Table of Contents

Factions Extension Documentation

Add factions and relations in your game using C++ or Blueprints

This plugin is for Unreal Engine 4 and has support for versions **4.20**, **4.19** and **4.18**.
You can download this **Test project** to see and test the API

Introduction

What are "Factions"

When we say Faction we refer to a **group of entities or actors** that share something in common.

Almost every single type of game uses factions in different ways. For example, in Shooter games there will be enemies and friends, in RTS games every player will be a faction by itself, and in Open-World games you will have factions fighting each other while you run around.

This plugin fulfills the needs of this feature in UE4 with a very flexible tool that will make the implementation, editing and design of your own factions a 5 minutes thing.

"What" can have a faction?

Everything!... well not everything, but pretty much. **All actors of any type can have a faction.**

Installation

Manually

This are the general steps for installing the plugin into your project:

1. Download the last release from here
Make sure you download the same version than your project

2. Extract the folder “FactionsExtension” into the **Plugins folder** of your existing project (e.g "MyProject/Plugins")

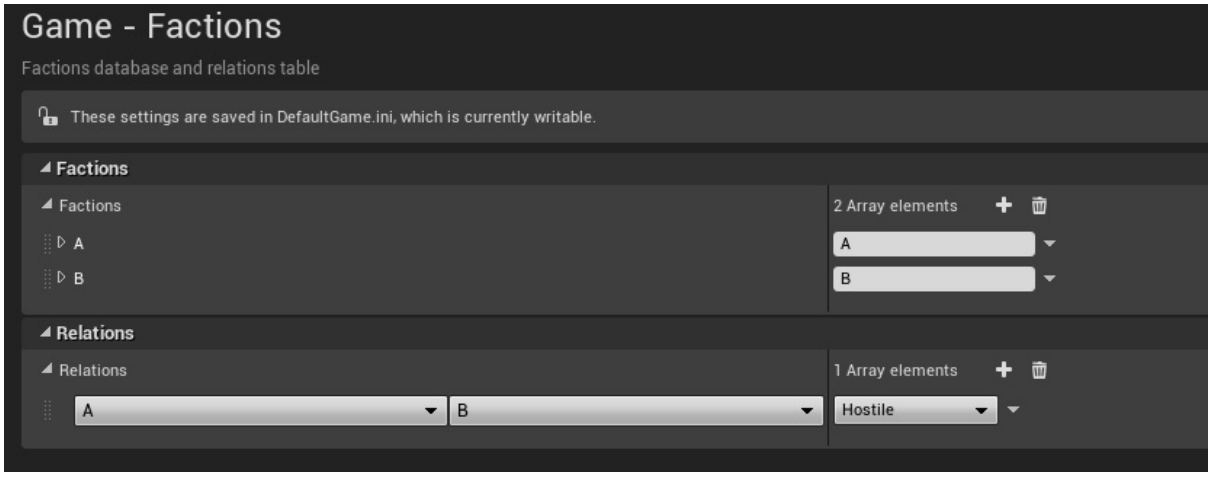
2. Done! You can now open the project

From Marketplace

Install from the launcher! AVAILABLE HERE

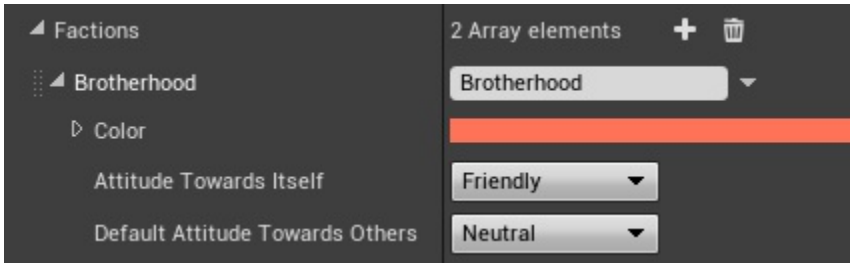
Usage

Factions can be edited from *Project Settings*, inside *Game* category, *Factions* tab.



Editing Factions

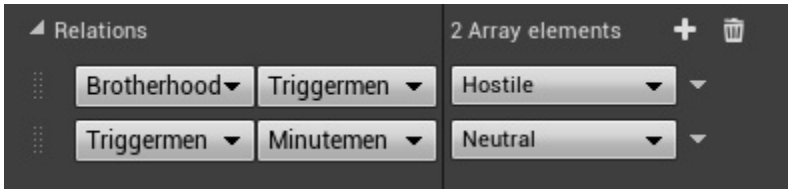
Factions are defined as an array to be edited and every faction will have its own properties like color or default attitudes.



This information is available on runtime though

Editing Relations

Relations define how will two factions interact to each other.



By default there are three possible attitudes: **Hostile**, **Neutral** and **Friendly**

Implementation

Blueprints

*Check the **Test Project** for a detailed example*

Adding the interface

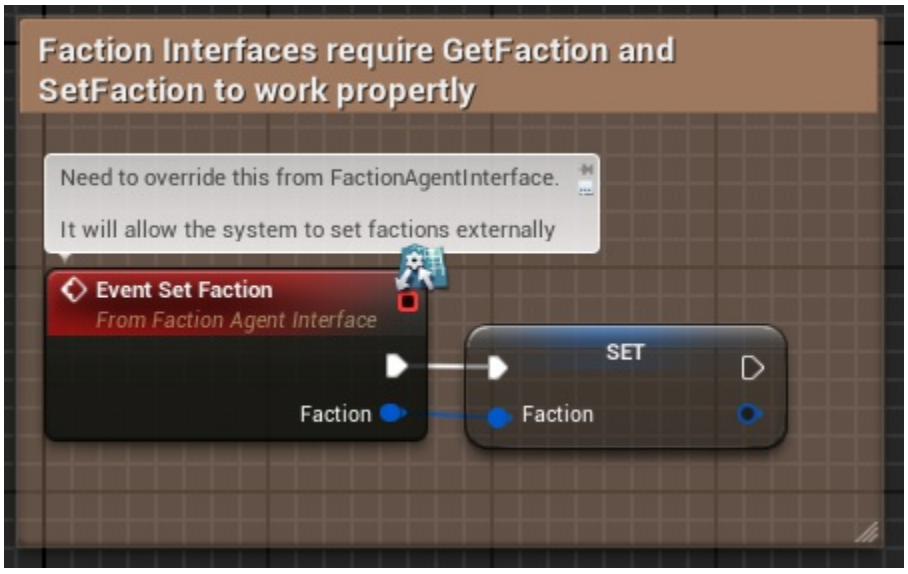
For an actor to have a faction, we need to add a FactionAgentInterface. This will allow the system to set and get the current interface.

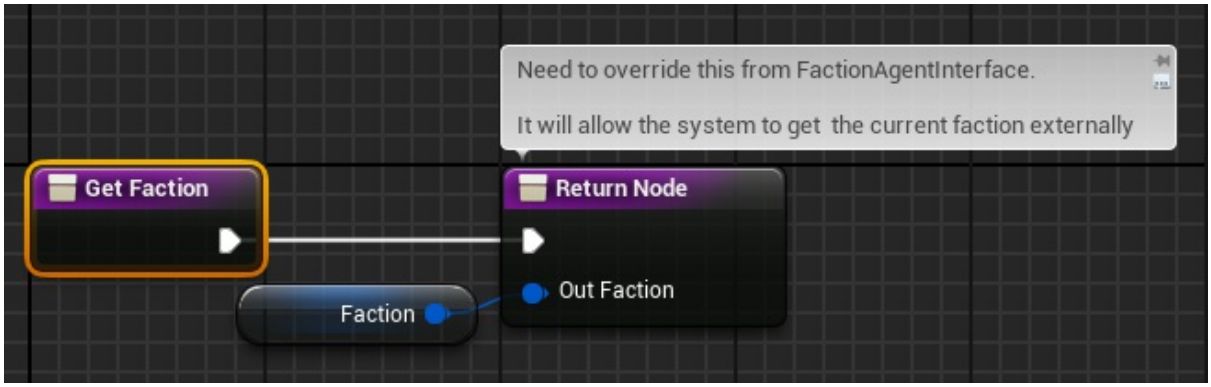
To add an interface we have to go into **Class Settings**

From here we click on **Add**, inside Interfaces and search for FactionAgentInterface

Getting and Settings the Faction

The system needs you to tell him how to set the faction and how to get it. This is done by overriding two functions as seen below. The Faction variable that is seen is a normal blueprint variable, that can be edited normally.





Factions in Controllers

Sometimes you may want a controller to share a faction with its controlled pawn or character. This process is exactly the same as before, except that instead of getting and setting a variable, we will get and set the faction of our pawn.

C++