The Future of Browser-based 3D – No Additional Plugins Needed!

Bill Bandrowski, M.Ed., Craig Clark
Concurrent Technologies Corporation
Bremerton, WA
band@ctc.com, clarkc@ctc.com

Philip Double
Concurrent Technologies Corporation
Durham, NC
doublep@ctc.com

ABSTRACT

In the realm of educational technology, the desire for immersive user experiences mimicking real-life situations drives the use of sophisticated 3D imagery in instructional design. However, the ability to “place” the user in the middle of a scenario, common enough in the gaming world, has faced proprietary software barriers up to this point. Currently, in order to have immersive 3D game experiences, client software or browser plugins must be installed—a significant barrier to implementation within Government organizations.

With the near-ubiquitous Adobe® Flash® player no such barrier exists. According to Adobe, 98% of computers already have the Flash player (plugin) installed; users don’t even consider it a plugin. Early in 2011 Flash announced it will offer technology (graphics processing unit (GPU) hardware acceleration) and performance similar to more 3D-centric game engines.

Working with a beta version of the Flash 3D technology, the authors tested this claim. The tests included 1) a comparison of standard Flash-based 3D content with the same content migrated into the new Flash 3D player; and 2) a side-by-side comparison of 3D content (developed using the Unity 3D game engine) with the same content migrated into the Flash 3D player.

Results indicate that Flash-based content performs as well as content created in the Unity 3D game engine, hinting at significant advantages for the development of interactive training within Government organizations. Additional benefits may accrue from Unity’s recent decision to enable their 3D content to publish to the new Flash 3D player, simplifying the process of sharing high-end 3D content with the widest possible audience; this application has not yet been released.

Flash’s developer-friendly framework could make working with 3D as easy as working with video or other content, reducing the cost of development, and learners within Government organizations could access more sophisticated interactivity, serious games, and high-fidelity simulations, including games on mobile devices.

ABOUT THE AUTHORS

Bill Bandrowski is a Manager for Learning and Human Performance in the Bremerton, WA office of Concurrent Technologies Corporation (*CTC*), an independent, nonprofit, professional services company that provides technology-based solutions to clients in state and federal government and the private sector. Mr. Bandrowski has over 25 years of education and training, instructional technology development, and project management experience.

Craig Clark is an award-winning web interaction developer with experience ranging from the airline industry to the Japanese factory automation industry. He is a certified Adobe Flash Designer and a certified Adobe Flash Developer. He works as a Senior Instructional Developer in the Bremerton, WA office of *CTC*.

Philip Double is a seasoned web developer specializing in e-learning, data visualization and Human-Computer Interaction. His experiences range from work in the film and television industry to 3D meeting spaces to innovative e-learning technologies. He is currently employed by *CTC* as a Senior Web Developer, in Durham, NC.

The Future of Browser-based 3D – No Additional Plugins Needed!

Bill Bandrowski, M.Ed., Craig Clark
Concurrent Technologies Corporation
Bremerton, WA
band@ctc.com, clarkc@ctc.com

Philip Double
Concurrent Technologies Corporation
Durham, NC
doublep@ctc.com

INTRODUCTION

In the realm of educational technology, the desire for immersive user experiences mimicking real-life situations creates a push for sophisticated 3D imagery and its use in instructional design. However, the ability to "place" the user in the middle of a scene requiring situational responses, common enough in the gaming world, has faced proprietary software barriers in the world of educational technology up to this point.

Currently, in order to have immersive 3D instructional or serious game experiences, client software or browser plugins must be installed. This installation requirement is a significant barrier to implementation within Government organizations and will continue to present serious challenges.

Our training/educational developers recently experienced this first-hand in a project that designed prototype training (using the Unity 3D game engine) in which the user takes on the role of an inspector "walking" down a hallway and "interviewing" participants after a disruptive episode on the job. It was recognized that future implementation would present challenges but even stakeholder attempts to review the training were frustrated at every turn by the need for installation of software.

Prior to the beta Flash® Player 11 technology, there was no true 3D support in Flash. Several teams, including the Papervision and Away3D teams, made valiant efforts, but what they worked with was not true 3D using Graphic Processing Unit (GPU) hardware acceleration. Instead, their "3D" scenes were actually scaled and skewed 2D interpretations which often overpowered the Computer Processing Unit (CPU) and resulted in mediocre or even un-usable products. The hardware and plugin simply couldn't render quickly enough for effective use.

Help Is On the Way

Two major announcements this past year will have a significant impact on instructional game development. Adobe® announced the beta release of the next generation of Flash (code named Molehill) and Unity announced that it will support exporting its 3D content to the new Flash player. Flash developers can finally implement true 3D content and Unity developers can now take advantage of the ubiquitous Flash player.

While Flash-based 3D has been around since Flash itself, the old methods proved to be very limiting. With the new Flash player, Flash developers can manipulate 3D content with GPU hardware acceleration and achieve performance similar to more 3D-centric game engines. This means that learners can have access to greater levels of interactivity, serious games, and high-fidelity simulations, including games on mobile devices. In addition, Flash's developer-friendly framework could make working with 3D as easy as working with video or other content, reducing the cost of development.

Terminology

To minimize confusion, this paper refers to Flash Player 10 as FP10 and the new Flash 3D player (code named Molehill) by its more formal designation of FP11 Beta. However, in reporting on non-*CTC* research, reference may be made to Molehill. Unity's 3D development products version 1, 2, and 3 are all known in the industry as Unity 3D, while its player is termed the Unity Player.

Frames per second (fps) refers to the speed with which images are drawn (become visible) onscreen. The higher the number of fps the quicker the images appear and the smoother their motion, improving the user experience. And, of course, users respond favorably to high-fidelity images. However, rendering high-fidelity images decreases the speed at which the images can be drawn. Software strives to load high-fidelity images as quickly as possible without negative impacts.

THE RESEARCH PROJECT

Adobe claims that FP11 Beta will finally enable Flash 3D applications to render high-resolution 3D at up to

60 frames per second (fps) as opposed to low resolution at 30 fps and often much less. Our team of engineers wanted to validate the Adobe claims and conducted a research project focusing on FP11 Beta in order to test the Flash 3D technology.

Our engineers recognized that if Flash-based content performs in the tests as well as or better than content created in the Unity 3D game engine, the impact upon developers of serious games and instructional simulations will be significant.

Performance implications may be especially important to developers experiencing the challenges associated with plugins.

Test Objectives

Specific objectives of the testing:

- Comparison of currently standard Flash-based 3D content to the same content migrated into FP11 Beta.
- Side-by-side comparisons of 3D content (initially developed with the Unity 3D game engine) to the same content migrated into the Flash 3D framework.

Setting up the Tests

FP11 Beta is not yet released as a mature plugin but is available for experimentation. Anyone wishing to replicate the tests will need the following:

1. **FP11 Beta plugin**
http://labs.adobe.com/downloads/flashplatformruntimes_incubator.html
2. **beta Flex SDK 4.5.0.19786**
http://opensource.adobe.com/wiki/display/flexsdk/download?build=4.5.0.19786&pkgtype=1
3. **flashplayer_inc_playerglobal_022711.swf**
http://labs.adobe.com/downloads/flashplatformruntimes_incubator.html
4. Away3D Actionscript Library (beta release)
http://www.away3d.org
This library allowed engineers to avoid writing the low level instructions to the graphics card.

Test data was collected on a Macbook 2.66 GHz Intel Core i7, 4GB RAM, integrated NVidia GeForceGT 330M GPU.

Test Results

Test #1: FP10 vs. FP11 Beta

Engineers conducted tests involving a simple 3D room within the earlier version of Flash (FP10) and also in FP11 Beta. The FP10 version is produced with an older version of the Away3D library. The second version was produced on the latest beta version of the Away 3D library and running on FP11 Beta and has some incomplete elements such as clipping and proper model import. These will be fixed in upcoming releases.

In the tests, frames per second rendered, memory (RAM) usage, and polygon counts were collected dynamically as the user experienced the 3D environment.

- Fps: The number of frames per second indicates how smoothly the animation is running – lower fps rates produce choppy motion
- RAM: The drain on random access memory indicates the load on the processing unit – using excessive RAM may have negative impacts
- Polygon count: The number of polygons indicates the complexity of the animations being rendered

These statistics are visible to anyone viewing the demos online by reviewing the data box in the upper right-hand corner. Figures 1 and 2 display screen shots of the two versions of the room.

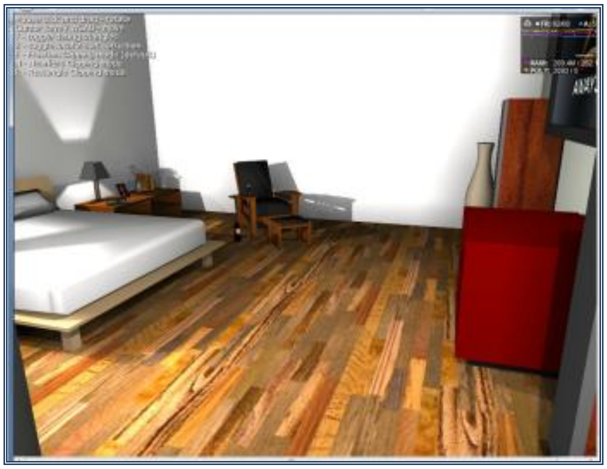


Figure 1. FP10 Version of Room

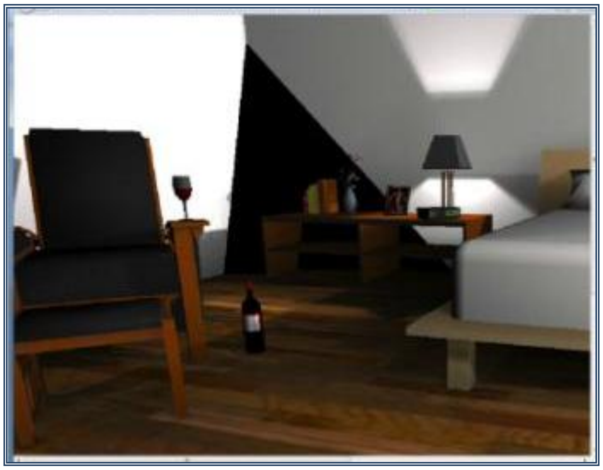


Figure 2. FP11 Beta Version of Room

It should be fairly obvious to anyone navigating the 3D room that the FP11 Beta version provides a much smoother experience. Specific data is shown in Table 1. Keep in mind that our testing indicates considerable variation across operating systems, browsers, and hardware including the GPU environment, but consistently points to excellent performance by FP11 Beta.

Table 1. Summary of Test #1 Data

	FP10	FP11 Beta
FPS	8	60
RAM	252MB	50MB
Quality	Med	High
Count	5,500	5,500

Test #2: FP11 Beta vs. Unity Player

Engineers then compared 3D capabilities associated with the FP11 Beta to those of Unity3D and the Unity Player. For the first comparison (Test #2a), engineers imported the same 3D room model into Unity, shown in Figure 3, and compared it to the one rendered in FP11 Beta.

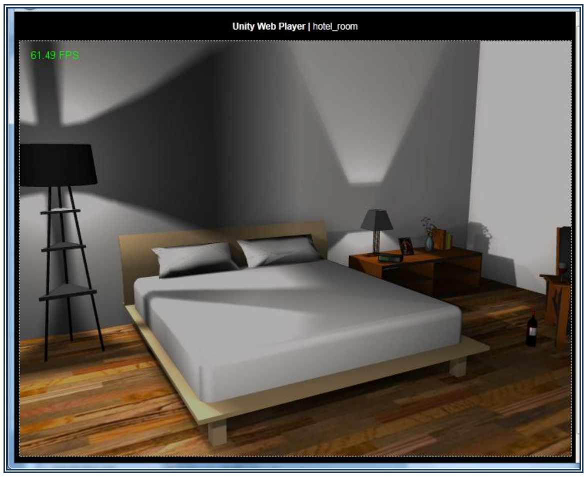


Figure 3. Unity Player Version of Room

There is no question that the Unity Player renders at a smoother frame rate, getting roughly 70fps, while the FP11 Beta version gets about 60fps. At some point, the human eye is incapable of recognizing an increase in the fps. No threshold has been defined and every eye is different but 50-60 is the general consensus. Television/video is at 30fps and film at 24fps. At 60-70fps, the user experiences smooth motion and any difference is probably insignificant. Specific data is shown in Table 2.

Table 2. Summary of Test #2a Data

	FP11 Beta	Unity Player Room
FPS	60	70
RAM	50MB	94MB
Quality	High	Very High
Count	5,500	5,500

.

For the second comparison (Test #2b), engineers imported a 3D head model into both the Unity Player and FP11 Beta and added textures to the model to further increase the load on the plugins.

Figures 4 and 5 display screen shots of the two versions of the reflective head.

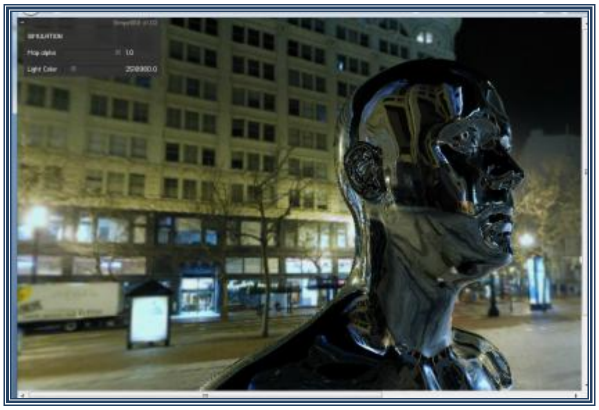


Figure 4. FP11 Beta Version of Reflective Head

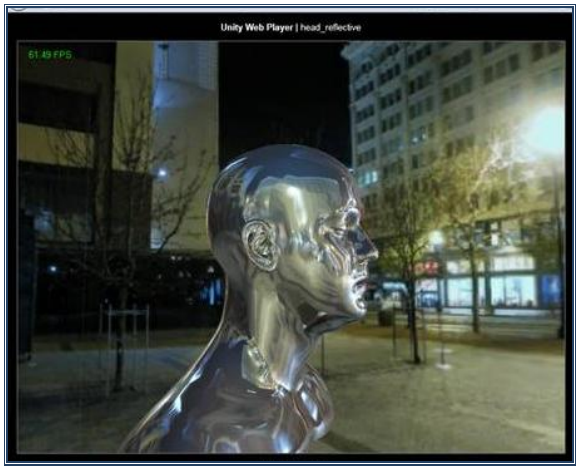


Figure 5. Unity Player Version of Reflective Head

Similar performance results were obtained in these tests which are summarized in Table 3. In both cases playback was smooth and seamless displaying a high depth of resolution, even with the fairly complex model.

Table 3. Summary of Test #2b Data

	FP11 Beta Head	Unity Player Head
FPS	60	70
RAM	84MB	94MB
Quality	High	Very High
Count	53,100	53,100

Limitations

Keep in mind that all these tests are limited and do not assess the full performance of FP11 Beta in all possible rendering situations. It is very rare that a scene consists of a single model as was tested here. More frequently there will be a multitude of less-detailed models and rendering would probably end up being slower than it was with our sample.

OTHER FLASH COMMUNITY RESEARCH

It is not surprising that, as our engineers were conducting this research, others within the 3D developer community were asking the same questions and conducting similar tests to validate Adobe's claims. Below is a synopsis of some excellent tests performed by members of the iFlash3D community that provide additional results pointing to significant performance gains using FP11 Beta.

Other Research: FP10 vs. FP11 Beta

Marco Scabia posted details on iFlash3D.com (a website dedicated to Flash 3D development), of his research in February, 2011 on the question of 3D rendering speeds when comparing the current Flash player and FP11 Beta. http://iflash3d.com/performance/how-fast-is-molehill/

Scabia posited that a valid comparison was possible and provided links to his research, encouraging others to replicate his results on other machines. He cautioned that each successive load on the computer required successively longer setup times and that this was to be expected–is focus was more on the comparison of rendering performance.

His first quantitative comparison involved a 20K triangle model of a "stone spring" using the CPU-based drawTriangles call in FP10. The resultant scene, shown in Figure 6, ran at 10 fps.

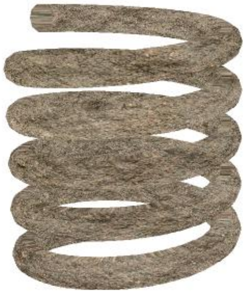


Figure 6. FP10 Version of 20K Triangle Model

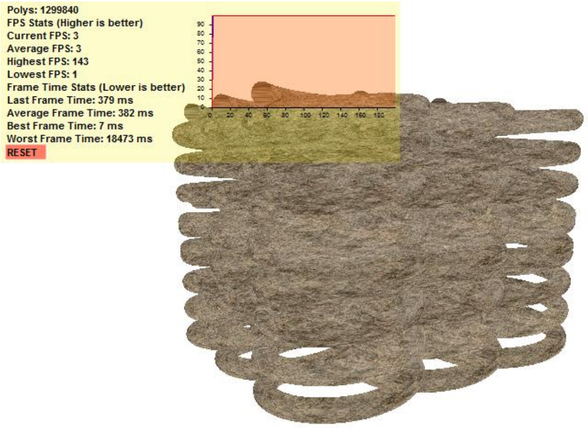
The same model rendered in FP11 Beta ran smoothly at 60 fps. http://iflash3d.com/flash-samples/mole20k/

The second test was of an 80K triangle model, which ran smoothly at 60 fps in FP11 Beta, but was not even attempted in FP10, as it would have slowed the computer to a crawl.

Using the single spring model ran into a limitation caused by FP11 Beta's Vertex Buffers–they limit models to 64k vertices. To continue testing the rendering speed of the new Flash player, Scabia switched to a multi-spring scene, and rendered a scene with 360K triangles (four springs, each with 80K triangles). FP11 Beta handled this easily as well, running still at 60 fps.

Doubling the number of springs to render a scene of 640K triangles, FP11 Beta ran fairly smoothly once setup was taken care of, though it slowed to 30 fps.

A third doubling led Scabia to counsel patience while the scene is being created. However, once uploaded to the GPU and set up, the frame rate for what he terms the Behemoth Test, was about 15 fps. With 1.2 million triangles on the screen, shown in Figure 7, this was still an impressive accomplishment for FP11 Beta.



```
Polys: 1299840
FPS Stats (Higher is better)
Current FPS: 3
Average FPS: 3
Highest FPS: 143
Lowest FPS: 1
Frame Time Stats (Lower is better)
Last Frame Time: 379 ms
Average Frame Time: 382 ms
Best Frame Time: 7 ms
Worst Frame Time: 18473 ms
```

Figure 7. FP11 Beta Version of 1.2M Triangle Model

Scabia concedes that this test was very preliminary, both in terms of its setup variables (which were limited) and the choice of scene. He notes that it's rare that a scene would be as simplistic as the one he chose to render. In spite of these caveats, the test conclusively demonstrated a dramatic improvement in the rendering performance of FP11 Beta when compared with the currently standard FP10. He concludes:

Nonetheless this test is a good rule of thumb reference point: in the simplest rendering case, Molehill can render a single textured model with 80000 triangles like a piece of cake. A scene with 320k triangles can be rendered quite smoothly, and a 640k triangles is still ok to render. A 1.2 Million triangle scene is a bit extreme, but it doesn't completely crawl the computer.

Other Research: Unity Player vs. FP11 Beta

Michael Thompson, also on iFlash3d.com, took the comparison of old and new Flash players the next step in his post of March 2011, comparing Molehill with Unity 3D. (http://iflash3d.com/performance/unity3d-vs-molehill/)

Unable to make use of the same spring models, as Unity 3D places an implicit limit of 64000 triangles per model, he chose to work with a simpler 20K-polygon spring model. In an attempt to establish a level playing field, Unity 3D's rendering optimization capabilities were curtailed, and he used shaders with no lighting, and texture only. He notes that, once the new Flash player matures a comparison of optimizations and effects will be in order, but that for the moment, it is the only rendering performance that can be fairly evaluated.

The first test went beyond Scabia's Behemoth Test, experimenting with 100 of the 20K-polygon springs, shown in Figure 8.

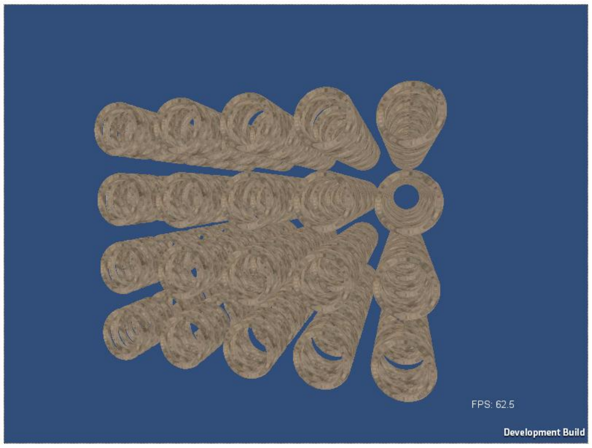


Figure 8. Unity Player Version of 2M-Polygon Model

This 2 million polygon rendering task was readily handled by Unity 3D with an average of 28.5 fps. Given Scabia's test results of 1.2 million triangles at about 30 fps, Thompson concludes that Unity 3D is "comparable to Molehill in graphics performance."

Thompson and Scabia revised their Behemoth Test to load more quickly and ran the same 100 20K-polygon springs, shown in Figure 9, that were used in FP11 Beta.

Figure 9. FP11 Beta Version of 100 20K Polygon Model

There was only a 0.5 fps difference between the rendering performance of FP11 Beta and the Unity Player.

Thompson notes that both platforms are taking advantage of the accelerated video hardware in similar ways and concludes:

Now Molehill is still somewhere in a murky pre-beta phase of development. Who knows what might emerge as a mature 3D development tool? Ultimately, a rigorous head-to-head comparison would pit two mature versions of the software – with optimizations in play – across a variety of scenes. Note, too, that as of this writing, Unity3D has a Flash-player export in the works. Will it prove to be as compelling as the native Unity3D plugin?

It might be that in the end the competition will be about the quality of the development/editing tools available or even better it might actually be about interesting content!

UNITY 3D CONTENT PUBLISHED TO FP11

As Thompson notes in his conclusion, Unity's Flash-player export may prove to be compelling. Instead of rebuilding projects like our "disruptive episode" training in the new Flash development environment in order to take advantage of the FP11 Beta plugin, developers can simply export from Unity to a .swf file. As soon as Unity releases technology to support this export, our engineers will be testing to review this compelling deployment option.

DEVELOPMENT FRAMEWORK

Another aspect of the future of browser-based 3D will be the development framework. The quality, flexibility, and familiarity of the development/editing tools will have a significant impact on adoption and implementation.

Adobe announced FP11 Beta and Flash 3D at the Flash gaming conference in February 2011. Directly after their announcement, Unity was the first presenter to take the stage and announced that they will be exporting/publishing to the Flash player (i.e., Unity building .swf files) and that they have been working together with Adobe to make this happen. (http://www.examiner.com/technology-in-san-francisco/molehill-goes-public-unity-goes-flash).

Now that Unity will output to the Flash plugin, this is a win-win situation. Flash enthusiasts can begin to develop true 3D using the framework they are comfortable with and Unity developers can continue to use their framework but deploy via the Flash plugin. The result will be more and better content to learners and end users.

At times there may have been a Flash vs. Unity sentiment, with developers choosing one side or the other and a recognition by the Flash community that the Unity approach was much "cooler and faster." But with FP11 Beta and the upcoming release of Flash CS6 supporting 3D creation, developers who are accustomed to working within Unity will continue to do so, but the doors will be opened to thousands more developers and designers who are accustomed to working within the Flash environment. Since there are significantly more Flash developers than Unity developers, it may be more economical and efficient to bring 3D to projects using the Flash framework.

In addition, if developers are creating 3D content that is not a game, but more typical content with some 3D elements, then Flash may be the best tool. Also, according to http://www.bytearray.org/?p=2810, the power of FP11 Beta does not end with 3D. Think of it as a new rendering engine tied to the GPU. If the application, website, or game is built correctly (by leveraging the GPU) then FP11 Beta will be useful in many situations.

In regards to mobile, demonstrations are now being published showing FP11 Beta running on mobile platforms. According to http://www.bytearray.org/?p=3053, FP11 Beta has been designed from the ground up with mobile in mind. Developers can leverage FP11 Beta on mobile platforms for 2D and 3D rendering. The same code is reused for the desktop version and then pushed to mobile. With the release of Flash Builder 4.5 and its ability to publish to iOS, whole new worlds of opportunity are going to be available to a very large developer base, already familiar with the Flash environment.

Implications for Training/Educational Developers

While Flash has its drawbacks, at this time Adobe Flash "owns" 2D e-learning development. However, initial tests (even though the FP11 is just a beta release), indicate that the next generation of Flash could have significant impacts:

- Allowing training/educational developers to realize their dream of high-fidelity 3D gaming and simulation content built in a familiar framework
- Enabling efficient deployment within government organizations using the ubiquitous Flash player

While the additional functionality in FP11 Beta will require that Flash 2D developers learn new skills, these new skills may represent significantly less effort than re-tooling to work in the more technical Unity 3D game-engine development environment.

Training/educational developers are empowered by this new technology, regardless of whether they prefer Unity or Flash as a development environment. This next generation of Flash offers an effective bridge between the two.

CONCLUSION

FP11 Beta appears to be a true "game changer," similar to Flash's introduction of video in 2004. This new option for creating immersive 3D content leverages an already accepted instructional tool (Flash) with an acknowledged developer-friendly framework that could make working with 3D as easy as working with video or other content.

The benefit to Government organizations is the potential for learners to access more sophisticated interactivity, serious games, and high-fidelity simulations, including games on mobile devices, concurrent with a reduction in the cost of development.

LINKS TO MODELS

The link for each model shown in this paper is listed in Table 4.

Table 4. Links to Models

Figure Number	Link to Model
1. FP10 Version of Room http://edutech.ctc.com/Bremerton3DDemo/OldHotel/HotelRoom.html	
2. FP11 Beta Version of Room http://edutech.ctc.com/Bremerton3DDemo/HotelRoom/Bremerton3D_HotelRoom.html	
3. Unity Player Version of Room http://edutech.ctc.com/Bremerton3DDemo/unity/hotel_room/	
4. FP11 Beta Version of Reflective Head http://edutech.ctc.com/Bremerton3DDemo/Terminator/Bremerton3D_Terminator.html	
5. Unity Player Version of Reflective Head http://edutech.ctc.com/Bremerton3DDemo/unity/head/	
6. FP10 Version of 20K Triangle Model http://iflash3d.com/flash-samples/20kfp10/	
7. FP11 Beta Version of 1.2M Triangle Model http://iflash3d.com/flash-samples/1200k	
8. Unity Player Version of 2M Polygon Model http://iflash3d.com/unity-samples/performance-test-20k-v2/	
9. FP11 Beta Version of 100 20K Polygon Model http://iflash3d.com/flash-samples/spring20k2/	

REFERENCES

Christensen, Ryan (2011). *Adobe Flash vs. Unity: The Near Future of Hardware Accelerated Web Games.* Retrieved May 4, 2011 from http://drawlogic.com/2011/01/12/adobe-flash-vs-unity-the-near-future-of-hardware-accelerated-web-games/

Crossley, Rob (2011) *Enemies to Friends: Unity Supports Flash.* Retrieved May 13, 2011 from http://www.develop-online.net/news/37149/Enemies-to-friends-Unity-supports-Flash

Doherty, Mark (2011). *Flash Player Versions 10-11*. Retrieved May 6, 2011 from http://www.flashmobileblog.com/2011/03/16/flash-player-versions-10-11/

Higgins, Tom (2011). Tom Higgins Talks Unity and *Flash*. Retrieved May 4, 2011 from http://learnunity3d.com/2011/03/tom-higgins-talks-unity-flash/

Imbert, Thibault (2011). *Changing the web, again*. Retrieved May 6, 2011 from http://www.bytearray.org/?p=2810

Rinaldi, Brian (2011). *Cool Stuff with the Flash Platform - Molehill 3D Special Edition.* Retrieved May 11, 2011 from http://www.remotesynthesis.com/post.cfm/cool-stuff-with-the-flash-platform-molehill-3d-special-edition-3-4-11

Scabia, Marco (2011). *The Fellowship of the Flash* Retrieved May 6, 2011 from http://iflash3d.com/news/fellowship-of-the-flash/

Scabia, Marco (2011). *How Fast is Molehill?* Retrieved May 13, 2011 from http://iflash3d.com/performance/how-fast-is-molehill/

Thompson, Michael (2011). *Unity3D vs Molehill Performance.* Retrieved May 4, 2011 from http://iflash3d.com/performance/unity3d-vs-molehill