



## **SOFTIMAGE | XSI Case Study: Capcom - DEVIL MAY CRY 4**

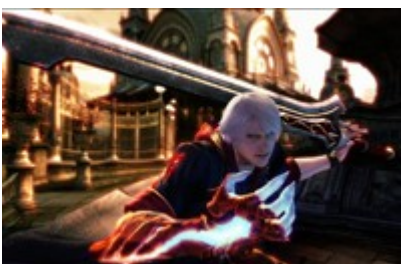
By Takashi Umezawa

Released on January 31, 2008 for the PlayStation 3 (PS3) and Xbox 360, Devil May Cry 4 (DMC4) leads the pack in the "stylish action" genre, having already racked up approximately 2.3 million sales. The game takes its place as one of Capcom's many million-selling hit titles. About 6.9 million games in the Devil May Cry series have been sold in the past. Including the new release, sales will soon reach 10 million—evidence of the incredible popularity of this series. The producer, Hiroyuki Kobayashi, explained that the worldwide market was their target all through the series, which explains the use of English dialog and why non-Japanese actors were used in the motion capture recordings. Capcom has offices in North America and Europe, and the team made full use of this network in creating a game with a global perspective that took into account differences in language and culture. These overseas offices also helped in developing a nuanced portrayal of Sparda, the religion that was created especially for the game.

In this project, the team continued to use MT Framework, a game development environment that they developed for themselves in-house. But the birth of next generation of hardware systems changed how they worked. DMC4 was the company's first title for the PS3, so every stage of development was a novel experience. The initial goal for all members of the development team was to create a fast action game for the next generation consoles at 60 frames per second (FPS). That may seem audacious, but Capcom has always strived to stay at the forefront of innovation by taking game development to its limits. We saw Capcom's enthusiasm for research and development when they showed the making of Onimusha at Siggraph a few years ago, and they have grown even more passionate since then.

Development started on DMC4 before the launch of the Sony PS3. During development, they were constantly testing out new things. Their trial and error operations involved the new hardware, HD video production, the new characters and general issues regarding production data. Work began in earnest on development for the PC version after development for the PS3 and Xbox 360 versions had been completed. With MT Framework, this work in itself was not so difficult. But the checking work for compatibility with a wide range of PC hardware and operating systems took a long time.

The PC version was launched on July 24, and to mark this occasion we spoke to various people who were involved in the development.



Makoto Tanaka had the following to say. "At first we all started out with the figures, saying let's use X number of polygons for the characters, X polygons for the backgrounds, X joints for the characters and X bones for facial animations. We were all wrong! (Laughs) We were dreaming when we started to create the characters. But this process did give us a vision of what we wanted to do. We created motion data by actually moving elements and using XSI to repeatedly add or take them away, while maintaining quality. This work was very challenging."

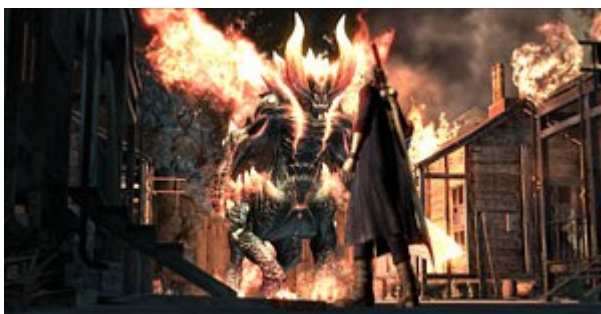
During development, each team faced serious issues, came up against challenging obstacles and had to resolve major problems. But thanks to XSI's useful functions for game development, the project went smoothly, and they maintained quality and data size while finishing the huge project on time.

### Real-time Demo

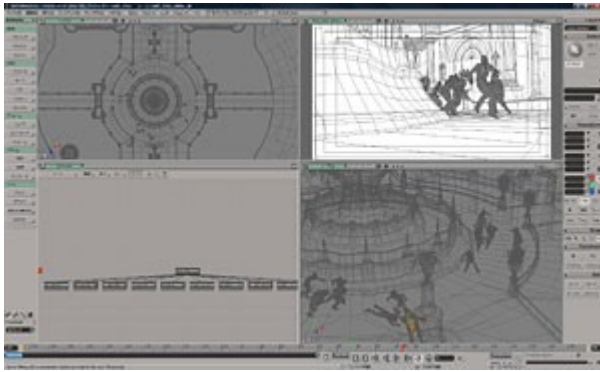
Takayasu Yanagihara was in charge of the production process for the real-time demo and progress control for subcontractors. He said, "Actually, although the game itself is 60 FPS, the real-time demo was 30 FPS. In the real-time demo, we gave priority to the images." This seemed to conflict with their initial target. When we inquired further, Mr. Yanagihara said that they came to this decision after repeated trial and error. Of course, this does not mean that they did not prioritize the images in the 60 FPS gameplay: this is obvious just by looking at the actual game. So why did they use 30 FPS in the real-time demo?



In a movie demo, the most important thing is the beauty of the video. Since this was their first PS3 development, it was difficult for them to set a target and predict the amount of work that was required. For this reason, while developing DMC4, the team looked at titles that were being released by other companies and inspired by these, they set new targets for themselves. By repeating this process, they rapidly improved quality. Initially, their target was to use 60 FPS for the real-time demo too. But to make Kyrie look cute or Dante look cool, they had to adjust the lighting as appropriate and so the required volume of lighting grew. But while lighting improves the appearance, it also increases the processing load. After seeing the progress of this work, director Hideaki Itsuno took the bold decision that if they could not create the event movie with satisfactory quality at 60 FPS, then they should use 30 FPS.



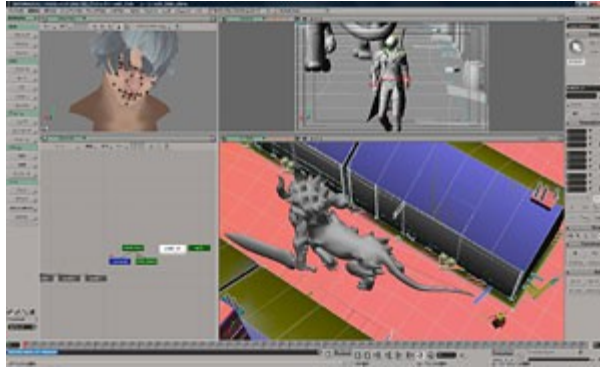
When SOFTIMAGE|XSI software was used to create such an event scene, they created it with a single scene for each cut. They asked Yuji Shimomura to continue his work from DMC3 and direct each cut. By creating pre-visualization with live-action video, it was easy for the team to understand the required images, which led to a smooth production process. The result is video scenes with a good tempo that links together with the gameplay in an enjoyable way. Each cut is made up of several seconds of animation. The fact that they could divide work between team members for each cut made the work very efficient and was a great advantage. Even though a cut is only a few seconds, loading all the background, character models and animation involved a considerable amount of data. But the team said that even under such conditions, the operations in XSI were stable and smooth.



Event scene on XSI



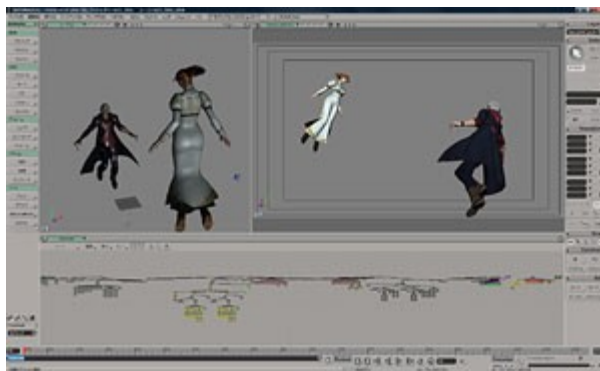
Actual event scene



Event scene on XSI



Actual event scene



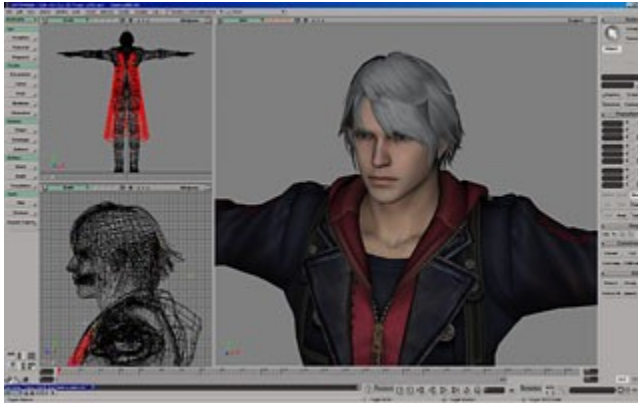
Event scene on XSI



Actual event scene

## Character Modeling

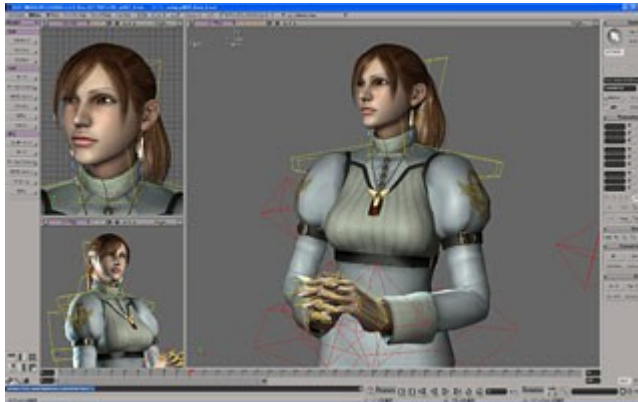
Although DMC4 had great significance as Capcom's first PS3 title, elsewhere at the company advance work had already started on a different project for a next-generation console. They used data from this other project as a reference for creating models, which enabled them to successfully finish the development without changing their initial target. The actual data size was three times that of the previous PS2 title. The polygon data size increased from 5,000 to 15,000 polygons, while the texture increased from 256x256 to 1024x1024. Hiromitsu Kawashima was in charge of creating Nero and Dante, Naru Omori was in charge of creating Gloria and advertising materials, and Jun Ikawa was in charge of creating Kyrie. In all, nearly 100 different character types had to be created, including the main cast, enemies and ordinary citizens who appear in the event scenes (such as old men and women, children, fat people, religious knights). But thanks to the non-destructive environment and intuitive modeling interface in XSI, the work proceeded very smoothly. SOFTIMAGE|XSI software allowed for flexible changes in all stages of work; for example, reducing the number of bones in models for ordinary citizens, or editing the apex even after weight is added to completed models for use in creating other models.



Nero model in XSI



Completed Nero image



Kyrie model in XSI



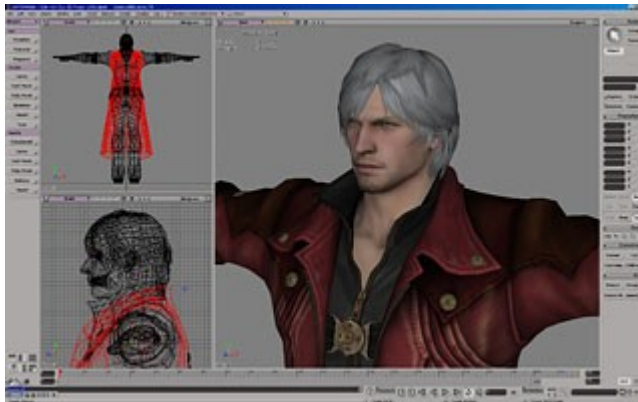
Completed Kyrie image



Gloria model in XSI



Completed Gloria image



Dante model in XSI



Completed Dante image

The team said that the same model data and texture for all the characters that appear in DMC4 was used everywhere: in the gameplay, event movies and advertisements. In other words, normal maps were used even for models in the gameplay. When textures for models in the gameplay were imported from the archive, they were used after being scaled down in real time. This enabled the effective use of hardware memory and fast movements at 60 FPS. However, the textures were not blindly scaled down to a uniform size; rather, appropriate adjustments

were made depending on the map type. Maps such as defuse maps were scaled down to a 1/4 size. But normal maps lose detail if they are made too small, so they were scaled down to 1/2.

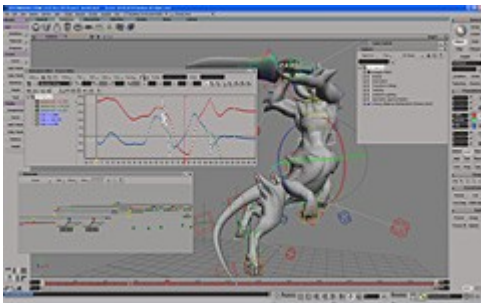
In the PC version that was released on July 24, the team set their own quality controls to allow gameplay with their own original texture size. DMC4 is next-generation entertainment, and as such it can be played on hardware with the most advanced specifications.



Difference in texture resolution on the PC version (right)

### Fastest Motion in Gaming History

One unavoidable problem when creating character animation is gimbal lock. When an animator tries to move an IK bone above a certain value, the chain rotates un-naturally. Because stylized action is a major attraction of the Devil May Cry series, we asked the team how they avoided this problem when setting high speed motion.

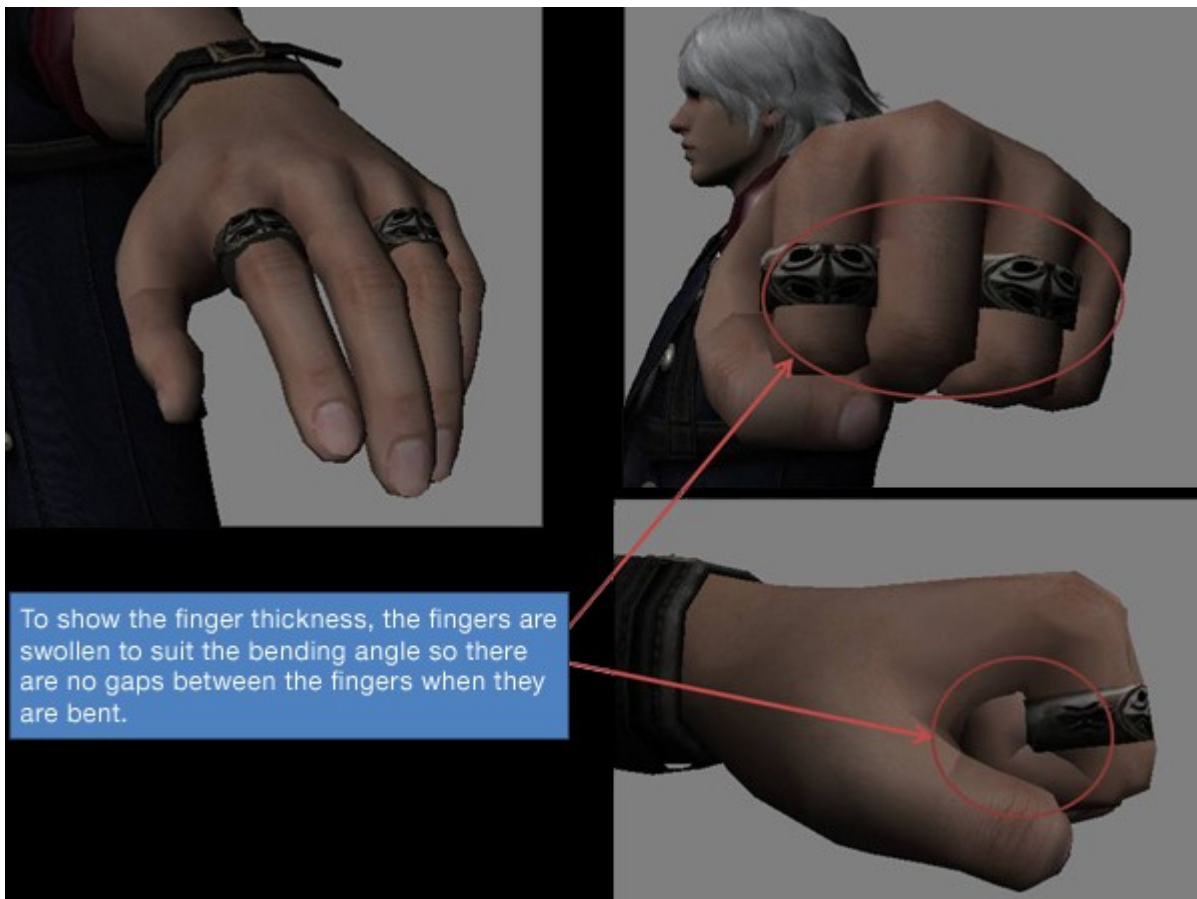


Team members said that they used an FCurve link mechanism with an auxiliary Null so that only the X rotation was retrieved for the character rig. If the X rotation exceeded a certain value, a different auxiliary Null would be used for control. However, even with this mechanism, there is a limit to the control possible with the three axes of XYZ. This meant that the auxiliary Null rig often failed

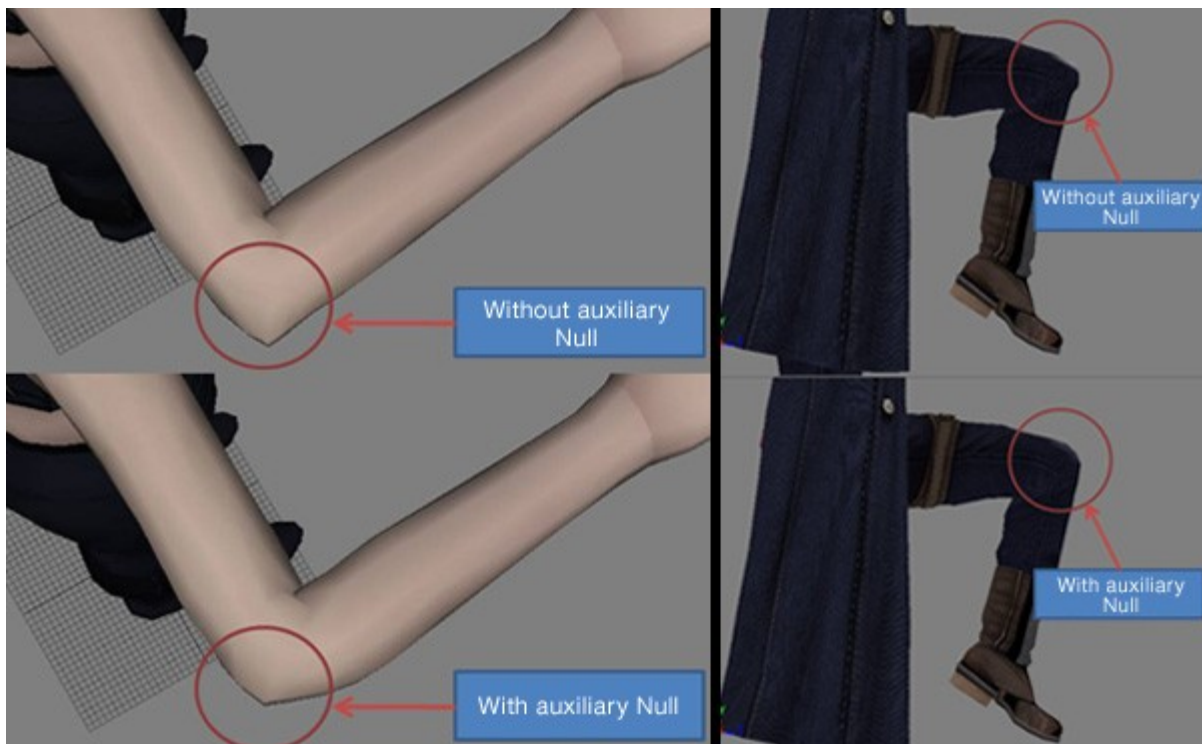


But if they added a tool to avoid the gimbal lock, it would also inevitably result in some sort of restriction. A rig should make it easy for animators to set the movements that they want. But the team knew that a rig for the gimbal lock would suffocate the creativity of the animators and put them off their work. For this reason, to avoid interfering with the stylish work of the animators, the team decided against introducing a rig to resolve the gimbal lock issue. When gimbal lock occurred, they corrected it by manually inputting the figures, or they corrected the FCurve from the plotted results using the Animation Editor. Normally in XSI, the rotation order is set as XYZ, but in DMC4, work was performed with a YXZ setting where the Y axis rotation is prioritized. The team also used a mechanism to ensure the optimum envelope condition when characters bend their fingers or knees. They also

embedded many other mechanisms, but this is confidential company information.



Swelling when bending the fingers is adjusted



Comparison of elbows and knees with and without auxiliary Null

**So how did they create the motion rigs, which are so beloved of animators?**

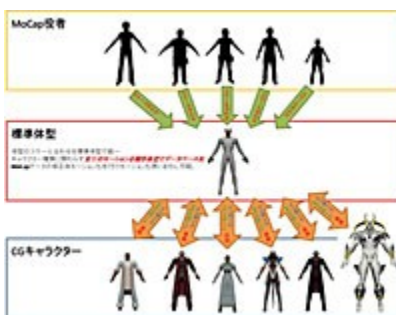
The team said that in past productions, work was very laborious because the rig structure and the number of models and bones were different for each model. For this reason, in DMC4 three key staff members came together to decide the optimum workflow. These were Jun Ikawa in charge of modeler management and Kyrie; Hiroyuki

Nara in charge of motion management and creator of Dante's player motion; and Takayasu Yanagihara, who set up the motion and facial rigs and was in charge of event demo management. The result, as mentioned earlier, was the decision to use the same model texture and rig for everything: the gameplay, event demos and data for advertising media. The only exceptions were the demo movie and the number of bones for character faces in gameplay. All other structures and number of bones were exactly the same, to eliminate wasteful work. This rig was created by Mr. Yanagihara and he succeeded in constructing a workflow that was fully optimized and did not fail.

Mr. Yanagihara explained that when the team started rig creation for DMC4, they had not yet finished any models. For this reason, they based the rig on the body shape of Dante's model in the previous DMC3 production and then performed a wide range of modifications. This was the standard template.

Mr. Yanagihara himself created a script that automatically performed corrections to this standard template to make it compatible with other characters that appear in the game, who have different hand, leg and body sizes. Of course, model scale adjustments were performed manually, but only two adjustments were required, the first for the scale from the ground to the stomach and the second from the ground to the knees. When the script was executed after performing these adjustments, a model with a controller was completed after about a minute, even when the character sizes were different. This meant that the team could start to add motion to character models as soon as they were finished by the animators.

Although the team used a large amount of motion capture data in the event demos, their target was to work with XSI and the standard template, without using Motion Builder. Usually with motion capture data, motion for specific characters is shot using specific actors, and then the animation data is created. Because the body sizes are different, reusing this data for different characters would involve shooting the motion capture again for a new rig, or performing adjustment using functions such as motion retarget.



**Diagram showing the reuse of motion with the standard template**

To avoid this work, Mr. Yanagihara requested that the captured data be delivered in a form suitable for the standard template. This meant that all captured data could be reused for all characters and that even if a character's model data had not yet been created, other team members could press ahead with the capture work. Of course, the motions added by animators could also be output to the animation presettings, enabling reuse for all characters with a single click of the export button.

This workflow also had many other unanticipated merits. The first was that using the same mechanism allowed changes right up to the final stages of production: usually, in game production it is strictly forbidden to change elements such as hand length or leg length partway through the project. Although the bending of hands required adjustment, motions that did not involve sliding the legs could be converted automatically within a few minutes. This resulted in stress-free work for both modelers and animators. The second merit was that the workflow was extremely useful when importing data from subcontractors into the in-house specifications. The Capcom team provided subcontractors with the same tool set and asked them to work according to the same workflow. However, on some delivered data, a large-volume Null was used over the standard template. This was because, for example, if an animator wanted to move the body of a character as the character had his hands on a desk, the animator would add a Null to set the desk and constrain. Even this could be set as an animation presetting, usable with the standard template with one click of the export button.

For hand control, the team used four more bones than in the previous title, making a total of 20 bones. The extra four bones were added to give a curved motion to the back of the hand during hand movements such as touching the little finger with the thumb. When setting hand animation, the team was able to use synoptic views. They could call up presettings such as shapes for holding a gun or sword. They imported the default pattern and then

performed detailed adjustments for the finger joints. In the same way as the character rigs, the synoptic views for the hands could be switched to any character with a simple tab operation. Even with many different characters, this made it easy to create animation for hand gestures that suited each individual character. The document that describes this work method runs to 50 pages and enabled both in-house and external designers to perform the best possible work in the shortest possible time.

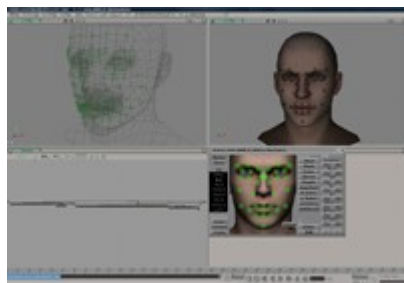
### Easy-to-Develop Synoptic Views and Scripts

The team had to outsource a lot of work for this game development and demo movie to external CG companies and freelancers, because they were under pressure to create a huge amount of data. In this context, the synoptic view was extremely useful for unifying the workflow between the in-house team and external subcontractors. Game data are created under extremely strict conditions: if you get only a single value incorrect, the result will be completely different from what you intended. To avoid this problem, all the required functions were embedded as a script function within synoptic views, so that even XSI beginners could work without making any mistakes. As such, the production environment was exactly the same for the in-house and external teams.

For example, during motion creation, one function allowed the automatic setting of animation keys simply by operating a rig controller and without opening Explorer or Schematic. Another allowed character body, hand or face movements to be set as presets and the data to be output to the PS3. Of course, the team also created other automated processes, such as setting noise to camera movements.

The team also used XSI in DMC3, when they created a large number of scripts. This accumulated know-how was invaluable in DMC4. When the team faced a serious problem, they were able to quickly resolve it, thanks to a script. This problem was an error in the rig controller. The parent-child relationship was cut and movement in the buttocks area became unnatural. The model team had already finished dozens of models, so they had gone too far to redo them. Instead of starting again from the beginning, the team gave the modelers and animators a script that performed the correction. By executing this script, they were able to quickly resolve the problem. In this way, XSI's expressions, constrains and scripts were extremely useful in making their workflow effective. Without these functions, their desired workflow would probably not have been possible. Mr. Yanagihara said that the scripts were also extremely useful in avoiding problems in advance.

### Challenge of Facial Animation



If you take a look at the demo scenes, you'll see that facial expressions such as surprise are extremely realistic. Mr. Kawashima created these character facial animations, Mr. Nara created the presets, and Mr. Yanagihara created the actual facial rigs. They knew that facial animation would take the most time, so they created various mechanisms to simplify the work and make it more efficient. For DMC4, they constructed a new system that was inspired by SOFTIMAGE|FACE ROBOT.

Mr. Yanagihara said, "We used a three-stage system where about 30 markers, 150 bones and a separate slider could be used." The facial rig setup for the characters was automated in the same way as for the standard template. When they imported the rig and model, pressed a button and waited for about one minute, the rig adjustment was performed automatically to suit each face model. This was then applied to the face model, enabling the facial animation settings to be performed immediately. In the same way, there was only one synoptic view type, so it could be switched to any character using a tab within the synoptic view.

So, how did they create facial animation using these rigs and synoptic views? Manual setting for the nearly 150 bone animations in a face model would be very inefficient. Instead, first they developed sliders for controlling vowel sounds and emotions. By moving these sliders, they could generate rough facial animations. These sliders were connected to the markers. The moved markers controlled the various bones in the surrounding area based on the set expressions. For example, when a marker was moved to close the corners of a mouth, the lips would bulge out before they shut. Or when the line of sight of a character was moved, areas such as the eyebrows would move in tandem and the nostrils would swell. If the slider adjustments did not produce satisfactory results, they performed further adjustments by adding marker offsets on top. If the adjustments were still not satisfactory after moving the marker again, they performed fine adjustments by moving individual bones to control the expressions. On the PS3, animations involving these 150 bones move at 30 FPS.



Basic emotional expressions for each character were registered in the synoptic view. These could then be called up with a single click of a button. Had this not been available, each designer would have had to create emotions from scratch each time, and a smiling face for Nero would look completely different depending on which designer created it. By using standard, uniform expressions and then performing fine adjustments, they minimized variations between different designers, enabling everyone, including subcontractors, to work more efficiently.

Other useful functions for facial editing were also included, such as mirror-image reversal, and reversing the X axis only.

Such high-quality facial animation was the direct result of Mr. Kawashima's hard work in performing weight adjustments for close to 150 bones for each character. He said that XSI helped him hugely because the non-destructive environment meant that he did not have to start adding weights from scratch each time. Also, XSI allowed him to add weight to only half a character and then perform symmetry copy to finish the process.

### Large-scale Background Production

The world of DMC4 looks like a mixture between Asia and Europe. They decided to go to Turkey to shoot materials because the director was keen to go, Capcom's texture library was from the PS2 era, and the amount of information in 256-resolution was too low. As for the results of this shooting trip... well, take a look at the game screens and you'll see.

Their superb depiction of buildings has a unique atmosphere; not quite Asia, but not quite Europe either. The shadows and textures are also exquisite. However, had they applied the real-time lighting process to all objects, the load would have become too heavy. So with the exception of certain stages such as the jungle stage, the real-time shadow process was only used for the characters.



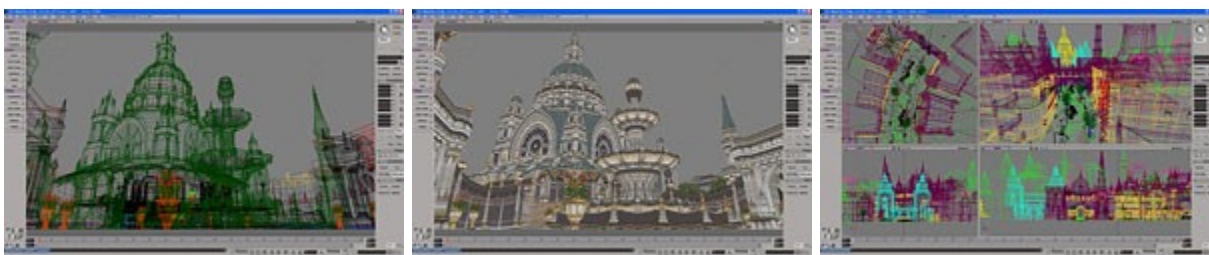
Capcom developed a mechanism for displaying pre-rendered results in the game using the integrated mental ray® renderer in XSI. This allowed them to create light maps with XSI for the background and surrounding objects and then use them in the game.

The information from final gathering and ambient occlusion during mental ray rendering could be baked in a light map. UV was then generated for the resulting texture with unique UV and they were used together to output apex color.



This enabled them to reduce the number of textures, and achieve a gameplay of 60 FPS while minimizing the size. Although the details are confidential company information, Mr. Tanaka did say, "Without XSI's render map, render vertex color and light map, we would not have been able to achieve such high quality." Mr. Tanaka was the coordinator for the background team in DMC4, and he also performed modeling work himself.

In DMC4, they created 56 scenes as new background data. Including past background scenes that they reused, there are 70 scenes in all. When playing the game, sometimes opening a door is a trigger to jump to a new scene; these switch scenes are included in the number of scenes. Although the size of each scene was 300 MB to 400 MB, thanks to the gigapolygon core, smooth work was possible even in 32-bit environments.



Finally

If you still haven't played this game, check out the Test Version and Benchmark. If you play the game after reading this article, you'll see the buildings and characters in a slightly different way. As you gaze at the mosque-like building in the opening scene, or at the shadows and texture of the fountains, your eyes will doubt that such beauty comes from prosaic data. And as you look back to the real world, you may feel a little uneasy and find it difficult to decide which world is more beautiful, or more real...

The team that we interviewed



Capcom Co., Ltd.

(Front row from left)

Hiroyuki Nara, Character & Motion Section Leader;

Takayasu Yanagihara, Event Section Leader;

Naru Omori, Model Manager

(Back row from left)

Makoto Tanaka, Background Section Leader;

Jun Ikawa, Model Section Leader;

Hiromitsu Kawashima, Model Facial Manager