

Postcard from GDC Europe 2005: The First Generation of The Next Generation: *AProject Gotham Racing 3* Postmortem

By Kieron Gillen

It's that time again. The first generation of next generation developers who've hacked and slashed their way through a world of still-in-development Software Development Kits (SDKs) appear before their peers to bring a simple message: the next generation is *hard*.

The lecture was presented by Bizarre Creations' effusive tag-team of associate producer Chris Pickford and design manager Gareth Wilson. At the time of speaking, they happily admit they're still inching towards the final release in time for Xbox 360's launch window, so cheekily subtitle the lecture "Flogging The Horse Before It's Dead." That the process isn't over yet doesn't mean they haven't useful experiences to share about the journey so far. Their lecture took the form of a meander through the various aspects of development and the lessons they learned in each.

Practical Issues

Pickford and Wilson stressed the importance of keeping open the lines of communication, both with other developers and the console manufacturer. For the latter, it's essential you make SDK bug reports as quickly as possible, due to the irregularity of SDK updates. Don't expect the bug to be reported and fixed by someone else, or you could find the SDK two-months down the road still possesses it.

For the former, when you hit a wall, a trip to one of the developer forums can reveal that someone else has already ran face first into that bricky surface and already found a way to chip through. In particular, the developers stressed that having the game running ahead of time, even on the Alpha kits or PCs, and having a modular development system to allow turning off whole sections when SDK fixes cause it to stop working.

Wrestling with content is the second section. You cannot underestimate the amount of artwork you'll require – and this amount of artwork requires some serious server space. Bizarre spent £80,000 upgrading theirs, with separate servers for the 20,000 photos of each city in their game, the models and so on. Similarly, with the size of art assets, export time causes problems. Tools to allow quicker testing of how a model may look in game before exporting saved huge amounts of time.

Similarly, the *Project Gotham Racing 3* creators described their successful experience with outsourcing. They relied on a demo-test to make sure the outsourcers were capable of the task required, then the writing of an extensive tutorial brief for all the staff. The investment of time here of a couple of weeks to make it properly thorough will pay back by having the outsourced material require the absolute minimum of fixes. After all, what's the point of outsourcing if the resultant assets require such extensive examination and reworking when they get back as to almost equal the time it'd have taken your art staff to build them in the first place?

Interfacing Properly

On the coding side, specific attention must be paid to user-interface issues. With the increased importance of online and multi-media aspects for the Xbox 360, the ability for these to be easily adjustable on the fly is absolutely paramount. To that end, extensive use of high-level scripting language was used to link together the front end, with the backend in C. Similarly, game-logic issues. With so much to worry about elsewhere in terms of coding, it's important to gain time and flexibility here. Other coding issues centered around good practice – sensible variable naming, not hard coding variables, commenting code, etc - which have been always important but are even more so with the increasing size of projects.

In terms of process, Bizarre made a mistake of not nailing down their essentials early enough, such as memory management or vision checking. Later, they needed to rewrite the whole thing, which led to two weeks where no artist could genuinely see if their material worked in-game. On the more successful side, they had two game versions constantly available, selectable from the boot screen. Either a fast or a pretty version. The pretty version has all the graphic modes on, existing for screenshots and to remind people what it'll look like after optimization. The fast version can actually be played properly, though often with grey-untextured cars. Both aspects of the game will be brought together for final release, but you'll never have them both simultaneously prior to that when making a next-gen game. They also kept a safe-build available which while not be the latest code, was relatively stable and could be showed to any journalist who turn up unannounced. The fiends.

Design Pitfalls



In terms of design, it's important to remember the absolute fundamentals: you are making a game. Easily said, but it's easy to get distracted by all the possibilities of the hardware. They embarrassedly reveal the bug where cars started 30 feet in the air, facing the wrong way around, which existed for months in the game, until artists found a work-around.



Bizarre Creations' *Project Gotham Racing 3* for Xbox 360 in action.

It helps if there's a formal meeting between the design and coding team heads once a week to make sure things are synching. However, they also warn of the possibilities of too many people at too many meetings. Meeting length increases exponentially according to how many people actually are in the room. Carefully chosen, articulate team-members who can disseminate the ideas to the rest of their group are a must.

Also, allow three months for balancing. Or – as Chris coyly put it in mock quotations, “three months,” recognizing that this will vary wildly. However, if you actually plan for this, it becomes a real deadline in the minds of the team, so motivates them to get the work done early. In standard development, they run a cycle of design, implement, play, then iterate. When they reach the endgame, it alters to experience, evaluate, polish then iterate, with a group of 10 people dragged in off the streets to give this unjaded feedback. It's something bearing in made throughout – they advised getting the artists to play the game, as they'll notice things which the designers never would. They use the example of the roadside turn indicators, which the designers were used to spotting but none of the artists were able to make out, so all went crashing into the first bend. Problem located, and then fixed.

However, as an ultimate lesson, the Bizarre Creations duo return to an earlier, extremely valid point. When you're making a next-gen game, don't forget that it's not the “next-gen” which is the most important part but the “game”. An obvious lesson, perhaps, but it's something that can't be said enough, and something we'll find out the results of this holiday season for *PGR3*, hopefully.

[Return to the full version of this article](#)

Copyright © 2016 UBM Tech, All rights reserved