Capture And Save (Lite)

CaptureAndSave (Lite) plugin is very useful if you want to capture current screen and save that to camera roll/gallery. There are many APIs which help you to capture screen as a texture, save your texture to gallery.

Integration Guide:

Follow these steps to integrate CaptureAndSave into your existing project

1). Import CaptureAndSave plugin into your project.

2). Check these files should be there
- /Assets/CaptureAndSave/Documentation/
- /Assets/CaptureAndSave/Example/
- /Assets/CaptureAndSave/Plugins/CaptureAndSave.dll
- /Assets/CaptureAndSave/Plugins/CallNative.dll
- /Assets/CaptureAndSave/Plugins/iOS/CallNative.dll
- /Assets/CaptureAndSave/Plugins/Android/RefreshGallery.jar
- /Assets/CaptureAndSave/Plugins/Android/RefreshGalleryWrapper.cs
- /Assets/CaptureAndSave/Plugins/iOS/libCaptureAndSave.a
- /Assets/CaptureAndSave/Prefab/CaptureAndSave.prefab

2.1).

For **Unity 5.x** users
- Select /Assets/CaptureAndSave/Plugins/iOS/CallNative.dll and uncheck "Any Platform" in inspector and check (mark) "iOS" only.

- Select /Assets/CaptureAndSave/Plugins/CallNative.dll and uncheck "Any Platform" in inspector and check (mark) "Editor","Android","Standalone" only.

3). Drag **CaptureAndSave** prefab into your hierarchy and set values in inspector.

FILENAME_PREFIX : This is name prefix of screenshot, final name will be followed by date and time.
ALBUM_NAME : Album name where all image will be saved.

5). Default directory where screenshot will save-
- Window (My Pictures) : C:\Users\<USERNAME>\Pictures
- MAC (Pictures): /Users/<USERNAME>/Pictures
- iOS : Camera Roll
- Android (with SDCard) : Pictures folder on SDCard, can be found in gallery
- Android (without SDCard) : /Data/bundle-identifier/files/(installation directory/files), not be there in gallery

Notes :

- ALBUM_NAME will be appended after default path, if there is no ALBUM_PATH then default directory will be final directory where all screenshot will be saved.

iOS Specific :

- ALBUM_PATH will not work on iOS.
- Add NSPhotoLibraryUsageDescription key in info.plist in xCode if you are using xCode 8.x.x, see this link to how to add keys in info.plist http://unitydevelopers.blogspot.in/2017/05/add-keys-into-infoplist.html

Other points to remember :

- See the Example scene for more details of function calling.
- Deploy your project on iOS to see your captured image into camera roll, on editor it will not work.
- For android, write permission should be given in Player Settings.

How to use:

 # Get reference of CaptureAndSave script

CaptureAndSave snapShot = GameObject.FindObjectOfType<CaptureAndSave>();

 # Set album path where all screenshot will save (optional)

snapShot.SetAlbumPath(albumPath);
Ex: albumPath = "D:/MyData/Pictures"; and similarly for each platform.

See : point no.(5) in integration section for default paths. If this is not set then default path will be consider.

 # Save full screenshots

snapShot.CaptureAndSaveToAlbum();

snapShot.CaptureAndSaveAtPath(string path); // save on a particular absolute path, will not work on IOS

 # Save particular area of the screen

snapShot.CaptureAndSaveToAlbum(int x, int y, int width, int height);

snapShot.CaptureAndSaveAtPath(int x, int y, int width, int height, string path); // save on a particular path, will not work on IOS

 # Save texture at path

snapShot.SaveTextureAtPath(Texture2D tex2D, string path);

Note : For IOS path should be Application.persistentDataPath\<fileName> or
Application.persistentDataPath\<Folder>\<filename>

: For Android it can be /storage/sdcard0/<folder>/<filename> or any path you want.

: For PC and MAC any path you want like /users/admin/Pictures etc.

 # Save texture in gallery

snapShot.SaveTextureToGallery(Texture2D tex2D);

 # Transfer your pre saved image from Document directory to CameraRoll, simply call this function

snapShot.TransferToCameraRoll(string path)

where path is the full url of the image saved in document directory.

 # Get full screenshot

snapShot.GetFullScreenShot()

 # Get specific screenshot

snapShot.GetScreenShot(int x, int y, int width, int height); // particular screen

***Note** :*
snapShot.GetFullScreenShot() and snapShot.GetScreenShot() will fire OnScreenShot event when screenshot ready.

***Events** :*

CaptureAndSaveEventListener.onError += OnError; // add event
CaptureAndSaveEventListener.onError -= OnError; // remove event
CaptureAndSaveEventListener.onSuccess += OnSuccess;// add event
CaptureAndSaveEventListener.onSuccess -= OnSuccess; // remove event
CaptureAndSaveEventListener.onScreenShotInvoker += OnScreenShot; // add event
CaptureAndSaveEventListener.onScreenShotInvoker -= OnScreenShot; // Remove event

```
void OnError(string error)
{
     Debug.Log ("Error : "+error);
}

void OnSuccess(string msg)
{
     Debug.Log ("Success : "+msg);
}

void OnScreenShot(Texture2D tex2D)
{
     Texture2D tex = tex2D;
}
```