# Table of Contents

- Introduction .................................................................................................................................................. 3
- Features ........................................................................................................................................................... 4
  - Ability to Create for both Pro and Free ........................................................................................................ 4
  - Easy Setup .................................................................................................................................................. 4
  - Full Window Setup .................................................................................................................................... 4
  - Capturing Moments to Unity .................................................................................................................... 4
  - Capture Importing and Exporting ............................................................................................................... 4
- Getting Started ............................................................................................................................................... 4
- Before You Start ........................................................................................................................................... 4
- Sprout’s First Scene ....................................................................................................................................... 5
  - 1. Download the Unity SproutSDK package ................................................................................................. 5
  - 2. Open the TutorialStart Project ................................................................................................................ 5
  - 3. Add the Cameras and Sprout Manager to Your Scene ............................................................................... 6
  - 4. Set Up the Mat Canvas ............................................................................................................................ 7
  - 5. Writing the Capture Script ....................................................................................................................... 8
  - 6. Setting the Timer ...................................................................................................................................... 11
  - 7. Configure Project ..................................................................................................................................... 13
  - 8. Build and Run Your Project .................................................................................................................... 15
- Developing on a Non-Sprout .......................................................................................................................... 15
  - Multiple Screens ....................................................................................................................................... 16
  - Capturing ..................................................................................................................................................... 16
- Camera Setups ............................................................................................................................................... 17
  - Separate Views .......................................................................................................................................... 17
  - Fitting Each Camera to the Screens ........................................................................................................... 18
  - Culling ......................................................................................................................................................... 19
  - Single View ............................................................................................................................................... 20
- Unity GUI Tips ............................................................................................................................................... 22
  - 1. Use ScreenSpace-Camera on Your Canvas ............................................................................................... 22
  - 2. Set Your Canvas to Match Height ........................................................................................................ 22
- Creating the Installer ................................................................................................................................... 23
Introduction

Unity 3D is a powerful game creation tool that allows you to create innovative 2D or 3D software. The Sprout Unity SDK has been built to harness this power and give you access to the innovative, creative toolset of the HP Sprout. With the SDK, you are able to access the Sprout’s Mat screen and 3D camera with easy-to-use captures.

This document helps get you up and running with the Sprout SDK for Unity. It helps you learn how to capture using the built-in camera and set up for two-screen displays.
Features

Ability to Create for both Pro and Free

Create Sprout applications on both the Pro and Free version of Unity 5.

Easy Setup

Camera and Sprout Manager prefabs make preparing a scene for Sprout simple; just drag two prefabs onto the Unity scene.

Full Window Setup

The mat and vertical screen are set up for game runtime, for small player previews inside Unity Editor, and for full Sprout player preview inside Unity Editor.

Capturing Moments to Unity

Captures can be done in Unity (with an optional configurable capture countdown); images are converted to Unity textures and can be used in scenes.

Capture Importing and Exporting

Captures can be saved to a local folder (as is done with other HP Sprout apps) and can be imported into a scene later.

Example Scene

See the example scene on how to do capturing and manipulating game objects within Unity.

Getting Started

Before You Start

You need Unity 5 or higher to run the Sprout SDK. You can download it from http://www.unity3d.com/get-unity. You also need the 1.2 or above build of Sprout and the Sprout SDK which you can download from https://sprout.hp.com/developer/develop/download/.


This guide leads you through the creation of a Sprout application with Unity.
**Sprout's First Scene**

This section helps you create your first scene with multiple cameras and capturing.

1. **Download the Unity SproutSDK package**

We have created a Unity package with everything you need to get started making a Unity application for Sprout. You can download it here: [https://sprout.hp.com/developer/develop/download/](https://sprout.hp.com/developer/develop/download/)

Note: The package supports Unity 5 and up.

2. **Open the TutorialStart Project**

On the Unity Start window click Open other and find the folder where you saved the Unity SproutSDK. Click on the TutorialStart folder and click Select Folder.

![Find the TutorialStart project](image)

Next, import the Sprout Unity asset package. In the Unity SproutSDK folder double-click on the SproutSDK.unitypackage. A popup within Unity opens; make sure everything is selected and click Import.
3. Add the Cameras and Sprout Manager to Your Scene

First, in your hierarchy look if there is a Main Camera object and if so delete it. Next find in your project tab the Sprout -> Prefabs folder. Drag and drop the Cameras and Sprout Manager prefabs into your hierarchy.
These two prefabs give you all the access to Sprout

The Cameras prefab has two premade cameras for you to use, one for the Vertical screen and the other for the Mat.

The Sprout Manager prefab holds all of the code that calls to the Sprout including capturing the mat. The default values work for this first scene.

4. Set Up the Mat Canvas

Now we want the Capture button to be displayed on the mat. Select the current Canvas and rename it Mat Canvas. Then change the Render Mode to Screen Space - Camera and drag the Mat Camera from the Cameras object to the Render Camera.
5. Writing the Capture Script

Capturing is a huge part of the Sprout experience, and it is simple to use with the SDK. We are going to create a new C# file called CaptureTest.cs. In your new file you want to change it to this.

```csharp
using UnityEngine;
using System.Collections;
using Sprout;

public class CaptureTest : MonoBehaviour
{
    private SproutManager sprout;

    void Awake()
    {
        sprout = GameObject.Find("SproutManager").GetComponent<SproutManager>();
        if (sprout == null)
        {
            Debug.LogError("Unable to find Sprout Manager");
        }
    }

    void OnEnable()
    {
        SproutManager.OnCaptureObjectReady += SproutCaptureObjectReady;
    }
}
```
void OnDisable() {
    SproutManager.OnCaptureObjectReady -= SproutCaptureObjectReady;
}

public void StartCapture() {
    sprout.CaptureMoment(false, true);
}

public void SproutCaptureObjectReady(CaptureListing listing) {
    float aspectRatio = listing.MatWidth / listing.MatHeight;
    transform.localScale = new Vector3(aspectRatio, 1.0f, 1.0f);
    GetComponent<Renderer>().material.mainTexture = listing.Texture;
}

This script captures the mat and changes the texture of the object the script is attached to in the scene.
To do a capture you need a SproutManager object and call the CaptureMoment(bool captureMatImage,
bool captureChildrenImages, string saveToFolder) function which has optional parameters.
Drag and drop the CaptureTest.cs script onto the CaptureTestHolder object.
Drag `CaptureTest` to the `CaptureTestHolder`

Next select the `Capture Button` in the hierarchy and set the `OnClick()` to the `CaptureTestHolder` with method `CaptureTest.Capture`. 
6. Setting the Timer

The last thing we want to do is create a countdown timer so the user knows when the capture is about to happen and they can move their hands away.

Click the SproutManager prefab and find the SproutManager Component. You will find a list for Countdown Textures. Place your textures in the order that they’re going to be shown. We have provided you with some default ones that you can use that are already set.
The game view
7. Configure Project

Since the Sprout uses two screens we need to configure the player to display the screens correctly. Open the Player Settings by going to Edit -> Project Settings -> Player Settings.

In the Resolution and Presentation panel we need to change two things:

1. Uncheck the Default is Full Screen option.
2. Change the Display Resolution Dialog to Disabled.
In the Other Settings panel under Optimization change Api Compatibility to .Net 2.0.
Open the Build Settings by going to File -> Build Settings. You need to change the Architecture to x86_64 to run on Sprout.

The Unity Build Settings window (Ctrl + Shift + B)

8. Build and Run Your Project

Add the current scene to the Scenes in Build by pressing the Add Current button. Now that your project has your scene and is properly configured you can click Build and Run. Choose a save location and run your project.

Developing on a Non-Sprout

You can develop for a Sprout even when you are on another Windows machine. You won’t have access to capturing from the camera but you can set up for a two screen setup.
Multiple Screens

To emulate two screens within the editor you can click Sprout -> Show Mat Preview. This opens up a dockable window of the Mat screen. You can place this window anywhere in the editor or set it up on a second screen. Note that this screen does not receive any input.

The mat preview window

Capturing

Since there is no camera to capture from, we have provided a way to use pre-made sample data. In the SproutManager prefab you can see the array of SampleCapture structs that are composed of a Texture2D and an XML file. We have provided you with some examples.
The Sprout has a unique setup for displays allowing for many configurations for your cameras, let’s look at the two most popular.

**Separate Views**

This is the most common setup where you have two cameras in your application. This can cover either two views of the same scene, or use one view for your scene and another for a GUI, or select from a range of other possibilities.

The easiest way to set this up is to use Cameras prefab and adjust each camera to the view that you want.

1. Go to the Sprout->Prefabs folder in your project view and drag the Cameras prefab into the hierarchy.
2. Configure each camera by setting their position, rotation, field of view, and culling to your choosing.
Fitting Each Camera to the Screens

To use your own cameras you need to set the Viewport Rect to fit the screens. This is so you can have each camera display on its own screen.

Click on your Camera and find the Viewport Rect. You need to change these values to match your intended result.

Here are the values for our cameras:
Culling

To make sure that the camera draws only what you want it to draw, set its culling mask appropriately to
the layers that you want drawn.

1. Choose your game object and in the inspector set the layer you want the object to be on.
2. Set your camera’s Culling Mask to the Layers that you want it to draw.

![Camera Culling Mask settings](image)

*The Culling Mask settings in the Camera Component*

**Single View**

Single view is the easiest since all you need is a single camera in the scene. This extends to both screens. Be careful about putting elements outside of the Mat screen area, as the two screens are at different resolutions (the Mat has a lower resolution than the vertical screen).

Be careful though, on the mat the resolution stretches the images so it appears bigger in real life - that is not shown in screenshots.
Single camera view
Unity GUI Tips

Unity has recently added its new GUI system; they are discussed in this document. Here are some best practices we use when implementing GUIs.

1. Use ScreenSpace-Camera on Your Canvas

To be able to set your GUI to a certain screen, it is easier to use the Screen Space - Camera setting. This allows you to set which camera draws the GUI so you can have separate ones for each screen.

The Canvas component

2. Set Your Canvas to Match Height

Unity’s GUI uses anchoring to help with multiple resolutions of screens. To help get a better view of what your GUI will look like on a Sprout, set the resolution to the screen you are targeting (1920x1080 for the vertical and 1024x768 for the mat). Next set the Match slider to Height; this helps while setting your GUI in the editor where your editor windows can change.

The Canvas Scaler component on the Canvas object
Creating the Installer

Once you are done with your Unity application you need to get it ready for the HP Sprout Store.

1. Find the directory in which you built out your app. Use the data folder and .exe file and follow the installer guide on our website sprout.hp.com/developer to create an installer.
2. Unity does have one additional step while using the Visual Studio Setup Project. Exclude all .tlb files, so your project can avoid errors while building.

API Reference

`SproutManager.CaptureMoment`

```csharp
public void SproutManager.CaptureMoment(bool captureMatImage = true, bool captureChildrenImages = true, string saveToFolder = "")
```
Description

Capture real life objects using the Sprout’s camera. When it is done processing it sends the events SproutManager.CaptureObjectReady then SproutManager.CaptureComplete when the capture is finished.

```csharp
using UnityEngine;
using System.Collections;
using Sprout;

public class CaptureTest : MonoBehaviour {
    private SproutManager sprout;

    void Awake() {
        sprout = GameObject.Find("SproutManager").GetComponent<SproutManager>();
        if (sprout == null) {
            Debug.LogError("Unable to find Sprout Manager");
        }
    }

    void OnEnable() {
        SproutManager.OnCaptureObjectReady += SproutCaptureObjectReady;
    }

    void OnDisable() {
        SproutManager.OnCaptureObjectReady -= SproutCaptureObjectReady;
    }

    public void StartCapture() {
        sprout.CaptureMoment(false, true);
    }
}
```
public void SproutCaptureObjectReady(CaptureListing listing) {
    float aspectRatio = listing.MatWidth / listing.MatHeight;
    transform.localScale = new Vector3(aspectRatio, 1.0f, 1.0f);
    GetComponent<Renderer>().material.mainTexture = listing.Texture;
}
}
**SproutManager.OnCaptureObjectReady**

```csharp
public static event SproutManager.OnCaptureObjectReady(CaptureListing listing);
```

**Description**

OnCaptureObjectReady is called after a capture has been initiated and the captured object has been processed by Sprout. The Listing class that is passed contains the image and metadata for one object from the capture. Multiple objects can be in a capture which calls the OnCaptureObjectReady event multiple times.

```csharp
using UnityEngine;
using System.Collections;
using Sprout;

public class CaptureList : MonoBehaviour {
    void OnEnable() {
        SproutManager.OnCaptureObjectReady += SproutCaptureObjectReady;
    }

    void OnDisable() {
        SproutManager.OnCaptureObjectReady -= SproutCaptureObjectReady;
    }

    void SproutCaptureObjectRead(CaptureListing listing) {
        Debug.Log(listing.Texture.name);
        Debug.Log(listing.MatPosition);
        Debug.Log(listing.MatWidth);
        Debug.Log(listing.MatHeight);
    }
}
```
**SproutManager.OnCaptureComplete**

```csharp
public static event CaptureComplete OnCaptureComplete();
```

**Description**

OnCaptureComplete is the final call when a capture is initiated. This event is used if you need to call functionality once a capture completes itself.

```csharp
using UnityEngine;
using System.Collections;
using Sprout;

public class LoadingSpinner : MonoBehaviour {
    SproutManager sprout;
    GameObject loadingIcon;

    void OnEnable() {
        SproutManager.OnCaptureComplete += SproutCaptureComplete;
    }

    void OnDisable() {
        SproutManager.OnCaptureComplete -= SproutCaptureComplete;
    }

    void SproutCaptureComplete() {
        StopCoroutine("SpinLoadingIcon");
    }

    public void StartCapture() {
        sprout.CaptureMoment(false, true);
        StartCoroutine("SpinLoadingIcon");
    }
}
```
IEnumerator SpinLoadingIcon() {
        while (true) {
            loadingIcon.Rotate(Vector3.forward * Time.deltaTime * 50f);
            yield return null;
        }
    }
}

SproutManager.IsSproutMachine

class public static bool IsSproutMachine()

Description

Check to see if user is on a Sprout Machine.

using UnityEngine;
using System.Collections;
using Sprout;

class public class CheckSprout: MonoBehaviour {
        void Start() {
            if (SproutManager.IsSproutMachine()) {
                Debug.Log("It's a Sprout");
            }
        }
    }

CaptureListing

Description

The data holder for each capture that is captured by the Sprout.
## Variables

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Texture</td>
<td>Texture2D</td>
<td>The image captured from the Sprout camera.</td>
</tr>
<tr>
<td>Mat Position</td>
<td>Vector2</td>
<td>Position on the mat in pixels of the upper left point of the object.</td>
</tr>
<tr>
<td>Mat Width</td>
<td>float</td>
<td>The width of the object in pixels.</td>
</tr>
<tr>
<td>Mat Height</td>
<td>float</td>
<td>The height of the object in pixels.</td>
</tr>
<tr>
<td>Contour Points</td>
<td>Vector2</td>
<td>List of points in pixels of the outline of the object. Used for mesh creation.</td>
</tr>
</tbody>
</table>
Troubleshooting

These are some known issues with current workarounds.

**DllImportException**

Issue: Unity is throwing a DllNotFoundException for SproutCSharpBindingSWIG101 in the editor.

When trying to use the Sprout SDK features in the editor Unity sometimes does not find the dll. To fix this, Right - Click on the Sprout folder and click Reimport. If this does not fix the issue try restarting Unity.

**Crash While Capturing**

Issue: The Unity Editor keeps crashing when I try Capturing from within the Editor.

This may be due to a Sprout Process crashing or being in use by another program. Try restarting your computer.

**Main Screen View Appears on Mat Partially**

Issue: When I play the app from editor, I see some of the main camera view showing up on mat

Sometimes the layout inside editor needs to be reset. In order to fix this issue, on the main menu go to Window -> Layouts – 2 by 3

**Conclusion**

This should get you up and running creating Sprout applications using Unity 3D. We have gone over setting up your environment, placing cameras in the scene, and using the capture functionality.

Check within the Sprout Unity package for more examples.

For support ask on the official HP Sprout developer forum

https://sprout.hp.com/developer/forum/