Adding Custom Missions to Campaign

Introduction

This guide will outline some of the various ways to insert custom made scenarios into the game. In the example below, we will add a custom mission to the base Mechwarrior 5 campaign. It will demonstrate a basic usage of the editor's tools, but the methods and tools examined can be further leveraged to create large-scale campaigns and mission arcs, similar to the base campaign content in Mechwarrior 5.

In the example below, we will take a custom made mission and place it on a specific location in the starmap.

Common Terms Quick Reference

There are multiple asset types involved in bringing content into a campaign. This is a quick reference for many of those terms.

- **Campaign Arc:** This is the equivalent of a "quest".
- **Metagame Objective:** An "incoming transmission" that contains text and reward data.
- **Campaign Arc Action:** An arc action is an event that happens in the campaign itself. These can be stringed together to offer missions, messages, and other campaign related events.
- **Metagame Objective Header:** A "category" which is applied to the quest, allowing it to be sorted and tracked in the Mission Log.
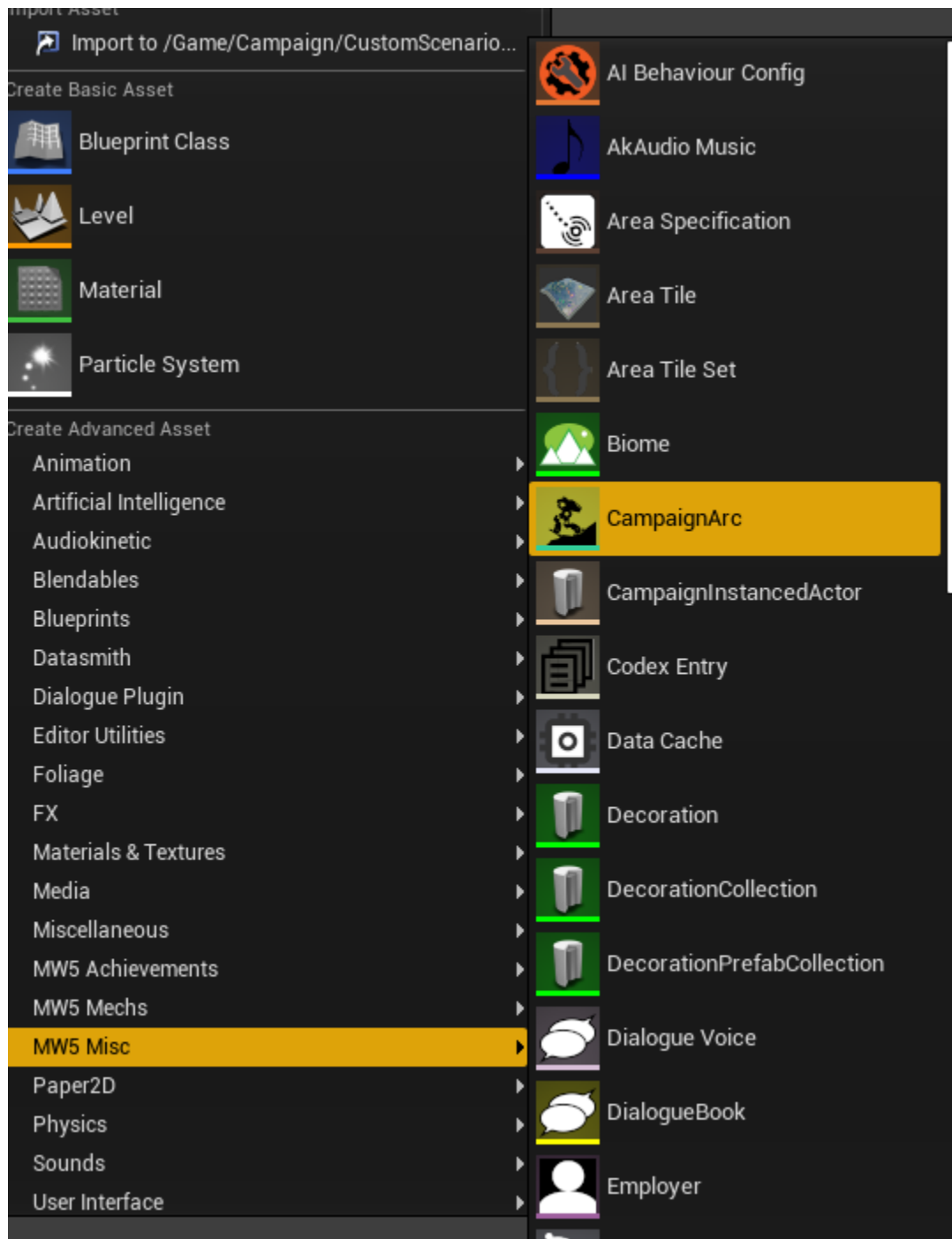
Step 1: Create the required assets

First, it's advisable to contain all the content in a single folder.

1.1 Campaign Arc

Start by creating a `Campaign Arc,` which will contain all of the mission content and trigger conditions.

To create one, go into the content browser and right click on the blank space, then under the `MW5 Misc` sub menu, select `CampaignArc`.


We will call it `CustomMission_Arc.`

1.2 Metagame Objective

Next, create a Metagame Objective, which is an objective that can be used to track progress, provide rewards, and sends a transmission that will appear to the player. This contains any story details and rewards, and essentially acts as the "quest giver".

To create one, go into the content browser and right click on the blank space, then under the `MW5 Misc` sub menu, select `MetagameObjective`.

Call this one `CustomMission_Objective.`
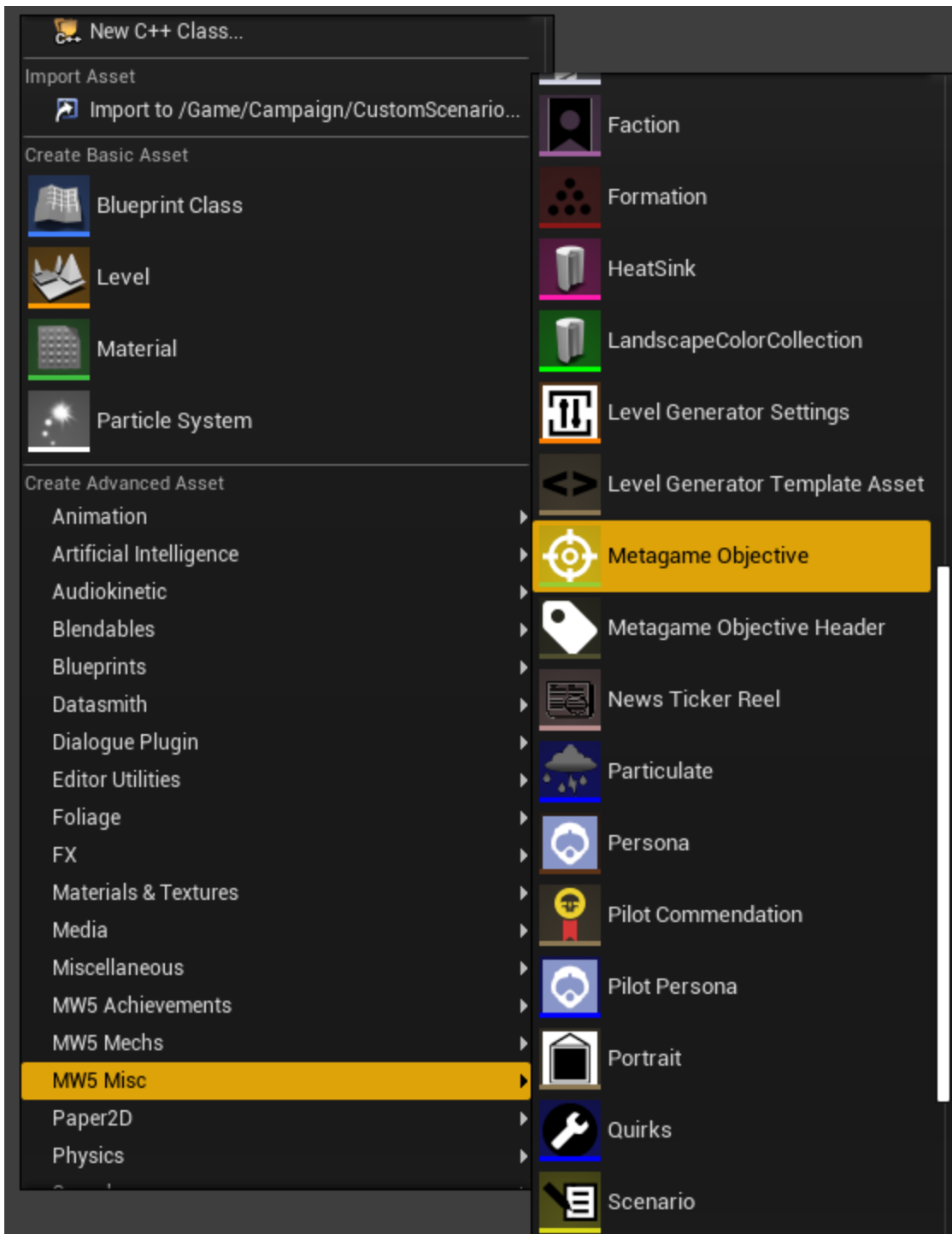
1.3 Campaign Arc Action

A Campaign Arc Action is an "event" that occurs during your quest. We need four for this simple questline. They will do the following:

1. Offer the quest
2. Place the mission on the starmap
3. Mark the quest as done when the mission is successful
4. Create another action that resets the mission if failed.

To create one, go into the content browser and right click on the blank space, and select `Blueprint Class`.
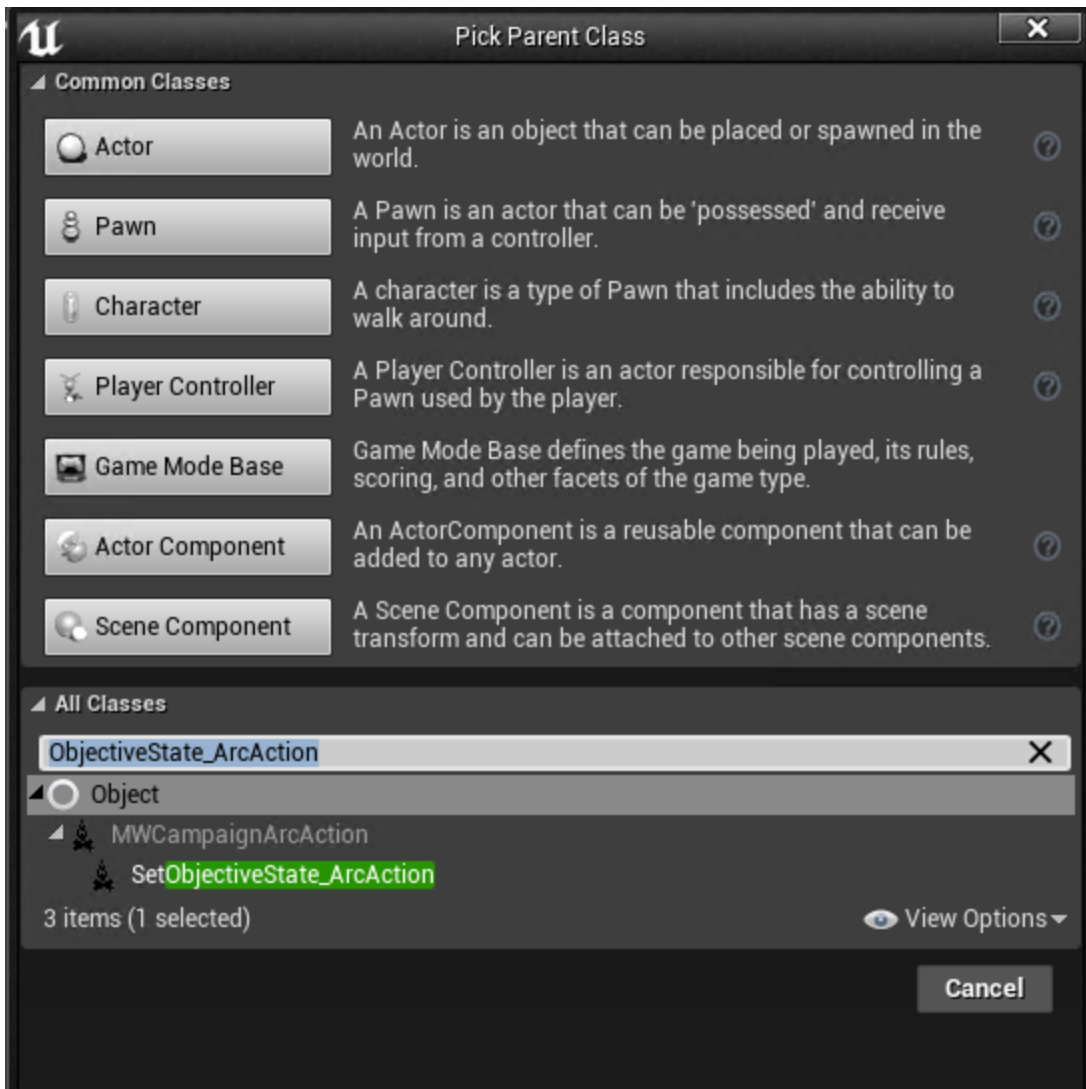
Create PlaceMission_ArcAction
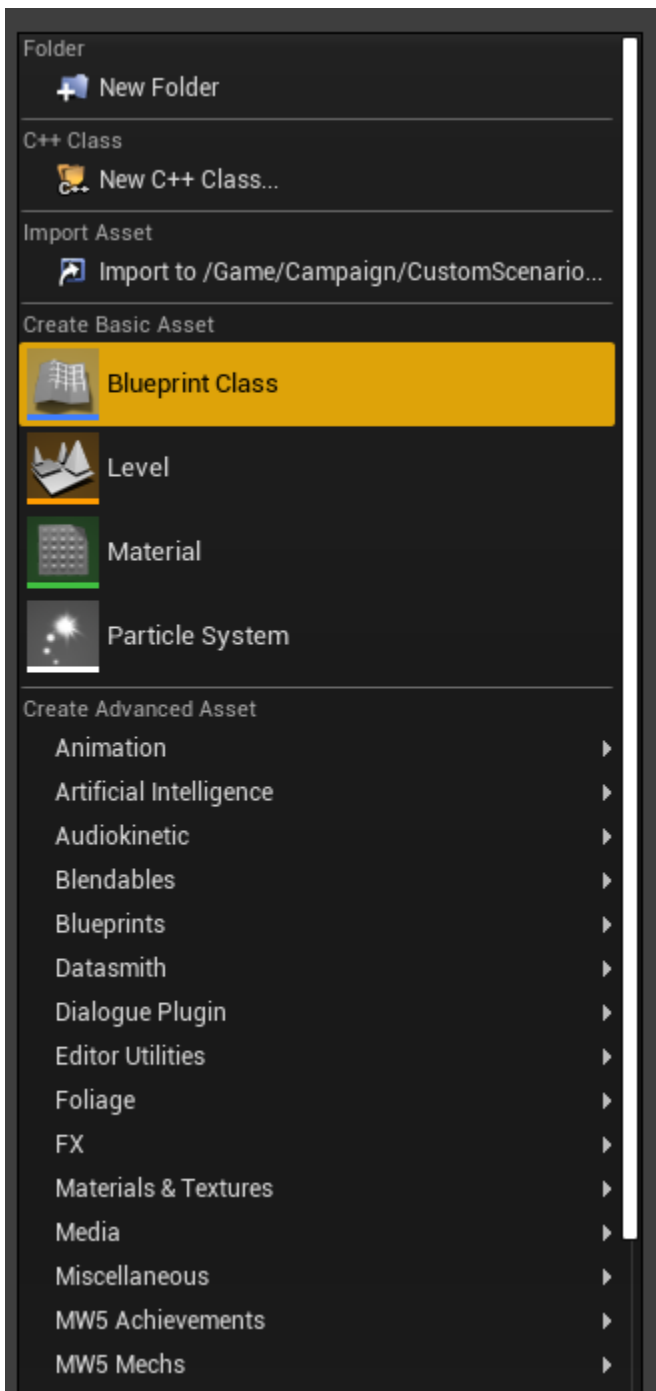
From there, we need to choose the Blueprint that will serve as the parent class. Under `All Classes`, type Place Mission Arc Action to filter the selection, and choose `PlaceMission_ArcAction` to select it.
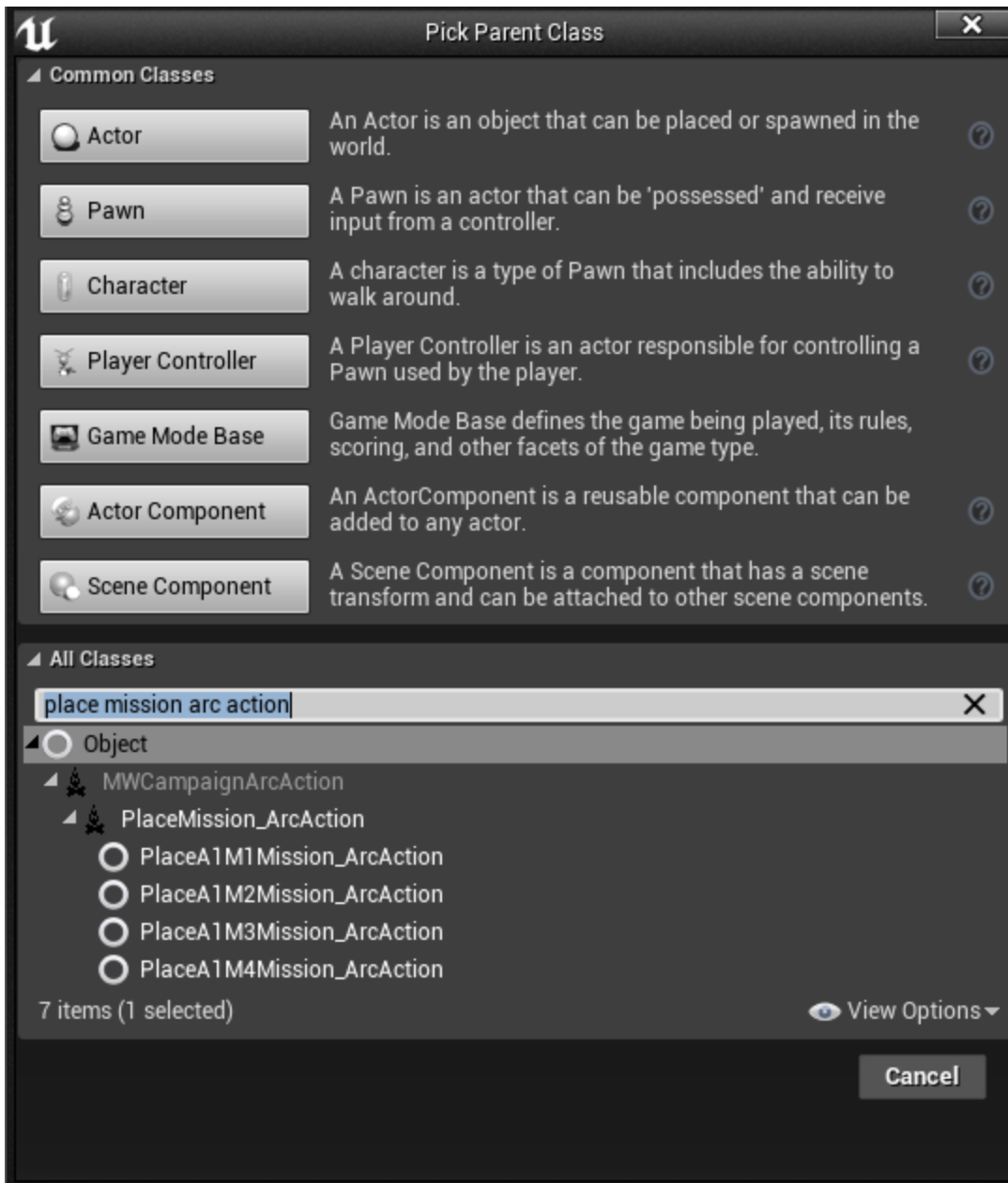
Let's call this `PlaceQuest_ArcAction`.

Create OfferQuest_ArcAction

Create another one called `OfferQuest_ArcAction`, and use `ObjectiveState_ArcAction` as its base.
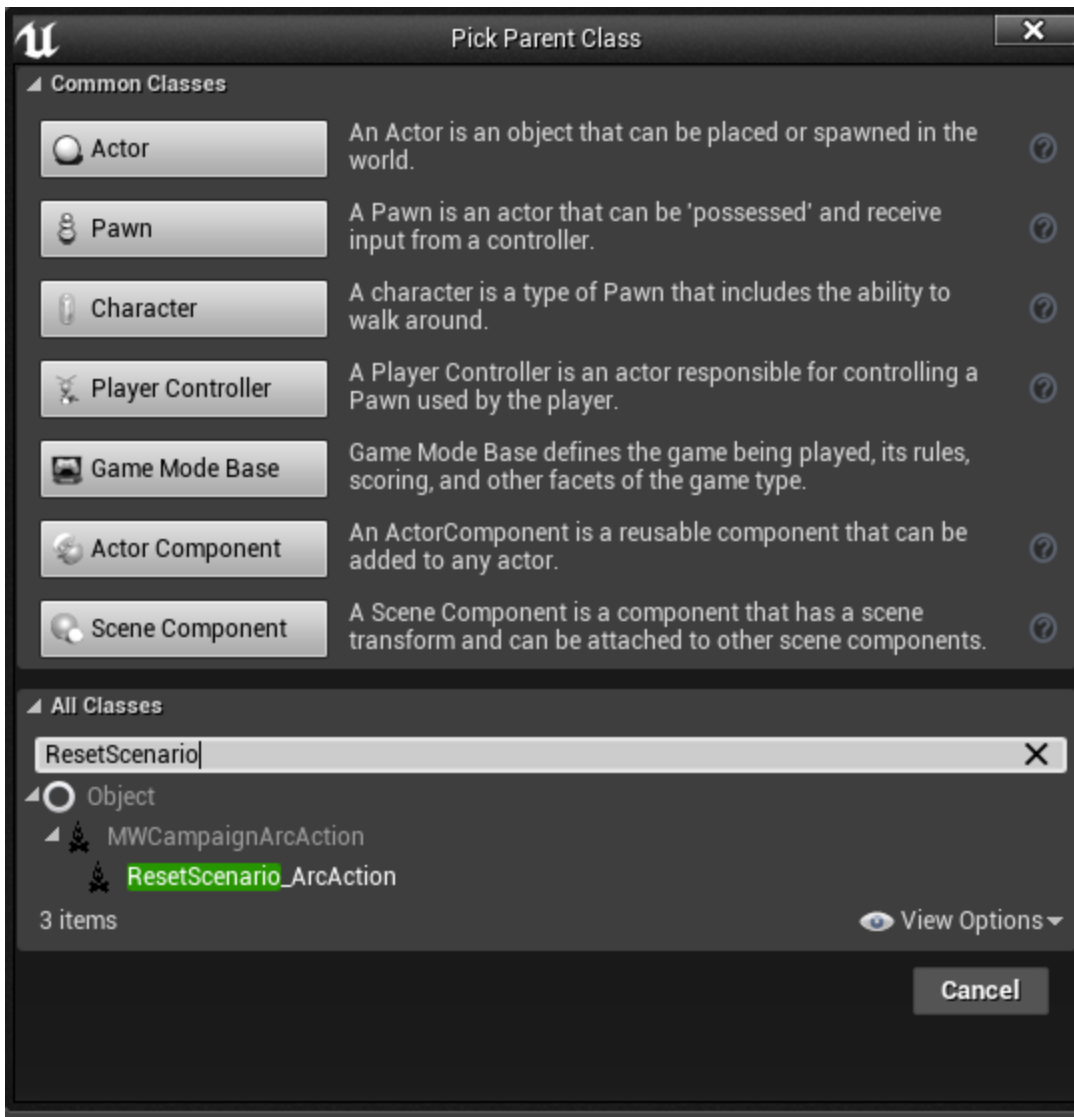
Create Complete_ArcAction

We need another Arc Action. Duplicate `OfferQuest_ArcAction`, and name it `CompleteQuest_ArcAction.`

Create ResetMission_ArcAction

Finally, we need one final Arc Action, that re-places the mission of the player fails it.

Use `ResetScenario_ArcAction`, and name it `ResetMission_ArcAction.`

1.4 Metagame Objective Header

The header is a "category" which is applied to the quest, allowing it to be sorted and tracked in the Mission Log.

To create one, go into the content browser and right click on the blank space, then under the `MW5 Misc` sub menu, select `Metagame Objective Header`.

Name this one `CustomMission_Header`.

While we're at it, open it up and give it some text. This will be the category name that the quest will fall under in the log. Usually it's best to give it the name of the quest you'd like to create.

Step 2: Fill out Metagame Objective Data

Open `CustomMission_Objective` and expand display info:

For our purposes, only the following fields are relevant:

- **Employer:** Metagame data for the employer faction, can largely be ignored for our purposes.
- **Screen Title Text:** The quest name that appears in the transmission.
- **Screen Sub Text:** The sub-objective that appears in the transmission.
- **Prompt Title Text:** The quest name that appears in the starmap UI. This can be shortened if the title is quite long.
- **Prompt Sub Text:** The sub-objective that appears in the starmap UI.
- **Objective Briefing:** The body text of the transmission. This is where the good flavor is added!

- **Has Rewards:** If you want to give the player a reward for completing the quest, mark this as true.
- **Objective Reward:** The rewards to give to the player upon completion.
- **Reward Briefing:** The text to show the player when the quest is complete.
- **Employer Agent:** The name and image of the speaker who appears in the prompt.
- **Only Show Currency Rewards:** If checked, the rewards (other than cbills and reputation) will be presented as question marks. This is used for when we want to hide rewards from players.

Fill it out the fields with whatever content you'd like (we'll look at rewards in a moment), which should look like the following:

Employer Agent

Expand the Employer agent, and then further expand the `Persona Data` dropdown.

These are the fields you can fill out to assign a character to the transmission.

You have two options- you can create a new character, by filling out the `Full Name` field, and selecting a `Portrait Asset ID` from the list of available assets presented in the dropdown.

Additionally, a large number of premade Persona assets are available for your use. A persona asset contains the name, portrait, and other character data. You can manually add one by selecting it from the `Persona Asset ID` dropdown.

For our use, let's select the pre-existing Sebastian Spears persona.

NOTE: You can also create your own `Persona Assets` and even `Portrait Assets` if desired.

Header

Under `Header`, be sure to select the new Header asset (`CustomMission_Header`) we just made.

Adding Rewards

If you'd like to add rewards to this quest be sure to mark `Has Rewards` as true, and then expand the `Objective Reward` dropdown.

In here, you can add CBill, Reputation and item, pilot and other types of rewards.

In the example below, I've assigned CBills, Rep, and even included a Mech as a reward. If you click the `Mech Data Asset` dropdown, a list of all available loadouts will be available. Other settings such as custom paint skins are available too.

If you are offering any sort of item or 'Mech reward, you need to generate a valid Item ID.

To do this, click the arrow next to the **Value** and select **Generate.**

You should see the value in the field change to a new GUID.

Step 3: Fill out the "OfferQuest" Campaign Arc Action

Now we need to fill out the data in the Arc Actions, which act as "events" in the questline.

Open up **OfferQuest_ArcAction**. This Campaign Arc will send the tranmsission and start kickoff the objective.
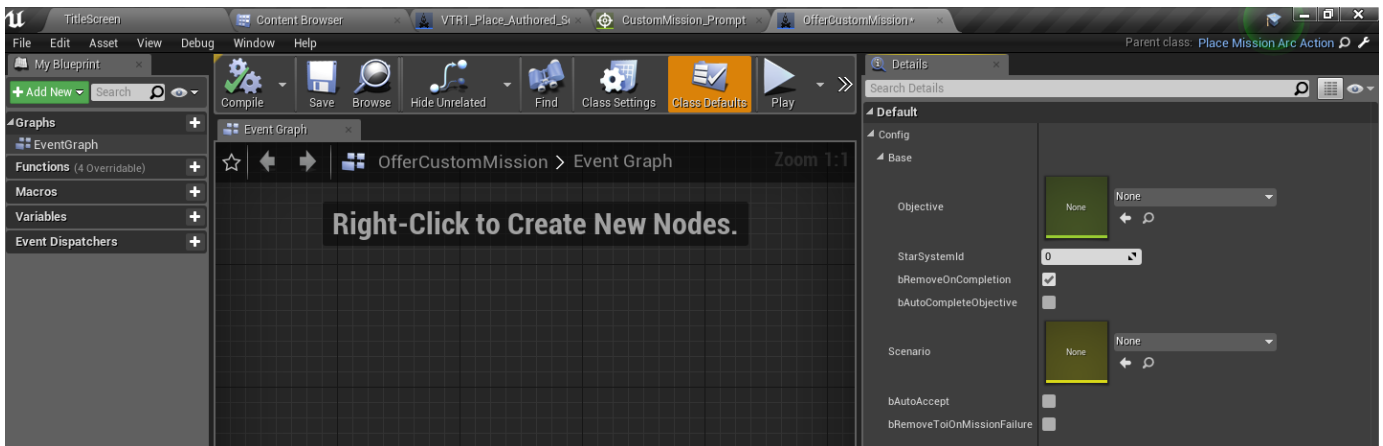
In the Objective field, add the Metagame Objective **CustomMission_Objective.**

Under operation, leave it set as "Offer", meaning the selected objective will be offered when this event is triggered.
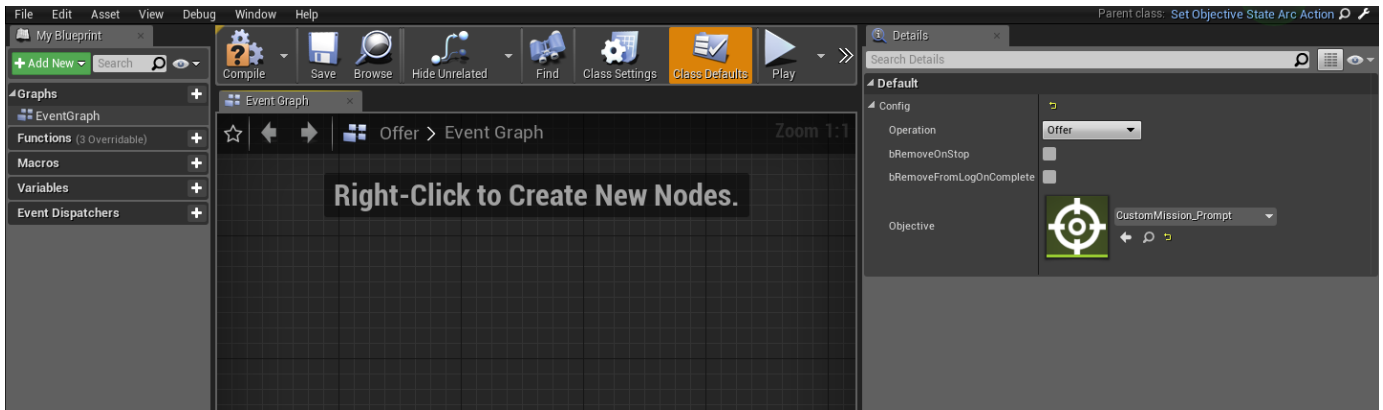
Step 4: Fill out the "PlaceMission" Campaign Arc Actions

First, open up the `PlaceMission_ArcAction` Campaign Arc. The logic for this blueprint actor will place the mission in the Starmap for the player. All that is required is to hook up the necessary assets.

Fill out the following:

- **Objective:** This is where the `Metagame Objective` (CustomMission_Objective) is added. Place it here.
- **StarSystemID**: This is the location in the Starmap where the mission will be placed. Each star system in the Starmap has a numerical ID.
    - To find an ID, navigate to to `Content/InnerSphereData` and open up `MW5_InnerSphereData`.
    - This is a data table that contains the information of every system on the map. You can search by name to retrieve the ID.
    - In the example below, the System "Brundage" has an ID of 2065.

- **Scenario:** Link to the custom mission you'd like to place. For here, we're just pointing to a campaign mission from the base Mechwarrior 5 campaign.

    At the end of the process, the Campaign Action should look like the following:
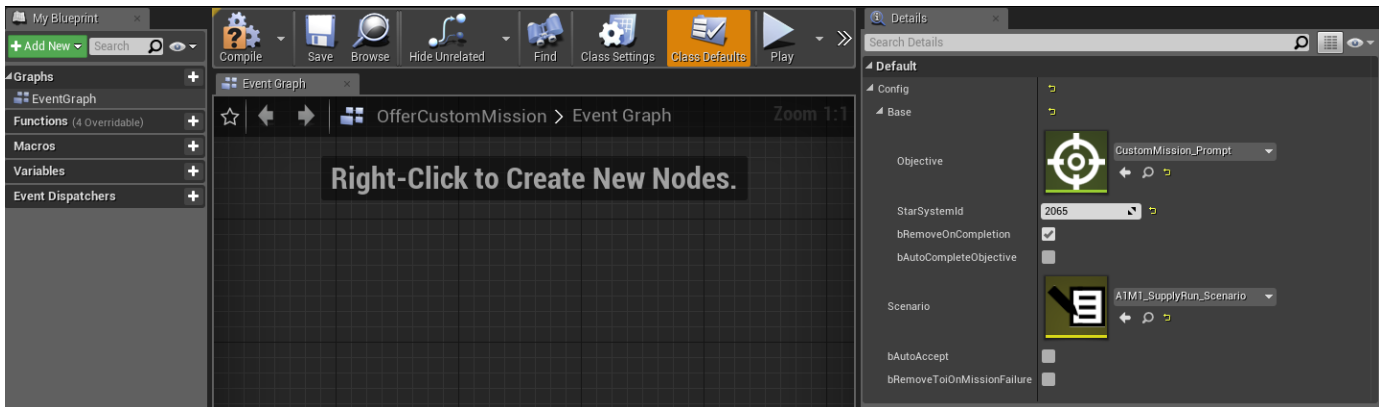
Step 5: Fill out the "CompleteQuest" Campaign Arc Action

Finally, we need to fill out `CompleteQuest_ArcAction`

Open it up, and in the Objective field, add the Metagame Objective `CustomMission_Objective.`

Under operation, this time set it as "Complete". What this means is that the selected objective will be marked as "complete" when this event is triggered.

Step 6: Fill out the "ResetMission" Campaign Arc Action

If the player fails your custom mission, it will be removed from the Starmap. We will need to place it again on the map under that circumstance.

Finally, we need to fill out `ResetMission_ArcAction`

Open it up, and in the scenatio field, add the scenario of your choice. This should be the same scenario you added in `PlaceMission_ArcActio n.`

Under operation, this time set it as "Complete". What this means is that the selected objective will be marked as "complete" when this event is triggered.
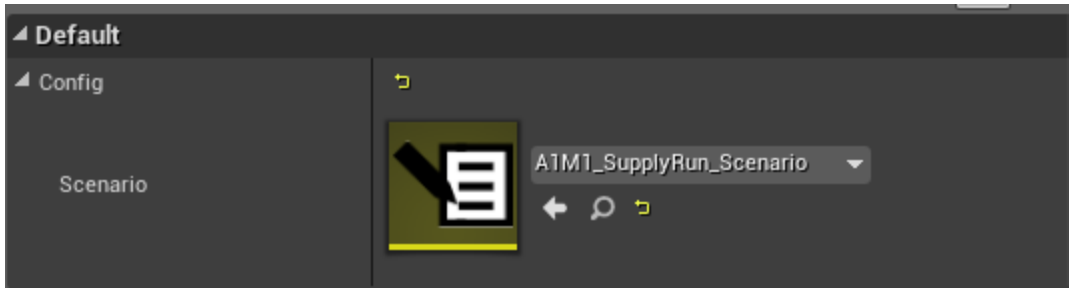
Step 7: Add the Campaign Arc Actions into the Campaign Arc

Now all we need to do is chain together our three Arc Actions and assign conditions for how they trigger.

Open up `CustomMission_Arc`, and you will see the following:
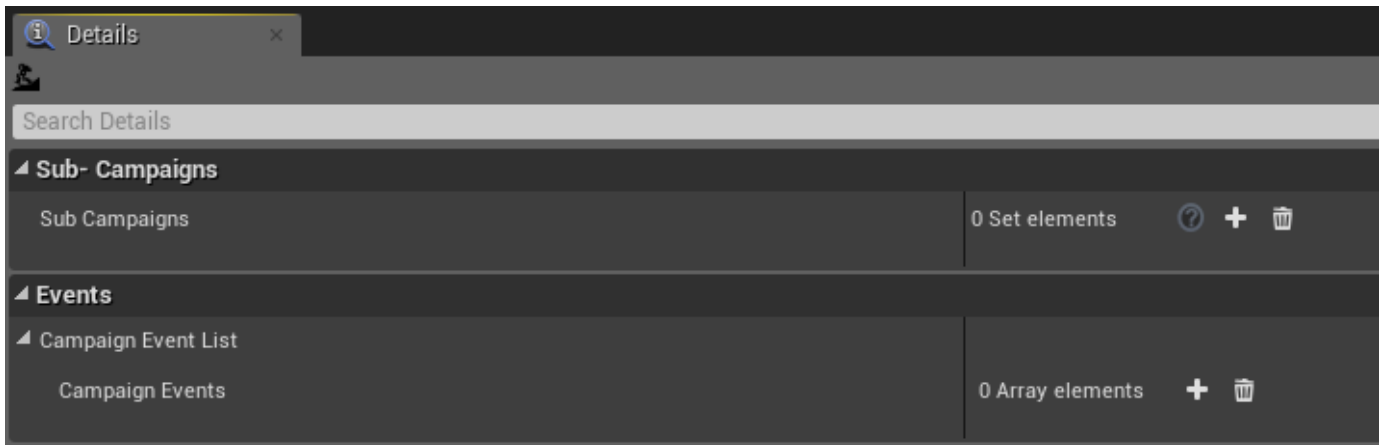
The top area called will allow you to add other campaign arcs within this current one. This is useful for nesting other campaigns inside a broader asset, but for now we don't need to worry about this.

The Events section is where the conditions and actions are determined.

For each Action, we'll make a new entry inside `Campaign Events.` Each event has a wide array of conditions and logic to apply to it.

Campaign Arc Conditions

Before we hook things up, let's take a look at the trigger conditions:

Event Name: The name of the event inside a quest. These can be referenced by other events for checking completion.

Trigger Conditions: Contains the various conditions that must be met in order for the event to occur.

- You will see a number of different variables inside trigger conditions, such as Rep Level, Date, the amount of Cbills the player has earned, and even the completion state of other objectives and missions.
- The `Trigger Type` variable provides a number of types of conditions:
  - Immediate: Only triggers once, and immediately marks itself as resolved when the criteria is met.
  - One Off: Ony triggers once, but marks itself as resolved when any criteria is not true.
  - Always: Triggers when criteria is met, resolves itself if any criteria is not true. However, this can trigger again if the criteria is met again.

With that in mind, let's add the arc actions.

Add the "Offer Quest" Action

Let's make the first event, which will "Offer the Quest".

Name the event **OfferQuest.**

Set **trigger type** as immediate.

For this sample quest, let's make it appear immediately for players. We'll do this by checking **ReputationLevel**, and setting it to Greater or Equal to 0. This means if the player is above Rep 0, this event will trigger. If you'd like other conditions, such as triggering after a certain Stamap Date, reputuation level, or anything else, feel free to experiment.

Under **Campaign Actions**, let's add **OfferQuest_ArcAction**.

This will result in the **OfferQuest_ArcAction** triggering once, immediately for the player.
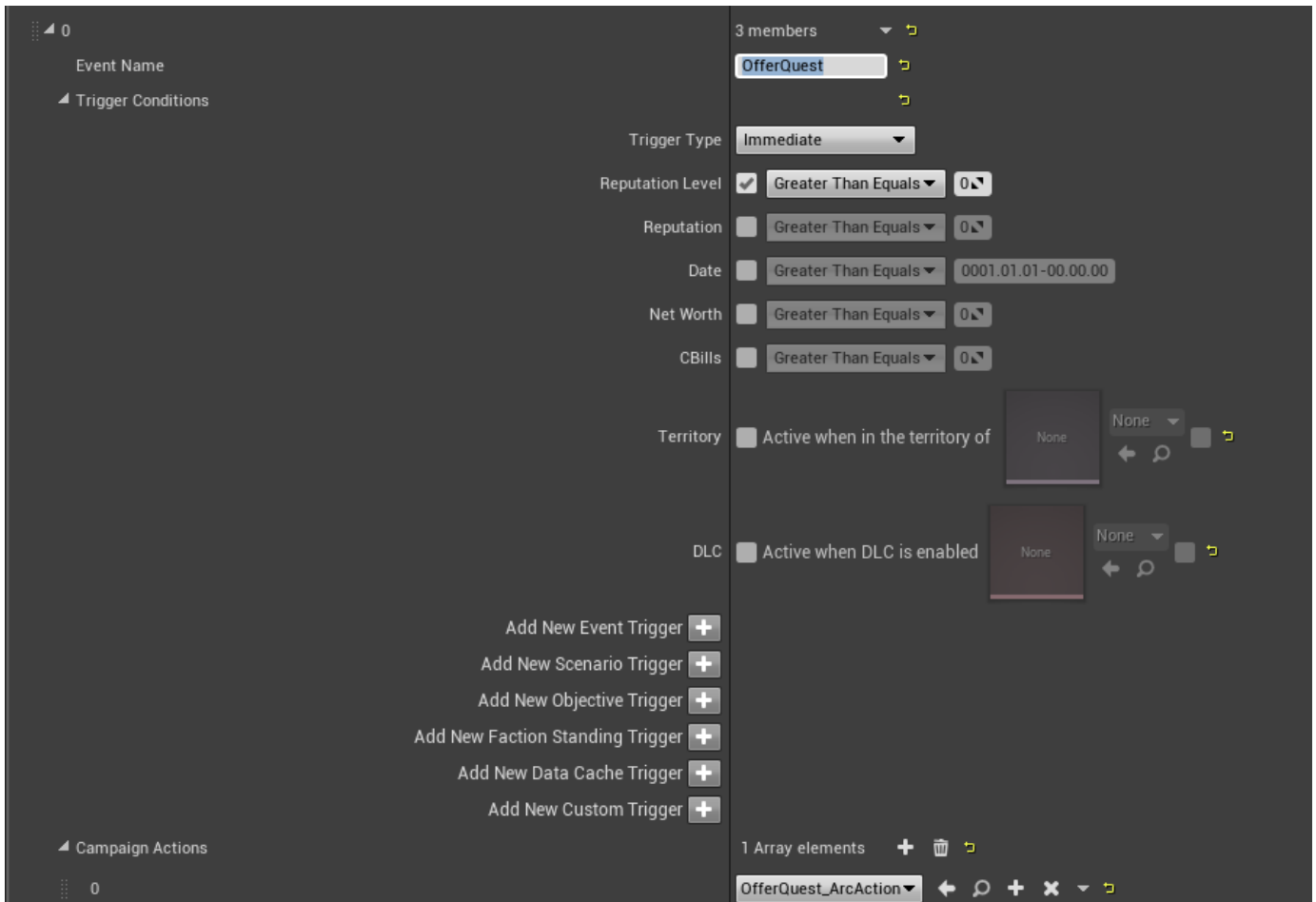
Add the "Place Quest" Action

Next up, if the player has accepted the transmission, we want to then place the quest. We will do this by making another event called "Place Quest", which triggers once the player has accepted the transmission.

This is done by adding an `Objective Trigger`. Do this by hitting the plus icon, and then linking to the transmission we made, `CustomMission_Prompt`.

Next to it, you will see a drop down with different options. Select `In Play`. (More on this in a moment)

Under `Campaign Actions`, link to the `PlaceQuest_ArcAction`.

The event should look like this:
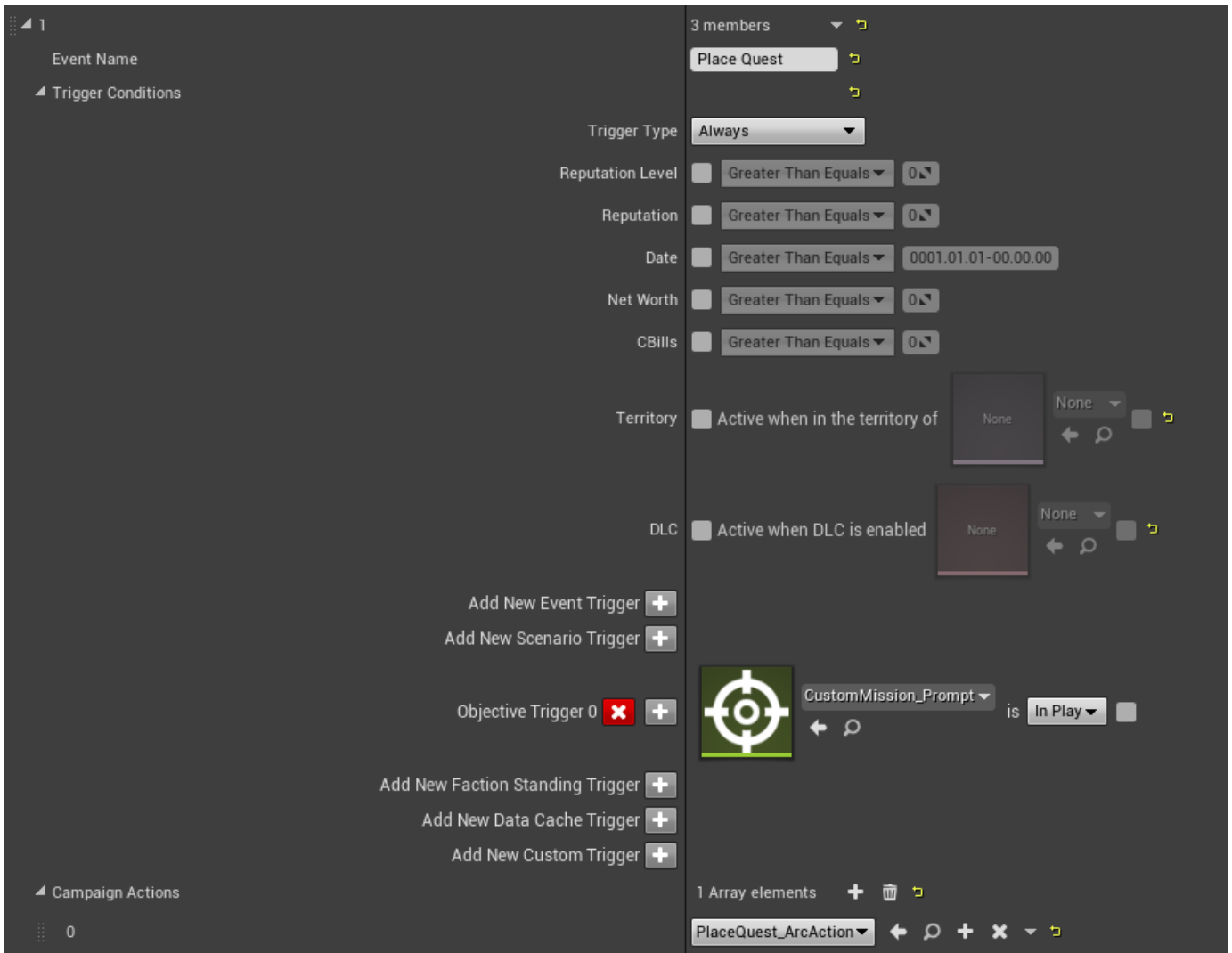
This will result in placing the mission on the specified planet, only when the player has accepted the transmission.

Next to the Objective Trigger condition is a dropdown with some options:

- In Play: Triggers if the objective has been accepted, and is running.
- Offered: Triggers if the objective has been offered or complete (ie, the transmission being sent), but not yet accepted by the player.
- Active: Triggers if the objective has been offered but not yet complete.
- Complete: Triggers if the objective is offered

Additionally, there is a checkbox next to the condition that inverts it- meaning if you check it, next to In Play, you'd create a condition where it triggers if the Objective is NOT In Play.

These can be leveraged to create branching arcs via conditions- IE, you can offer a specific mission, only if you have or haven't completed a previous objective, etc.

Add the "Mission Succeeded" Action

We need to mark the quest as complete when the player finishes the mission successfully.

Create the following event:
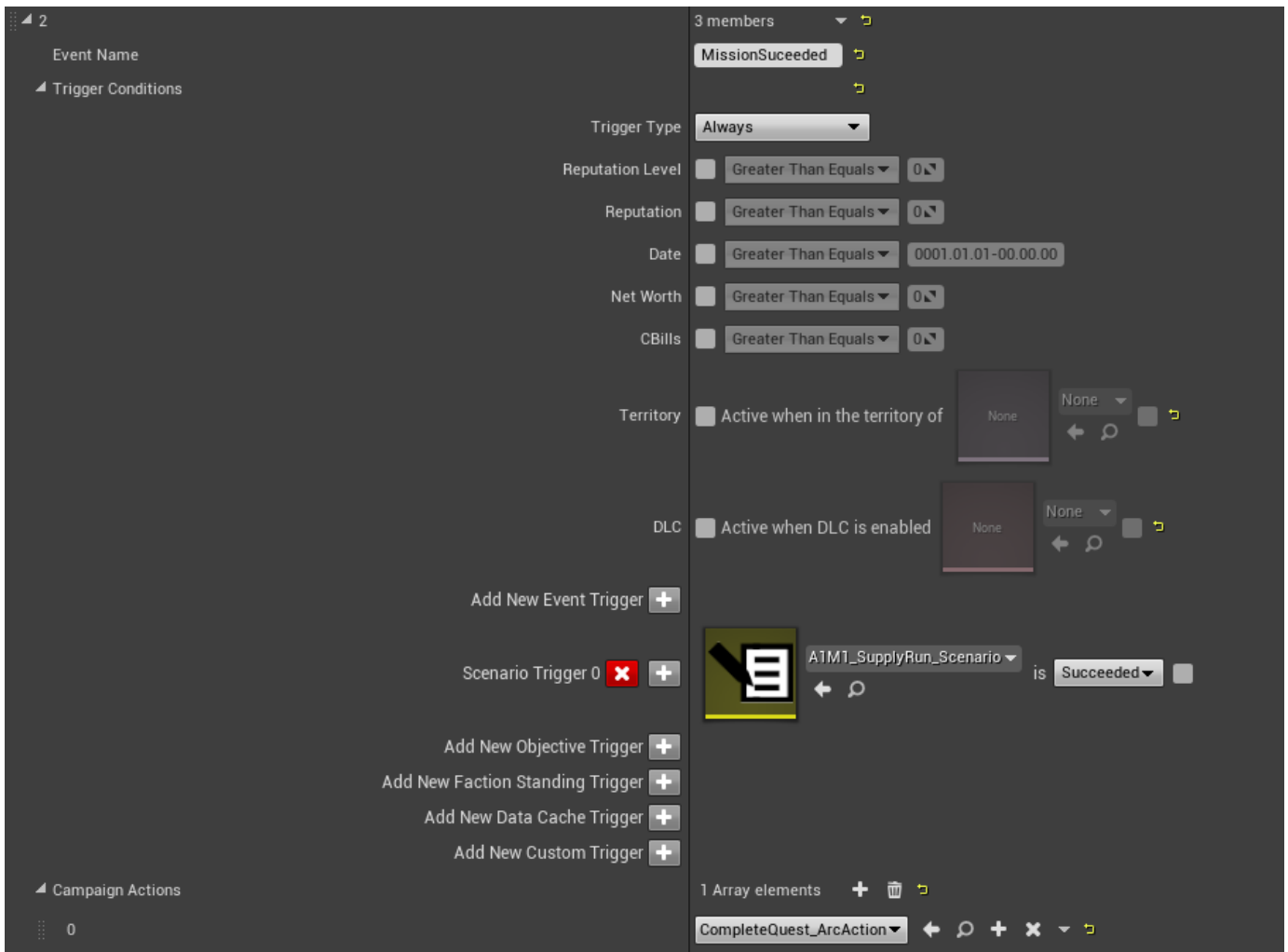
This time, place a condition in `Scenario Trigger`. Make sure it's selected as succeeded.

Add the `CompleteQuest_ArcAction` to the event.
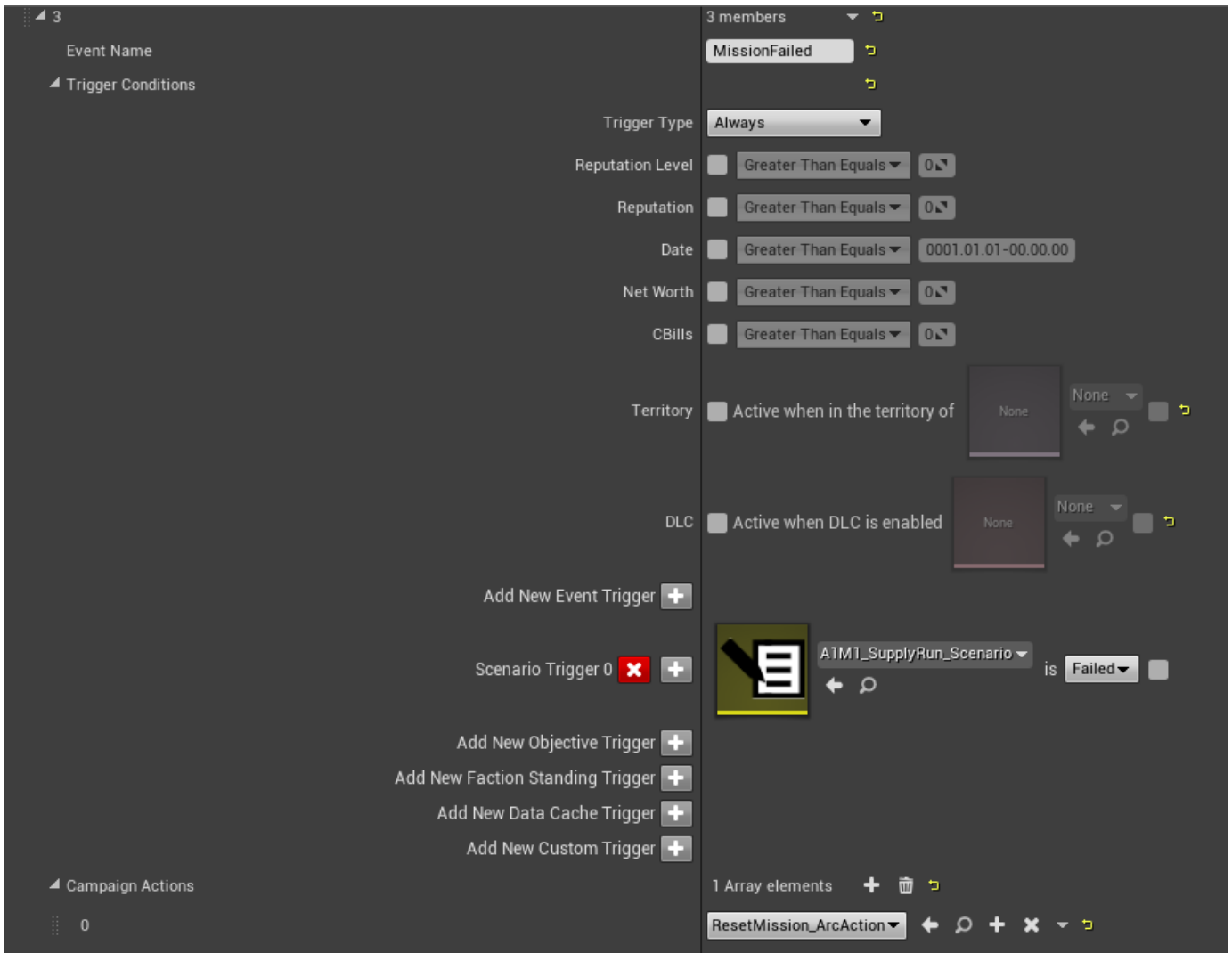
This will result in us calling `CompleteQuest_ArcAction` when the mission of our choice is a success.

Add the "Mission Failed" Action

Finally, we need to handle the failure state. We are simply re-planting the mission at the location, but you can offer other missions, quests, and transmissions as desired.

Create the following event:

This closes out our simple quest.

Step 7: Insert the Quest Into the Game

Everything is now complete, and all the logic is nested within `CustomMission_Arc.` All that remains is adding it to a campaign.

You are able to place this in different campaign types. Where you decide is up to you, and you can even create your own entire custom campaign filled with unique start conditions and missions.

In general however, the most common places to inject new story content would either be into the base Mechwarrior 5 campaign, or the Career Mode.

For our purposes, let's add this to the base MW5 Campaign:

Navigate to `Content/Campaign/CampaignArcs` and open up the file `MW5CoreCampaign`.

This is simply a larger, more complex Campaign Arc that contains all the information and logic for the main campaign:
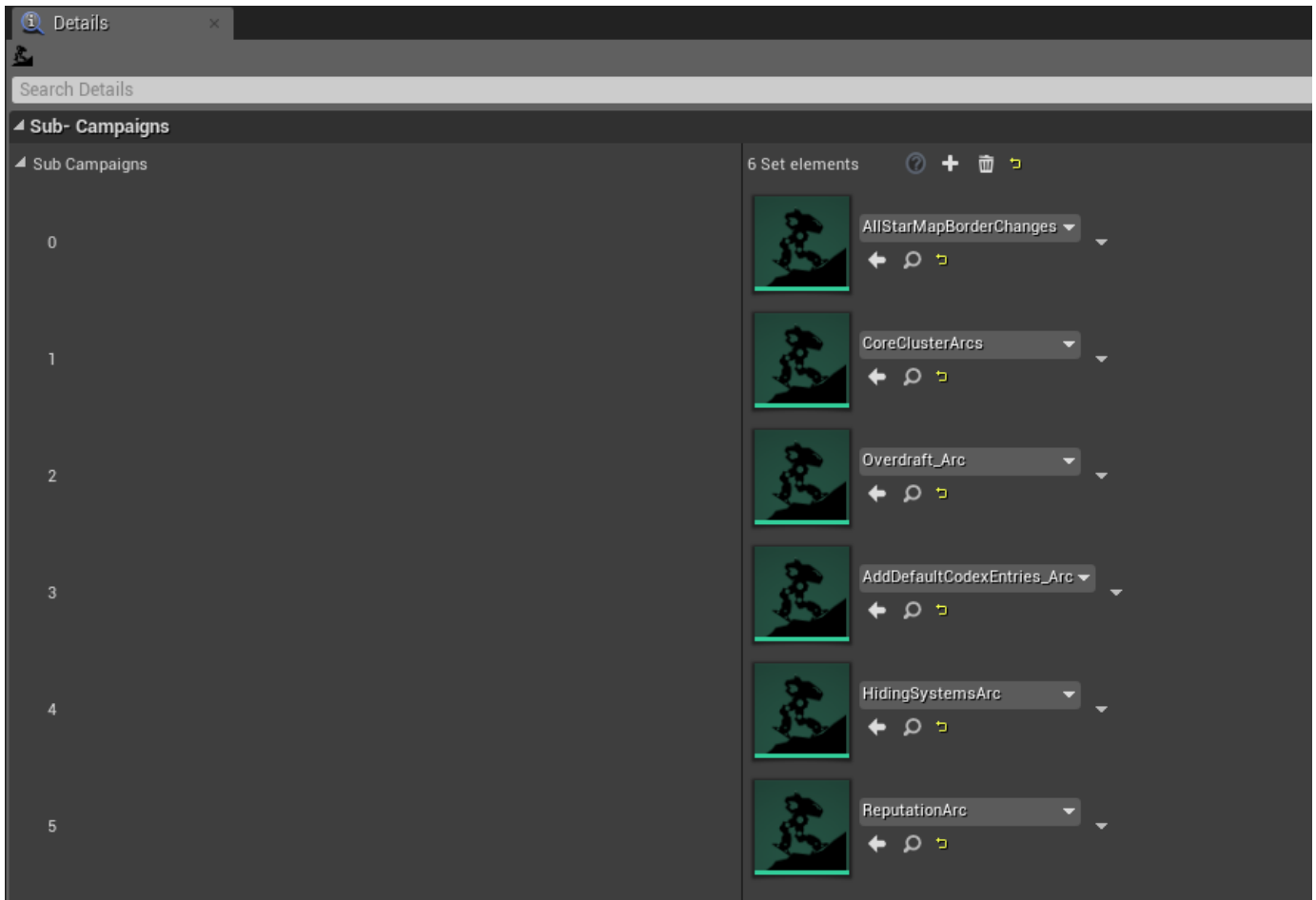
For the sake of modularity, the core components are broken into smaller campaign arcs, and nested within the **Sub Campaign** sections. These contain things like controlling the starmap borders, offering the main story quests, handling end-game states and more.

We don't want to remove anything, but instead just inject our new content to the campaign. Leave everything as is, but add a new sub campaign.

Insert `CustomMission_Arc` into the new sub campaign:
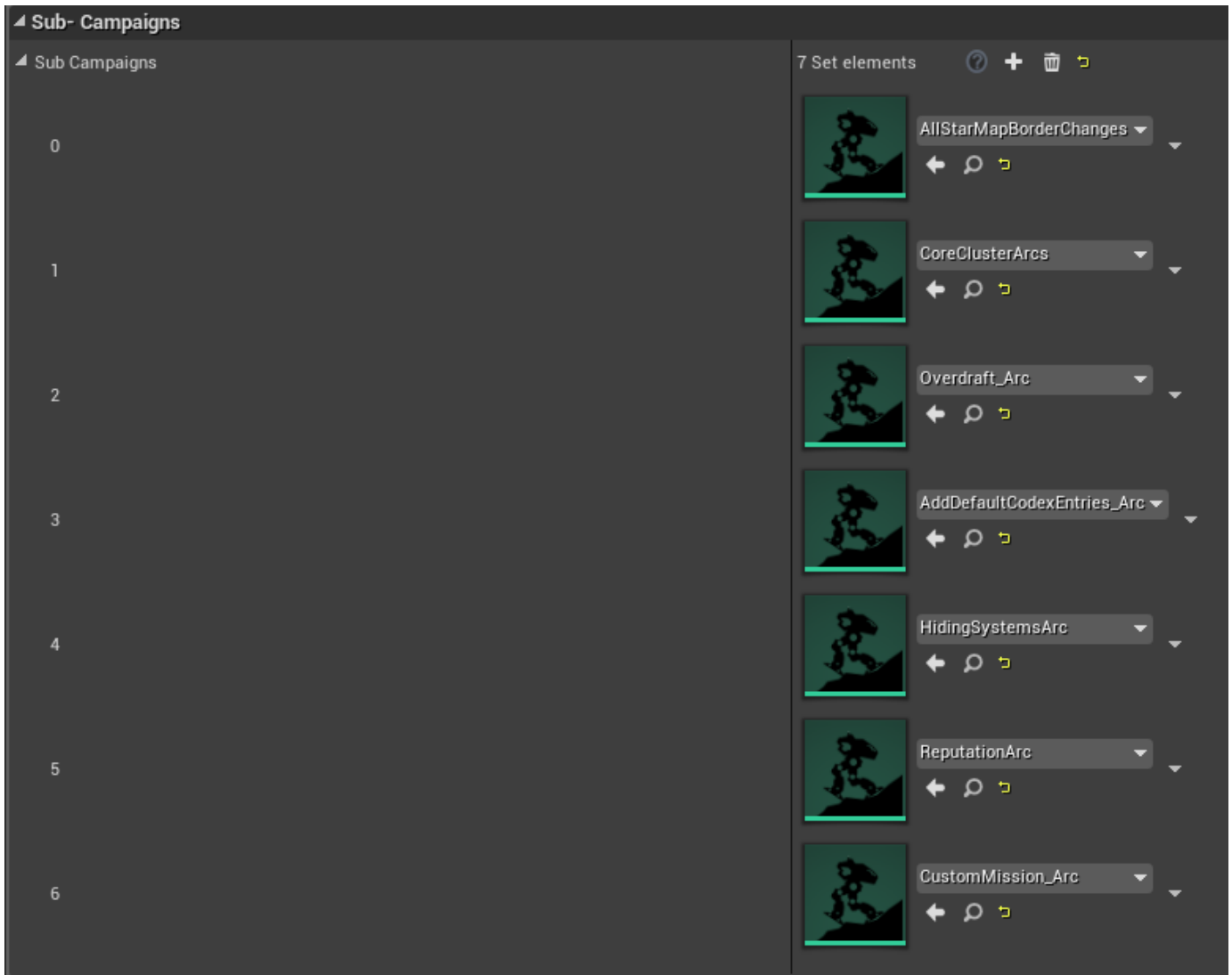
The order is not taken into account, and all the logic for triggering this is all handled in the Arc we just created.

Additional content and events can be further added to the campaign by adding them into this Arc.

Step 8: Test the Content

If everything has been setup correctly, you should be able to instantly see the quest in game. You can start a new campaign, or load an existing one, provided it's the base campaign per our example:

INTERSTELLAR EXPEDITIONS
Alright player, I need you to do a bunch of stomping for me. Go to the marked location and get ready to lock and load!

CURRENT OBJECTIVES

Complete the mission!

COMPLETION REWARDS

500,000 C-Bills   500 Reputation   UrbanMech UM-K9

ACCEPT