**GAMEDEVREJECTS:**

**C# Programming in Unity**

**Orlando Unity3d Development Group**

**Date: 18-Oct-21**

**All Copyright GAMEDEVREJECTS©2021**

**Contents**

[**3.1.0** **C# Programming in Unity** 3](#_Toc85433659)

[**3.1.1** **Why C# in Unity ?** 3](#_Toc85433660)

[**3.1.2** **What is C# ?** 3](#_Toc85433661)

[**3.1.3** **Programming terms** 3](#_Toc85433662)

[**3.1.4** **The Unity Package Manager** 3](#_Toc85433663)

[**3.1.5** **The Unity Asset Store** 4](#_Toc85433664)

[**3.1.6** **Prefabs** 4](#_Toc85433665)

[**Glossary** 5](#_Toc85433666)

[**Resources** 6](#_Toc85433667)

# **C# Programming in Unity**

The objectives:

* **Create C# Script, explores its structure & syntax**
* **Declare, initialize, execute custom variables, methods/functions to reference classes and/or components for the behaviour**
* **Manipulate behaviors through input from an input device.**

# **Why C# in Unity ?**

So why do we use C#. I believe the reason why Unity decided to use C# was because it is an easy language to learn, C# is a lot more intuitive in a way compared to C++, and it’s powerful, it completes a lot of functionality in the background for you i.e. you don’t need to know how they work, you just need to know it works.

# **What is C# ?**

In a nutshell C# is

* **an object-oriented programming language**
* OOP is a programming design pattern where we structure all the program code into "reusable" code blocks called classes and objects. By organizing the code into groupings of classes and objects, you’re not restricted to running your code in line-by-line sequential order, where you have to execute the code on Line1 first, before Line 2, before Line 3, before line 100 and so on. Object Oriented Programming therefore makes it far easier to debug and maintain the code base.
* OOP uses the principles of **Abstraction, Polymorphism, Encapsulation, and Inheritance (APEI)**
* **is another component you add to the Game Object in your Scene.**
  + 1. **Programming terms**
* **Classes**
* **Objects**
* **Abstraction**
* **Polymorphism**
* **Encapsulation**
* **Inheritance**

**[ 6:51 ]**

* + 1. **The Unity Package Manager**

**The Unity Package Manager –** is used to **list, track, view, install, remove, or update assets, plugins, built-in assets,** linked to your Unity Id, and downloaded stored on your PC, and imported to your specific project.

Now, at this stage we recommend you just stick to the official release’s tags, because it’s just going to be safer !

* + 1. **The Unity Asset Store**

To access the Unity Asset store is Unity’s online marketplace for 3d models, assets, plugins etc.

To download any Unity Assets from the Unity Asset Store, go online. Click the link below

For now, make sure you select Model, which contains a 3d model, it contains all the components, in your import settings, because you still have a lot of geometry options. as to what you want to use. For demonstration purposes, we’ll just to stick to selecting all for now. Note when we add this 3d model to the hierarchy panel, it is now a Game Object an instance, a copy of the 3d model in our scene.

This model is an .FBX model. Please be advised Unity only accepts .FBX, .OBJ models. The biggest difference between an .OBJ and an .FBX is that .OBJ’s are usually imported to be static objects, that doesn’t have a rig applied like a character whilst an .FBX will support a deformation, and information. You choose which one you prefer.

It also supports are Software 3d software like blender .blend files because blender has a .fbx convertor. In the background what blender does is convert your .blend file into .fbx. Which is great, because what you can do is update your 3d model in blender then it will automatically update in Unity. Though, you must be careful especially if you need embed.

* + 1. **Prefabs**

Prefab is a template or a copy for a game object. For situations we need to create multiple instances of our game object or we don’t want to create the gameobject from scratch in our hierarchy. Once a prefab is created you literally drag and drop or import into the Hierarchy and view it in the Scene View.

/ IN 1:37

/END

# **Glossary**

Package Manager is used to track, view, install remove or update **assets, plugins, built-in assets,** in your Unity Project

# **Resources**

Scripting API: MonoBehaviour.StartCoroutine - Unity – Manual

<https://docs.unity3d.com/ScriptReference/MonoBehaviour.StartCoroutine.html>

/End