



Author Once, Deploy Anywhere

Unity's ability to run content virtually unmodified on multiple platforms is helping indies concentrate on content first and platform second. **Thomas Grové** talks to two such users to find out how it makes a difference...

Author once, deploy anywhere – every developer wanting to avoid rewriting core parts of their engine, or completely re-authoring their content for another platform has dreamt of this.

At Unity we've taken this quest seriously and we're starting to see it pay off. The value in being able to publish your game for multiple platforms with minimal tweaks is evident: a lower cost of development and a larger potential customer base.

But that's just the beginning – it turns out that when you take the pain out of cross-platform deployment, developers are much more willing to experiment with different platforms,

spend time on new features that take advantage of a platform's unique characteristics, explore interoperability between different platforms, or to

“Take the pain out of cross-platform and developers are much more willing to experiment.”

create free-to-play versions of their games as an advertisement for the other versions.

That's exactly what Press Play – developers of the award winning *Max and the Magic Marker* – have done: they have a free to play web version of their game that entices would be customers to buy the full version; either on WiiWare or via a PC or Mac download.

But Press Play isn't alone in the laboratory; we're increasingly seeing developers taking advantage of Unity's ability to target multiple platforms.

I've asked two of these pioneers about their experiences: John Grden, a notable flash developer who recently picked up Unity to make his dream game and Hayden Scott-Baron, an artist who was recently able to 'turn indie' by using Unity...

➤ **Max and the Magic Marker**; available for WiiWare, Windows, OS X, and the web. Play it for free at: maxandthemagicmarker.com/

➤ **Tumbledrop**; available for iPhone and the web. Play it for free at: tumbledrop.com/

➤ **Star Wars: Trench Run**; available for iPhone and the web. Play it for free at: www.starwars.com/games/playnow/trench_run/

➤ To learn more about using Unity to deploy your games to multiple platforms, visit unity3d.com

John Grden

Senior Developer at Infrared5, creators of *Star Wars: Trench Run*



You created *Star Wars: Trench Run* for the iPhone and then released the web version. What does being able to author once and deploy anywhere mean to you?

As a smaller shop it means everything to us. Time equals money, and the development costs come right out of our pocket. So, if Unity didn't do this, people would probably never experience *Trench Run* on the web at all. Just because of the sheer cost and probable noticeable continuity issue we'd see between the two versions. I mean, Unity allows us not only to deploy easily, but to be able to afford it and maintain continuity all around.

How hard was it to modify your game to work on the web?

It was very easy. I made sure to keep the hooks for turning and input nice and neat. The day when we moved it over, it nearly worked right off the bat with minor code updates to strip out the iPhone specific API calls.

What types of features or content did you add, subtract, or alter?

We added real lighting for one thing – the trench features real lighting and shadows on the obstacles and ships. We added lights for the lasers that fire down the trench as well as the cannons shooting at you.

We also used the Detonator package for the explosions which look amazing; flying through the smoke and the trails of the explosions just rocks.

But beyond polish with some shaders, lighting, and control input, the game is identical to the iPhone version. We certainly didn't want to stray from that.

Anything else to brag about?

The experience went so well with this application, that we've been approved and are currently working on v2.0 for the iPhone and web which will release simultaneously this time around.

I can't give details away, but I can say that we listened to fan feedback and I think people will be blown away by not only seeing what they asked for, but the extras we've added that have not been done before in a *Star Wars* title – ever!

Also, because of how well this has gone, I now get to work on *Star Wars* and Unity full time. I haven't touched Flash in over a year now, but this is exactly the type of work I'd always wanted to be doing.

Hayden Scott-Baron

Founder of Starfruit Games, creator of *Tumbledrop*



What was the development timeframe for the web version of *Tumbledrop*?

The original version of *Tumbledrop* was put together in eight weeks, working only part time, with no prior experience of Unity.

How many different version have you made?

I've made two versions so far, one for browsers and one for iPhone. There was also an additional branded version of the browser game made exclusively for the Cartoon Network.

What kinds of changes did you make to the iPhone version?

I spent several months working on improvements for the iPhone, so the game became much more like a sequel. Tilting islands and balloon type objects introduced far more complex behaviours into the

puzzles, and the medals allowed keen players to challenge themselves for much more replay value.

One of the biggest difficulties was the lack of a mouse pointer, and realising that the player wouldn't want to keep their finger on the screen because it obscures the gameplay. The game now has the ability to queue up your next move, which allowed players to tap the screen when they were ready, rather than always waiting for the timer.

Are you considering deploying to any other platforms?

I'm currently working on *Tumbledrop Deluxe* for PC, Mac and iPad. The larger screen on the iPad means that I can create more complex levels, and the PC and Mac version should be almost identical on the iPad.

I'm also presently looking into the options for a Nintendo Wii version of the game, because of the pointer interface.

What does the ability to author once and deploy anywhere mean to you?

Developing in this way allows me to focus on the game design more than the hardware, and it gives me more opportunities to present the product to the right audience. It also affords me the freedom to change my target platform midway through the development cycle.