

develop

GAME DESIGN | CODING | ART | SOUND | BUSINESS

SEPTEMBER 2014 | #153 | £4 / €7 / \$13
WWW.DEVELOP-ONLINE.NET

ROMERO RELOADED

THE DOOM DESIGNER
ON REINVENTING
THE SHOOTER

INSIDE ▶ WARGAMING'S VICTOR KISLYI • GAME MUSIC • CLOUD GAMING • REGION FOCUS: LIVERPOOL

NEW KEY ROLE:
**TALENT
OFFICER**

INTERESTED?

contact: trw@playground-games.com in confidence.



The difference engine

Japanese middleware outfit Silicon Studio has lifted the lid on the first consumer version of its Paradox engine. **Will Freeman** caught up with the team to find out just what they hope the tech will bring to the table



Top to bottom: Lead architects Virgile Bello and Alexandre Mutel, and engineer manager Colin Magne

This is Paradox's debut as a commercial game engine. What motivated its creation?

Virgile Bello, lead architect: We started working on Paradox around three and a half years ago, and at that stage it was more of a research project. We were doing it to investigate a lot of things. One of the most important was collaborative editing, as well as shader languages, physical rendering and multi-threaded rendering.

But on top of that, we also wanted to bring our own vision to the engine, with a good mix of ease of use, performance and support for numerous platforms, allowing users to scale from mobile to far beyond. We felt that at the time there were a lot of engines that were accessible, but lacking in features or having a lot of features but without the flexibility and ease of use. We wanted to find that sweet spot, blending the two together.

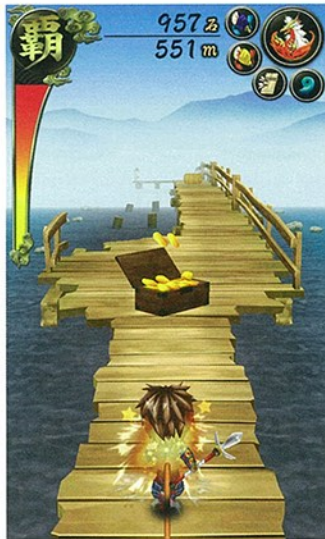
And at the very early stages, we wanted to target C#, which was kind of new at the time, and do that while giving the developer a better opportunity for productivity.

Does the engine today deliver on those early, research-phase ambitions? Or have you seen it evolve over the years?

Bello: Certainly flexibility and scalability became more important to us, especially after

“In Paradox, everything is apparent to developers and they can control it the way they want.”

Colin Magne, Silicon Studio



the first six months. We realised we wanted something that was very flexible, and not just a black box that was very hard to control.

Alexandre Mutel, lead architect: Over the last year we've been working on making this into a real product, rather than just the research project it was at first. The difference there is we started working on and digging into the new ways there are to use a game engine. We've spent a year making Paradox something that provides a good experience, so we are ready for the public version.



Bello: It has already been used for one commercial project, which is already on the market. That allowed us to test the engine in the real market, and we have grown the team far bigger so we can support our users.

And you've mentioned flexibility. What exactly do you mean?

Bello: We didn't want Paradox to block users at any part of the process of making a game.

Colin Magne, engineer manager: What we mean by flexibility is the access to the customisation of the engine. That customisation is very easy, and the freedom of the programmer there is very high, as is the freedom of the artist.

Flexibility means a user can reach their goal without barriers on the way. It's the black box thing we mentioned. In Paradox everything is apparent to the developer, and they can control it the way they want without being restricted.

So it's integral to Paradox that the engine is open to extension and customisation by the user?

Bello: Yes, and, unlike some engines, you won't need a big team to do that. Paradox targets those wanting a flexible engine that is suitable for customisation by a small team.

Mobile is clearly also very important to Paradox. Why and how do you cater to those platforms?

Mutel: It's important for many reasons, but partly because the mobile industry has been going so strongly in recent years. Mobile



gaming has been increasingly important to Silicon Studio.

Around two years ago our CEO made a clear statement to the company; that we should go forwards towards more mobile development. So when we started work on Paradox, we were mostly concentrating on high-end features; those that were most common on high-end console and PC.

Then quite quickly we saw what was happening with the mobile market, so we looked at targeting those developers as well, meaning we could offer a competitive product to developers looking to cross-target many platforms.

Paradox has been built with everything that makes it easy to develop in a cross-platform way, and that includes mobile.

Bello: And we have a longer-term vision beyond mobile and console, to target browser maybe. We are considering any platforms developers want to target.

Magne: And early on we were doing a lot of research into high-end graphics. This really built up a strength in the platform, and it was one we could keep when we started to move the engine more towards supporting mobile. Paradox is really strong in terms of its architecture supporting high-end graphics, but it is also optimised for mobile. We believe it is future proofed in that way, as smartphone platforms evolve and offer developers much more power in the future.

Going back to the focus on C# and the support for .NET libraries, why is that of particular relevance to developers in today's games industry? Why does that matter so much?

Bello: It comes down to the fact that we wanted to offer our users power, but in a place where they could be very productive too. C# allows us, and our users, to have very fast iteration times when they are processing ideas and concepts.

It means they can integrate new ideas as they develop, and it will give them much better ideas as C# will allow them to be so productive, giving them more time. It frees up more time for developers to look at bug fixes, memory leaks and so on.

On top of that it brings some nice features, such as asynchronous scripting, which will let a developer work on a second task while waiting for results from another, for example, and in much nicer ways than many other languages allow.

You've also conceived the engine to support collaborative workflows. What does that mean in the case of Paradox, and how is it realised?

Bello: Part of our general vision from the start was to support collaborative development. It's something we will introduce more over further releases; during the research phase we were looking at things like collaborative real-time editing, but that's something that needs more time.

But in the first release we have an asset management system that works well for versioning. It's easy to understand and use, everything there is text-based, so it's easy to read and merge, and all assets are kept as



The Paradox engine has been developed as part of Silicon Studio's research into high-end graphics, something the firm is determined to bring across to mobile games

"We are open to having people help us make Paradox even better so the engine will be free."

Alexandre Mutel, Silicon Studio

separate files on the hard drive. It should help with all versioning concerns.

Magne: Artists, coders and planners using Paradox can work on games without blocking each other. That's the overall impact of our versioning and asset management features.

Bello: And in the future we would like to implement real-time collaborative editing in a scene editor, so that developers can edit together. That is a longer-term goal.

Similarly, the DirectX 11 support must be something you hope will appeal to a broad spectrum of developers?

Magne: The Paradox engine's architecture is basically equivalent to the DirectX 11 feature

set, so even though we target mobile, the architecture is DirectX 11. What that means is that in the close future we will see – with new releases coming – things like compute and tessellation added. So Paradox is already ready for that.

So with all these features, are you aiming Paradox at everyone from indies to large-scale teams?

Mutel: We are aiming this more at the indie teams and small-to-medium-sized studios. We are not, for now, targeting triple-A teams, but we will, in the future, try to attract those larger studios. But that is quite far away from today. For now we're offering Paradox for free to small and medium-sized teams. For the first versions the whole engine will be free, even for commercial projects, though that will change over time as we introduce more features.

Further down the line, we will work out offering commercial and open source licenses. The whole engine will be made available as an open source engine, so all will be able to contribute to it, perhaps through something like a GitHub project website. We are very open to having a wide range of people help us make Paradox even better. ■ www.paradox3d.net