

Item Drop

version 2.0.1

requires: [uMMORPG Remastered](#) v2.36 and [TextMesh Pro](#)

TABLE OF CONTENTS:

ITEM DROP	3
Description	3
Preparing for installation	3
How to drop an item	5
How to pick up an item	5
Gold sharing	5
MONSTER LOOT	5
Description	5
COMMANDS	6
Description	6
ITEM RARITY	6
Description	6
ITEM LINKING	6
Description	6
How to link an item into chat	6
ITEM SAVE CYCLE	6
Description	6
CORRECTION OF ITEM POSITION AND ROTATION	7
Description	7
EXTRAS	7
Individual Loot	7
Gold Loss	7
Item Loss	7
SETUP WIZARD	8
Description	8
How to install	8
How to uninstall	9
INSTALLATION MANUAL (uMMORPG 3D Remastered)	10
Step 1	10
Step 2	11
Step 3	11

Step 4	12
Step 5	12
Step 6	13
Step 7	13
Step 8	14
Step 9	14
Step 10	14
Step 11	15
Step 12	16
Step 13	16
Step 14	17
Step 15	17
Step 16	17
Step 17	17
Step 18	17
Step 19	17
Step 20	17
Step 21	18
Step 22	18
Step 23	18
*OPTIONAL STEPS	18
Step 24	18
Step 25	18

ITEM DROP

Description

Item Drop has been designed as an addon for [uMMORPG](#), which means that uMMORPG is required and must be installed first as well as [TextMesh Pro](#) (in Unity's top menu: Window > Package Manager). Item Drop allows you to drop items on the ground and modifies the way monsters drop loot to be individual items on the ground that can be picked up.



Items can be picked up and dropped only if a player state is set as idle or movement.

Item labels are displayed in World Space with automating repositioning and are using object pooling.

Only 2 draw calls for all item labels on the screen.

Go to the [table of contents](#) to find out information about all of the features.

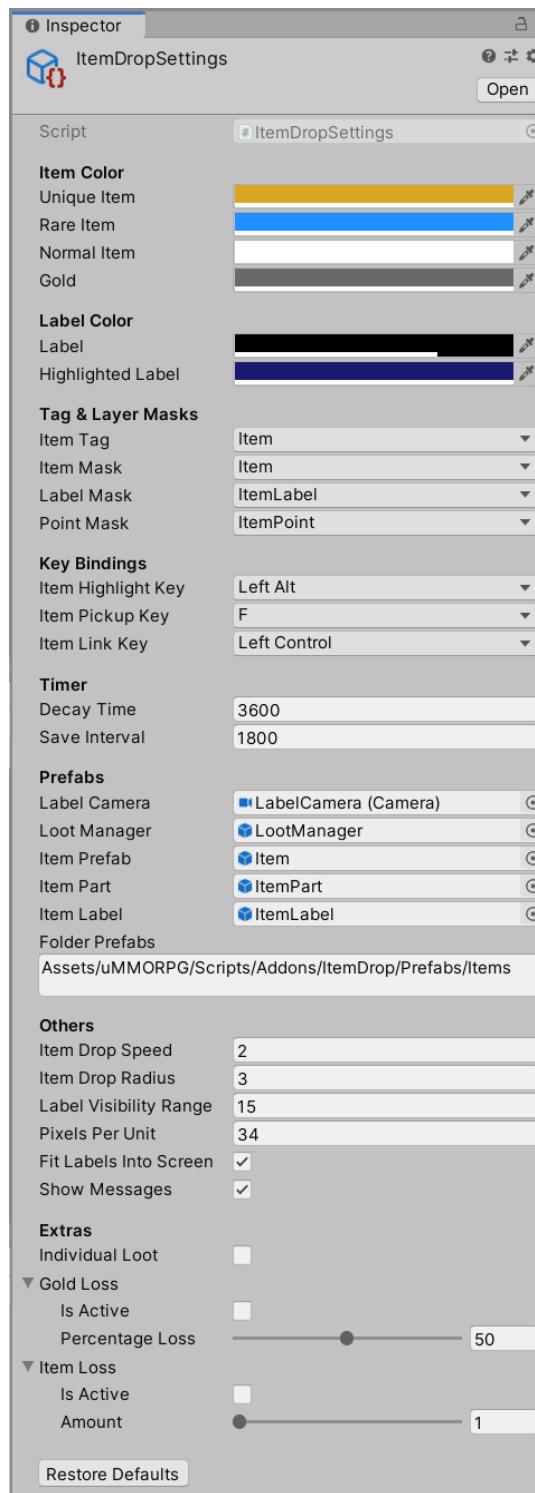
Preparing for installation

Make sure you have TextMesh Pro installed, if not you can find it in Unity's top menu: Window > Package Manager. Then click on Window > TextMeshPro > Import TMP Essential Resources.

Only then can you safely download this addon. After that use the [Setup Wizard](#) or install this addon manually as described in the [Installation Manual](#).

ItemDropSettings.asset stores all settings related to this addon. Configure it after the installation is complete.

The default settings can be restored by clicking on the **Restore Defaults** button.



How to drop an item

You can drop items from the inventory and equipment by just drag an item and drop it on the ground. Your character will try to drop the item in front of him. If this position is obstructed, it is chosen randomly around your character. If not, the item will be dropped normally. If the item destination is already occupied by another item, it will be placed on top of it up to a height of 1.25f (counting from the player's position to the control point on the last item). If this value is exceeded, the item will be randomly dropped around your character.

Pet and mount items cannot be dropped.

Items dropped by players are always visible to everyone.

How to pick up an item

Hold down the `LeftAlt` key to see all item names on the screen for items within range 15.

Hovering the cursor over an item's label will highlight it when all of them are forced to display by holding down the key.

Hovering the cursor over an item will highlight it in green and display the name and amount.

Click on an item or label to pick it up. You can also use the `F` key to pick the nearest item up within range 15.

Items can be picked up when you have enough space in your inventory, with the exception of Gold.

If the character is out of the pickup range, it will move closer automatically to pick the item up.

If items are placed on top of each other and one of them is taken from underneath, others will change their position.

Gold sharing

All gold picked up can be shared among nearby party members.

The amount of gold is calculated by the formula from uMMORPG.

Use the `P` key and turn on the `Gold Auto Share` option in the panel just like in uMMORPG.

MONSTER LOOT

Description

Loot will be dropped on the ground in a small distance, randomly where there is an available position for players on the NavMesh around a killed monster.

Each player sees the same loot and can pick it up. However, you can turn on the [Individual Loot](#) option.

Loot dropped by monsters is not saved. After a certain amount of time it is destroyed and disappears. 60 minutes by default (Decay Time).

COMMANDS

Description

If you want to save or delete all items instantly there are two functions for this in the PlayerItemDrop.cs script: CmdMsgSaveItems(); and CmdMsgDeleteItems(). These functions are intended for the admin and are not used anywhere by default.

ITEM RARITY

Description

Item name color based on the rarity of the item. There are three levels of item rarity: Basic, Rare and Unique.

Select the rarity (**Basic, Rare, Unique**) in the **ScriptableItem**.
To choose a color go to **ItemDropSeettings.asset** in the Inspector window.

You can always add more rarity levels as described in the [*OPTIONAL STEPS](#).

ITEM LINKING

Description

Each player can link an item into chat. The item name will be visible in the message in color based on the rarity of the item and marked in bold, placed between two brackets.

Hover the cursor over a linked item to see an instant pop up box showing information about it.

Writing brackets [and] are disabled by default in the message input area, the exception is the method described below.

How to link an item into chat

Hold down the **LeftCtrl** key and click on an item from the inventory. Item link will be added to the message input area of your chat, between two brackets. If an item link is already in the message input area and you will click on another item, that link will be replaced with a new one. This means that only one link is allowed for one message. Use the **Send** button and your item link will be already active in the sent message.

ITEM SAVE CYCLE

Description

Item is marked if the position is equal to its ending point (if it falls on the ground).

All items dropped by players that have been marked are saved every 30 minutes (Save Interval) and will be loaded from the database after server restart. After saving, the saved item is no longer marked, which means that it will not be taken into account during the next save cycle.

CORRECTION OF ITEM POSITION AND ROTATION

Description

Additional fields to set in the ScriptableItem: Offset X, Offset Y, Offset Z for position and Rot X, Rot Y, Rot Z for rotation.

They have been added to correct the standard models from uMMORPG. If you prepared your models the right way*, before import them to Unity engine, you should not need to change anything here but for the uMMORPG models it is necessary.

Everything is prepared in the script, so after the installation is completed, the position and rotation for each default item from uMMORPG will be automatically set appropriately.

Unity engine uses different axis than most standard 3d modeling software which can cause problems with scale and rotation, especially for new users.

**Scale Factor from your model should be 1 after import, blue arrow as front of your prefab, scale (1, 1, 1), rotation (0, 0, 0).*

EXTRAS

Individual Loot

All items dropped by monsters are assigned to the individual player. These items only appear on clients connected to the server and are automatically destroyed when the player logs out.

Gold Loss

Characters drop a percentage of their gold when they die. The lost gold remains on the server and will not be destroyed until someone picks it up.

Item Loss

Characters drop a random item of their inventory when they die. The lost item remains on the server and will not be destroyed until someone picks it up. You can choose the number of items to drop.

ItemDropSettings.asset stores all settings related to this addon. The default settings can be restored by clicking on the [Restore Defaults](#) button.

SETUP WIZARD

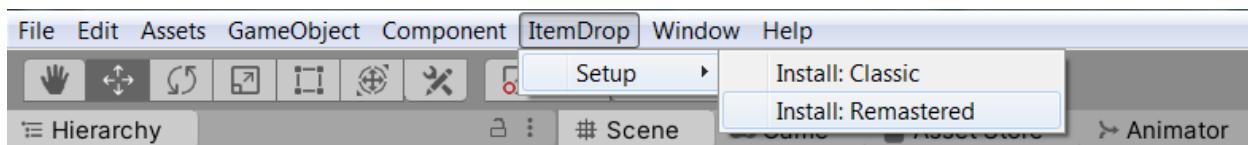
Description

Version 2.0.0 comes with the Setup Wizard. Now you can install everything in a quick and easy way.

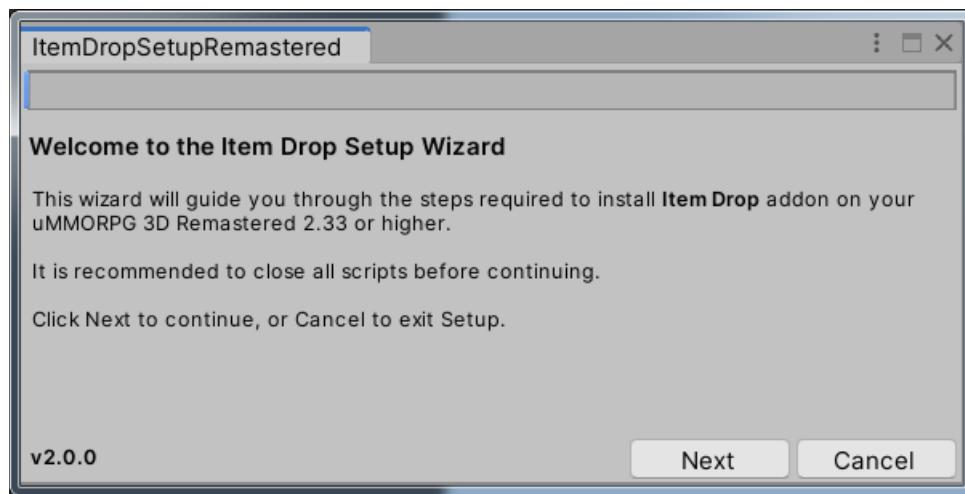
A few clicks to install the entire addon or uninstall as needed. Just click a button to replace or add code, components and prefabs. However, if you have previously made any changes that can conflict with the installer you must install this addon manually. See the [INSTALLATION MANUAL](#).

How to install

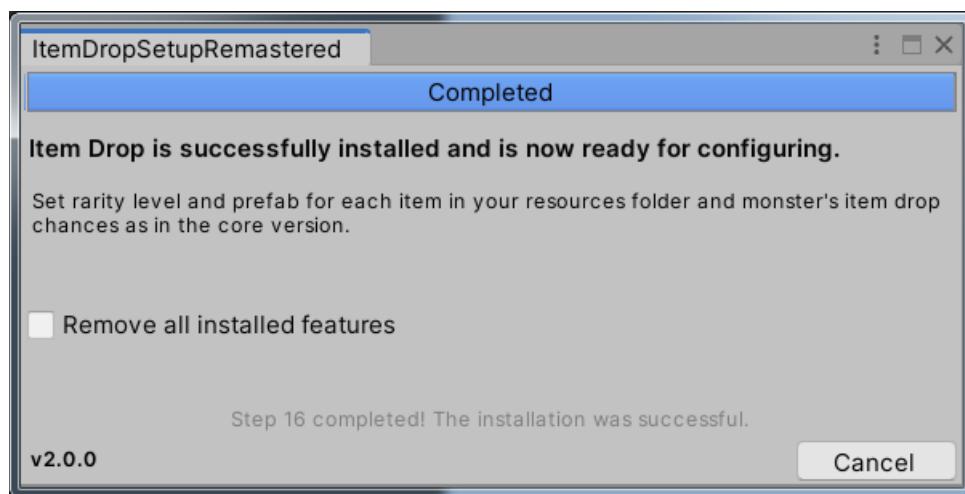
Unity's top menu: ItemDrop > Setup click and choose the remastered version of uMMORPG (Install: Remastered).



This wizard will guide you through the steps required to install Item Drop addon on your uMMORPG.



Follow the instructions displayed on the window and click **Finish** at the end. If everything works, you can skip the [INSTALLATION MANUAL](#).

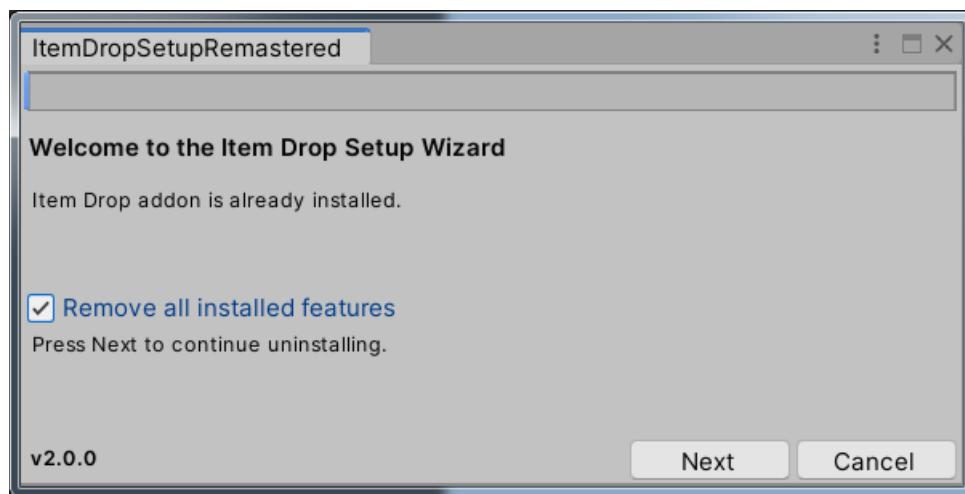


After the installation is completed, the ItemDropSettings.asset Inspector window will open automatically, where you can configure all the settings.

How to uninstall

Open the Setup Wizard in the same way as for installation. In Unity's top menu: ItemDrop > Setup click and choose the classic version of uMMORPG (Install: Remastered).

If this addon is already installed you can check “Remove all installed features”. Follow the instructions displayed on the window and click **Finish** at the end.



After uninstalling, you can safely remove the ItemDrop folder from your project:
Assets/uMMORPG/Scripts/Addons/ItemDrop.

INSTALLATION MANUAL (uMMORPG 3D Remastered)

Step 1

Edit the [PlayerCharacterControllerMovement.cs](#) Script in
“Assets/uMMORPG/Scripts/MovementSystems”

Add this coroutine

```
System.Collections.IEnumerator GoToItem(Vector3 destination, float stoppingDistance)
{
    Quaternion rotation = Quaternion.LookRotation(destination - transform.position);
    rotation.x = 0;
    rotation.z = 0;

    while (player.targetItem != null)
    {
        yield return AddonItemDrop.FixedUpdate;

        if (!player.IsMovementAllowed())
        {
            player.targetItem = null;
            yield break;
        }

        if (GetInputDirection() != Vector2.zero)
        {
            player.targetItem = null;
            yield break;
        }

        if (stoppingDistance >= Vector3.Distance(transform.position, destination))
            yield break;

        bool blocked = UnityEngine.AI.NavMesh.Raycast(transform.position, destination, out var hit, 1);
        if (!blocked)
        {
            float step = runSpeed * Time.deltaTime;
            transform.rotation = Quaternion.Slerp(transform.rotation, rotation, step);

            Vector2 inputDir = new Vector2(0, 1f);
            Vector3 desiredDir = GetDesiredDirection(inputDir);

            if (state == MoveState.IDLE) state = UpdateIDLE(inputDir, desiredDir);
            else if (state == MoveState.RUNNING) state = UpdateRUNNING(inputDir, desiredDir);
            else if (state == MoveState.AIRBORNE) state = UpdateAIRBORNE(inputDir, desiredDir);
            else if (state == MoveState.SWIMMING) state = UpdateSWIMMING(inputDir, desiredDir);
            else if (state == MoveState.MOUNTED) state = UpdateMOUNTED(inputDir, desiredDir);
            else if (state == MoveState.MOUNTED_AIRBORNE) state = UpdateMOUNTED_AIRBORNE(inputDir, desiredDir);
            else if (state == MoveState.MOUNTED_SWIMMING) state = UpdateMOUNTED_SWIMMING(inputDir, desiredDir);
            else if (state == MoveState.DEAD) state = UpdateDEAD(inputDir, desiredDir);
            else Debug.LogError("Unhandled Movement State: " + state);

            if (!controller.isGrounded) lastFall = controller.velocity;

            controller.Move(moveDir * Time.fixedDeltaTime);
            velocity = controller.velocity;

            CmdFixedMove(new Move(route, state, transform.position, transform.rotation.eulerAngles.y));

            float runCycle = Mathf.Repeat(animator.GetCurrentAnimatorStateInfo(0).normalizedTime +
runCycleLegOffset, 1);
            jumpleg = runCycle < 0.5f ? 1 : -1;

            jumpKeyPressed = false;
        }
        else
        {
            player.indicator?.SetViaParent(player.transform);
            player.targetItem = null;
            yield break;
        }
    }
}
```

```
public override void Navigate(Vector3 destination, float  
stoppingDistance)
```

Add these lines

```
if (player.targetItem != null)  
{  
    StartCoroutine(GoToItem(destination));  
}
```

```
void Awake()
```

Add this line

```
viewBlockingLayers = AddonItemDrop.DisableLayers(viewBlockingLayers);
```

Step 2

Edit the [PlayerNavMeshMovement.cs](#) Script in “Assets/uMMORPG/Scripts/MovementSystems”

```
void MoveWASD()
```

Add these lines above // create input vector, normalize in case of diagonal movement

```
if (player.targetItem != null)  
{  
    player.targetItem = null;  
}
```

```
void MoveClick()
```

Add these lines above Ray ray = cam.ScreenPointToRay(Input.mousePosition);

```
if (player.lockRaycast)  
    return;
```

Find this line

```
else Navigate(bestDestination, 0);
```

and insert below

```
if (player.targetItem != null)  
{  
    player.targetItem = null;  
}
```

Step 3

Edit the [Monster.cs](#) Script in “Assets/uMMORPG/Scripts”

```
protected override void UpdateClient()
```

Add these lines at the end

```
#if ITEM_DROP_R  
    UpdateClient_ItemDrop();  
#endif
```

```
protected override void OnInteract()
```

Comment these lines

```
/*
else if (health.current == 0 &&
    Utils.ClosestDistance(player, this) <= player.interactionRange &&
    inventory.HasLoot())
{
    UILoot.singleton.Show();
}
*/
```

Step 4

Edit the [MonsterInventory.cs](#) Script in “Assets/uMMORPG/Scripts”

Change public class to public partial class

```
public partial class MonsterInventory
```

Find and comment the following lines

```
/*
[ClientCallback]
void Update()
{
    // show Loot indicator on clients while it still has items
    if (LootIndicator != null)
    {
        // only set active once. we don't want to reset the particle
        // system all the time.
        bool hasLoot = HasLoot();
        if (hasLoot && !LootIndicator.isPlaying)
            LootIndicator.Play();
        else if (!hasLoot && LootIndicator.isPlaying)
            LootIndicator.Stop();
    }
}
*/
```

```
public void OnDeath()
```

Add these lines at the end

```
#if ITEM_DROP_R
    OnDeath_ItemDrop();
#endif
```

Step 5

Edit the [Player.cs](#) Script in “Assets/uMMORPG/Scripts”

```
protected override void UpdateClient()
```

Add this line at the beginning

```
UpdateClient_HighlightLoot();
```

Comment these lines in if (state == "IDLE" || state == "MOVING") part

```
/*
if (Input.GetKeyDown(cancelActionKey))
{
    // reset Locally because we use rubberband movement
    movement.Reset();
    CmdCancelAction();
}
*/
```

and insert below

```
if (Input.GetKeyDown(cancelActionKey))
{
    CancelAction();
}

if (Input.GetKeyDown(ItemDropSettings.Settings.itemPickupKey))
{
    if (!Utils.IsCursorOverUserInterface() && Input.touchCount <= 1)
    {
        FindNearestItem();
    }
}

if (EventMoveEnd())
{
    if (targetItem != null)
    {
        // check distance between character and target item
        if (AddonItemDrop.ClosestDistance(this, targetItem) <= 1.5f)
        {
            ItemPickup(targetItem);
        }
    }
}
```

```
public override void OnDeath()
```

Add this line at the end

```
OnDeath_ItemDrop();
```

Step 6

Edit the [PlayerEquipment.cs](#) Script in “Assets/uMMORPG/Scripts”

Change public class to public partial class

```
public partial class PlayerEquipment
```

Step 7

Edit the [PlayerInventory.cs](#) Script in “Assets/uMMORPG/Scripts”

Change public class to public partial class

```
public partial class PlayerInventory
```

Step 8

Edit the [ScriptableItem.cs](#) Script in “Assets/uMMORPG/Scripts”

```
public virtual string Tooltip()
```

Comment this line

```
//tip.Replace("{NAME}", name);
```

and insert below

```
tip.Replace("{NAME}", GetTitle());
```

Step 9

Edit the [Instance.cs](#) Script in “Assets/uMMORPG/Scripts”

```
void DestroyAllNetworkIdentitiesInInstanceBounds()
```

Find these lines

```
if (identity != null)
    NetworkServer.Destroy(identity.gameObject);
```

and change to

```
if (identity != null)
{
    if (identity.CompareTag(ItemDropSettings.Settings.itemTag))
    {
        if (identity.TryGetComponent(out ItemDrop item))
        {
            AddonItemDrop.DeleteItem(item.uniqueId);
        }
    }
    NetworkServer.Destroy(identity.gameObject);
}
```

Step 10

Edit the [UIChatEntry.cs](#) Script in “Assets/uMMORPG/Scripts/_UI”

Add the following lines

```
public UIShowToolTip tooltip;

void Awake()
{
    tooltip = GetComponent<UIShowToolTip>();
}
```

```
public void OnPointerEnter(PointerEventData pointerEventData)
```

Add the following lines

```
string stringToCheck = message.Construct();
int start = stringToCheck.IndexOf("[");
int stop = stringToCheck.IndexOf("]");
if (stop > -1)
{
    string output = stringToCheck.Substring(start + 1, stop - start - 1);
    int key = output.GetStableHashCode();

    if (ScriptableItem.All.TryGetValue(key, out var data))
    {
```

```

        Item item = new Item(data);
        tooltip.text = new ItemSlot(item).ToolTip();
    }
}

```

Step 11

Edit the `UIChat.cs` Script in “Assets/uMMORPG/Scripts/_UI”

Add the following lines

```

public bool itemLink;

void Start()
{
    messageInput.onValidateInput += delegate (string input, int charIndex, char addedChar) { return
MyValidate(addedChar); };
}

char MyValidate(char charToValidate)
{
    if (charToValidate == '[' || ']' == charToValidate)
    {
        if (!itemLink)
        {
            charToValidate = '\0';
        }
    }
    return charToValidate;
}

```

```
public void AddMessage(ChatMessage message)
```

Comment these lines

```
//go.GetComponent<Text>().text = message.Construct();
//go.GetComponent<UIChatEntry>().message = message;
```

and insert below

```

Text chatMessage = go.GetComponent<Text>();
chatMessage.text = message.Construct();

UIChatEntry chatEntry = go.GetComponent<UIChatEntry>();
chatEntry.message = message;

string stringToCheck = message.Construct();
int start = stringToCheck.IndexOf("[");
int stop = stringToCheck.IndexOf("]");
if (stop > -1)
{
    string output = stringToCheck.Substring(start + 1, stop - start - 1);
    int key = output.GetStableHashCode();

    if (ScriptableItem.All.TryGetValue(key, out var data))
    {
        chatMessage.text = chatMessage.text.Replace(output, data.GetTitle());
        chatEntry.tooltip.enabled = true;
    }
}

```

Step 12

Edit the `UIInventory.cs` Script in “Assets/uMMORPG/Scripts/_UI”

```
void Update()
```

Find and comment these lines

```
/*
if (itemSlot.item.data is UsableItem usable &&
    usable.CanUse(player, icopy))
    player.CmdUseInventoryItem(icopy);
*/
```

and insert below

```
if (Input.GetKeyDown(ItemDropSettings.Settings.itemLinkKey))
{
    UIChat.singleton.itemLink = true;
    Item item = player.inventory[icopy].item;

    string stringToCheck = UIChat.singleton.messageInput.text;
    int start = stringToCheck.IndexOf("[");
    int stop = stringToCheck.IndexOf("]");
    if (stop > -1)
    {
        string output = stringToCheck.Substring(start + 1, stop - start - 1);
        stringToCheck = stringToCheck.Replace(output, item.name);
    }
    else
    {
        stringToCheck += $"[{item.name}]";
    }
    UIChat.singleton.messageInput.text = stringToCheck;
    UIChat.singleton.itemLink = false;
}
else
{
    if (itemSlot.item.data is UsableItem usable && usable.CanUse(player, icopy))
        player.CmdUseInventoryItem(icopy);
}
```

Step 13

Edit the following chat prefabs in “Assets/uMMORPG/Prefabs/UI”

SlotChatMessageGuild.prefab

SlotChatMessageInfo.prefab

SlotChatMessageLocal.prefab

SlotChatMessageParty.prefab

SlotChatMessageWhisper.prefab

Add the `UIShowTooltip.cs` component in the Inspector window, and then disable this component. The `Tooltip Prefab` field is empty, you have to assign a prefab. To do this click `Select GameObject` and change the category to `Assets`. Find and click on the `Tooltip.prefab`.

Step 14

Edit the `Item.prefab` in “Assets/uMMORPG/Scripts/Addons/ItemDrop/Prefabs”

Click on the Inspector window Tag list, then Add Tag `Item` and set it on this prefab.

Create three new Layers. Click on the Inspector window Layer list, then Add Layer `Item`, `ItemPoint` and `ItemLabel`.

Set Layer `Item` on this prefab (select “No, this object only”).

Open this prefab and in Prefab Mode click on the child object and set Layer `ItemPoint` for it.

Drag and drop `Item.prefab` from “Assets/uMMORPG/Scripts/Addons/ItemDrop/Prefabs” to the NetworkManager Registered Spawnable Prefabs.

Step 15

Edit the `ItemPart.prefab` in “Assets/uMMORPG/Scripts/Addons/ItemDrop/Prefabs”

Set Layer `Item` on this prefab.

Step 16

Edit the `ItemLabel.prefab` in “Assets/uMMORPG/Scripts/Addons/ItemDrop/Prefabs”

Set Layer `ItemLabel` on this prefab (select “Yes, change children”).

Step 17

Drag and drop `LootManager.prefab` from “Assets/uMMORPG/Scripts/Addons/ItemDrop/Prefabs” to the World scene.

Step 18

Edit the `Loot` in “World/Canvas/Loot” on the World scene.

Disable the `UILoot.cs` component.

Step 19

Edit the `MiniMap Camera`’s Inspector window/Camera/Culling Mask on the World scene.

Uncheck `Item`, `ItemPoint` and `ItemLabel` layers from the Culling Mask list, then make sure `Item` and `ItemPoint` are selected in your main camera while `ItemLabel` must be unchecked.

Step 20

Edit the Project Settings from Unity’s main menu
“Edit/Project Settings/Player/Configuration/Scripting Define Symbols”

Insert `;ITEM_DROP_R` at the end and push the `Enter` button.

Now all lines of code related with the remastered version are active.

Step 21

Configure the `ItemDropSettings.asset` in
“Assets/uMMORPG/Scripts/Addons/ItemDrop/Resources”

Click the `Restore Defaults` button (at the bottom in the Inspector window) to set Tags and Layers you added in step 16 and set all prefabs for the default items from uMMORPG. Your project will also be saved.

`ItemDropSettings.asset` stores all settings related to this addon. You can configure it however you want.

You will also find three additional options here:

Individual Loot All items dropped by monsters are assigned to the individual player. These items do not have a unique ID and only appear on clients connected to the server and are automatically destroyed when the player logs out.

Gold Loss Characters drop a percentage of their gold when they die. The lost gold remains on the server, has a unique ID and will not be destroyed until someone picks it up.

Item Loss Characters drop a random item of their inventory when they die. The lost item remains on the server, has a unique ID and will not be destroyed until someone picks it up. You can choose the number of items to drop.

Step 22

Set rarity level and prefab for each Scriptable Item in “Assets/uMMORPG/Resources/Items”.

Step 23

Edit all monster prefabs in “Assets/uMMORPG/Prefabs/Entities/Monsters”

Set `Loot Gold min` and `Loot Gold max` as in the core version. Add items, but remember to not add the `Gold` Scriptable Item!

*OPTIONAL STEPS

You can easily add more rarity levels.

Step 24

Edit the `ItemDropSettings.asset` in “Assets/uMMORPG/Scripts/Addons/ItemDrop/Resources”

Add this line

```
public Color magicItem; // choose a color in the Inspector window
```

Step 25

Edit the `AddonItemDrop.cs` Script in “Assets/uMMORPG/Scripts/Addons/ItemDrop/Scripts”

```
public enum Rarity
```

Insert here

```
Magic,
```

```
public string GetTitle(int stack = 1)
```

Insert the following lines into the same place as other colors

```
case Rarity.Magic: color = ItemDropSettings.Settings.magicItem; break;
```