

## Postmortem: Black Rock Studios' MotoGP'07

By David Jefferies

[In this Gamasutra-exclusive postmortem, David Jefferies of UK-based Black Rock Studios (formerly known as Climax Racing) follows up [last year's postmortem](#) of MotoGP'06 with an honest look at this year's THQ-published Xbox 360 iteration, covering optimization, frontend, crowd systems and more.]

*MotoGP'06* was released in June 2006 for the Xbox 360. Developed by Climax Racing and published by THQ, it enjoyed good commercial and critical success including receiving a BAFTA nomination. As soon as we'd got it out the door we started work on its sequel, *MotoGP'07*.

As it transpired, it was to be our final outing with *MotoGP* franchise because Disney acquired Climax Racing in the fall of 2006. Our name was changed to Black Rock Studios and we were tasked with working on original IP for Disney. But before that, we needed to finish *MotoGP'07*. Now that this was our swan song, we were determined to end on a high.



## What Went Right

**The Spectacle** - One of our objectives on *MotoGP'07* was to try and get across more of the spectacle of the race. The tension, noise and hubbub on the grid just before a real race are incredible, as is the color and palpable excitement of hundreds of thousands of fans. I think we all recognised that one of the shortcomings of *MotoGP'06* was its lack of dynamism around the track -- the tracks looked great but they could also be very static.

In order to achieve this spectacle we invested in pre- and post-race cutscenes, a new crowd system and some new audio to go with it.



The crowd system in *MotoGP'06* was a mixture of a few dozen animated instances and tens of thousands of static billboard characters rendered using the same technology as our volumetric grass. This time we wanted to have tens of thousands of animated crowd members decked out in the colors of their favorite rider, and when you got close to them they needed to turn into 3D models.

Crowd systems can be controversial on development teams because they require a large investment yet don't provide any gameplay advantage. You also can't see them when you're powering along at 200mph. All that is true, but we believe that, especially for sports titles like ours, they provide an essential level of immersion that just isn't there with a load of static billboards.

The system was based on a highly optimized CPU skinner that meant all the 3D crowd members could be rendered with a single draw call, coupled with an impostor system, which meant all the billboard characters in the distance were always animating. The system applied

random clothing, skin tones and accessories (sunglasses, cameras etc) to the characters but the artists could also specify grandstands as having allegiance to certain teams and riders so, for instance, all the Ducati fans would be stood together in a grandstand wearing their red shirts.

---

As well as the crowd we included cutscenes, rendered in real time using the game engine, to add to the pre-race build-up and to capture the post-race celebrations. These turned out a big success, but as we'll see in the What Went Wrong section did not go as smoothly as hoped.

**Frontend** - We all know the frontend is a banana skin upon which even the best-run development teams can slip. It's common to underestimate its scope, and it seems to be a common testing ground for unwise software methodologies or ill-considered experiments in data-driven software. Some of these problems are caused by lack of experience in the frontend team, because all too often it's staffed by graduates, while the more experienced coders and artists work on the game.

Here at Black Rock we've made all those mistakes in the past before eventually settling on a formula for the *MotoGP* frontend that worked. Data-driven software is usually a good thing, indeed much of our game is data-driven, but with a front-end the coupling between the data and the game code is particularly complex. Instead of this we employed a more coder-driven system where the screens were defined by a set of templates and as the templates were updated all the screens using them were updated. This means the programmers and artists had to work very closely together.

The original *MotoGP* had won an award for its implementation and we had ported it across pretty much verbatim for the *MotoGP'06*. It was clear and logically laid out, but in the intervening 6 years things had moved forward and it was looking quite tired.



The first thing we did was hiring a specialist graphics designer, which was something we'd never done before. He had a lot of experience working for large companies such as U.K. mobile provider Orange but had never worked in the games industry. I think this worked to everyone's advantage because he was able to take a totally fresh look at how we visualize our frontend and came up with a very clean and compelling style.

Secondly, we decided not to change the functionality of the frontend. We liked the way it worked, just not the way it looked. In those situations it's tempting to start implementing from scratch, but instead we opted for a re-skin. Every screen was completely redesigned from a visual point of view, but still maintained the same functionality.

In this way we were able to break one of our golden rules and staff the frontend code team with a single graduate. A very good graduate, for sure, but still a grad. Code and art made a great team and even colleagues with years of experience playing *MotoGP* were fooled into thinking we had created an entirely new frontend.

---

**Optimization** - One of the things that *MotoGP'06* was criticized for was the screen tearing. This occurred when the game wasn't able to render the scene in one 60<sup>th</sup> of a second. At this point there are two options. Firstly, the game can display the scene as soon as the rendering is finished, or secondly it can wait until the next video refresh.

The problem with the first solution is that the TV may be halfway through showing the previous scene and so you get an image that is half old and half new -- hence there is a tear at the boundary between the two. The problem with the second solution is that you've just halved your framerate, which can be crucially off-putting to the player, especially if they are navigating a tricky corner.

We opted for the first route judging that it was more important to the player to keep the gameplay intact than have some screen tearing. Unfortunately the paying public weren't happy that their brand new LCD HD displaying what appeared to be graphical corruption.

Maintaining 60 FPS on a near-launch title is hard because we only have final hardware for a very short amount of time before the game's release. As developers, we were pretty happy with maintaining 60fps for 80 or 90% of the time.



However, the public, quite rightly, when they are paying £50 or \$60 for a game, don't want the developer to be "pretty happy" with their effort. They expect them to get it right.

Optimization is always a big deal to us but this time we also had a point to prove.

In order for the artists to improve the look of the tracks, we had upped their polygon budget from 500,000 per track to 750,000 per track and this -- combined with the additional spectacle requirements -- meant we were drawing twice as much geometry... and we needed to draw it faster than *MotoGP'06*!

This time round there was no silver bullet, because our engine had already been through four months of optimisation on *MotoGP'06*. We had to go through every bit of CPU code, reducing complexity and increasing cache coherency, and put a huge amount of effort into our asset exporters so that by the time the geometry reached the game engine it had good LODs and used the minimum number of shaders.

Our tracks are modeled using higher-order surfaces, which means we can build flexibility into our exporter and tessellate them down to a pre-determined number of polygons. By the time the tracks reach the game they consist of a total of about 30 million polygons, which would be far too many for the artists to optimise by hand.

---

## What Went Wrong

**Cutscenes -** The pre- and post-race cutscenes were something our studio hadn't tackled before. A decision we made early on was that we wanted hundreds of people on the grid -- so we were going to concentrate on quantity rather than high polygon count. At the time we didn't have the technology to render perfectly featured faces and hair, so we wanted to keep the camera cuts quick and the character count high.

We had 18 races each with pre- and post-practice, pre- and post-qualify and pre- and post-race cutscenes. That's 108 different scenes. We needed to use the in-game engine for the scenes rather than recording them, so that we'd be able to correctly represent the player's place and team on the grid. We also didn't want a load screen between cutscene and race, so all the assets had to be loaded up front.

The cutscenes were a departure for our studio, and we underestimated the workload. We had a single programmer whose job it was to write all the cutscene art tools and the in-game support and a single animator working with a character artist. The workload for this small team was just way too high and unfortunately we were never able to free up additional resources to help them out.

This meant the team ran late and were subject to unfair amounts of stress. That the cut-scenes turned out so well is testament to their skill and perseverance -- the number of characters on the grid is staggering and it really adds to the sense of immersion.





**Buils** - The build was in the 'What Went Wrong' section of the *MotoGP'06* post-mortem so its indefensible that it appears again. One thing that the game development community is learning is that that building the volume of assets required for 360 and PS3 games is a tough problem.

We made a serious effort to engineer a good build system for *MotoGP'07*. A system that could check out the code and assets, build the code and assets across multiple build machines for multiple SKUs, run unit tests on the code, run a series of tests on the game to make sure it was operating correctly, email us a status log, build DVD discs, archive a build environment (an archive of all the data, code and tools which was everything you need to build assets and run the game) and if the data was good (i.e. the automated tests had passed) it pushed the new version of the game out to all available devkits. It did this in less than five hours at the push of a single button. Sounds quite good, doesn't it?

Unfortunately it wasn't quite what we had hoped. Fundamentally our build system is built on top of Scons (a replacement for make) and Cygwin (a Linux-like environment for Windows) and it stressed them beyond their limits. In order to get the build time to less than five hours we issue tens of thousands of build commands over a large number of CPUs, and we exposed two rare bugs in the third party software.

Every few thousand times the fork command in Cygwin would fail and similarly Scons would hang. We spent weeks debugging the problem but never managed to solve it, and instead ended up with an unsatisfactory tangle of error-catching and retrying in the hope of getting some good data.



Our Core Technology Department is solving the problem by rewriting the build system using Python rather than Scons and eliminating the need for Cygwin.

Fortunately for us we had an enthusiastic build engineer who was happy to baby-sit the build, often after-hours, tinkering with it until it gave good results. As you can imagine this was far from satisfactory (especially for him!) A poorly-operating build system compromises the quality of the game and costs the team late nights and weekends and it is, therefore, inexcusable.

## Conclusion

We've made five versions of *MotoGP* in the past seven years and although we love the franchise I think we are all ready to move on. For the final game in the series we wanted to tie up all the loose ends, implement all the suggestions from the fans and really step up the graphics and handling to produce a fitting finale. With *MotoGP'07* I think we achieved that. We'll miss the races, the riders and the brolly girls but we'll leave the series knowing a lot more about motorbikes than when we started!

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

Copyright © 2016 UBM Tech, All rights reserved