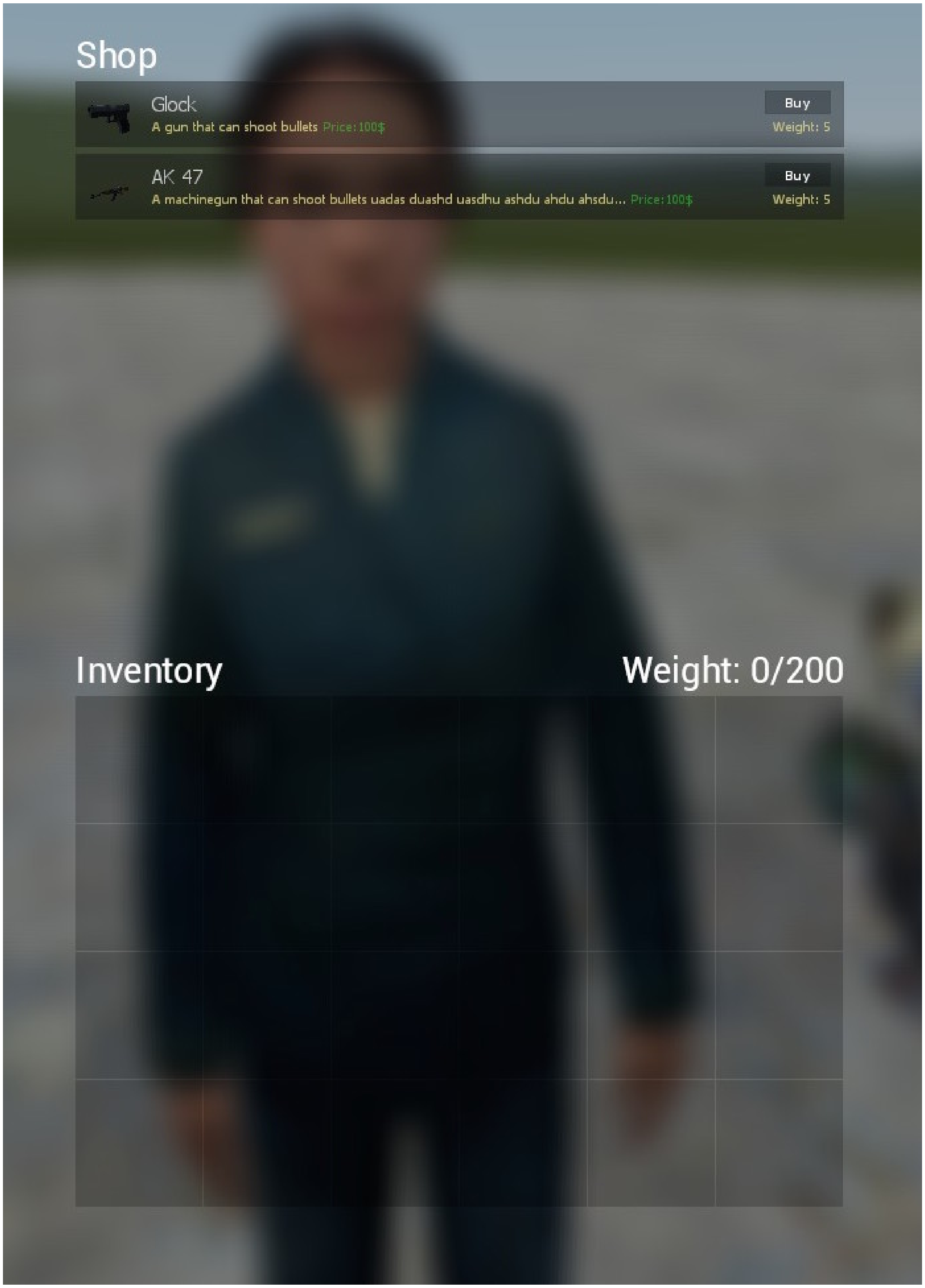
| Items |
|---|
| AK 47 |
| Glock |

Now you can spawn the shop (Left click). If you have forgotten any item then add it to the list and (right click) to refresh the shop. We have now a shop.

Remenber to save the shop by running “craftingmod_save_entities”.

If you want to remove the shop while not deleting other shops, then just use the remover tool and remove the npc and run again “craftingmod_save_entities”.


The name of my shop


Shop
Glock
A gun that can shoot bullets Price: 100$
Buy
Weight: 5
AK 47
A machinegun that can shoot bullets uadas duashd uasdhu ashdu ahdu ahsdu... Price: 100$
Buy
Weight: 5
Inventory
Weight: 0/200

You can use the shop restriction tool to make the shop only be able to let you sell specific items. This allows you to create a medic shop, miner shop and so on.

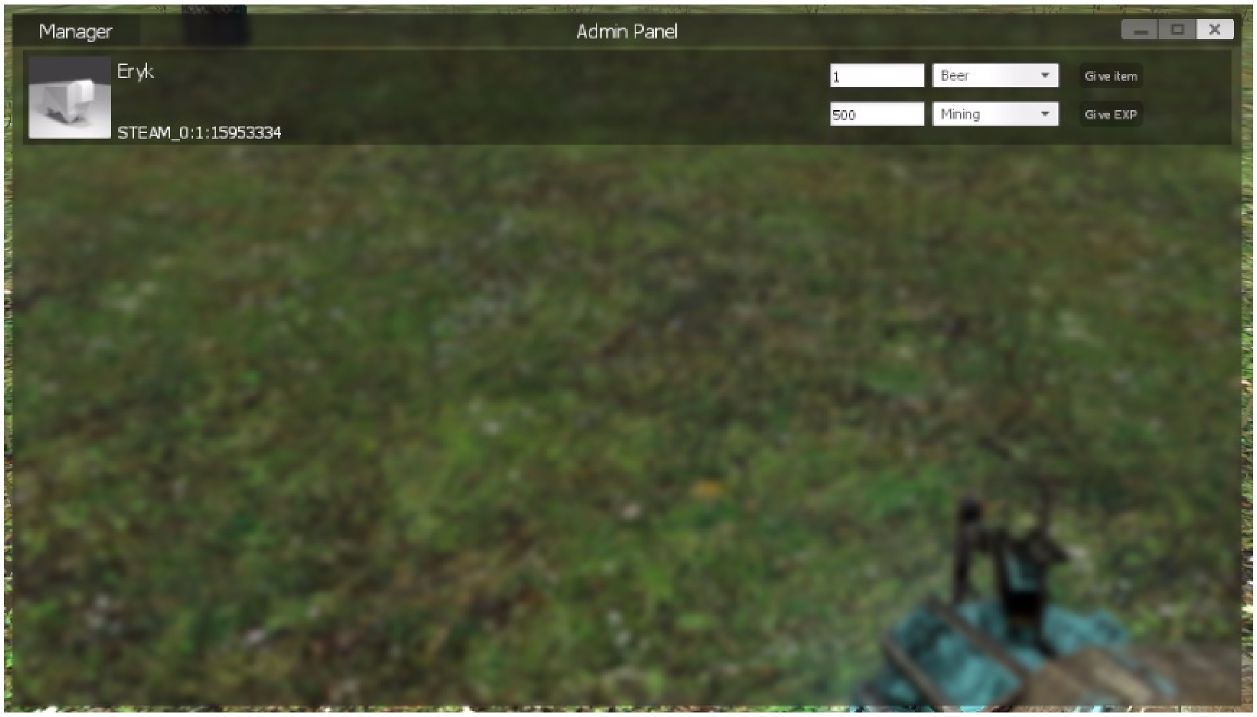
### 2.1.4 Admin Menu

#### Introduction

In craftingmod you can open up the admin panel if you're an admin. This panel helps you mostly with testing out new items that have been added. You can also use it for admin purposes as giving other players items or experience.

Its easy to use. It creates a panel for each player in game. With this you can then give each player an item or experience. Simple by putting an amount and then in a list you can choose the item or level and just press the button.

Have fun with the admin panel! (Any new content is added to the list on the admin panel)



### 2.1.5 Config

In the config you can change different variables to customize mechanics to your liking.

In the config there are two list. One to blacklist entities, so that they aren't able to be pickup. There is also a model list, which allows you to add new models for craftingmods npc model list.

## 2.2 References

They will soon come! **WIP**

### 2.2.1 AddExperience

You can use this code if you're a developer and you want to use some of craftingmods functions.

You can add experience with this command.

```
local steamID = ply:SteamID()

CRAFTINGMOD.User[steamID].Levels:AddExperience(string name, num experience, player␣
↪ply)
```

The name needs to be the exact name of the configured level. The experience is the experience amount and ply is the player var to send notifications back to the player

### 2.2.2 AddItem

You can use this code if you're a developer and you want to use some of craftingmods functions.

You can add items with this command.

```
local steamID = ply:SteamID()

CRAFTINGMOD.User[steamID].Inventory:AddItem(table tbl, num amount, player ply)
```

The table is one of the registered items/weapons/etc which you can give the player

CHAPTER 3

# Fishingmod

Welcome to the wiki site for fishingmod.

## 3.1 Guides

### 3.1.1 Getting started

#### Fishingrod

Left-Click: Throw / Retrieve Reload: Remove bobbler L-ALT: Open fishing inventory

With the fishingrod you can fish for different distances. You can load up throw power by holding down left click and then letting go when the wanted power is reached. If you dont want to retrieve the thrown bobbler, then you can use Reload to delete it. (Note that there is a 5 second cooldown) The inventory is used to equip baits and remove or drop items.

**PICTURES SOON**

#### Fishspot

This tool allows you to create custom fishspots. Player can now go and fish on the spots.

The tool has different variables that can be changed while using the tool. Most of the informations on how the tool is used, are adquired while in game. Remember that the radius has to only cover a water area.

I suggest that you run fishingmod_save_entities to save the fishspots. If there are any fishspots that you want to remove, then there is a list on the tool todo so. After any changes make sure of that you save.

**PICTURES SOON**

**WIP**

## 3.2 References

**WIP**