Hello, and welcome to Amazing Feats of Daring.

This talk is a post-mortem for *Uncharted: Drake's Fortune* which is Naughty Dog's first game for the PlayStation 3.

I'll be giving you an overview of how we work at Naughty Dog, the challenges that we faced in developing the game and the things that we struggled with, some of our successes, and the things that we plan to do differently next time.

My name is Richard Lemarchand, and I'm a game designer, originally from the west of England.

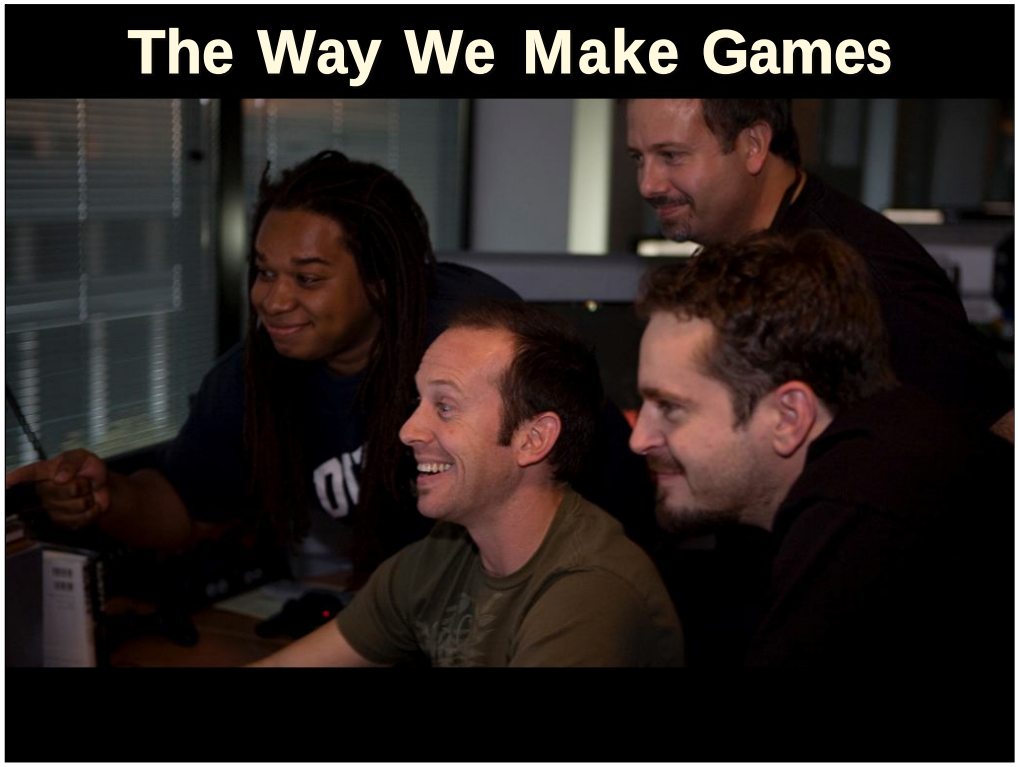
I've made character-action games the main focus of my career, although I've worked on some other kinds of stuff - these are the games that I've helped ship.

Naughty Dog is a small game studio of about 80 people based in Santa Monica, Southern California.

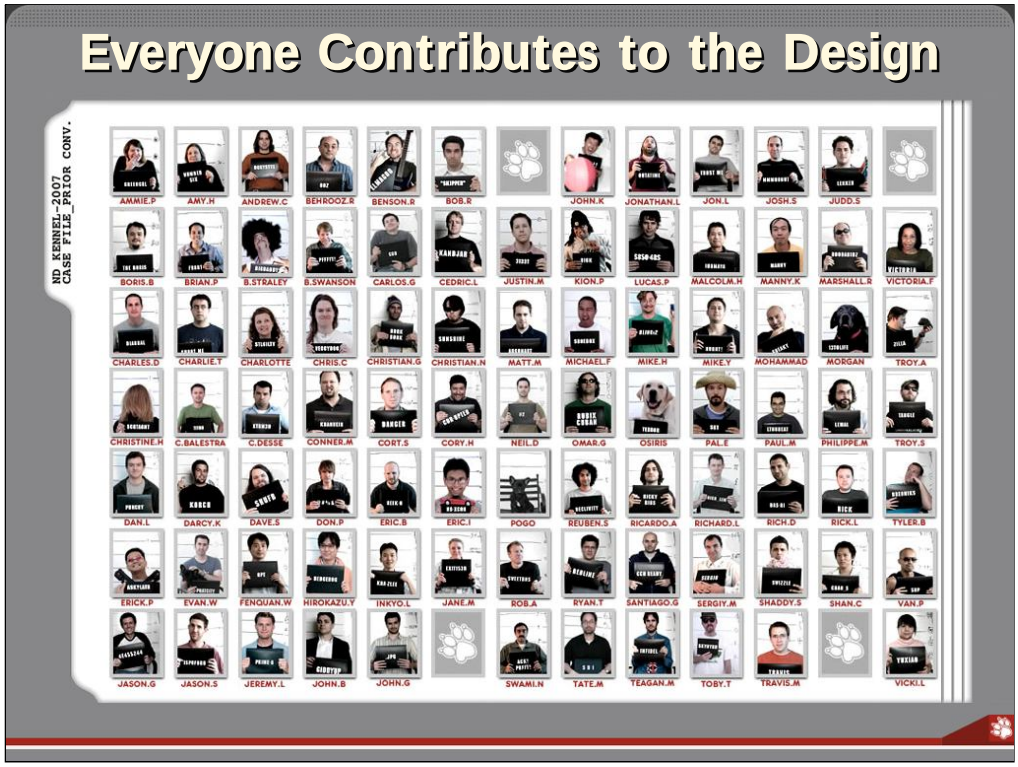
We were founded in the mid-'80s by those excellent chaps Jason Rubin and Andy Gavin while they were still at high school.

We are the creators of *Crash Bandicoot* and the *Jak and Daxter* series of games - the *Jak* games have sold over 7 million units worldwide.

We've been wholly owned by Sony Computer Entertainment since 2001, and the studio is now headed up by Co-Presidents and long-time Naughty Dogs Evan Wells and Christophe Balestra.



So before we get going I want to tell you a bit about the way we make games at Naughty Dog, because we think it’s a bit different from the way that a lot of people in the industry do things.



One of my favorite things about Naughty Dog is that it has an open, meritocratic studio culture.

Ideas and constructive criticism about the design and production of whatever we're working on are always welcomed from absolutely everyone on the team.

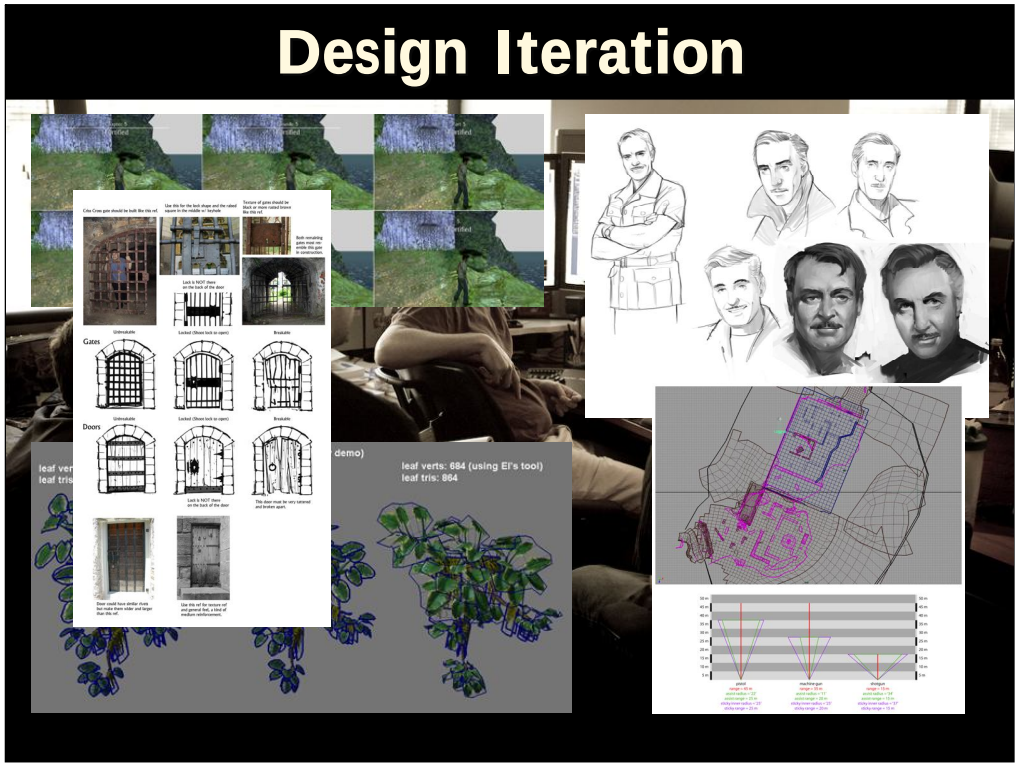
No Producers In-House

We’re very management-light, and in fact no-one in-house at Naughty Dog has the job title of Producer - though we do have some great external producers at Sony.

The game is produced by the team, by anyone who wants to step up and take some responsibility for an aspect of the game.

Although it’s sometimes messy and chaotic, it’s messy in a good way, since the people with the responsibility are the same people actually making the game, which helps us stay very focused on quality and fun.

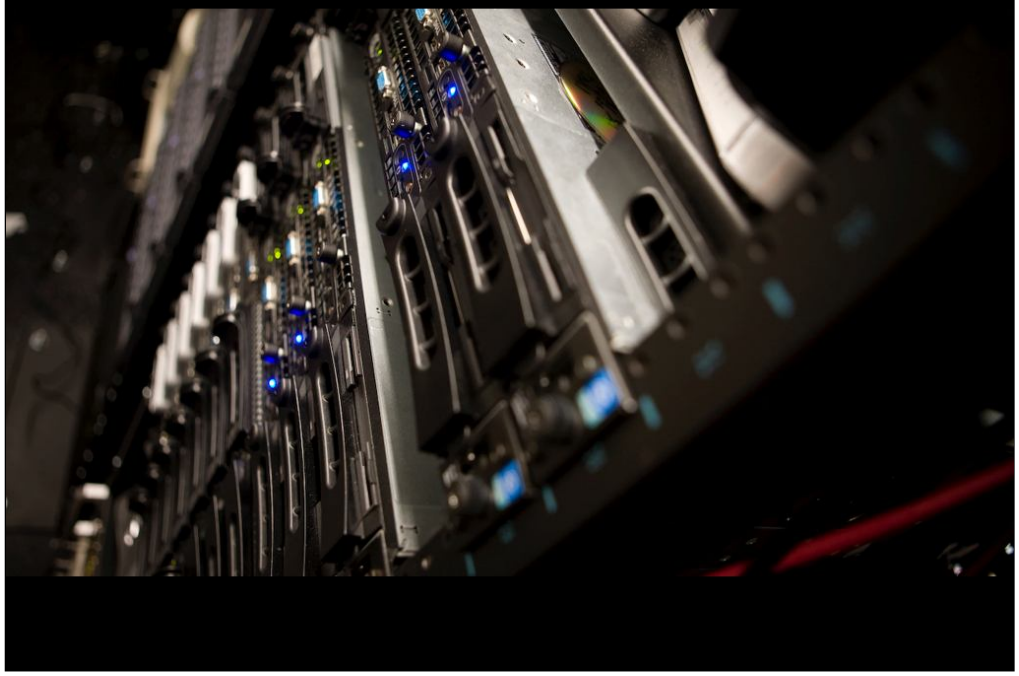
We develop concentrically, which is to say that we implement the fundamentals first to a good level of polish, working outward, iterating on secondary and tertiary mechanics, building modularly and ditching stuff that fails as early as we can.



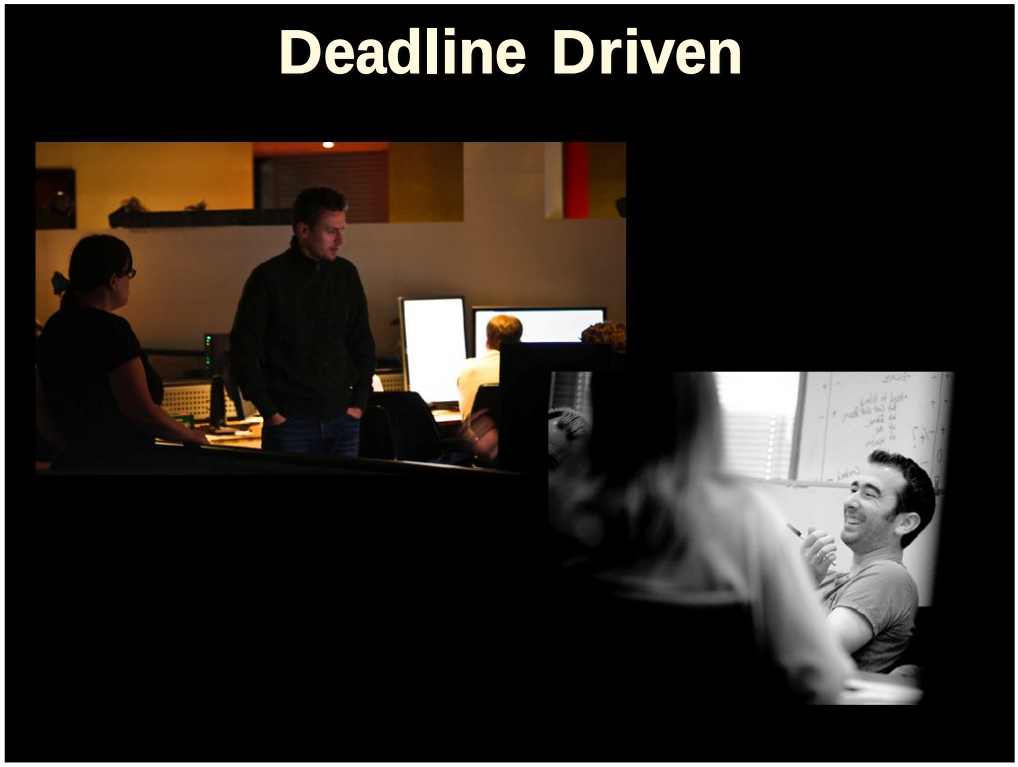
We don’t do a lot of documentation – just enough to keep track of our basic ideas.

We come up with a lot of our design and polish our games iteratively, sitting down together and trying things, and making incremental changes until we’re happy with what we’ve got.

Healthy Code

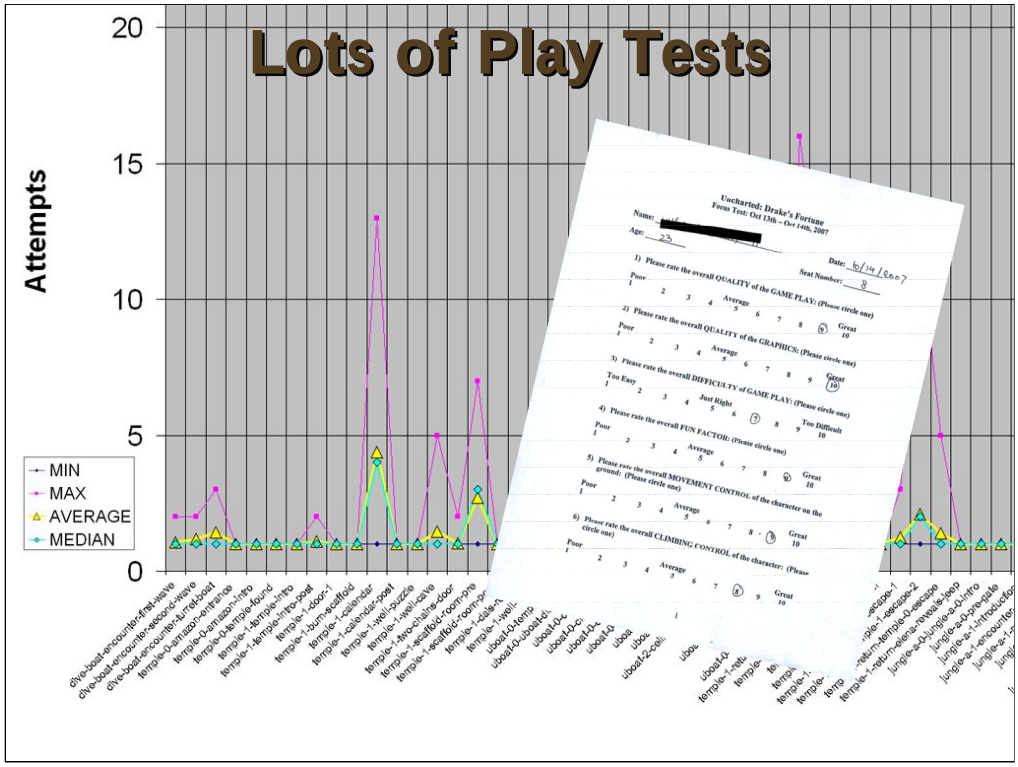


We always try and keep the game code and content running healthily and as free of crash bugs as possible, which allows the team to always be able to move forwards with building the game, and has other advantages that we'll get to later.



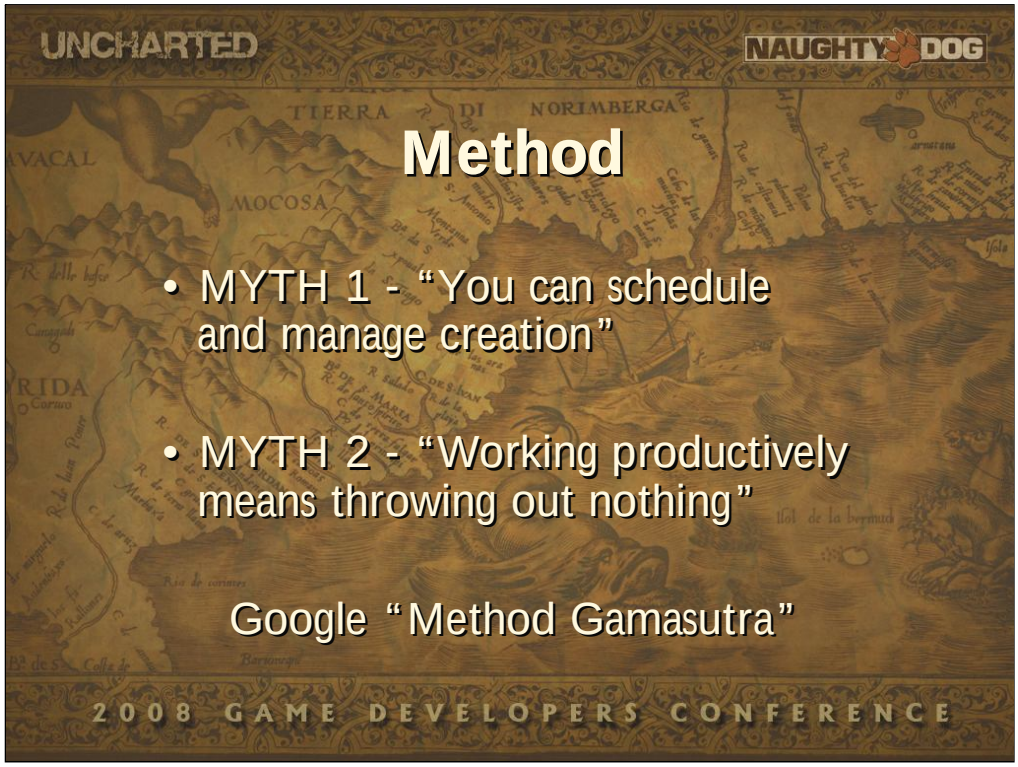
We’re very deadline driven, and we’ve never missed a ship date in the history of the company.

Our presidents and game directors keep an eye on the big picture, and alert us to things that are coming down the pipe.



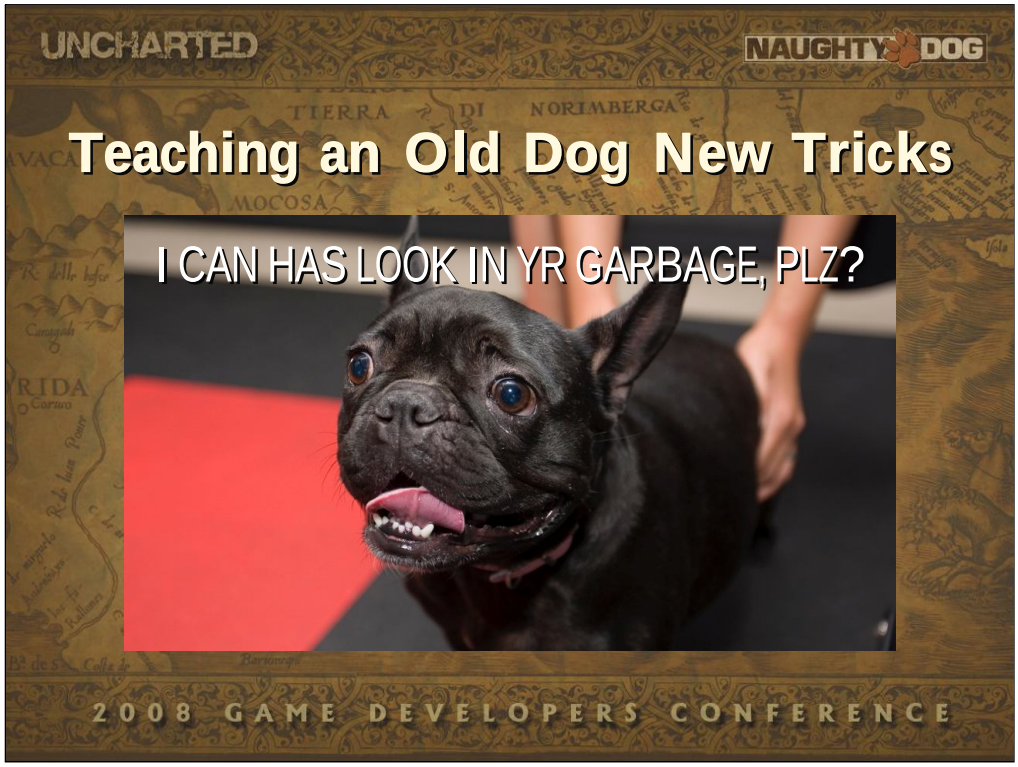
We do a lot of play testing with statistically large-ish groups of new players, and we try to be as scientific as possible with this process, extracting data from the play testers' save games, graphing it and studying it, mainly to tune the level of challenge.

We also get our testers to fill out questionnaires that we tabulate and take note of, which can be useful on an anecdotal level.



The way we makes games owes a lot to The Method – a game development strategy laid out by Mark Cerny and Michael John in the mid-90s.

If you Google "Method Gamasutra", you can hear a lecture by Mark Cerny on the subject, which I strongly encourage you to listen to if you're interested in ideas about better ways of making games.



We were going to try and learn a lot of new tricks for *Uncharted*.

We were going to be using new types of production methodologies and technologies, and even though we had a great PS2 code base, for a number of reasons we set out to create an entirely new game engine and tools, starting from ZERO lines of code!

This would come back to bite us in the butt later on.

Here's a really quick overview of our production timeline:

1 year of preproduction

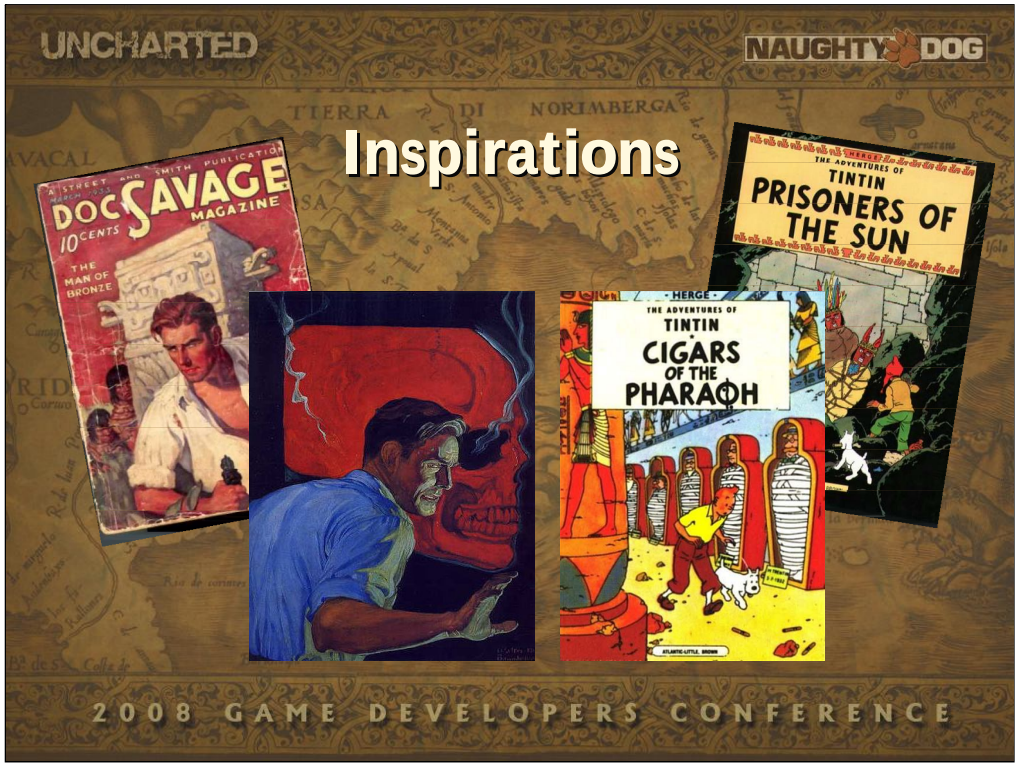
2 years of full production

This might seem like a lot of time, but we still found ourselves very crunched at the end of the project, because of how long it took us to get everything going.

So as we got into Preproduction, at the start of 2005, we began to choose some core design principles to base our game on.

We very soon decided that we wanted to set our game in a re-imagining of the classic pulp action-adventure genre.

From the early adventure tales like Robinson Crusoe and Treasure Island, and cowboy stories…



…to the prototypical pulp heroes – Doc Savage from the 1930s in particular, who was an evil-fighting scientist-adventure-explorer…

…and Tintin, Herge's intrepid boy reporter, was a great favorite for many of us.

We looked at a lot of the early adventure movies and B-feature serials that so famously inspired George Lucas and Steven Spielberg to make the great Indiana Jones movies.

…and of course we grew up with Doctor Jones, and had loved more recent explorations of the same themes, like The Mummy and the contemporary historical mystery of National Treasure.

As we studied these kinds of stories, we saw some common themes emerging.

All that action - running, jumping, hanging, climbing, and exploration of mysterious places, gunplay and bare-fisted fights, and chases – either chasing someone or being chased…

…casts of great, interesting, often eccentric characters with shifting allegiances – friends becoming enemies and vice-versa, cliffhangers and reversals of fortune and out-of-the-frying-pan-into-the-fire kind of situations.

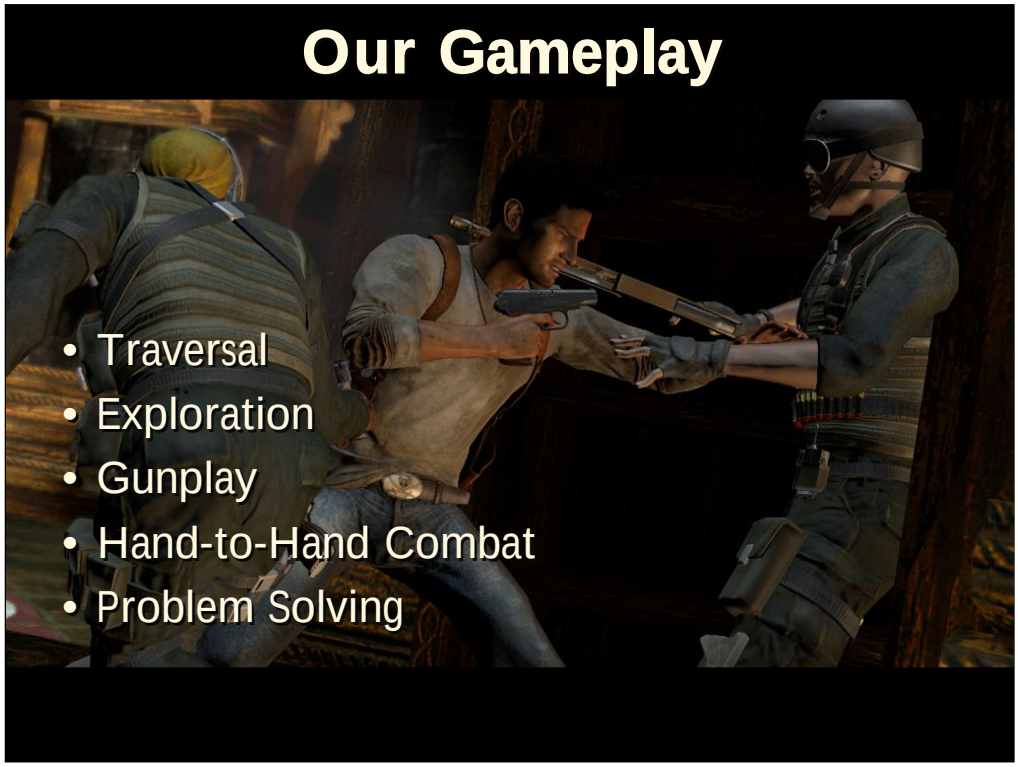
These are all themes that provide a really great basis for an action game - “action” being the operative word - the player’s participatory action in a dynamic world.

The Tone of the Game

Another major theme that we extracted from the classic pulps was a certain lightness of tone - the feeling of a world that was bright and cheerful, colorful, and of course humorous.

So we tried to express this in the game whenever it was appropriate, often through the color palette.

Aside from being what we wanted, we felt that this could help us stand out in the market , since a lot of games these days are so moody and overwrought and kinda emo.



So we eventually boiled down all this research into gameplay that we could all buy into:

Traversal (including vehicles)

Exploration - a kind of expression of traversal

Gunplay - our core combat mechanic

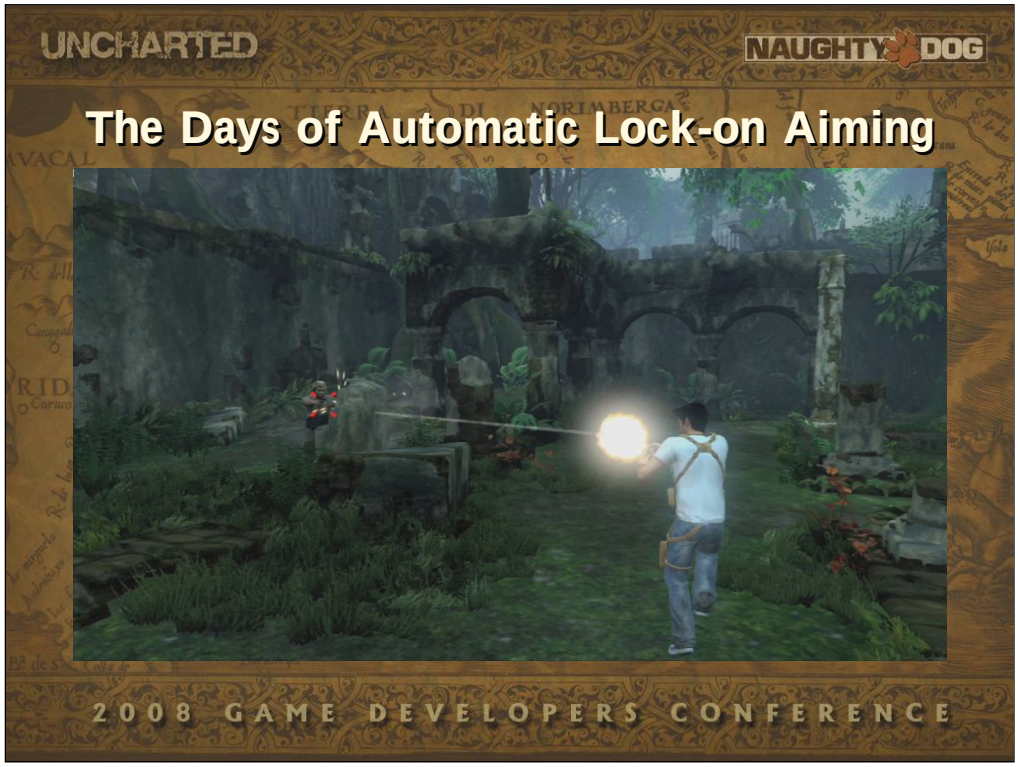
Hand-to-Hand Combat - to add some spice to the combat

Problem Solving - both traversal problem solving and puzzles

We didn't really know exactly what mix of these things we'd end up with, but we knew that these were the components and roughly what proportions of them we wanted.

Cover-based gunplay had been a big part of our conception of the game from the very beginning, because it's such an important thing in the sources we were drawing from.

We thought that *kill.switch* had done a great job of showing that a videogame could have really top-notch cover-based gameplay, and so from the start we were committed to the animation and control effort that pulling it off well would imply.

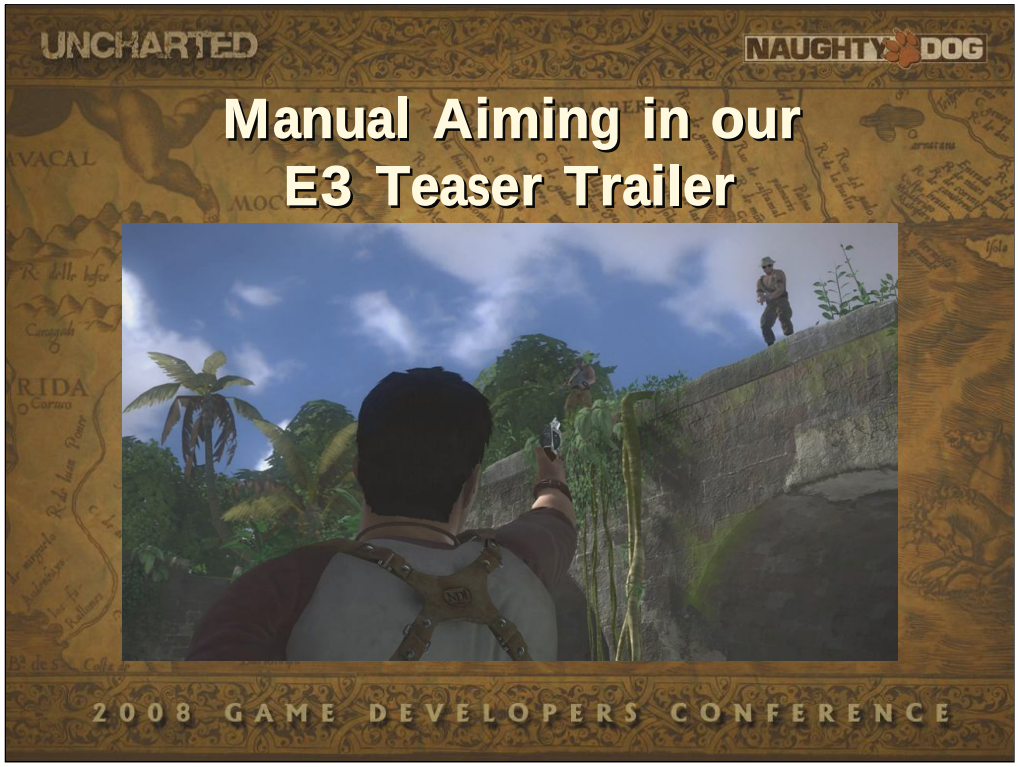


For a long time, in fact through the late winter of 2006 (through our first year of full production), we'd planned to use an automatic lock-on aiming scheme.

We were skeptical that manual aiming could create the fast-paced kind of gameplay we wanted to go for, and we were worried that the transition between the *Mario 64*-style "follow" camera we were using for traversal

and the over-the-shoulder camera that manual aiming implies would be jarring.

We tried every possible lock-on auto-target selection mechanic and target-selection control scheme perturbation that we could think of, and lock-on aiming didn't give us the visceral fun of taking a bead and loosing a round that we wanted to capture. Also, it wasn't challenging and interesting enough from a gameplay point of view.



A number of people on the team felt that manual aiming could work, and we were already big fans of *Resident Evil 4*.

So in Naughty Dog's spirit of experimentation and iteration, we tried it – and liked it! We managed to solve the camera transition issues, making it nice and snappy so that the game didn't feel like it had two gameplay modes, and was one seamless experience instead.

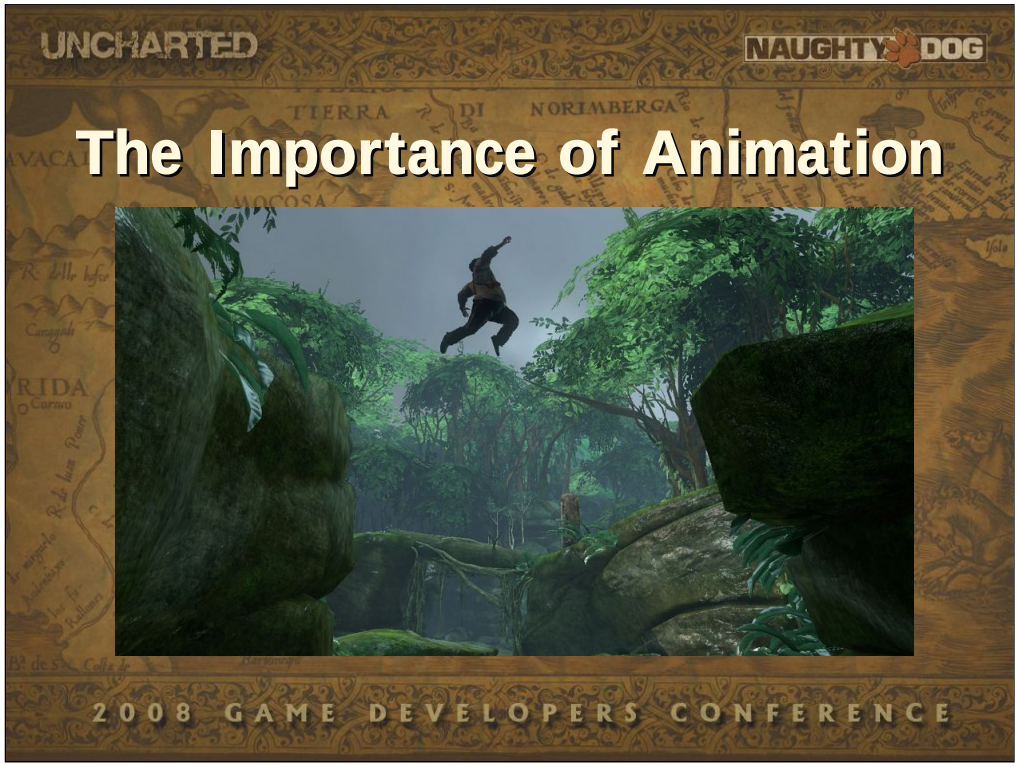
It seems like we were telling ourselves even back in the days of our first E3 teaser trailer, then that manual aiming would be possible. Though we then had a lot of new issues to puzzle over…

Manual aiming in a Third-Person Character-Action Game proved to be a lot harder than we thought it would be.

We knew that we would have to have an assisted-aiming, or Enemy Adhesion, system that was as superb as that in Halo – so that when the player aimed near a target, the game would have to subtly, almost imperctibly, move the reticle onto the target, and travel with the target, to help the game be optimally fun.

We also knew that we'd have to overcome the issues associated with manual-aiming in a third-person game as well as *Gears of War* had done (which is a game that we really love, BTW) - issues like those arising from the differences between the line of sight from gun-to-target and camera-to-target, among many other things.

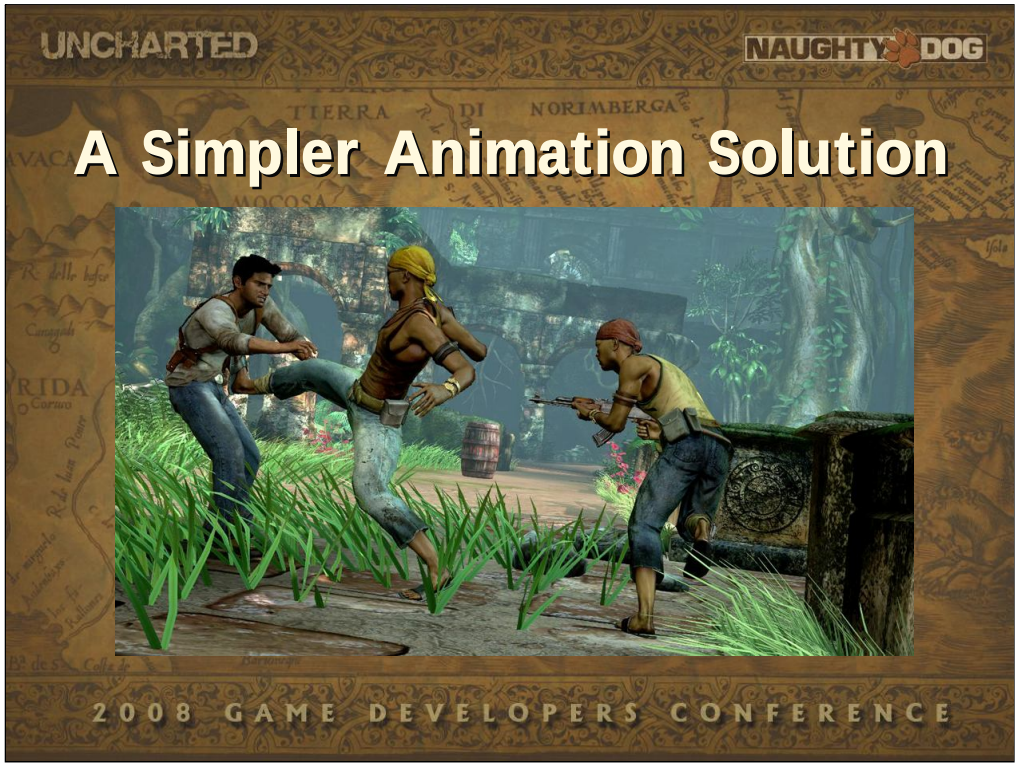
So we studied those games a lot, and we eventually managed to get it right.



Animation was very important for our game from the beginning - to make the player-character's humanity and vulnerability believable, since that's such a big part of the pulp adventures, and to help sell the reality of the world.

So we always planned to use a lot of complex animation techniques that I'll talk about when I show you a movie in a moment.

We also thought that excellent facial animation would be the key to depicting a relatable hero that you'd really want to root for, with nuanced emotions for the characters that are as recognizable as they are in our favorite movies.



With our first approach we set out to create a very complex animation system - we were trying to solve the general problem of complex human animations with the environment.

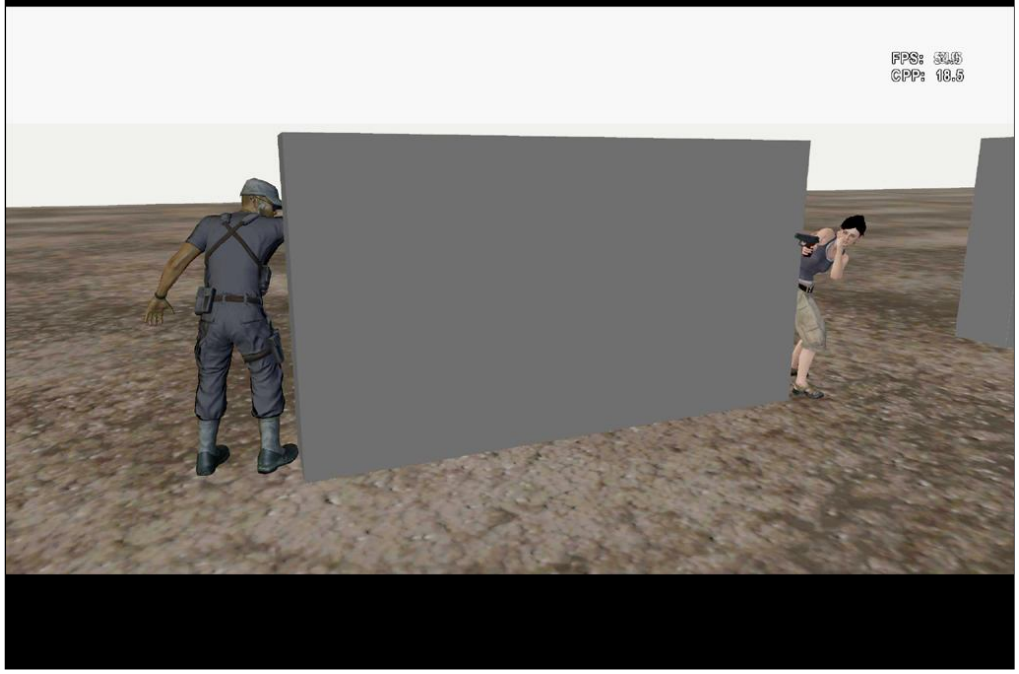
But ultimately, that approach proved to be too complex - we had gotten hung up on esthetics, and wanted to find perfect solutions for everything.

So instead we implemented a simpler solution that had more in common with the way we made the *Jak* games although it was much more complex and incorporated a lot of new features, and we shifted our focus to the kinds of activities associated with the type of character-action game we had in mind.

AI was also always very important – both for the enemies and the friendly characters that Drake would meet and play along side in his journey.

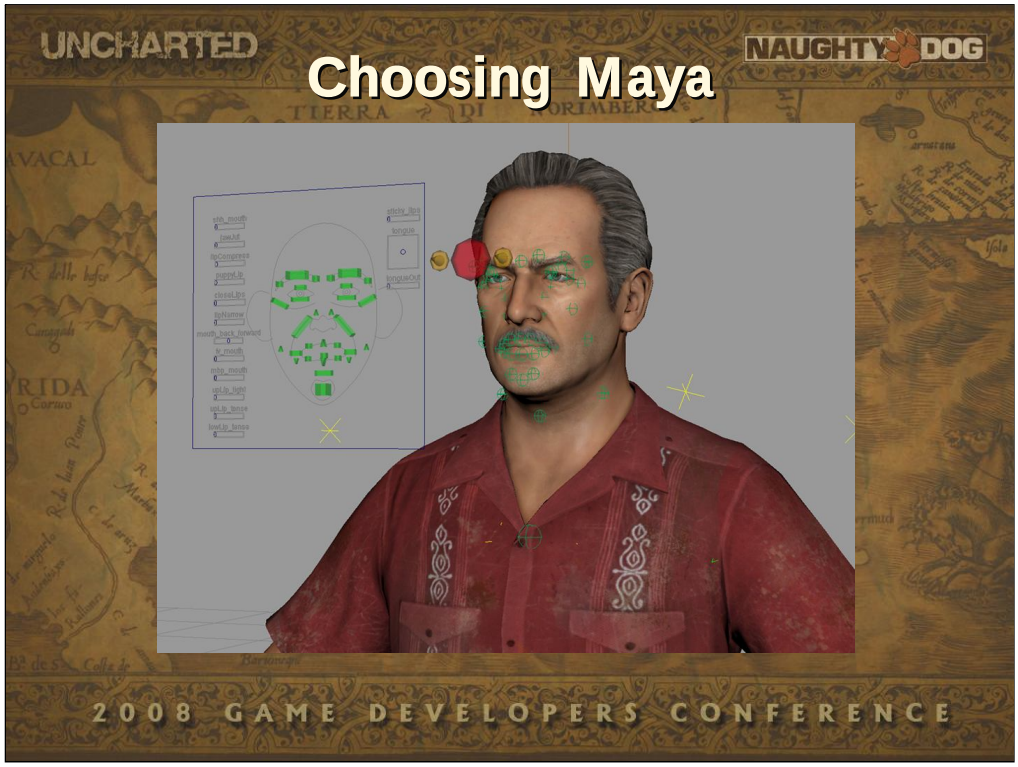
It was important in order to make the combat and traversal gameplay fun, and we wanted to make sure to avoid the potential problems that we'd read about in the post-mortems of developers who'd worked with AI for allied characters (*Half Life 2*, in particular).

Targeted AI Approach



Our initial work in this area anticipated very sophisticated AI, with complex agents that would play out their behaviors in a kind of general, systemic way in the world.

It helped us explore the space, but again, it would ultimately prove to be too sophisticated for us, and so we adopted a somewhat simpler and more scripted approach targeted more towards our game's needs.



Early on, we chose to stay inside Maya for all of our animation – both for characters and for events in the game world.

To stay in just one package, especially a great tool like Maya, was a good idea, and we wrote a lot of Maya tools to make this possible.

Hopefully you caught Jeremy and Judd's talk yesterday, where they went into great detail about our tools and animation process.

We used zBrush and some Mudbox to create our character models, which we then imported into Maya for rigging and animation.

Catch Rich Diamant's speech tomorrow morning at 10:30 on 'Autodesk Mudbox and its integration with Maya and Max'!

A recurring them in our post-mortem process has been that we often didn't commit to technological or implementation processes early enough, and that we should have got more basic stuff in and working earlier.

This applies to very many areas of the game, but for example, in the case of the enemy character models, we wanted to use a system of interchangeable body and clothing parts for our enemies, to get greater apparent variety in their look. This ended up holding up our whole enemy implementation process, until we finally realized we didn't have time to do it, and got on with making the great enemies that are in the game – and a lot of players never noticed that we didn't have a huge range of subtlety different enemies.

Anyway, despite a few reservations, we felt that we developed a strong character modeling pipeline, and got great results like those you see here.

Collaboration Is the Key

We were very happy with the way that all of our game animation, control and AI turned out.

It was the result of lots of iteration and some very close collaborations between programmers and animators (two of whom moved seats to sit with the programmers they were working with!) and between the team and Lead Character Technical Director.

It really highlighted for us how critical close collaboration between disciplines is for us at Naughty Dog.

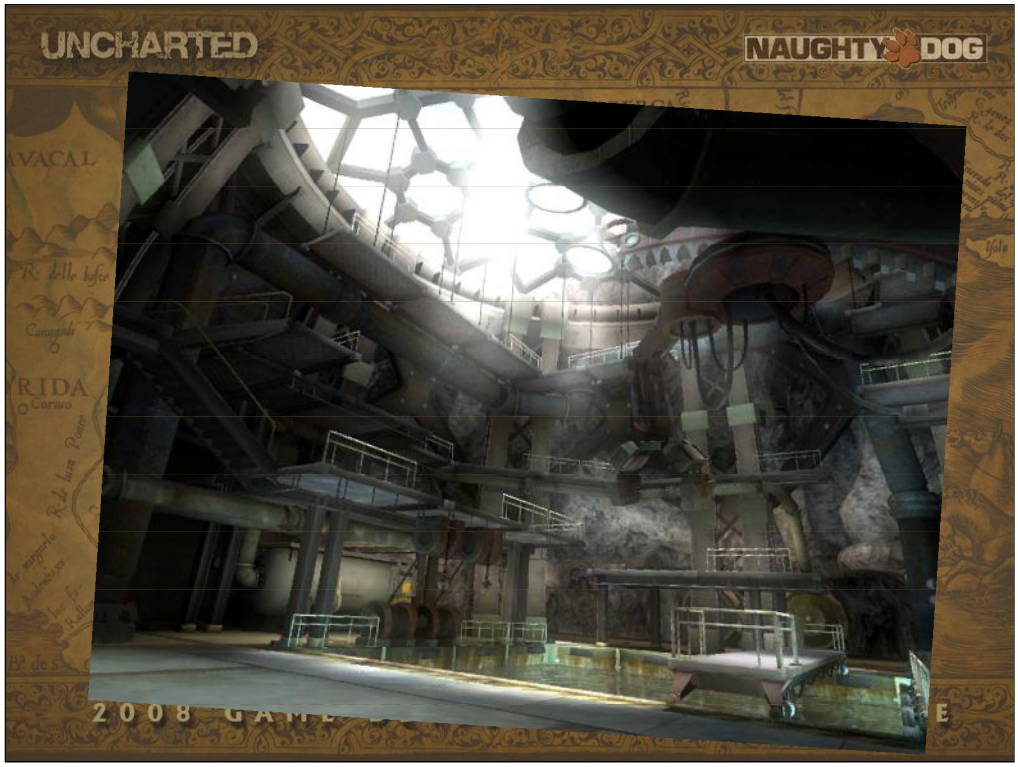
If you're interested to hear more about the techniques we used, including a system to provide a layer of indirection between control and animation, please go hear Christian Gyrling's talk, Creating a Character in *Uncharted: Drake's Fortune*, at 9am tomorrow in Room 2001 in the West Hall.

For *Uncharted*, we were going to be using motion-capture for the first time, in combination with the kind of traditional hand-keyed animation we'd used for our previous games.

In May of 2005 we did our first Motion Capture research, going to see different studios and how they did things. We finally chose House of Moves, the well-known mo-cap studio in Los Angeles, who were great to work with.

A couple of months later we did our first test shoot with a stunt performer we'd chosen to play Drake.

The rest of 2005's work involved a lot of concept art creation and design research, and coming up with story ideas and moments like the Uboat-in-the-jungle you see here which made it into the finished game, albeit in a different form…



…building character models, the programming of tools (animation tools in particular) and of our core gameplay, and our first experiments with building environment art.

In October 2005 we prepared a Concept Movie", using concept art and a temporary soundtrack.

What's remarkable is the way that this early concept movie accurately represents a lot what's in the final game that would ship two years later.

As a result of this movie, we received a Green Light from Sony.

Up until the fall of 2005, although we had already strong look for him dialed-in, our hero Nathan Drake's personality had been hard to get a handle on for a lot of the team.

Around that time, someone had the idea that maybe Drake should be kinda like Johnny Knoxville from *Jackass*. So we modified the character design for Nate slightly to a design loosely based on Johnny, reflecting Knoxville's cynical sense of humor, which we eventually moved away from a bit, and also to reflect his essential coolness and goodness.

We were all able to immediately grasp and get behind the tone that this idea embodied and from then on people really started to understand Nathan Drake.

So we entered full production at the beginning of 2006, and a few months into the year we were asked to make an *Uncharted* teaser-trailer for E3, where Sony would be unveiling a bunch for new stuff for the PlayStation 3.

We only had four weeks to create it and at the time, given the relatively early state of our tools, it seemed like an almost impossible challenge.

But we buckled down, and planned out a flow of action for the player-character in the test levels we'd been building…

…that would seem like a movie trailer, sort-of telling a short story about a confrontation between Drake and the South Seas pirates on a tropical island.

We thought carefully about all the things that Drake would do in the trailer…

…and made sure that they were all things that we wanted him to do in gameplay, in the game.

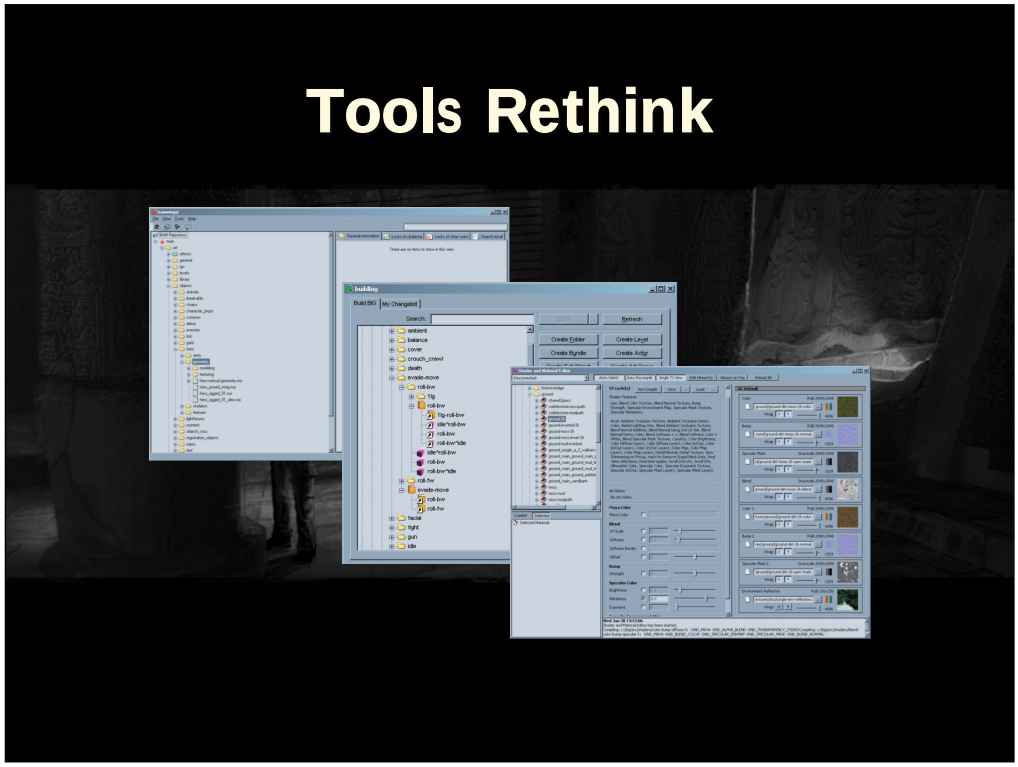
We were in our engine by this point, although it was in its infancy, and we weren't running at full frame rate yet for one reason or another.

So we captured it at lower frame rate and played it back at full speed, making sure that we were confident that we would we would, pretty soon, be able to do all this in real-time.

And when we look back at it now (you can see it on the internet), we're delighted that nearly everything Drake does in the trailer, you can actually do in gameplay in the game.

We ended up referring back to this movie all the way through development, to make sure that we were staying on track with our initial vision.

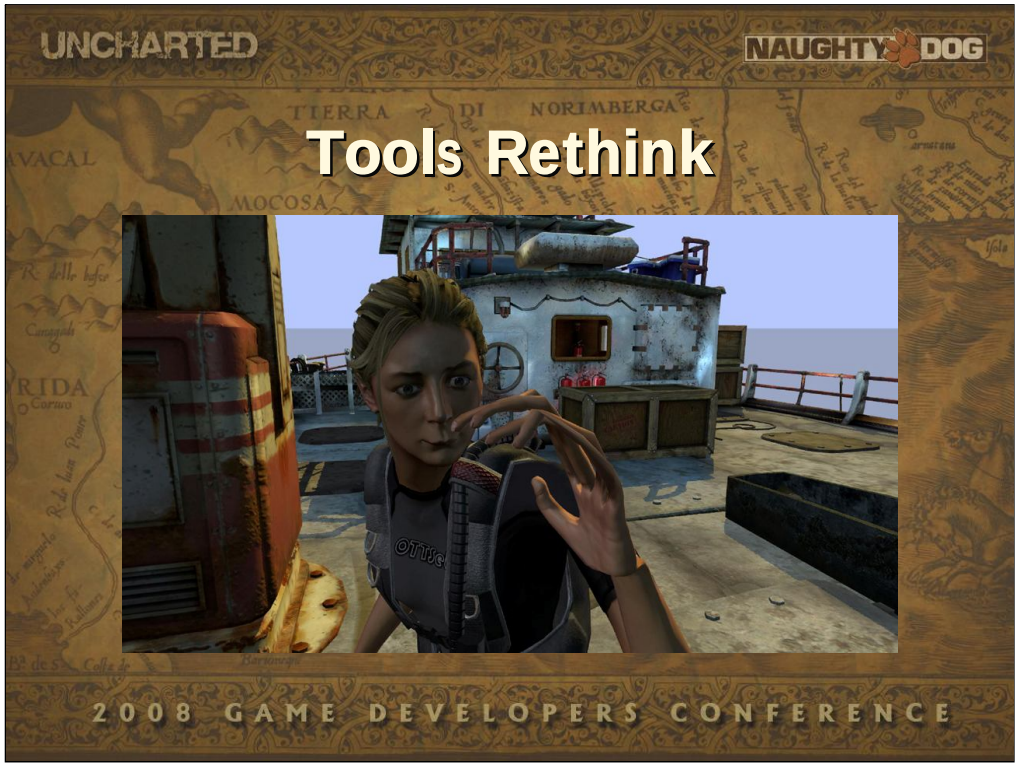
It was the first piece of major production coordination and intensely hard work that the team pulled together to achieve, and as such, completing it was tremendously inspiring, and helped us begin to gel as a team.



In June of 2006, the team faced a big crisis.

Creating the E3 trailer had exposed issues with the tools pipeline, and it was at this time that that we realized that we were in deep trouble.

Like I said, we'd developed all-new tools for *Uncharted*, including BAM, our asset management system, BuildBig, our first GUI build tool, and a shader and material editor called Shmeditor.



But there were numerous problems with the tools – they were slow and had other usability issues - and people were very frustrated by them. We realized that we'd bitten off more than we could chew with our tools approach.

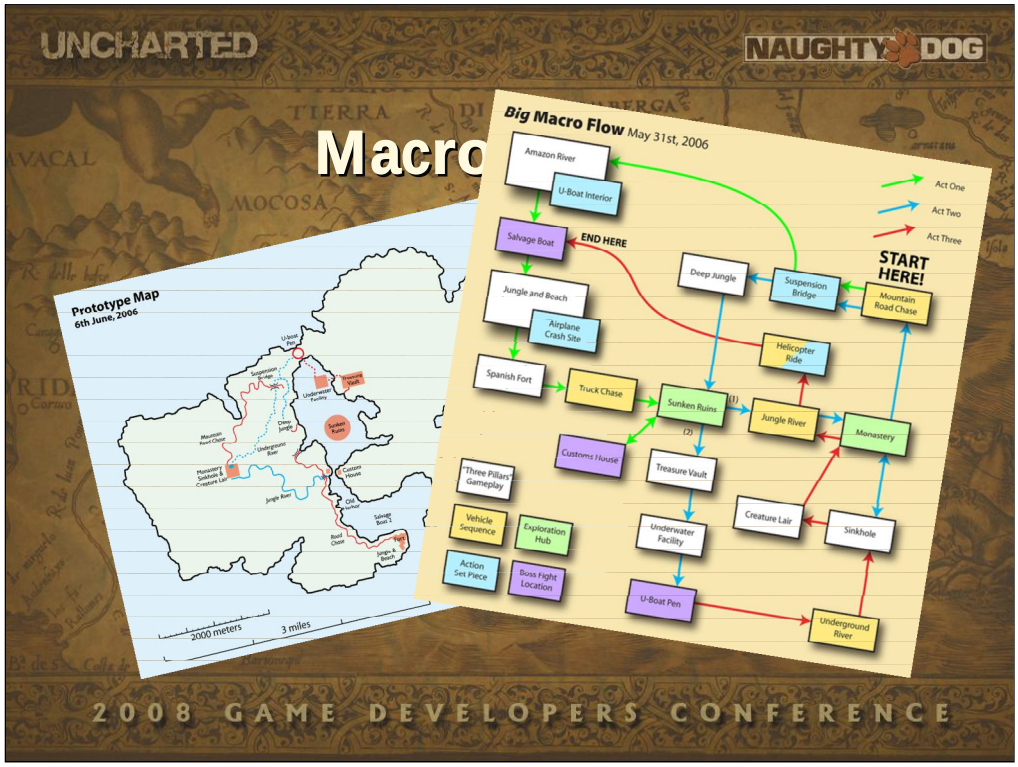
Basically we'd tried to be too clever – coming up with convoluted approaches that were intended to solve every last problem that anyone had ever had with each kind of tool.

To make it worse, we'd gotten distracted by our lofty aspirations, and left it very late to implement a tools pipeline that let people build and play levels.

In many, if not most, cases, we went back to our old familiar *Jak and Daxter* ways of doing things with tools.

And from then on, things got better and better, and we began to get really good traction on building out the game.

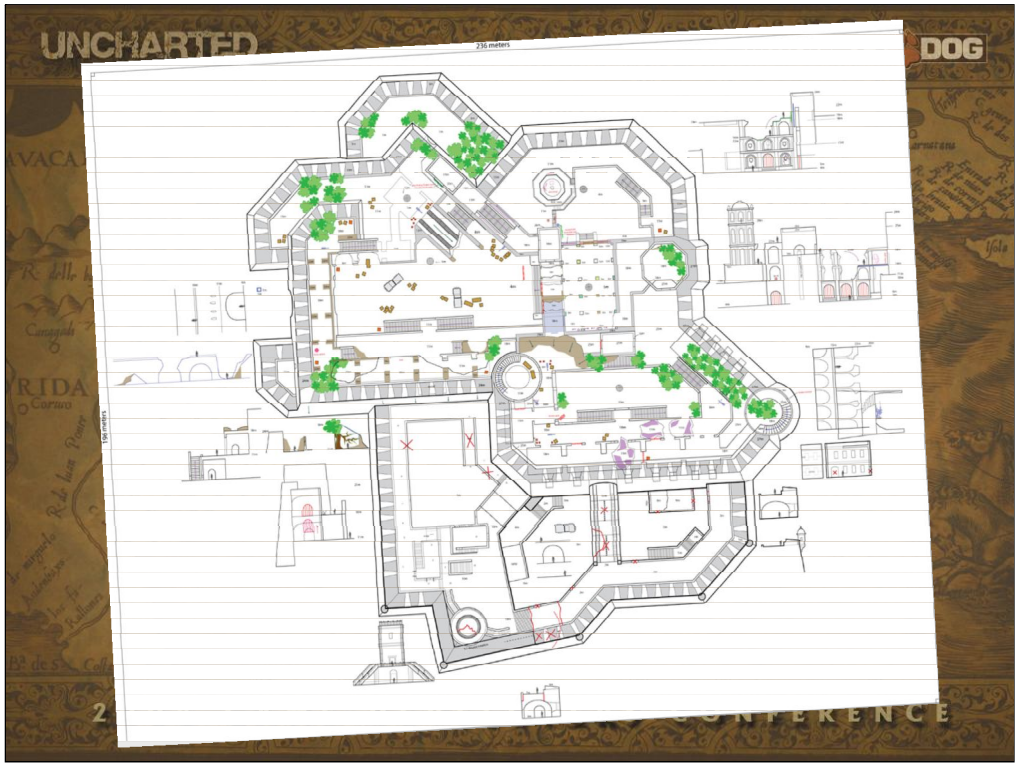
Our takeaway from this experience was: don't try to solve every last tools issue! It's good to aim high, but choose your battles.



Another thing that helped lift our spirits around this time was that the Macro Design had firmed up a lot.

Lots of details about the gameplay had by now been decided - all the levels were described and in the approximate order that they'd be in in the shipping game, and most of the major plot points and key moments were fleshed out.

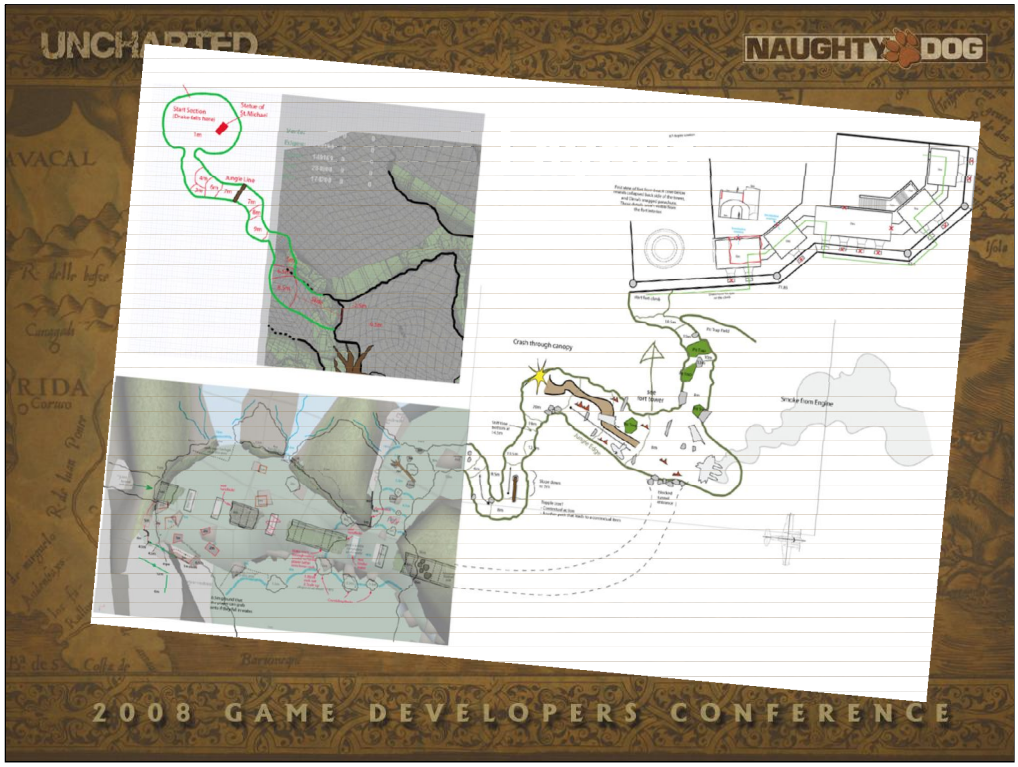
The macro continued to change throughout production in response to our schedule and the way the gameplay and story evolved, but people on the team could now see much more clearly where the game's design was heading.



We were well underway by this time with the level layout effort that would make the creation of the environment art possible. We like our mixed-media approach to layout - whatever worked well for the designer of the level was OK with us.

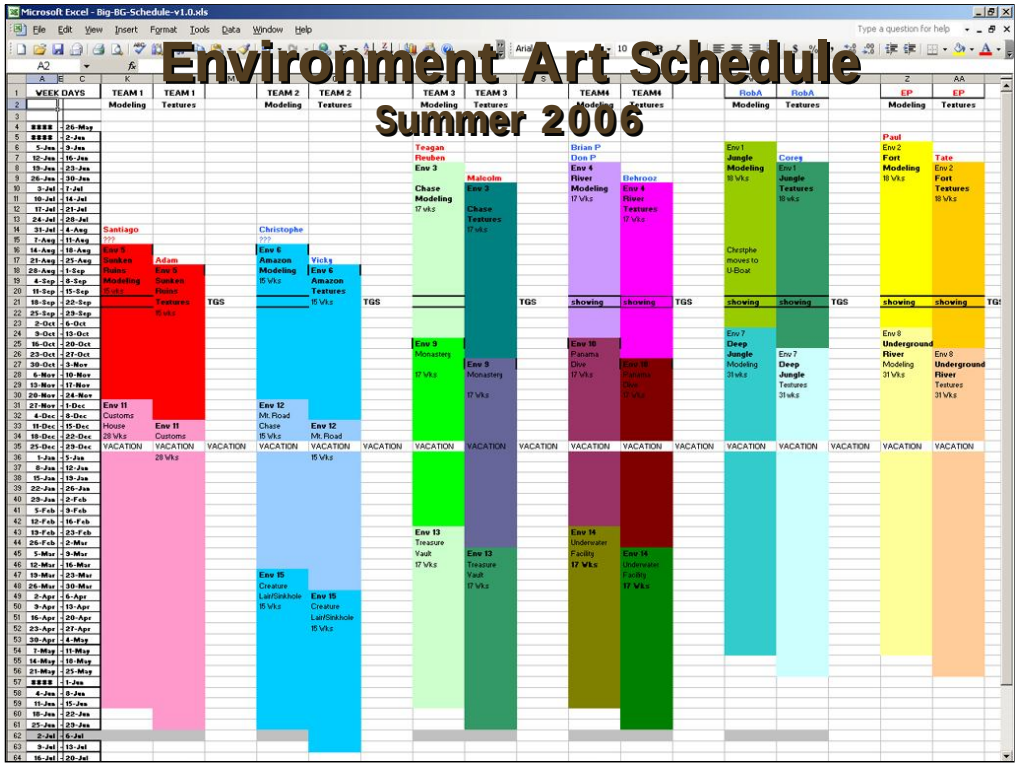
A lot of people started with physical sketches, drawings or maps on squared paper, but mostly we used a combination of Adobe Illustrator and simple prototype geometry, letting us iterate on the levels quickly and easily, and allowing us to easily add concept art or screen grabs as overlays to communicate our ideas about each part of the level with each other.

Some of our layout was quite simple, but some was as complex as architectural plans, like the one for the Fort you can see here.



We also have to plan at the layout stage for breaking the environments up into the chunks, so that we can stream the parts of the level coming up and, as in our previous games, get rid of load times.

If we hadn't been so pressed for time in the last year of the project, we would have taken an even more iterative approach with level layout, and that's what we plan to do in future, playing the levels in block mesh (or grey box) and rapidly making changes until it's right, before the final art gets made.



By July 2006 our Environment Art effort had begun in earnest.

This is the way that we schedule environment art creation at Naughty Dog. We don't really believe in Gantt Charts - with a production approach like ours that involves so much change, they really just make for busy-work, and for tasks that are measured in weeks, this kind of schedule in a spreadsheet is perfect.

In fact, we use spreadsheets for a lot of things; for example, the Environment Leads also used spreadsheets to track work and coordinate modelers and texture artists, and probably more than half of our design documents are spreadsheets.

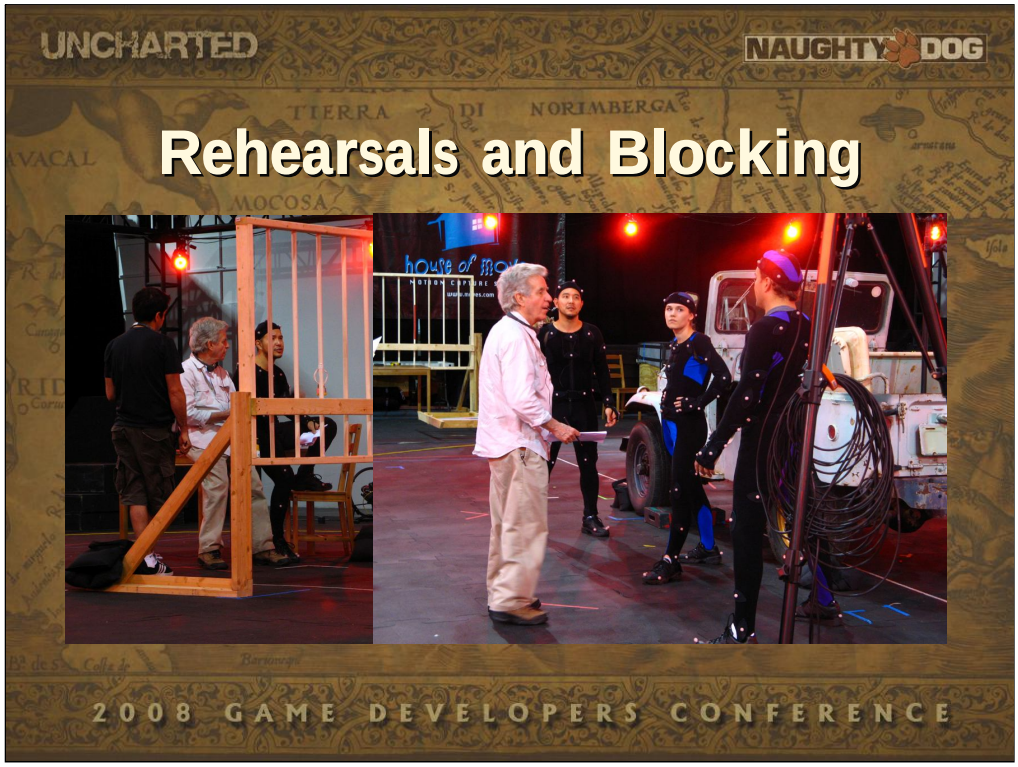
So, getting back to how we developed our Motion-Capture Production Flow: in the summer of 2006 we cast our game, and ended up with a great group of actors, including Nolan North and Emily Rose playing Nate Drake and his co-adventurer Elena Fisher.

The actors would be working with Gordon Hunt, our motion capture director, who has a tremendous amount of experience working in film, television and theater.

Because this was our first experience using motion capture, we had an open mind about how to do things, and our approach ended up owing more to the way a movie or theater production is staged than the way most people in the games industry do mocap, often using different actors to do the voice and physical performances. We used the same actors for both mo-cap and dialogue, at least for the cut-scenes (we used stunt people for some of the in-game animation).

In September 2006 - one year away from Beta - we did our first for-real cinematics motion-capture shoot.

We started out doing table reads with the actors and directors, so that everyone had a chance to discuss their characters and the situations they were in.

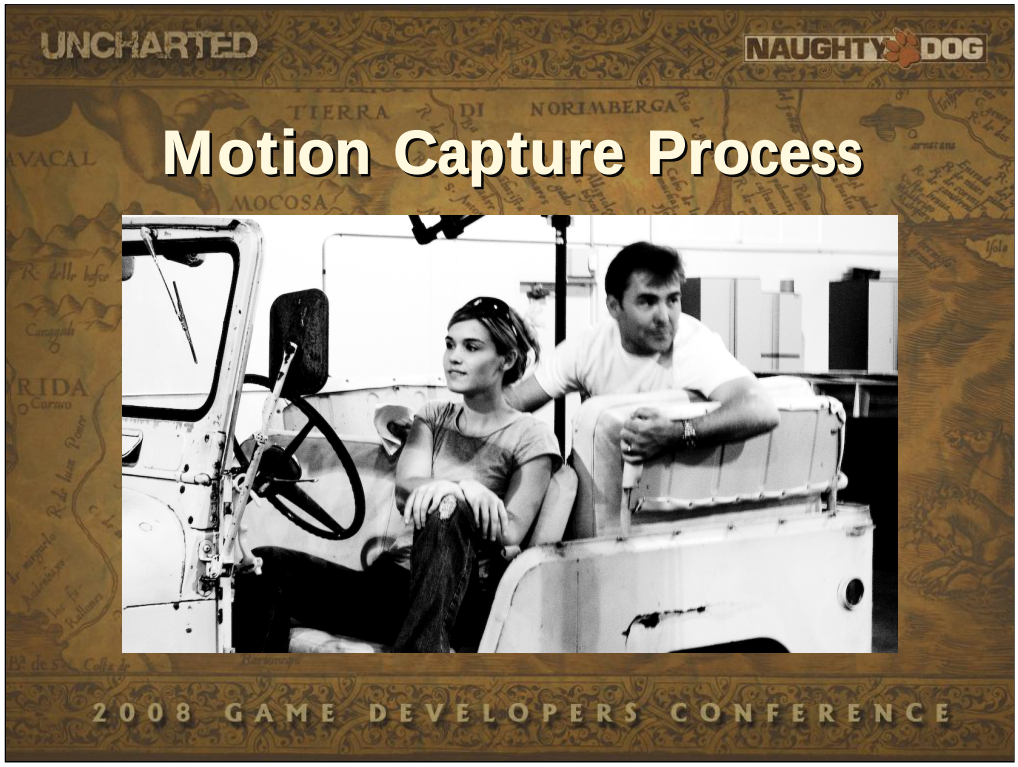


Then we blocked the scenes out on the stage, so that everyone could understand how the physical space was going to work with the acting performances.

One great thing about this, apart from ending up with great performances, was that it let us do a lot of improvisation with the actors.

Some of the best moments in the game were actually suggested by the actors, like this moment when Elena punches Drake for having left her behind at the docks in a previous scene, which Nolan and Emily came up with while they were rehearsing.

We had to do a lot of careful planning, since we needed to build out some stages so that the actors could interact with the environments properly, like the jail cell bars you see here, or the airplane cockpit that we built out so that the actors would be in the right places in the virtual scene.



Overall, it was a process of trial and error. We had to struggle with the process, learning from our successes and mistakes.

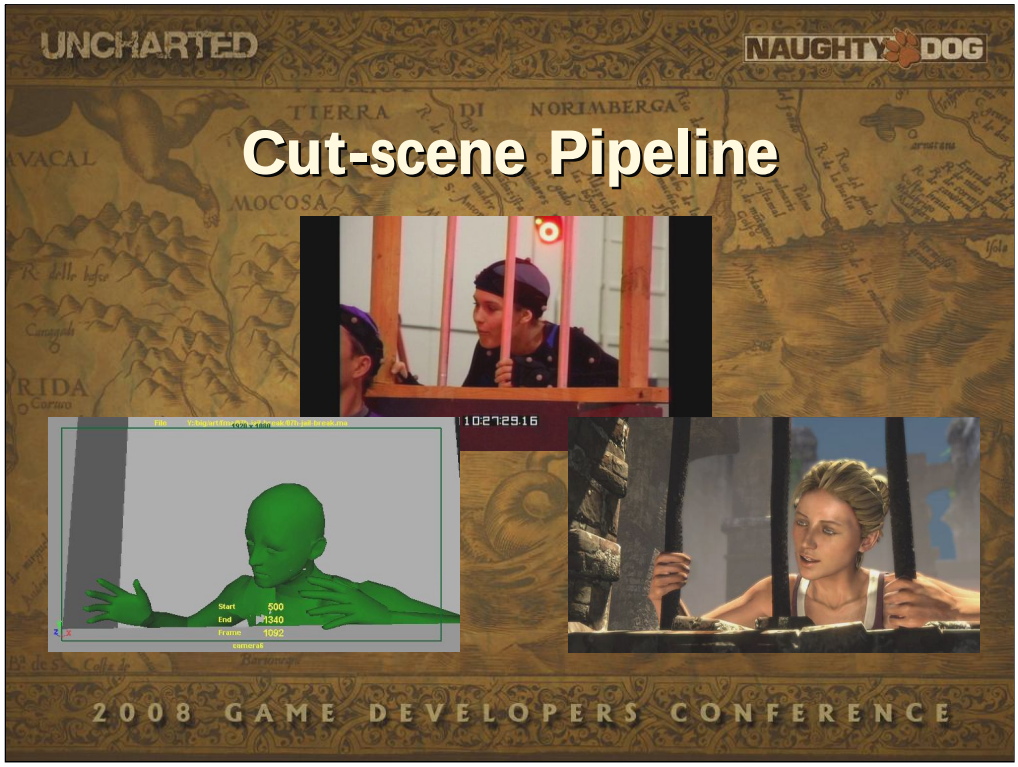
We were happy to hear from House of Moves that we running such a tight ship that we managed to get more useful minutes of mocap data per day than many TV and movie productions do.

But most of all, we were very happy with the authenticity and subtlety of the performances we got.

So when we got back to the studio, we set about making the cut scenes.

Another unusual aspect of the way we do things is that we have an in-house editor – someone with movie-editing experience.

We'd start by having him cut together the video and audio that we'd recorded live on the stage as the mo-cap was 'shot', into the kind of 'quad view' movie you can see a still of here, to provide reference for our animators, and also to provide a guide track for the dialog recording sessions where we got 'clean' takes of the audio.



We'd learned a lot from making the cinematics for the *Jak and Daxter* games, and we have a terrific team of in-house animators, who worked with the mo-cap data and did a lot of key-frame animation to finalize it.

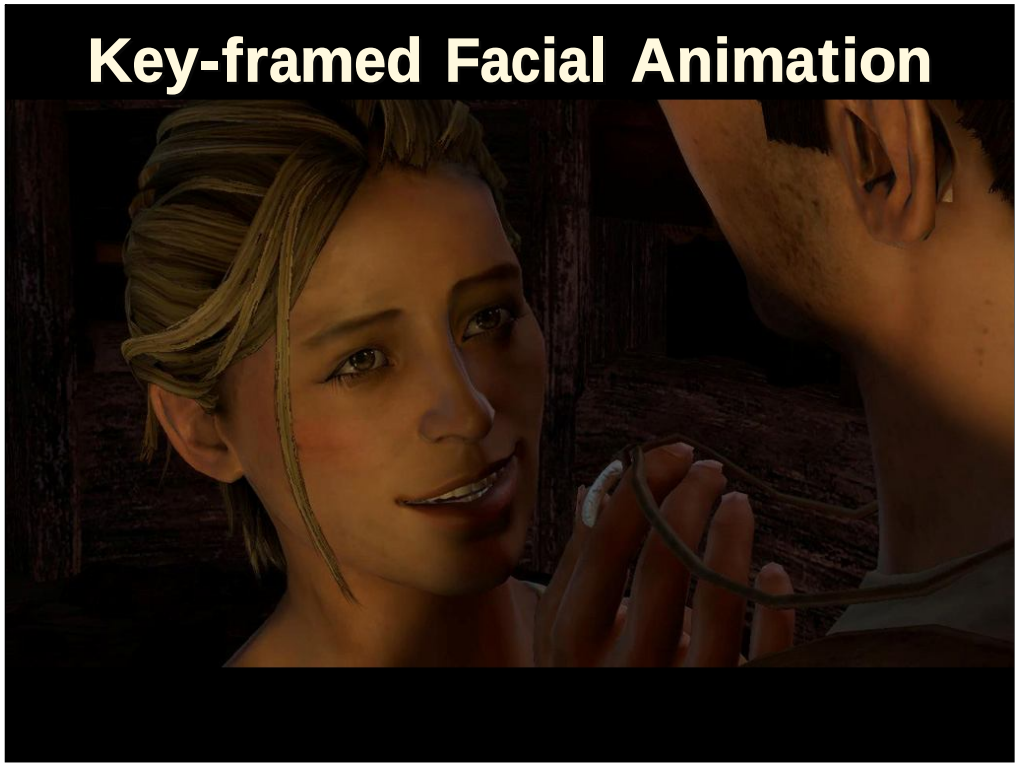
So once we'd got a clean audio track cut together, the animators blocked the scene out in animatic form, and then went back and forth with our editor, constantly refining the cuts and timing of the scene until it was right, before finally fleshing out the animation and creating the final movie.

The movies were rendered in the game engine, using the game's characters and environmental assets, although not in real-time.

We weren't driven by a desire to "cheat", but to be able to get into and out of the movies quickly, which we couldn't have done if we'd had to load a bunch of assets from disc.

We got a slow start and faced many tools and pipeline issues that required workarounds, and it was very hard to get everything done in time.

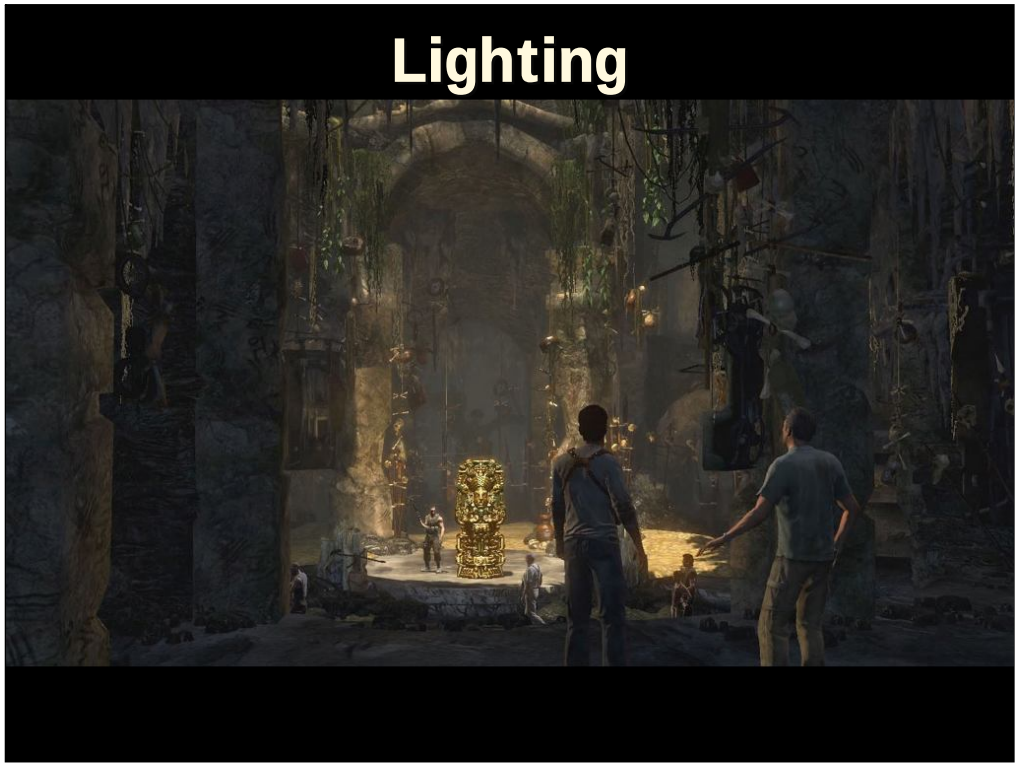
We couldn’t have done it without outsourcing about half the animation to Technicolor, who have a great animation team, and Technicolor also did the sound design, editing and mixing for the cut-scenes.



Our facial animation was all key-framed – we decided not to do facial mo-cap for *Uncharted*. Our animators studied hard to work with more realistic human characters, and pulled off some amazing work.

So overall, we were very happy with our production methodology for the cut-scenes.

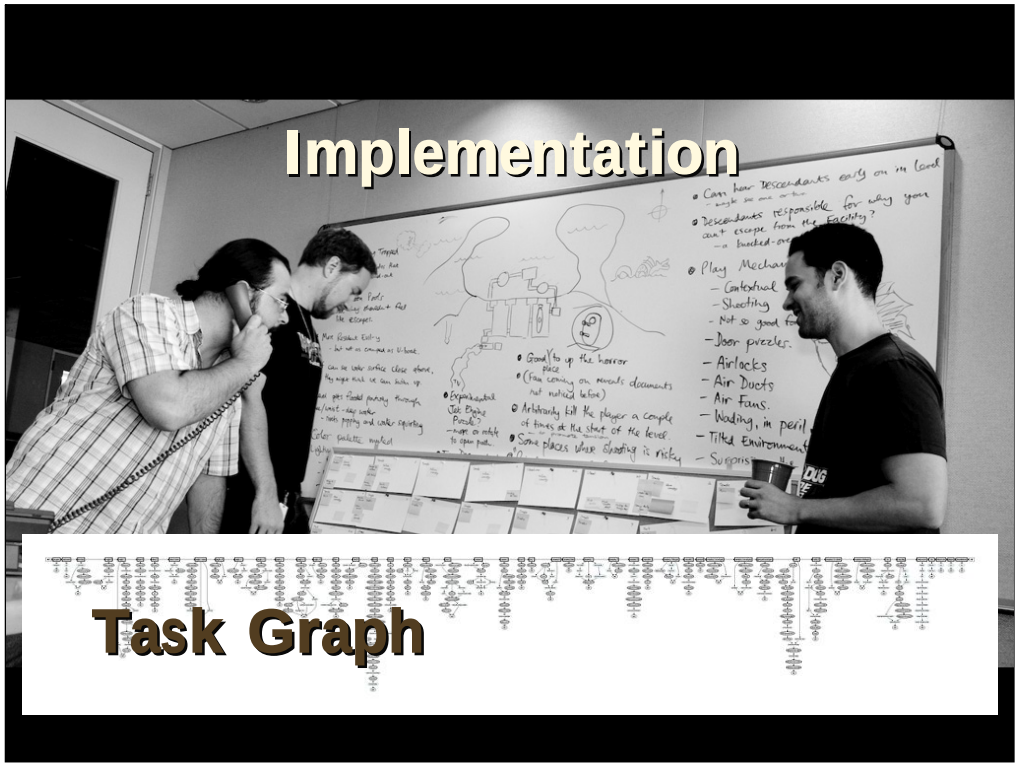
We localized our game into 8 languages for audio, and 13 for text (subtitles and menus), and we have some great localization tools.



Our lighting guys, both for in-game and the movies, did an amazing job. We lit the cut scenes like you would a movie, setting up special sets of rim lights and fill lights for virtually each and every shot.

The post-processing settings that we had to set up for each part of the game were an order of magnitude more complex than anything we'd done on the *Jak* games, and our course we were able to use a lot more run-time lights than before and use more complex shadowing systems thanks to the PlayStation's power.

I think the lighting is a very important part of the emotional tone of any game, and worth putting a lot of effort into.

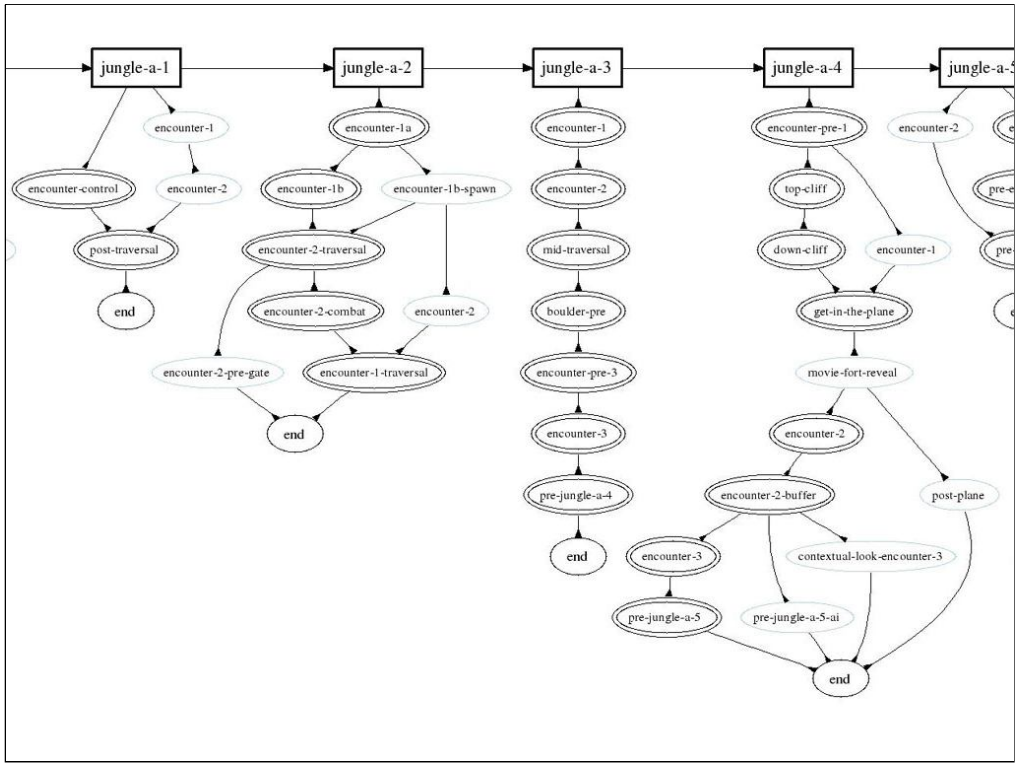


By the beginning of 2007 we only had about nine months left until Beta, and lots and lots of game left to build - the environment artists were all insanely busy, up to speed and kicking butt, and the game designers were implementing gameplay as fast as they could.

Now, our general implementation approach at Naughty Dog is that as soon as we have the tools and the assets to set up some part of the game, we do it. So, as each level came on-line in even a rough state, we began to set up cameras, place enemies and interactive objects, and write the scripts for the combat set-ups and spot interactions.

As the levels changed in the process of getting polished we'd have to make tons of tons of changes to the stuff we'd set up, but then that's what we do anyway, in the course of improving the gameplay, and the net result of getting more time to play with what we were building made it totally worth it.

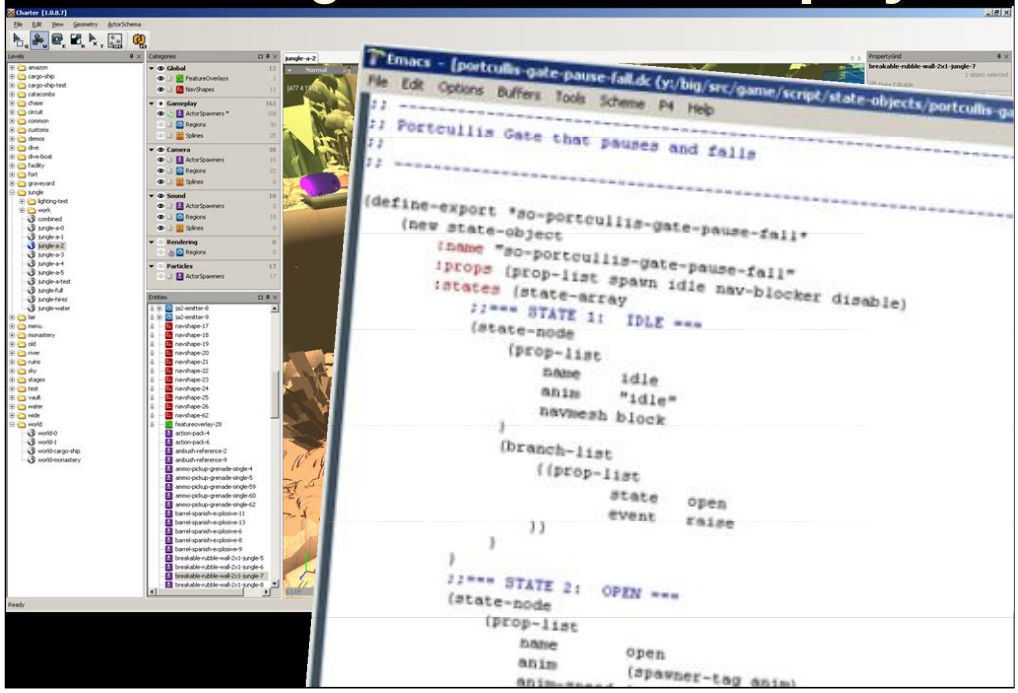
As part of this work, we had to set up our Task Graph, which is the thing that provides the logical backbone of the game.



It controls the appearance of enemies and other in-game events, and also keeps track of the game's save state.

You can see that it gets quite complicated, and some of the tasks have to run in parallel to let us get the kind of control we need.

Building Out The Gameplay



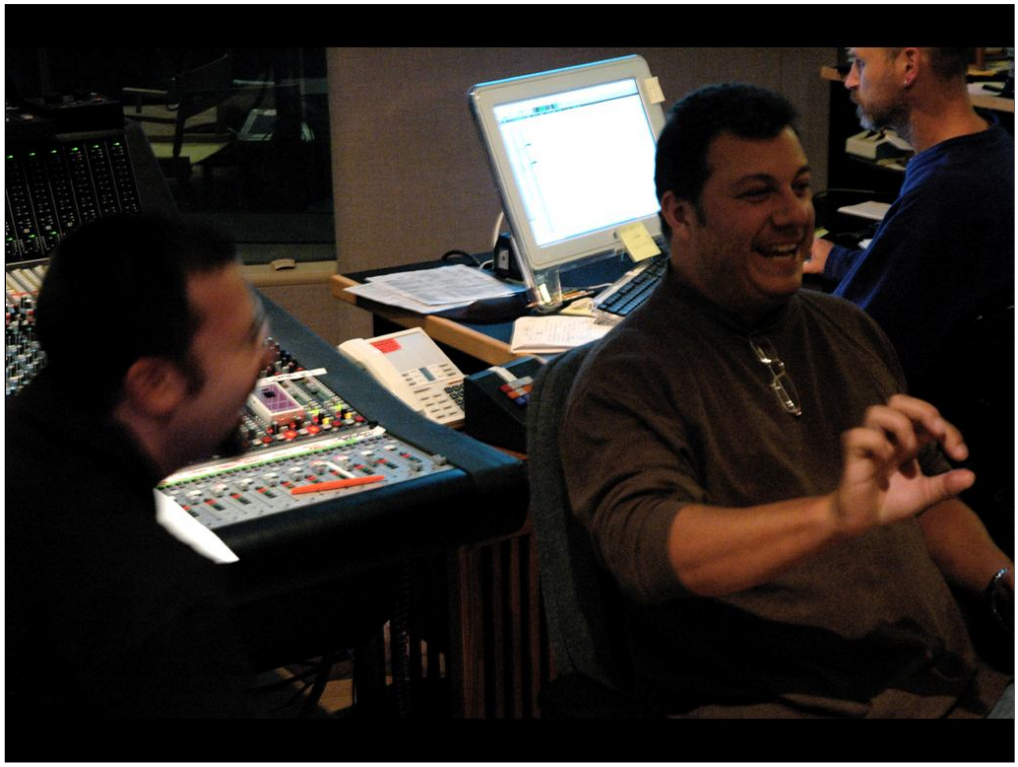
Some great tools had come together by this time that let us build and reload the game's content without re-running the game, letting us iterate more rapidly, including Charter, our level editing tool that you can see here, which we used to place objects in the game, as well at the signal regions that we used to trigger in-game events and control the streaming of the levels.

We had a great scripting system – the first time we'd done scripting at Naughty Dog. We got some excellent advice from the *God of War* team at Sony Santa Monica, and one of our brilliant programmers created what we called our IGC (or In-Game Cinematic) Scripting System.

We initially created the IGC system to script the behaviors of the player-character and the AI-controlled characters during in-game events, but we ended up using it for nearly all of the spot environmental interactions that you find in the game - like turning wheels to open doors in the u-boat, and crumbling platforms, and puzzles - and this let us set up a ton more gameplay that we would have otherwise been able to.

Music was of course very important in *Uncharted*, for creating mood and heightening drama, and to make sure that the music reflected the events unfolding the game very closely, we created an adaptive music system in-house at Naughty Dog.

But for the first time we farmed out the work setting up the adaptive music to the fantastic staff of the SCEA Sound and Music Group.

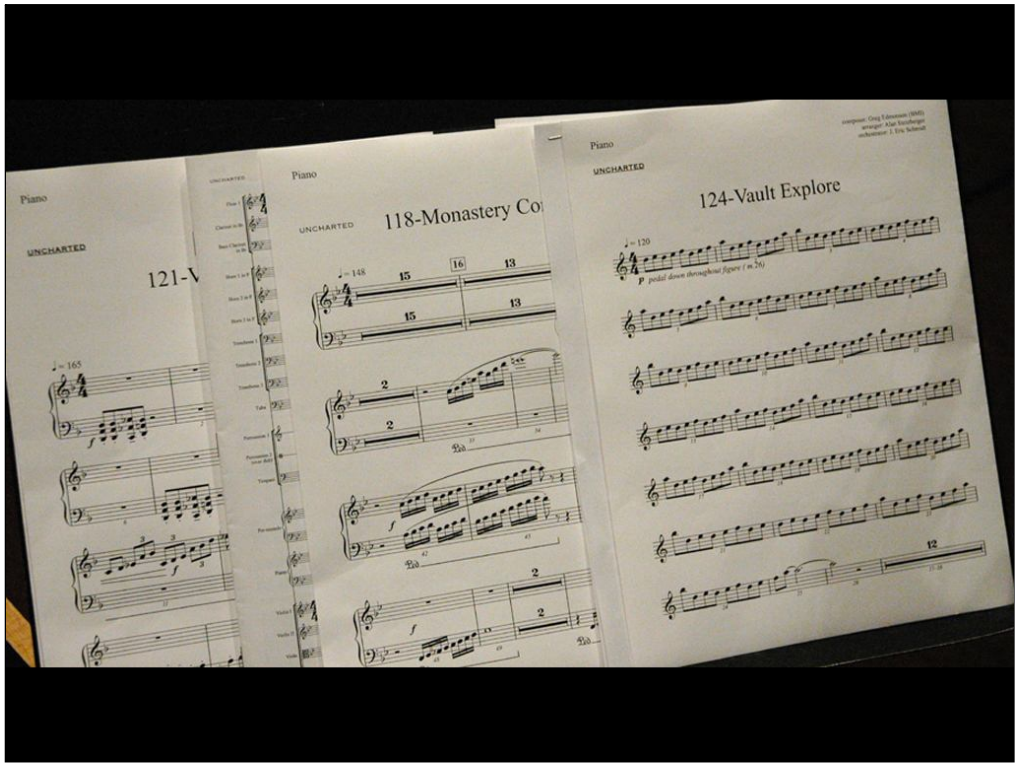


The Sony guys were very enthusiastic, and worked very closely with us, letting us set the creative and technical agenda for the music.

They scripted the adaptive, interactive part of the whole game, and even came to down to Naughty Dog at the end of the project to work in-house and finesse the adaptive scheme and its triggers.

We were very lucky to work with Greg Edmonson, the composer for Joss Whedon's *Firefly.* We loved Greg's work and the way he combines classical symphonic elements, contemporary instruments, and unusual instruments from around the world.

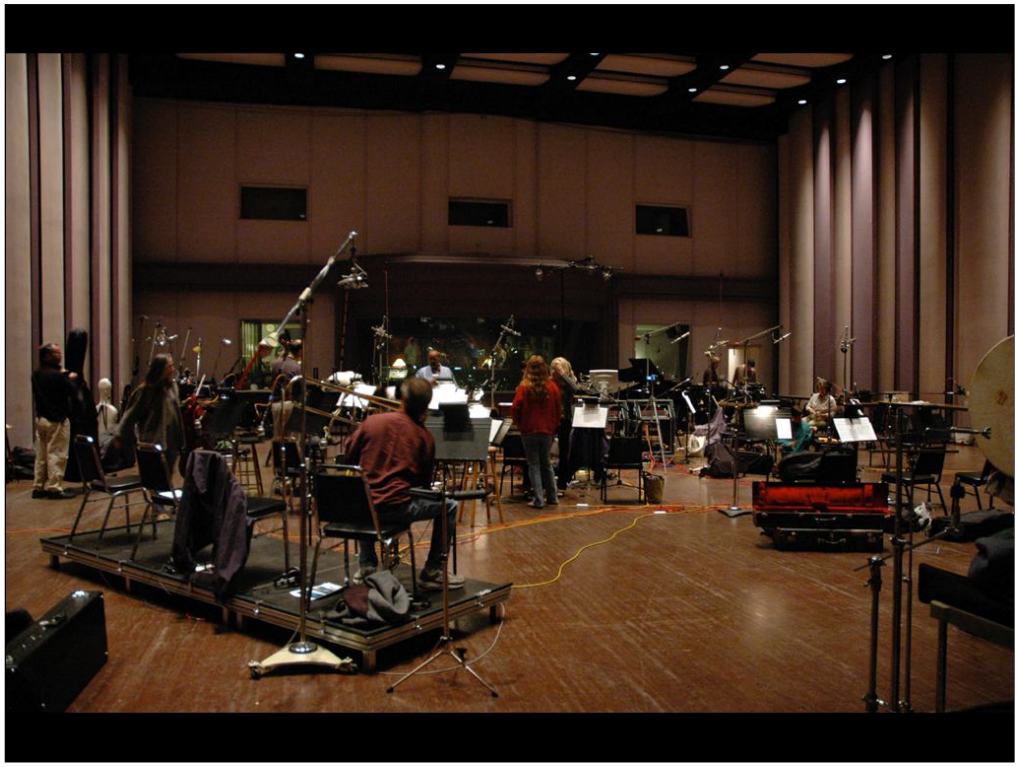
His work has an atmosphere that's both familiar and other-worldly, which was perfect for our game.



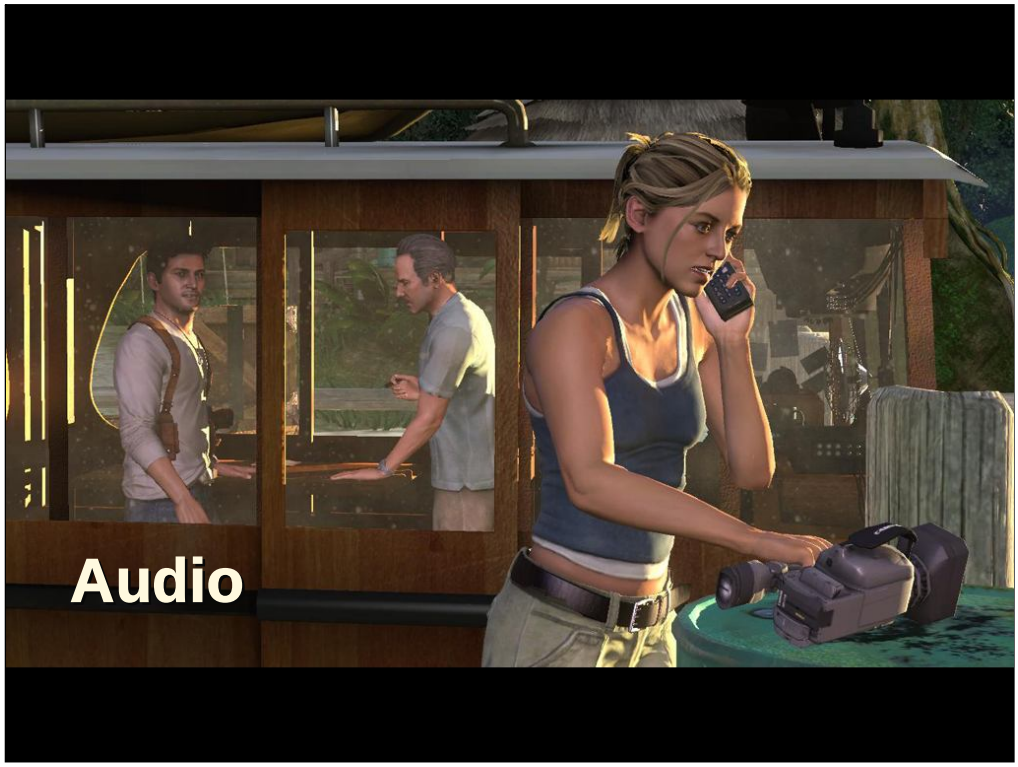
Greg hadn’t done an interactive project before, but he was very passionate from the beginning about understanding our vision.

We showed him the game and he quickly got very much into its spirit.

He was already very technical, and so he was able to work closely with the guys at SCEA to turn his linear scores into the branching systems that make up our adaptive score.



We did our live recording sessions at Skywalker Sound, where they have a great facility that lets them isolate different parts of the orchestra in soundproof booths linked together with video feeds, so that everyone can follow the conductor. This means that each part can later be muted or its volume changed, making the adaptivity of the music possible.



Our audio process, setting up sound effects for the game, was very down-to-the-wire, but both the Naughty Dogs and the Sony employees who helped us implement the audio in-house worked very hard, and we were very happy with the results that we achieved.

We took a data-driven approach to audio that was new for us, which let us implement our audio very quickly and make changes without having to recompile code. We ended up with over 1,200 lines of in-game dialogue, which we feel really helps reinforce the reality of the game world!

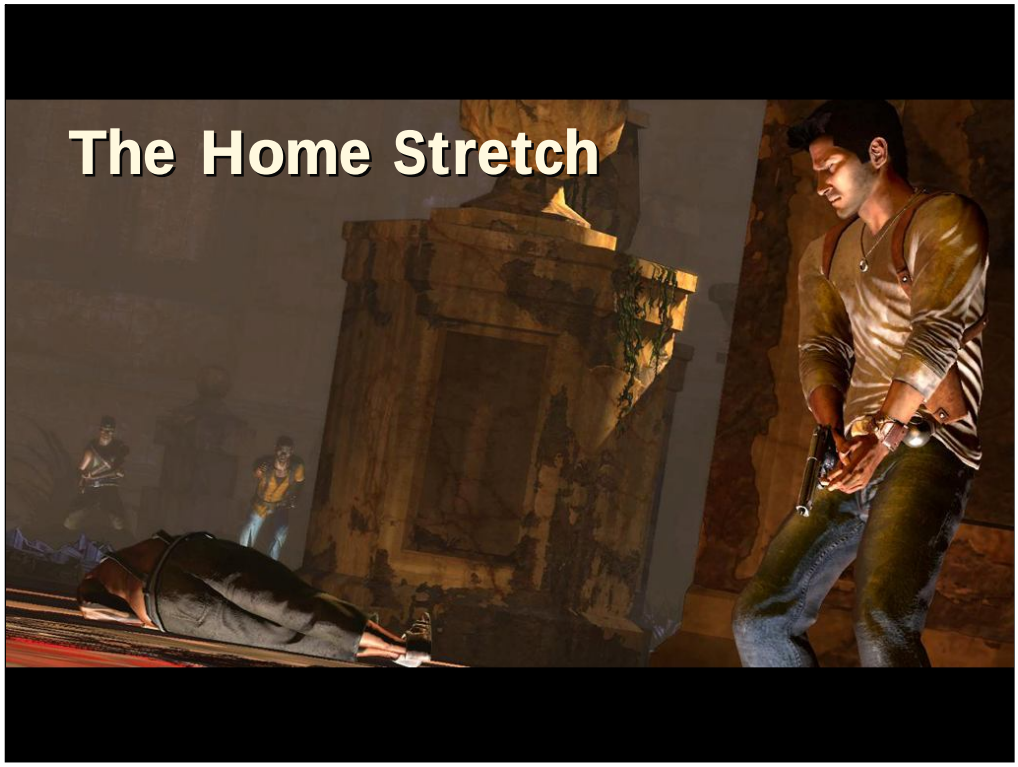
In the summer of 2007 we made a lot of demos – maybe 5 demos and 6 playtest builds.

We think that demos are something of a double-edged sword - it sucks having to make them, and an unpolished early demo can leave people with a bad impression of your game.

But aside from creating good buzz in the lead-up to a game's release, demos really help force the team to achieve concrete goals in terms of getting different parts of the game finished to a good level of polish.

Also, the feedback we get from people who play our demos, both the gaming press and the public, really helps focus our attention to the parts of the game that need the most work - for instance, we got kinda dinged for our aiming scheme at E3 last year - we weren't yet happy with it at the time, but the constructive negative feedback that we got ultimately helped us fix the issues we had.

I want to stress that we do very little throw-away work for our demos. Because of the task graph and its modular nature, we can excerpt a section of it with a relatively small number of script and code changes to the game. Always keeping the game unbroken really helps with this too. But next time we think we'll try to make a smaller number of demos and use them for longer.



About six weeks out, we decided to move Alpha up by two weeks.

We'd scoped the last few of the game's levels to get built as we approached Alpha, reducing them in size and complexity, which actually helped the game's pace a lot, in retrospect.

We hit Alpha well, and we think that even though it was tough at the time, moving Alpha up was extremely smart, since it gave us just enough time to polish what we had.

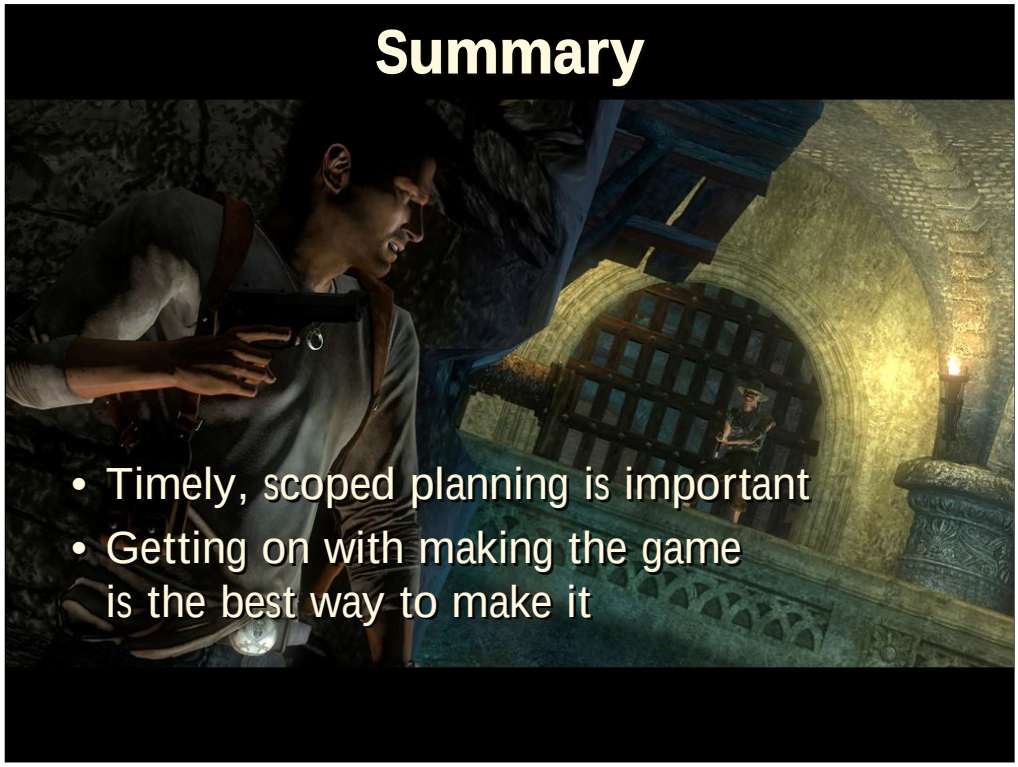
Finally, we hit Beta on September 14th, and from there on we were pretty much just fixing bugs, adding the last few bits of bonus content, and doing a final pass of gameplay tuning.

Just before Gold Master, we made a trailer using the track “Dissolved Girl” by Massive Attack, of whom we’re huge fans - we’d liked all of our trailers a lot, but we thought this one was something really special.

On October the 19th, we hit Gold Master, and this is pretty much how we felt!

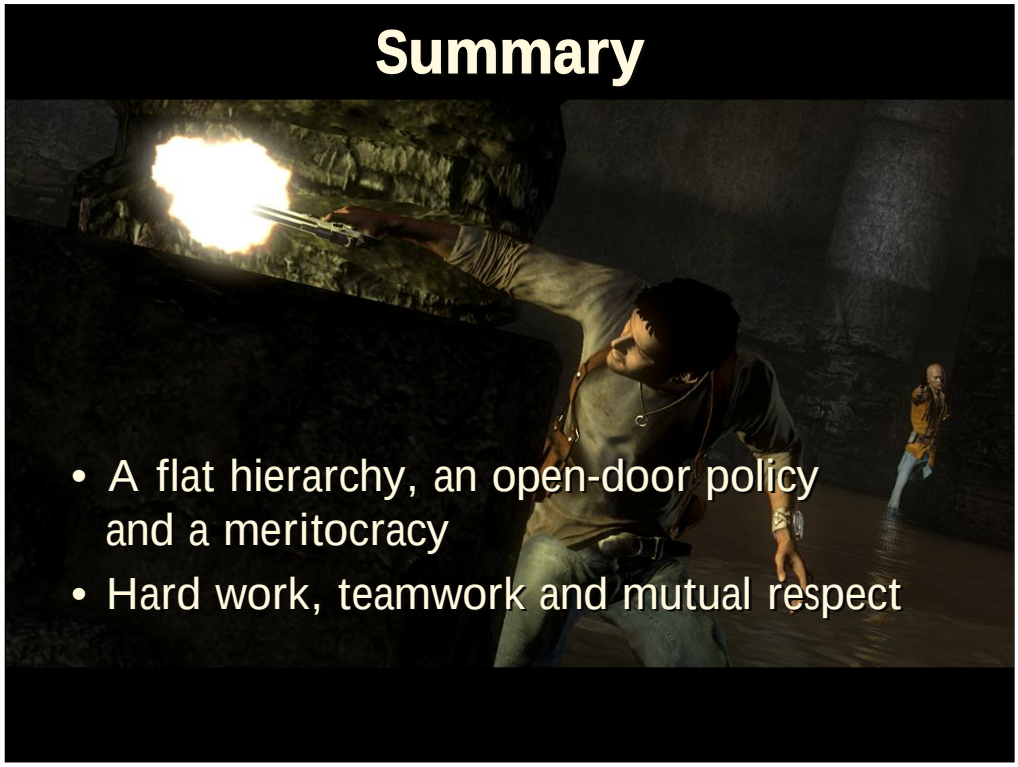
We were very happy with the way that everything came together.

We'd said throughout development that we thought *Uncharted* would be Naughty Dog's best game ever, but we were certain that this was the case by the time we shipped, and the critical and public reception we've received has been everything we'd hoped for.



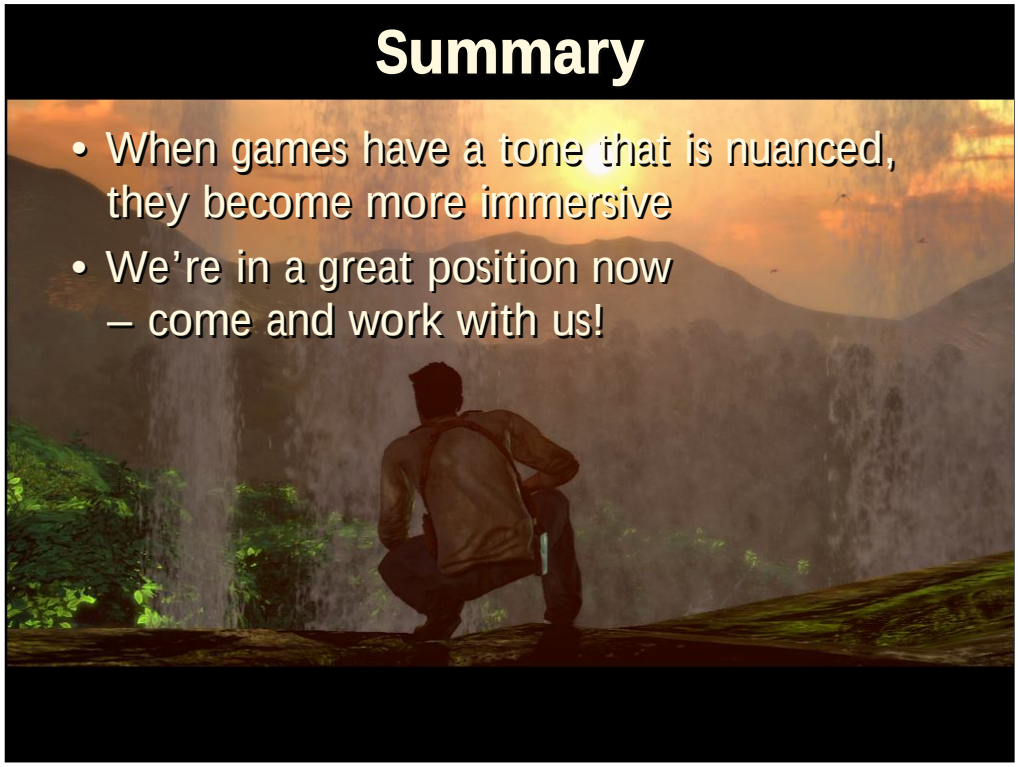
Timely, scoped planning is important. You have to come out of pre-production with a strong plan for your game, and create the micro design right before it's needed. However, don't over-plan, because you'll make discoveries as you go along that will lead to changes.

Getting on with making the game is the best way to make it. Build something bare-bones that gets the job done, and then expand and refine the things that work well or show potential. People become more creative within the structure of a limited tool or game mechanic that's solid, simply because they can do work and iterate.



A flat hierarchy, an open-door policy and a meritocracy are all really important to us. The freedom of the team to participate, be constructively critical, and innovate – when anyone can step up to the plate, get involved, and see an idea through to completion – is always inspiring, investing and motivating.

And the flip side of that is hard work, teamwork and mutual respect. People at Naughty Dog are generally very respectful of each other's design opinions, and that culture, of listening, thinking and discussing respectfully, staying focused on the objective merits of the game, propagates across all disciplines and parts of the production.



We believe that when games have a tone that is nuanced, they essentially become more immersive. Story games don't have to take themselves so seriously, be so melodramatic and be very long and difficult. Games can simply be charming and transporting. I think that settling down with a game that takes you away to somewhere amazing for a few hours is the kind of immersion that a lot of us are always chasing.

Despite having worked on the PlayStation 2 for 6 years, we caught up in terms of new technology (like shaders) and we hope that we've even done something to raise the bar. We were able to preserve our company culture despite an increase in team size, and we have better tools, play mechanics and AI code now than what was comparable at the end of our first *Jak* game.

We haven't announced our next project yet, but whatever it is, we're excited about it, and poised to make Naughty Dog's next best-game-evar! In short, we're in a great position now – so why not come and work with us?

So that's it – thanks for listening! Thanks to everyone who helped me make this presentation, and who helped us make *Uncharted*: all the Naughty Dogs, past and present, Sony Computer Entertainment, and all of our family and friends.