

Developing Content for the Web

Tom Higgins - Product Evangelist

User Experience

User Experience

Web-based content shares some of the same requirements as desktop or fixed-media (CD/DVD) content, but it requires you to prepare for and deal with them in different ways.

User Experience

Web-based content shares some of the same requirements as desktop or fixed-media (CD/DVD) content, but it requires you to prepare for and deal with them in different ways.

- Predictable Behavior

You're using a web technology that not everyone will have installed (yet!). Their first experience with your content and the Unity Web Player is critical.

User Experience

Web-based content shares some of the same requirements as desktop or fixed-media (CD/DVD) content, but it requires you to prepare for and deal with them in different ways.

- Predictable Behavior
- Smooth and Immediate
Folks browsing the web want the experience to be as smooth and fast as possible, long download times need to be avoided if at all possible.

User Experience

Web-based content shares some of the same requirements as desktop or fixed-media (CD/DVD) content, but it requires you to prepare for and deal with them in different ways.

- Predictable Behavior
- Smooth and Immediate
- Good Performance, Good Aesthetics
When dealing with a web-based audience it's important to remember that you're going to deal with a wide range of users and hardware, balancing performance and aesthetics is critical.

Development & Delivery

Development & Delivery

Let's talk about developing your content first...

File size is a critical element of any web-based content, and the largest contributors to file size are going to be the art assets you use. Additionally, it's important to remember that the audience you'll be reaching tends to be very broad and will have a variety of hardware (low-spec to high-spec).

Development & Delivery

Let's talk about developing your content first...

File size is a critical element of any web-based content, and the largest contributors to file size are going to be the art assets you use. Additionally, it's important to remember that the audience you'll be reaching tends to be very broad and will have a variety of hardware (low-spec to high-spec).

- Models

Watch your detail level and animation quality, use what's necessary to achieve your desired look.

Development & Delivery

Let's talk about developing your content first...

File size is a critical element of any web-based content, and the largest contributors to file size are going to be the art assets you use. Additionally, it's important to remember that the audience you'll be reaching tends to be very broad and will have a variety of hardware (low-spec to high-spec).

- Models
Watch your poly counts...
- Textures
Optimize the size of your textures, either prior to inclusion within your Unity project or afterwards using the Import Settings window.

Development & Delivery

Let's talk about developing your content first...

File size is a critical element of any web-based content, and the largest contributors to file size are going to be the art assets you use. Additionally, it's important to remember that the audience you'll be reaching tends to be very broad and will have a variety of hardware (low-spec to high-spec).

- Models
Watch your poly counts...
- Textures
Optimize the size of your textures...
- Audio
Use compressed audio formats where appropriate. Compressed audio formats are best used for longer format audio (background music), uncompressed are more appropriate for shorter, repeated audio (brief sound effects).

Development & Delivery

Let's talk about developing your content first...

File size is a critical element of any web-based content, and the largest contributors to file size are going to be the art assets you use. Additionally, it's important to remember that the audience you'll be reaching tends to be very broad and will have a variety of hardware (low-spec to high-spec).

- Models
Watch your poly counts...
- Textures
Optimize the size of your textures...
- Audio
Use compressed audio formats where appropriate...
- Use the Console window
When you create builds, Unity outputs a variety of data to the Console window, check there for information about how various in-game elements are contributing to your overall file size.

Development & Delivery

Let's talk about developing your content first...

Whether you're using small or large files, you can always offer an improved experience by utilizing streaming where possible. Unity 2.0 now offers the ability to stream your web player files, thus allowing you to offer your users a more immediate experience.

Development & Delivery

Let's talk about developing your content first...

Whether you're using small or large files, you can always offer an improved experience by utilizing streaming where possible. Unity 2.0 now offers the ability to stream your web player files, thus allowing you to offer your users a more immediate experience.

- When you build, select Web Player Streamed
- When your first level loads playback will begin
- After that simply check loading status of future levels using the `Application` class:

```
Application.CanStreamedLevelBeLoaded(...)  
Application.GetStreamProgressForLevel(...)
```

- When the desired level is ready, go! It's that easy...

Development & Delivery

Now, what about delivery?

Development & Delivery

Now, what about delivery?

Utilize a Unity Web Player detection routine in order to offer all users a consistent experience regardless of their platform and browser choices.

```
function DetectUnityWebPlayer () {
    var tInstalled = false;
    if (navigator.appVersion.indexOf("MSIE") != -1 &&
        navigator.appVersion.toLowerCase().indexOf("win") != -1) {
        tInstalled = DetectUnityWebPlayerActiveX();
    } else {
        if (navigator.mimeTypes &&
            navigator.mimeTypes["application/vnd.unity"]) {
            if (navigator.mimeTypes["application/vnd.unity"].enabledPlugin &&
                navigator.plugins && navigator.plugins["Unity Player"]) {
                tInstalled = true;
            }
        }
    }
    return tInstalled;
}
```

Development & Delivery

Now, what about delivery?

Internet Explorer on Windows is still the most dominant browser in use today and it is unique in how it handles “Active Content”. While Active Content is itself a long and involved topic, the net result for you the developer is not.

- When Windows Internet Explorer encounters embedded content, like Unity content, whose object and embed tags are inline within the HTML file, or whose tags are dynamically written by JavaScript found within the page, it requires the user to “activate” the content before they can interact with it.
- “Activating” the content requires the user to click on the content once to enable it to receive mouse and keyboard input (“breaking the glass shield” if you will).

Development & Delivery

Now, what about delivery?

Internet Explorer on Windows is still the most dominant browser in use today and it is unique in how it handles “Active Content”. While Active Content is itself a long and involved topic, the net result for you the developer is not.

- When Windows Internet Explorer encounters embedded content, like Unity content, whose object and embed tags are inline within the HTML file, or whose tags are dynamically written by JavaScript found within the page, it requires the user to “activate” the content before they can interact with it.
- Bypassing this need for activation is easy, simply use linked external JavaScript files to dynamically write the object and embed tags. Done!

Development & Delivery

Now, what about delivery?

While the default look, feel and behavior of the Unity Web Player is designed to have “mass appeal”, it may not work perfectly in all circumstances. Take control of the web player’s appearance and behavior to suit your needs or those of your client.

Development & Delivery

Now, what about delivery?

While the default look, feel and behavior of the Unity Web Player is designed to have “mass appeal”, it may not work perfectly in all circumstances. Take control of the web player’s appearance and behavior to suit your needs or those of your client.

- Customize the appearance of the Unity Web Player

The Unity Web Player allows you to specify the colors used by the Unity Web Player as well as to use your own graphics for the web player loading screen and progress bar. With Unity 2.0 we now allow full 32-bit RGBA PNG files.

(example on next slide)

Development & Delivery

Now, what about delivery?

While the default look, feel and behavior of the Unity Web Player is designed to have “mass appeal”, it may not work perfectly in all circumstances. Take control of the web player’s appearance and behavior to suit your needs or those of your client.

- Customize the appearance of the Unity Web Player

```
<object ...>
  <param name="src" value="MyDataFile.unity3d" />
  <param name="backgroundcolor" value="A0A0A0" />
  <param name="bordercolor" value="000000" />
  <param name="textcolor" value="FFFFFF" />
  <param name="logoimage" value="MyLogo.png" />
  <param name="progressbarimage" value="MyProgressBar.png" />
  <param name="progressframeimage" value="MyProgressFrame.png" />
  <embed ...
    backgroundcolor="A0A0A0" bordercolor="000000"
    textcolor="FFFFFF" logoimage="MyLogo.png"
    progressbarimage="MyProgressBar.png"
    progressframeimage="MyProgressFrame.png" />
</object>
```

Development & Delivery

Now, what about delivery?

While the default look, feel and behavior of the Unity Web Player is designed to have “mass appeal”, it may not work perfectly in all circumstances. Take control of the web player’s appearance and behavior to suit your needs or those of your client.

- Customize the appearance of the Unity Web Player
- Customize the behavior of the Unity Web Player

With Unity 2.0 the web player now allows you to hide the context menu, prevent the user from going into fullscreen viewing mode and/or prevent content-browser communication via JavaScript.

(example on next slide)

Development & Delivery

Now, what about delivery?

While the default look, feel and behavior of the Unity Web Player is designed to have “mass appeal”, it may not work perfectly in all circumstances. Take control of the web player’s appearance and behavior to suit your needs or those of your client.

- Customize the appearance of the Unity Web Player
- Customize the behavior of the Unity Web Player

```
<object ...>
  <param name="src" value="MyDataFile.unity3d" />
  <param name="disableContextMenu" value="true" />
  <param name="disableExternalCall" value="true" />
  <param name="disableFullscreen" value="true" />
  <embed ...
    disableContextMenu="true"
    disableExternalCall="true"
    disableFullscreen="true"
  />
</object>
```

Protecting Your Content

Protecting Your Content

Successful and popular web games attract a lot of attention which can be a very good thing, and a very bad thing...

Protecting Your Content

Successful and popular web games attract a lot of attention which can be a very good thing, and a very bad thing...

- Good: users will want to play your game and hopefully spread the word to their friends!

Protecting Your Content

Successful and popular web games attract a lot of attention which can be a very good thing, and a very bad thing...

- Good: users will want to play your game and hopefully spread the word to their friends!
- Bad: some content hosts may want to leech off that popularity without properly licensing the game for delivery on their site, piracy is a very real thing even for web games.

Protecting Your Content

Successful and popular web games attract a lot of attention which can be a very good thing, and a very bad thing...

- Good: users will want to play your game and hopefully spread the word to their friends!
- Bad: some content hosts may want to leech off that popularity without properly licensing the game for delivery on their site, piracy is a very real thing even for web games.

Why is this a bad thing?

Protecting Your Content

Successful and popular web games attract a lot of attention which can be a very good thing, and a very bad thing...

- Good: users will want to play your game and hopefully spread the word to their friends!
- Bad: some content hosts may want to leech off that popularity without properly licensing the game for delivery on their site, piracy is a very real thing even for web games.

Why is this a bad thing?

It denies you a fair opportunity to earn revenue off of your efforts.

Protecting Your Content

Successful and popular web games attract a lot of attention which can be a very good thing, and a very bad thing...

- Good: users will want to play your game and hopefully spread the word to their friends!
- Bad: some content hosts may want to leech off that popularity without properly licensing the game for delivery on their site, piracy is a very real thing even for web games.

Why is this a bad thing?

It denies you a fair opportunity to earn revenue off of your efforts.

In some cases it may actually cost you money as pirate sites utilize your download capacity which is something you pay for.

Protecting Your Content

Successful and popular web games attract a lot of attention which can be a very good thing, and a very bad thing...

- Good: users will want to play your game and hopefully spread the word to their friends!
- Bad: some content hosts may want to leech off that popularity without properly licensing the game for delivery on their site, piracy is a very real thing even for web games.

Why is this a bad thing?

It denies you a fair opportunity to earn revenue off of your efforts.

In some cases it may actually cost you money as pirate sites utilize your download capacity which is something you pay for.

What can you do to protect your content?

Protecting Your Content

- Ensure that the content itself is being delivered from your web site.
- Ensure that the content is being embedded in your web page.
- Ensure that your web page isn't being framed or included as a pop-up within another site.
- Ensure that the content comes from your server by "shaking hands" using a dynamic and encoded time-stamp.

Protecting Your Content

Through Unity's `Application` class you can check the content's absolute URL and therefore you can verify that it's being delivered from your web site.

```
// validate the game's absolute URL
if (Application.absoluteURL != "http://mysite.com/mygame.unity3d") {
    ...
}
```

Protecting Your Content

Through Unity's `Application` class you can also check the content's `src` attribute as specified in the `object` and `embed` tags and therefore you can verify that it's being embedded in your web page.

```
// validate the game's object/embed tag src attribute
if (Application.srcValue != "mygame.unity3d") {
    ...
}
```

Protecting Your Content

Unity itself does not offer facilities to verify that your content page is being displayed within a frame on another site or within a pop-up window. You can however utilize some simple browser-based JavaScript to do this detection for you.

```
<script language="JavaScript">
<!--
  if (top.location != document.location) {
    top.location.replace(document.location);
  }
  if (window.opener) {
    window.opener.location.replace(document.location);
    window.close();
  }
// -->
</script>
```

Note: the above technique assumes that the page containing your game content isn't hosted within a frame of any sort and that it isn't opened in a pop-up window. If that's not the case on your own site then you will need to either avoid using this technique or modify it appropriately to match your site structure.

Protecting Your Content

If you are using a server-side technology like PHP, ColdFusion, ASP, etc., then you can implement an encoded and time-stamped “hand shake” procedure to verify that the content is being delivered from your site.

Protecting Your Content

If you are using a server-side technology like PHP, ColdFusion, ASP, etc., then you can implement an encoded and time-stamped “hand shake” procedure to verify that the content is being delivered from your site.

- When the page loads, the server-side script should generate an encoded time-stamp and dynamically write that into the object and embed tags in such a way that the content can read and access that data.

An example being as a `src` attribute URL parameter:

```
<object ... >  
  <param name="src" value="mygame.unity3d?32923821">  
  <embed src="mygame.unity3d?32923821" ... >  
</object>
```

Protecting Your Content

If you are using a server-side technology like PHP, ColdFusion, ASP, etc., then you can implement an encoded and time-stamped “handshake” procedure to verify that the content is being delivered from your site.

- When the page loads, the server-side script should generate an encoded time-stamp and dynamically write that into the object and embed tags in such a way that the content can read and access that data.
- When the game content loads it should read in the encoded time-stamp and then submit that back to the server for validation.

In the example case of using a `src` attribute URL parameter:

```
// retrieve the encoded time-stamp
var tStamp = Application.srcValue.Split("?")[1];

// submit that back to the server
var tHandShake = new WWW("http://mysite.com/handshake.php?" + tStamp);
yield tHandShake;
if (tHandShake.data != "ok") { ... }
```

Protecting Your Content

If you are using a server-side technology like PHP, ColdFusion, ASP, etc., then you can implement an encoded and time-stamped “hand shake” procedure to verify that the content is being delivered from your site.

- When the page loads, the server-side script should generate an encoded time-stamp and dynamically write that into the object and embed tags in such a way that the content can read and access that data.
- When the game content loads it should read in the encoded time-stamp and then submit that back to the server for validation.
- The server should receive the submitted time-stamp and verify that it is within a time-window of acceptability and return a pass/fail result.

Protecting Your Content

Nothing I've suggested today is fool-proof or guaranteed to protect you so stay on your toes and be alert if this is something you're concerned about.

Questions?

Why the web?

Why the web?

- It's a powerful delivery mechanism that allows you to quickly reach a wide customer base.

Why the web?

- It's a powerful delivery mechanism that allows you to quickly reach a wide customer base.
- It's a flexible delivery mechanism that allows you to easily update your content after release.

Why the web?

- It's a powerful delivery mechanism that allows you to quickly reach a wide customer base.
- It's a flexible delivery mechanism that allows you to easily update your content after release.
- It's an accessible delivery mechanism that is within reach of every game developer.