

To differentiate the two crafting systems in this asset. “Simple” crafting refers to a crafting system that simply tells you what ingredients you need to craft each recipe, then you’re able to craft automatically with a single click. “Grid” crafting refers to a Minecraft style crafting system where you must place ingredients into a grid in a specific pattern in order to craft the desired item.

Getting Started

If you want to use the crafting system as-is, then all you have to do is drag the Crafting Table (Simple) prefab (*Prefabs/Blocks...*) into the scene somewhere the player can reach. As long as the player is tagged Player and is within proximity of the Crafting Table, you can press Tab to open up the crafting menu. This works best if you also have the Inventory dragged into the scene.

The crafting table is nothing but a few lines of code that turn on the Crafting Menu (Simple) prefab that’s attached as a child. This is where all the actual crafting happens.

To add recipes to the crafting menu, you’ll just need to drag them in from the Project tab into the list of Recipes within the Crafting Menu Simple component. Once the game starts, that’s when the crafting menu instantiates each recipe button and all the ingredient slots.

Recipes

Recipes are set up as Scriptable Objects, so inside the project tab you’ll be able to create a new recipe by right clicking and selecting Create > EZ Inventory > Recipe Simple.

Simple recipes require ingredients and a result.

Ingredients: size = how many unique ingredients. Then for each ingredient specify an Item (check Items guide) and how many. If we want a sword that requires 1 wood and 2 stone, that would be 2 ingredients.

Result: which Item the recipe will make.

Result Amount: how many of the result item to create.

Once you have the recipes you want, you’ll have to go into the Crafting Table or Crafting Menu and add the recipes manually.

Placing Inside Inventory

It’s very likely you might want to have the crafting menu as a permanent inclusion in the Inventory. This is very easy. Just drag the Crafting Menu (Simple) prefab (*Prefabs/UI...*) to be a child of Inventory Canvas > Inventory Main.

Customizing Layout

For info on changing the UI sprites to your own, check out the [Changing UI](#) guide.

Since the crafting menu is set up to instantiate all the recipes and ingredient slots, it might be a bit daunting to customize it yourself without diving into the code.

There are two very important variables for customizing the crafting menu: Recipes Parent and Ingredients Parent. These will be the parents for everything instantiated, so it's important to set them up correctly.

If you look at the child "Scroll Rect Image" you'll see how the recipes are currently set up using a Scroll Rect and Horizontal Layout Group. This is important to instantiating everything in an organised way. The child "Ingredients" is similar but uses a Grid Layout Group. You can change these however you want to fit your game, but the most important part is using Layout Groups.

Find and use all recipes automatically

By default, crafting menus are set up to use only the recipes you manually drag in to the list of recipes. This is to allow for the option of having, say a crafting table for armor, a crafting table for weapons, and so on. But maybe you'd prefer a crafting table to have access to all recipes; well we can do this pretty easily by changing up CraftingMenuSimple.cs slightly. Add the following code to the top of Awake() inside CraftingMenuSimple.cs.

```
recipes = new List<RecipeSimple>();
string[] assetNames = AssetDatabase.FindAssets("CRS_");
foreach (string assetName in assetNames)
{
    RecipeSimple recipe = AssetDatabase.LoadAssetAtPath<RecipeSimple>(AssetDatabase.GUIDToAssetPath(assetName));
    recipes.Add(recipe);
}
```

This is copied directly from CraftingMenuGrid.cs if you want to use that for reference, but here I've changed the names and types to better fit the current task.

Basically this bit of code will clear the current list of recipes, then with AssetDatabase it will find all RecipeSimple files that begin with "CRS_" (you can use whatever handler you want here, but I chose CRS for Crafting Recipe Simple).

All you have to do after putting this code in is to make sure you have "CRS_" at the start of every recipe's file name.