Unity3D Libraries of Life Content Management System

Adam Chmurzynski
3/28/2016

Contents

Features of Unity you need to understand in order to use the Content Management System:

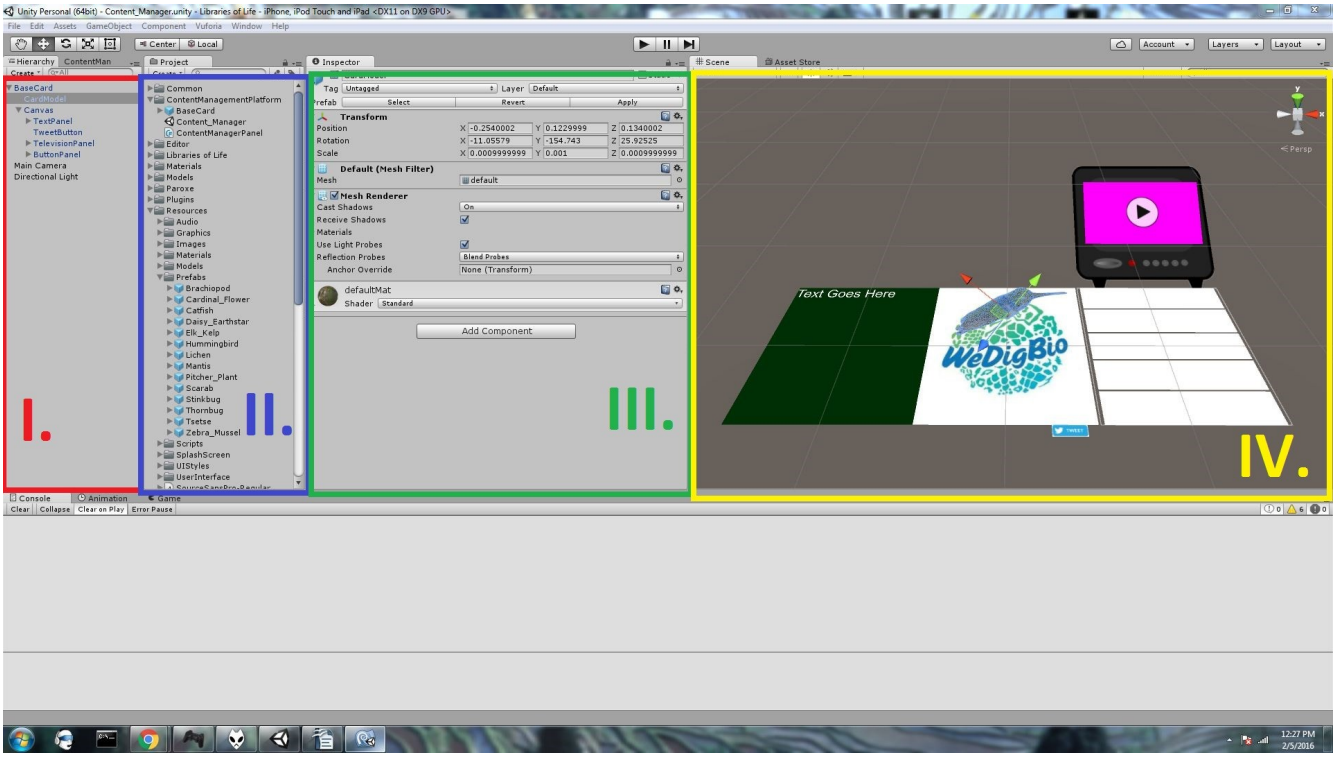
Windows:

- -Scene
- -Inspector
- -Project

GameObjects
Hierarchies
Parent/Child relationships

Transforms

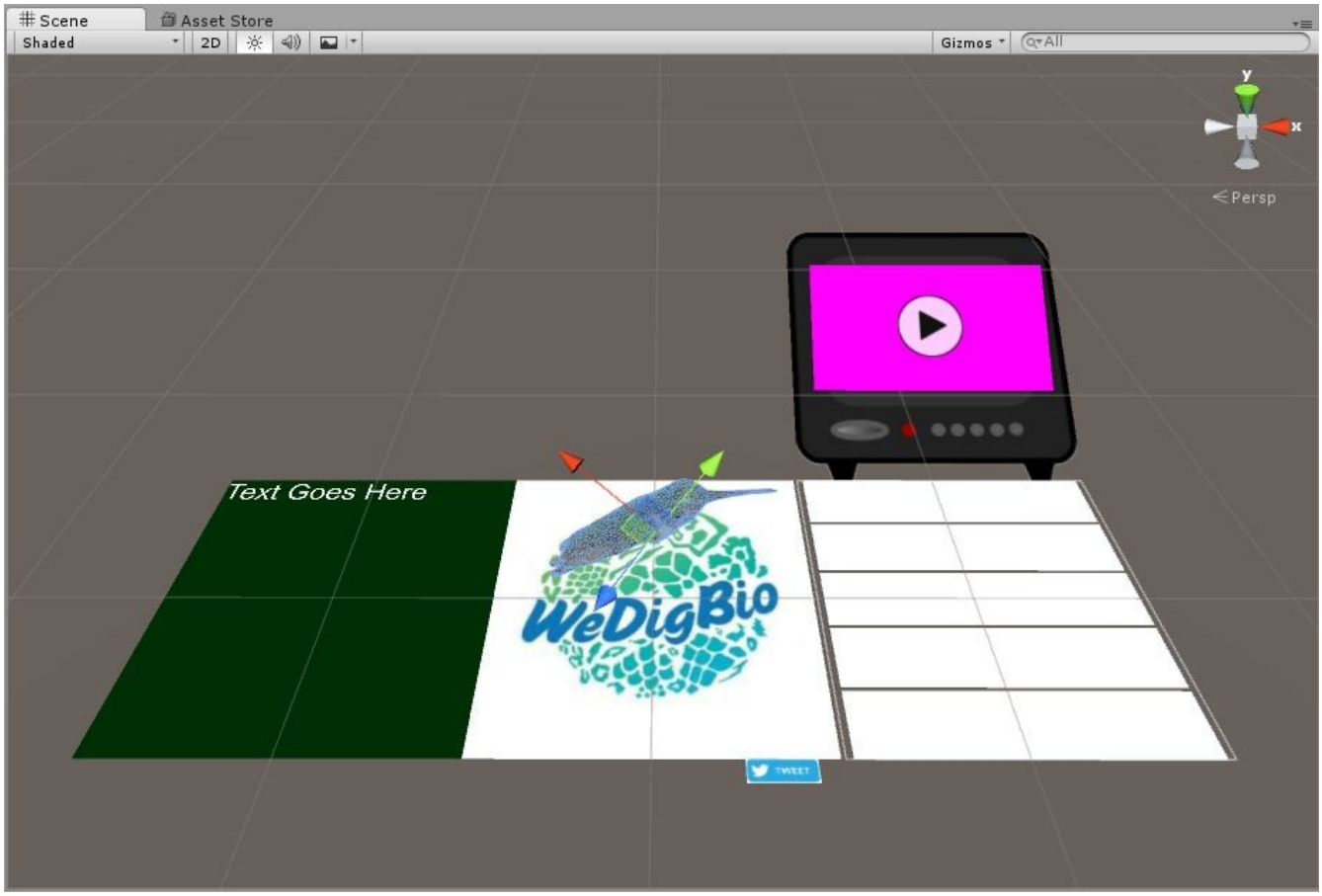
Components



I. The Hierarchy Window
II. The Project Window
III. The Inspector Window
IV. The Scene Window

Your Window layout may be different, and there are more Windows than I've shown here. You can drag and drop Windows to create whatever layout suits your workflow.

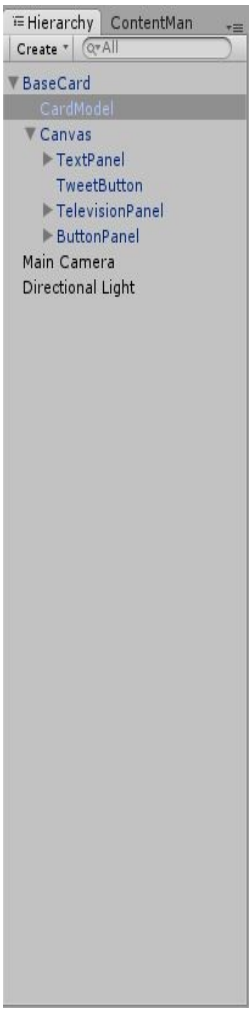
The Scene Window



The Hierarchy Window

A List of all of the GameObjects currently present within the Scene Window.

Clicking the name of a GameObject will make it appear in the Inspector.



The Inspector Window

Displays a list of Components attached to a particular GameObject. All objects have a Transform, which determines how they are oriented within the Scene Window.

The Project Window

Contains any assets imported into the Unity Project (models, audio, textures, images)

-Every asset you want to incorporate into the scene has to be attached to a GameObject

This is done via Components

GameObjects and Components

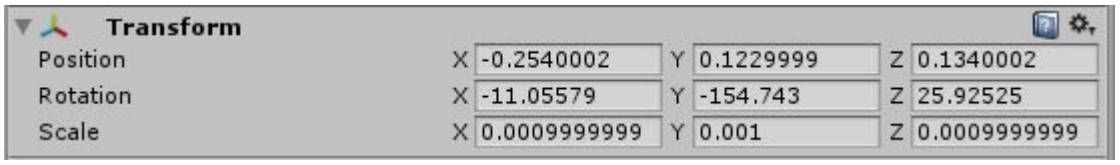
Components are visual elements corresponding to pieces of code (which you should never worry about) that let GameObjects do certain things

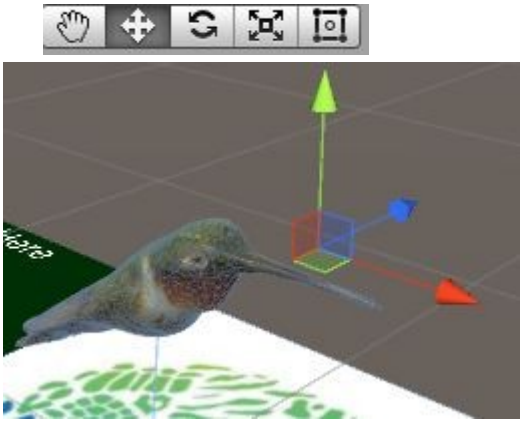
Every GameObject has at least one component: a Transform

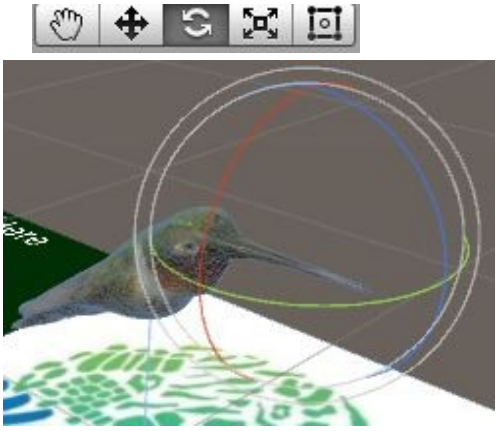
Transforms

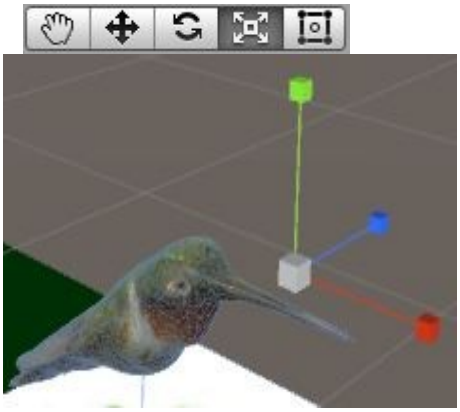
-Fields that should be familiar from Metaio: Position, Rotation, Scale

-These can be manipulated visually using widgets in the Scene Window









Other Components

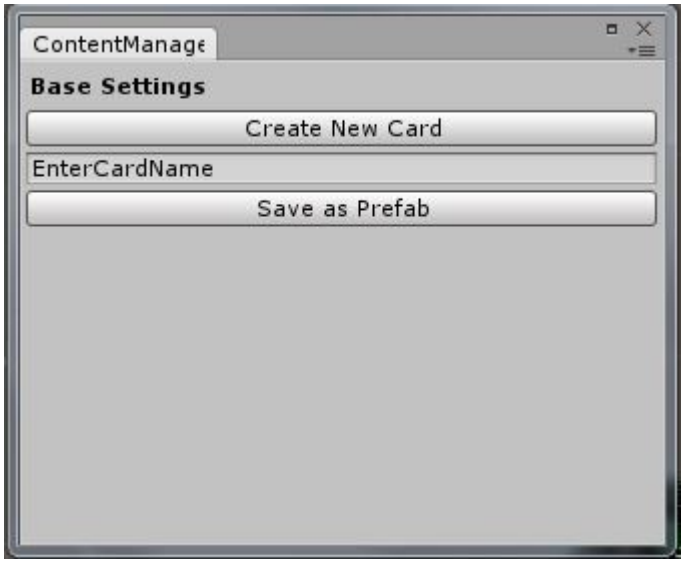
-Buttons
-Audio on objects
-Images
-Models
-Text
-Lights
-Hyperlinks/Tweets

These are all unified by the need to be connected to an object in 3D space.

The 'Content Management System' will be a simple extension to the Unity Editor. It is a custom Window, similar to the other kinds of Windows we have reviewed.
This Window has a simple interface for constructing a Libraries of Life card based on the prototype card I've shown you.

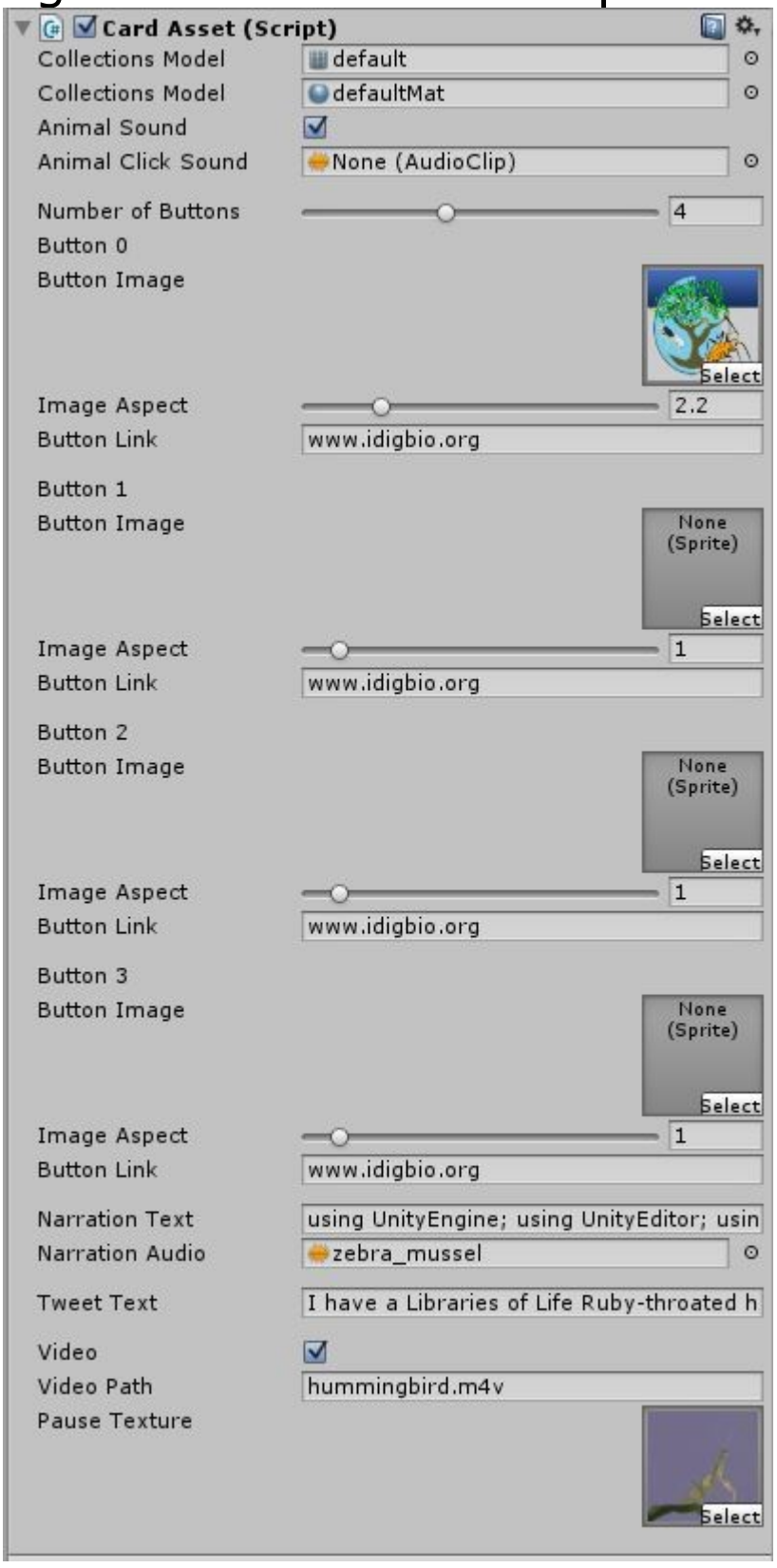
Opening the Content Manager

Under the 'Window' menu go to 'ContentManager'.
A new Window should appear.



Entering a name for the card and pressing 'Create New Card' should add a new prototype card to the Scene. Look for the card in the Hierarchy and click its name. A GameObject should appear in the Inspector. Double clicking the name will center the camera in the Scene view on the card in 3D space.

All of the changes you make to the card will be done through the CardAsset component.



Adding a Model

If you've imported a mesh and a texture into Unity, they should be accessible from the two fields on the CardAsset inspector shown above.

Selecting both should make the model appear with the proper texture on top of the card. If you import the texture into Unity separately, you will need to make a new material which uses the texture; we won't cover this point here.

Adding a click sound

Toggling the 'Animal Sound' field will reveal a new field where you can add a sound clip to be played when a user clicks/touches a card's model.

Adding Buttons

The slider at the top of this image determines how many buttons exist at the side of the card (up to 10, no less than one). Try moving the slider—you'll see buttons are automatically added and removed.
Simultaneously, fields will be added in the inspector, allowing you to specify an image and change its aspect ratio to make it more or less rectangular.

Each button is also associated with a link.

IMPORTANT: Links should never include the http/https protocol. Only the www.------ should be included with whatever URL appears afterward, as shown above.

Adding narration

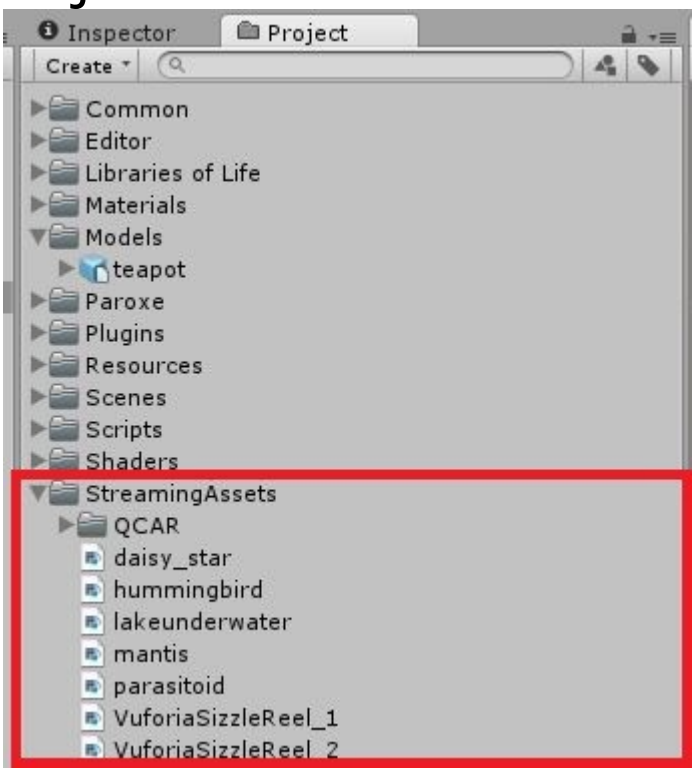
The 'Narration Text' field allows you to copy/paste a transcript of some narration onto a card. Try typing in the field to see the left panel on the card in the Scene view add the text.

The 'Narration Audio' field takes an audio file from the project containing the narration you'd like to play. This audio file will play automatically when users click the appropriate button on the top left-hand side of the card.

Adding a video

The video has its own television panel that sits on top of the side-buttons. You can toggle this television appearing on a card or not with the Video toggle.
Vuforia handles the video independently of Unity. For this reason, you must provide the name of the video so that Vuforia can find it within the project.

Specifying a video path
The video must be included in the Unity Project. The Unity Project is structured as a file system, and videos must be placed in a specific folder of the Project. Open the 'Project' window.
Find the StreamingAssets folder (if it doesn't exist, create it).
The video should be placed here. After this, in the 'Video Path' field of the Card's inspector, simply specify the video's name (with the correct file extension)



Finally, the texture that appears on the television screen when the video is not playing can be specified using the 'Pause Texture' field. These are textures added in exactly the same way as you add images to buttons.

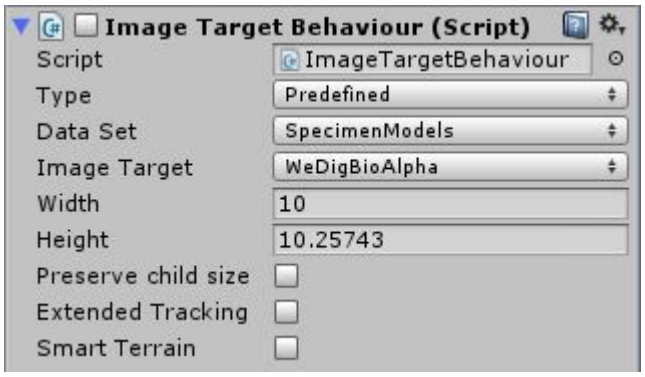
Adding a Tweet

In a way similar to the narration text, you can simply paste or type your tweet into this field. IMPORTANT: The following are already included automatically in the tweet:

iDigBio@iDigBio
" #idigbio",

Only include the actual text of the tweet – tags and hash tags are already included.

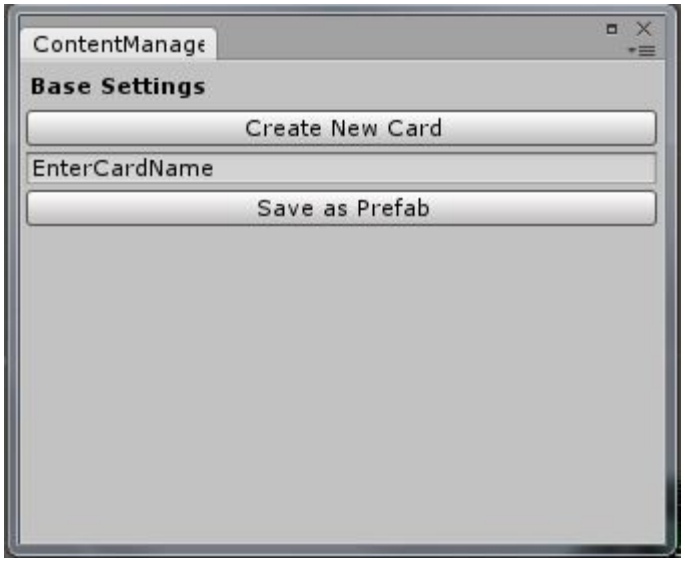
Adding a trackable
This assumes you have a dataset with the appropriate trackable already imported into Unity.



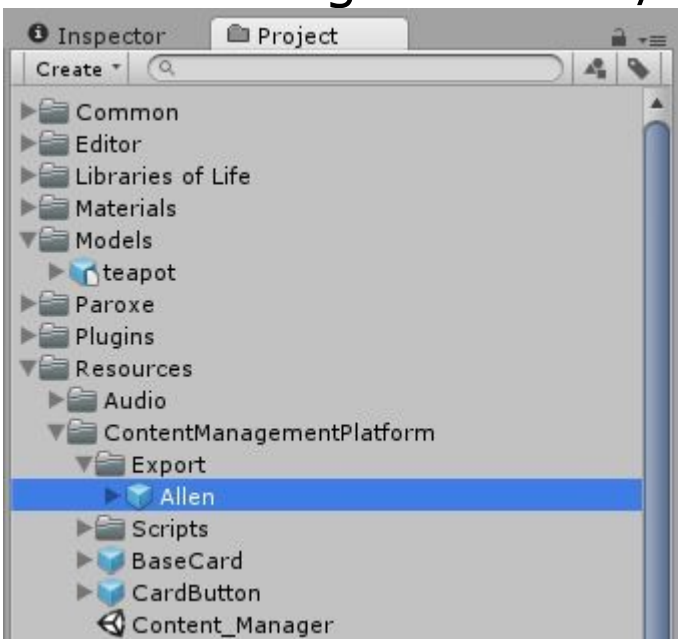
Once created, the card should come with this script attached, above the inspector containing the other fields.
We are interested in two fields:

Data Set: Make sure the data set containing your trackable is specified.
Image Target: Use the dropdown to find the trackable name corresponding to your image.

Saving your card
When you are finished editing, you can press 'Save as Prefab' as depicted below.



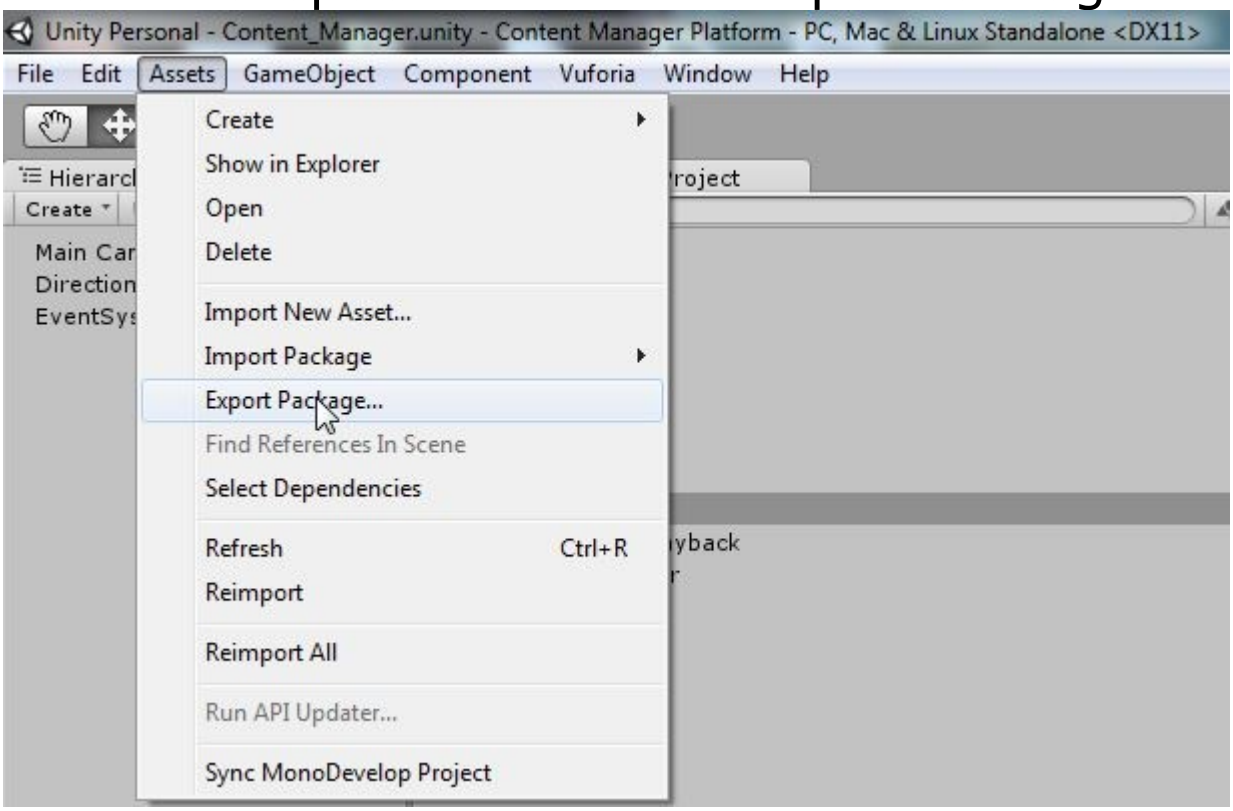
A prefab should appear in the following folder: Resources/ContentManagerPlatform/Export



This prefab can be immediately exported to any application, and is ready for use.

Exporting as a card as a .unitypackage

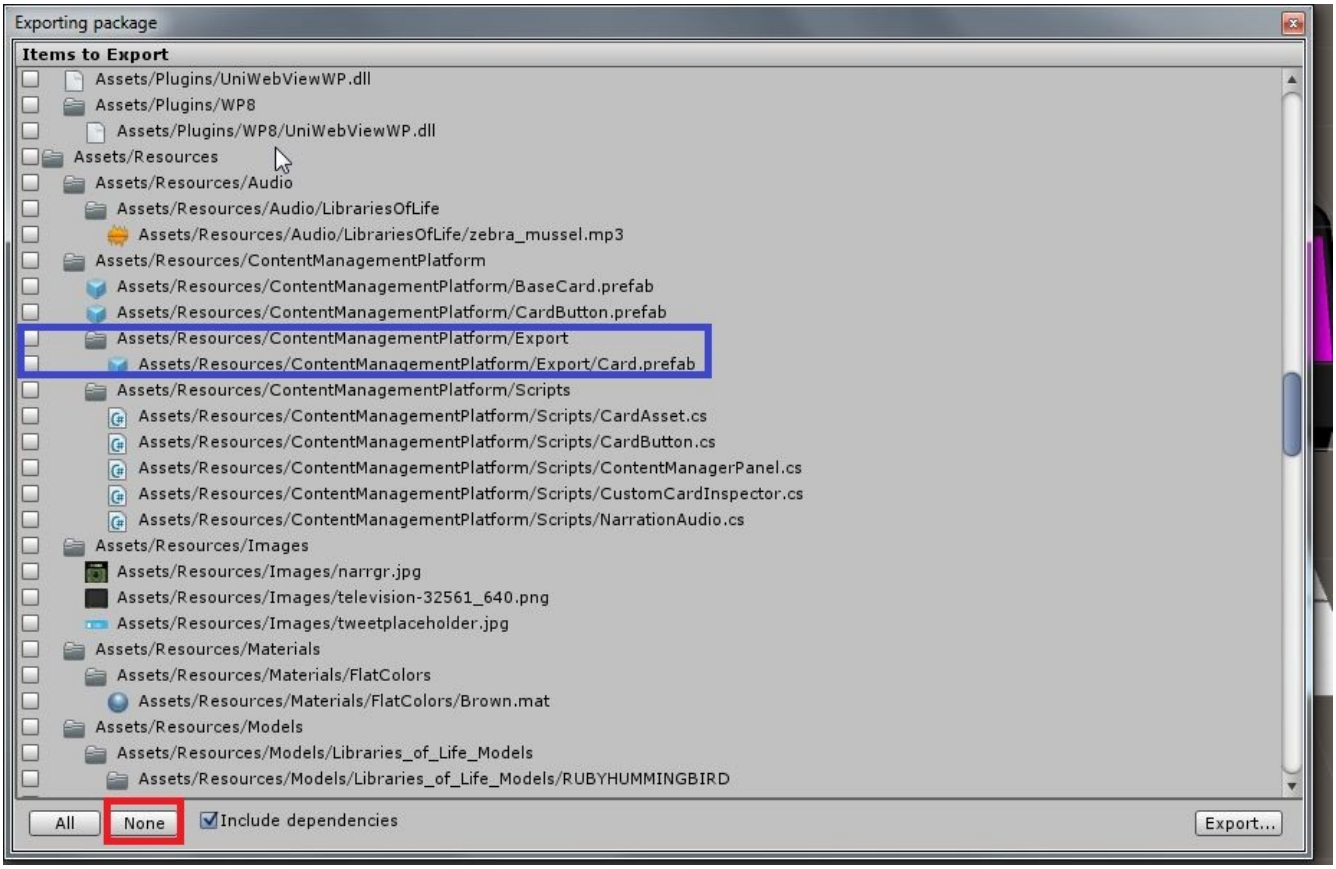
The newly created prefab can be transported to other Unity projects by creating a custom unitypackage. This file format allows one to quickly bundle the prefab for the card with all of its assets into a self-contained file which can be imported directly into any other Unity project. Go to the dropdown Assets->ExportPackage



Note: Make sure no folders are selected/highlighted in the Project view, or only these will show up in the Export screen.

Use the button at the lower-left (highlighted red)

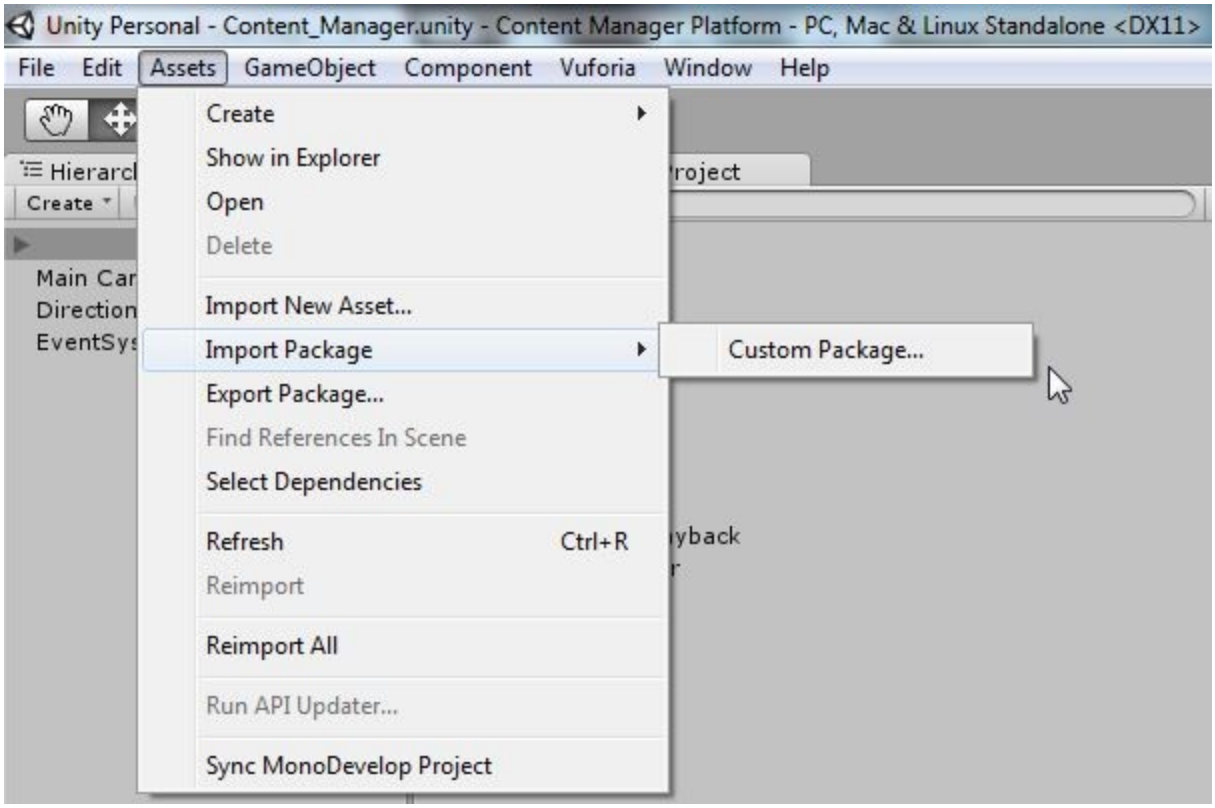
to “Select None”



Cards you save as a prefab will always appear as Assets/Resources/ContentManagementPlatform/Export/CardName.prefab

Select the checkbox next to the card to schedule it for export. **Importantly, all assets (images, models, materials, sounds etc) used on the card should be exported along with the .prefab, or the card will not work in the new project.** This is why it is important to have an organized Project structure to manage assets.

Importing the .unitypackage is straightforward. Go to the dropdown Assets->Import Package-> Custom Package... and use the file finder to navigate to the created .unitypackage



If all assets have been imported correctly, the card should be ready to instantiate in the new project.

Testing a card

These three buttons should be in the top-center of your editor. Pressing the leftmost will start Play Mode, and transfer the simulation to the Game window.
If the webcam is not enabled and the card is properly positioned, it should appear looking like below.

In either case, one can use the mouse to click various parts of the card in the same way as one would touch the components on the screen of some device.