Face Tracking in Unity Document

<http://www.cs.ccu.edu.tw/~damon/photo/,OpenCV/,Mastering_OpenCV.pdf>

THIS LINK IS IMPORTANT

# About

This purpose of this document is to track the progress I have made while studying during my professional practice module.

The project I have undertaken is face-tracking in the Unity Game Engine. During initial discussion about this project, there was a heavy emphasis on accurate jaw movement. However due to several issues that has arose while progressing through this project I was unable to find a suitable way if tracking the jaw specifically. I hope that the contents in this document can help the next person in continuing this project.

The key parts of this document is the Unity, EmguCv and OpenCvSharp sections.

# Unity

The [Unity](https://unity3d.com/) game engine is a powerful game engine which provides a wide verity of tools. Some key benefits of this engine are the augmented reality (AR), the virtual reality (VR) and the wide support of build capabilities.

However, one feature that this engine lacks is a face-tracking library. There are plugins available from the Unity store that can easily be imported to your Unity project. Unfortunately, this is a costly solution as it costs $95. More detail about this plugin can be found [here](https://www.assetstore.unity3d.com/en/#!/content/21088).

When creating scripts in Unity there is an option of two different programming options; C# and JavaScript. I have decided to program in C# as I have more familiar with the language and for other reason which will be talked about in the next section.

# OpenCv

[OpenCv](http://opencv.org/) is a native C++ library which deals with object detection. Due to its C++ nature, it is not easily implemented into Unity. A solution to this problem is to use one of the several wrappers around this library to allow it to be used in conjunction with another programming language.

When researching possible wrappers that could be used there was two criteria’s that I followed:

1. Needed to be free. (Could not afford a wrapper that needed paid licence)
2. Wrapper is in C# to be easily compatible with Unity.

Regardless of what wrapper is selected it should still be like the OpenCv functions; just with different method calls and slightly different structure names.

From the research undertaken the two potential wrappers that can be used with Unity is EmguCv and OpenCvSharp.

General information

Regardless of what wrapper is selected the main issue that will occur is the conversion between the Unity Texture2D format and the Mat structure that OpenCv would use. In the projects provided along with this document, there are a couple of different solutions to converting them, however, both have an impact on performance. Another solution is to use the OpenCv camera instead.

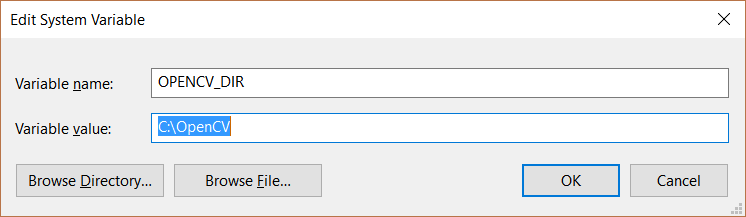
Training your own cascade

To use object detection, a xml file which contains the cascade classifier needs to be used. This file is what contains the data to help the computer know what object it is looking for. As part of the OpenCv library, there is a way to create the xml file.

To start off with you will need a series of positive and negative images, these are images with and without the object or feature you are looking for.

Installation

To use face-tracking I would highly advice installing OpenCv to your computer and adding it to your system path. The steps to do this are as follows:

1. Download and install [OpenCv](http://opencv.org/)
2. Go to Environment Variables, which can be found in system properties.
3. Create a new system variable, create a suitable name for it and add the path to your OpenCv library. For example:  
   

This should allow Visual Studio find the DLL for OpenCv allowing creation of OpenCv applications.

An easy way to find out if this is working is creating an empty C++ project. If you can access the OpenCv namespace, then it has been configured correctly.

For more information on setup of OpenCv and how to use its features please have a look at the tutorial section of their [website](http://docs.opencv.org/master/d9/df8/tutorial_root.html).

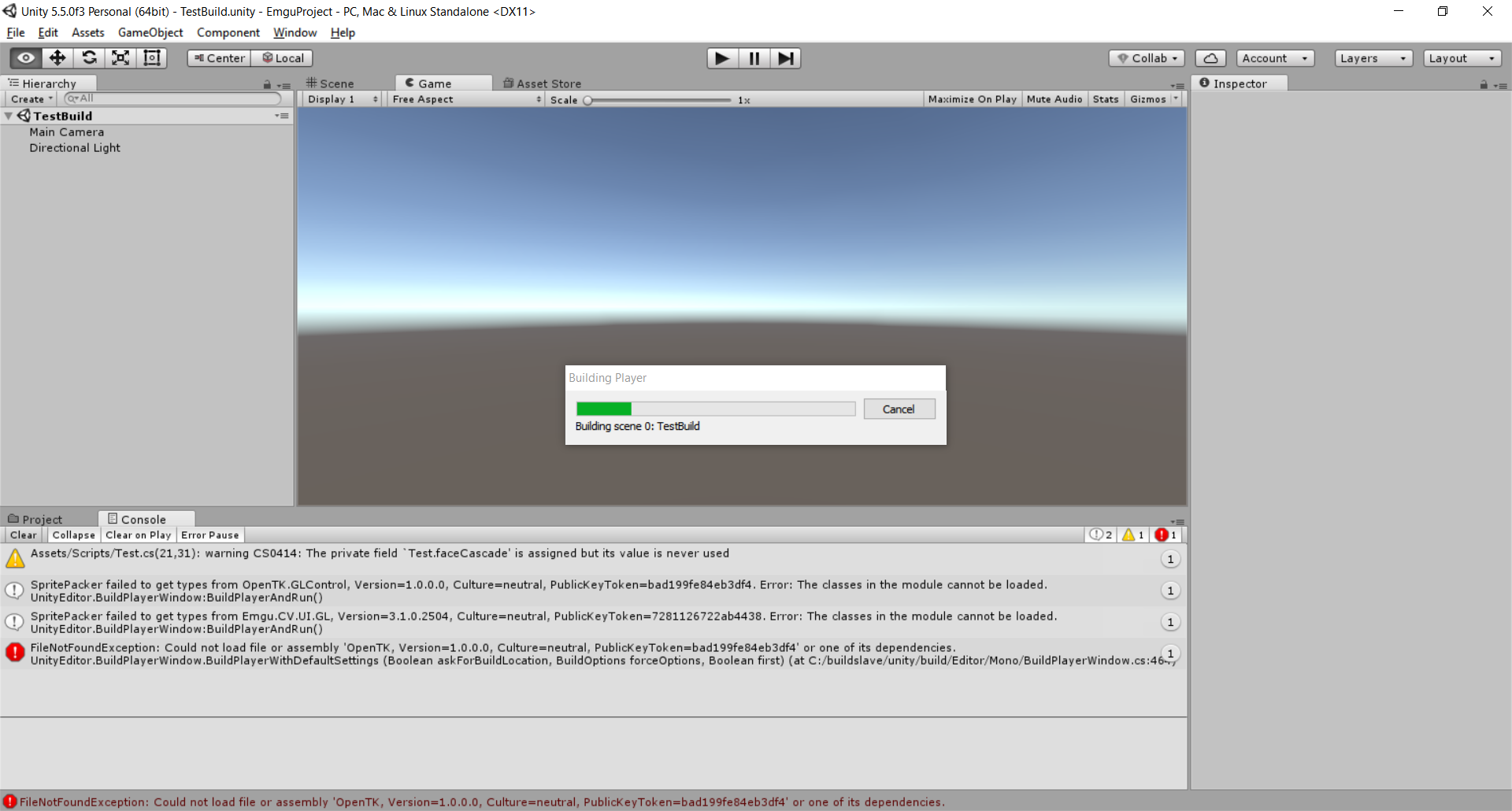
# EmguCv/Unity

[EmguCv](http://www.emgu.com/wiki/index.php/Main_Page) is the first of the two wrappers that I found while researching ways to easily implement face-tracking into Unity engine.

Installation

1. Download and install [EmguCv](http://www.emgu.com/wiki/index.php/Download_And_Installation) to a suitable location on your computer.
2. Once installed, add to the system path of your computer. For Unity, only the 32bit path needs to be included.
3. Add the required DLL’s to the plugin folder in your Unity project\*.

\* Create a folder in the assets folder of the Unity project and call it ‘plugins’. However, listing the dll’s required is in practical as it is quite a large list. What I would suggest doing is copying the plugin folder from the EmguCV project that I provide.

Remember to test if you can build the project afterwards to ensure that the setup has been configured correctly. If an error such as this occurs:

Then you are missing the required dll. To solve this, find the missing dll and copy it into the plugin folder.

The repository for the official EmguCv wrapper can be found [here](https://sourceforge.net/projects/emgucv/).

How to use

Examples

Contours

In the contours example I draw the contours found around the bounded area of where a face has been located.

Final evaluation

# OpenCvSharp/Unity

Installation

How to use

Examples

Final evaluation

# Other alternatives

Vuforia

Augmented reality application which has Unity support.

# Issues

# Links/ References.