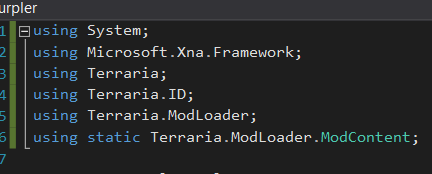
Terreria Mod Reference

Parts of a Mod code file:

## Using statements

At the top of the file the ‘using’ statements allow you to use other code in your code. This is called a dependency.



In this code file, you can access any non private or internal code defined in the namespaces above.

What is a namespace?

A fancy word that lets you organize your code into logical pieces. For instance



The namespace of my mod ‘PurpleNurpler’ would contain any and all code for this mod. As you can see in the using statements, you could make this even more granular or detailed. I could have a namespace called PurpleNurpler.Swords to contain all of my sword mods. I could get even more detailed and create something like namespace PurpleNurpler.Mods.ModItems.MeleeWeapons to really break out each part of my mod. What does this do? The only purpose is to allow you to organize the code into well defined parts. If you create a big mod that does a lot you will want to do this, but it isn’t required to write the mod.

What is a private or internal code?

In C# you can hide or prevent your code from being seen or (easily) used by other code. There are different reasons for this, but for a mod there isn’t any reason to use anything other than the public access modifier most of the time. You will see the access modifier on variables, properties and methods.

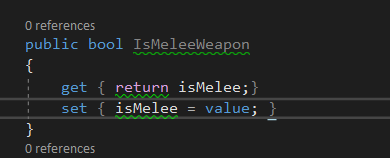
What is a variable?

In your mod you may see something that looks like this:



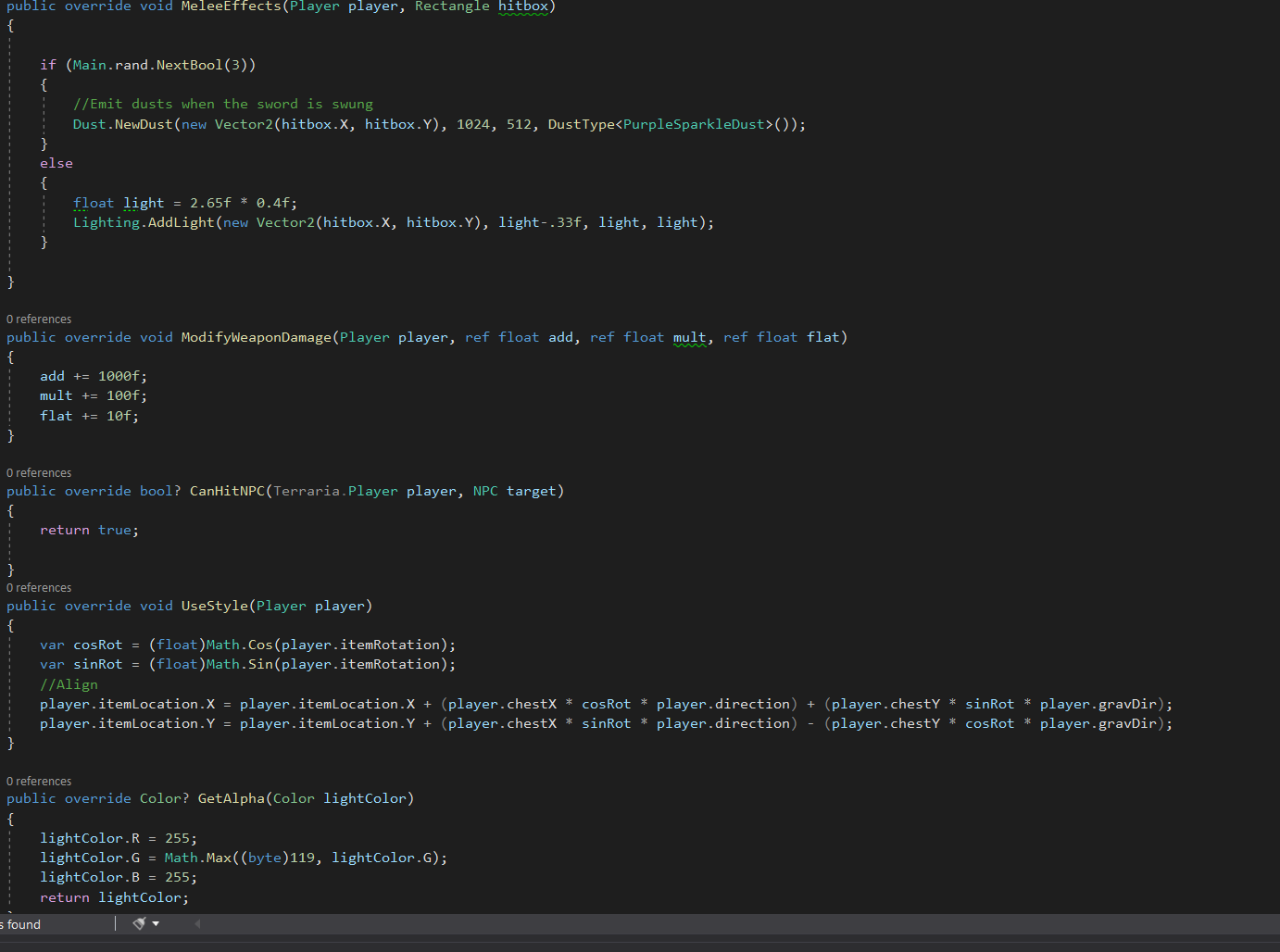
In this case isMelee is a variable that holds a true or false value(that is what bool means- short for Boolean). The variable is private which means its only visible in the current code file. And it is set to false.

What is a property?



Properties often contain the value of private variables and allow you to manipulate their value when they are retrieved or set. For a mod, there is not much difference between a property and a variable, but as you learn more coding you will see why properties exist and why they are important to use.

What is a Method?



A Method is what most people would look at and call ‘code’. It is a function that will take some input and modify and sometimes return the value it creates or modifies. A method can be defined to take no inputs as well. A simple method that counts to 10 would look like this:

public void CountToTen()

{

int i = 1;

while(i < 10)

{

i +=1;

}

}

The parts of that method are first the access modifier (public), the return type(void- that means no value is returned) and the method name (CountToTen). The open and closed parenthesis mean there are no values passed into the method. The first thing the method does is define a variable called i that holds an integer value and assigns the value of 1 to it. The next thing that happens is a loop is created by the ‘while’ keyword. What this word means is that everything below the while statement will run until the condition in the open and closed parenthesis is true. The first time the while statement is evaluated the variable i = 1 which is less than 10, so the += 1 means that i will be incremented to 2. At that point the first loop is complete, so the while statement will see if i is still less than 10. This process will repeat until the value is incremented to 10. After that the method is finished executing and will return back to the method that called CountToTen(). This part may be tricky to grasp at first but stick with it and you will get there.