What replaces Flash

We have four months or so before Flash goes away and the biggest question after why is it going away? Is what replaces Flash?

This is an important issue and not just for those people who are worried about the Flash games they play and what will replace them but also for people who are migrating old e-learning modules from older versions of Captivate and other e-learning tools.

So I thought it interesting to take a look and see what HTML5 technologies and APIs would replace some of Flash's functionality. It's by no means a complete list; it's what first comes to mind when thinking about migrating e-learning content. If you want to see additional elements covered, tweet the comment to @elrond25.

Note

These technologies may require additional expertise and resources to implement successfully. I'm not saying everyone should do it, I'm answering the question what HTML technologies and APIs replace Flash.

Encrypted and Unencrypted Video

In the early days, video on the web meant plugins and making sure that users had the right plugin for the type of video you wanted to play. It was also impossible to encrypt the video playing on the browser.

HTML5 provided the `video` and `audio` elements to playback media content across browsers without a plugin. It wasn't a perfect solution; since W3C members couldn't agree on a single format to support, competing interests made HTML video a daunting proposition with multiple formats required to support major browsers.

Since version 6, Flash provided encryption as part of their video offerings in the Flash platform. This made TV channels and movie studios happy since they could deliver their content without fear of piracy.

For the longest time, the web didn't provide an alternative to encrypted flash video content until Encrypted Media Extemsions came into the picture.

This Javascript API enables web applications to interact with content protection systems, to allow playback of encrypted audio and video.

Most educational material doesn't need access to encrypted media, but it's nice to know we have it when we need it.

2D Drawings and Animations

Animations and 2D drawings are (were?) one of Flash's strengths. So is there anything on the web platform that provides equivalent functionality?

The web equivalent is canvas.

Canvas can also be used to create very interesting animation like the examples below, taken from Codepen.

The first example shows mouse interactivity, using the right mouse button to rip the cloth from where it hangs.

The final example in this section is a simple animation of the earth on the solar system. The moon orbits earth and earth orbits the sun.

3D Models and Rederings

3D modeling and rendering are more complicated than the 2D equivalent. We still use Canvas to render the content but, instead of using the canvas API, uses WebGL, a Javascript API conforming to OpenGL ES 2.0 (for version 1) and OpenGL ES 3.0 (for version 2).

WebGL examples can't be easily embedded in a page so I've provided some links to examples below.

Above the clouds presents a combination of text, audio, and animations that show what you can do with WebGL on the web. This demo uses the Three.js library as a Javascript abstraction on top of the raw WebGL.

WebGL Water by Evan Wallace is a more technical exercise of what you can do with WebGL.

Interactivity

ActionScript provided a very rich language for creating interactions inside Flash applications, so what would replace this on the web?

There are four ways to answer the question of what would replace AS3.

We can use Javascript directly. Using the latest version (ES2020 as of this writing) provides a very strong foundation for interactive activities.

Dart and Typescript offer Javascript as compilation targets. Typescript, in particular, provides a strong type system to make your applications easier to debug.

The most complex option is to use whatever language you're comfortable with like C/C++, Go, C#, and covert it toWebassembly bytecode. With these technologies, we could leverage existing applications or parts of an application and make use of it on the web.

Both Unreal Engine and Unity provide exports from their proprietary code to WebAssembly so games can be played on browsers without plugins.