Plugin User Guide

To use it in Blueprints:

Adding the plugin into your project:

1. Add the folder with the plugin to Plugins folder in the root folder of your project or in the engine itself:

/[Project Root]/Plugins/[Plugin Name]/ or

/[UE4 Root]/Engine/Plugins/[Plugin Name]/

1. Switch the plugin on in UnrealEditor. Edit->Plugins->Advert->DynamicAdvert.
2. Restart the editor.
3. Place an Instance of DynamicAdvertActor in your Scene:

Content Browser->Add New-> Blueprint class -> Choose DynamicAdvertActor

1. Place it in a scene.

To use it in C++:

Adding the plugin into your project:

1. Include DynamicAdvertPlugin plugin in the build cs, so the build cs will end up looking something like this:

PublicDependencyModuleNames.AddRange(new string[] { "Core", "CoreUObject", …,”DynamicAdvertPlugin”});

1. Include DynamicAdevertActor class to your code

#include”DynamicAdvertActor.h”

1. Create DynamicAdvertActor instance or inherit own class from DynamicAdvertActor ;

Using the plugin:

FindAdvert function

The main method of DynamicAdvertActor is FindAdvert. Use it in DynamicAdvertActor blueprint to get the images and the video for the particular GameObject.

It has such parameters:

Id: you should pass here the index of media resources you need to get from the server;

Media Player: you need to select the MediaTexture that will show the advert video from the server. You should pass here the reference to MediaPlayer that provides Video to that MediaTexture.

Mesh: you should pass here the reference to StaticMeshComponent where you want to see the advert images from the server. The FindAdvert will add the textures to the material the StaticMeshComponent is using, one by one.

Parent Material: you should pass here the reference to Material that will be using as a base material when showing textures on the Mesh.

Need Hd: bool value that allows to choose which type of video to show. Pass true if you want to show hd, pass false if you want to show sd. If selected video type will be unavailable the plugin will ignore this and will show available format.

It returns true if the process started successfully.

It returns false in two cases:

if last request wasn’t successful

if the Game Session hasn’t started for some reason (Take a look at StartGame description).

StartGame function

It starts game session. You should call it on PlayBegin. If for some reason the request failed you got OnBlStartGameOnceMore event(StartGameOnceMoreEvent event in native C++) while trying to call FindAdvert. You should try to use it by yourself at that case. Also you may need to call it after EndGame in case you need to restart the session at runtime.

It has two parameters:

GameId: string identificator for your game

ServerAddress: string Server address, something like that:

http://c156057b.ngrok.io/sdk/advert

It returns true if the process started successfully.

It returns false if the session was already started.

EndGame function

It ends game session. This plugin calls it automatically on PlayEnd. You may need to call it by yourself in case you need to restart the session at runtime.

It has no parameters.

Plugin events

OnBlStartGameCompleted (StartGameCompletedEvent in native C++) can be fired after StartGame function call. It has bool parameter to show if Game Session started or not.

OnBlStartGameOnceMore (StartGameOnceMoreEvent in native C++) can be fired after after FindAdvert function call in case game session was not started yet.

OnBlVideoLoadingCompleted (VideoLoadingCompletedEvent in native C++) can be fired after FindAdvert call when video has loaded. It has bool parameter to show if it was loaded successfully.

OnBlTextureLoadingCompleted (TextureLoadingCompletedEvent in native C++) can be fired after FindAdvert call after one of the textures was loaded. It has has bool parameter to show if it was loaded successfully and int parameter - the loaded texture index.

OnBlEndGameCompleted(EndGameCompletedEvent in native C++) can be fired after EndGame function call. It has bool parameter to show if Game Session ended or not.