

# THE CODE/ART DIVIDE



## HOW TECHNICAL ARTISTS BRIDGE THE GAP

» AS GAME DEVELOPMENT BECOMES MORE COMPLEX, with bloated budgets and team sizes doubling over last generation development, the need to bridge the gap between art and programming has never been more pressing. It's already passé to rely on programmers to develop art pipelines and understand those needs. Game development today needs to be far more efficient, able to produce high-quality triple-A titles with team sizes comparable to those seen in the last hardware generation.

The technical artist is a new concept and role in the game industry that is starting to take hold. Every company has a different idea of what a technical artist's roles and responsibilities are. However, to really maximize the development process, a company must integrate the technical artist to the fullest capacity.

What follows is a blueprint for how we've integrated technical artists into our game development process at

Volition, which will hopefully give you ideas as to how your studio can do the same.

### CASE IN POINT

During development of SAINTS ROW for the Xbox 360, I was responsible for designing and developing many of the tools and pipelines used for getting art assets into the game. But before Volition had any technical artists, it was up to the programmers to design and develop these systems for the artists to use.

Like any studio that doesn't employ technical artists, by and large programmers are the ones who develop and support the art pipelines. Generally, an artist will submit a request, and at some point in the future they are presented a tool to use. Most of the time these tools are not easy to use from an artist's perspective, nor is the workflow clear. Moreover, diagnosing problems with the

---

**JASON HAYES** is a technical art director at Volition, working on an unannounced game. He was senior technical artist on SAINTS ROW for the Xbox 360 and has been in the industry for 10 years. Contact him at [jhayes@gdmag.com](mailto:jhayes@gdmag.com).

# THE CODE/ART DIVIDE

tools can become problematic because getting programmer time to address them is typically difficult, especially near the end of a project cycle.

Integrating technical artists into a studio frees up the programmers from being solely responsible for the development and maintenance of the game's tools and pipelines. While programmers still have a hand in the design (and sometimes implementation) of those tools and pipelines, the technical artist is the driving force behind them and is looking out for the best interests of both parties. This allows programmers to focus more on developing game code, and artists to focus on making the best-looking content they can with easy-to-use tools and workflows.

## THE ART OF DIPLOMACY

The following scenario, which took place near the end of development for *SAINTS ROW*, illustrates the importance of utilizing a technical artist.

Our game was having some serious frame rate problems, especially during nighttime gameplay. This happened for a number of reasons, but primarily because we were developing the game long before any hardware was available, and it was a new type of genre for the studio.

One of the many causes for the frame rate problems was our liberal use of per-pixel dynamic lighting. Since we were running low on time, many people on the programming side felt that it would be best to turn off the dynamic lighting at night and fake it with effects. On the art side, there was a desire to keep it because the lighting gave the night scenes a much better sense of believability and richness. All things being equal, programming would have won that fight because it's better to ship a game with stable frame rate than not.

Because of my knowledge of the engine and its capabilities and limitations, I proposed that we develop a hybrid solution to the problem. Dynamic lights would remain on at certain distances around the player, while further out, effects gave the impression of a much more well lit city without paying the GPU and CPU costs associated with the dynamic lighting. This

along with other optimizations improved the frame rate during nighttime gameplay dramatically, which kept the programmers happy. And visually, we still retained the believability and richness the artists wanted.

Cases like the one I described above happen often, sometimes daily. It's important to have an experienced technical artist between the two disciplines to negotiate what's best and important for not only each of the respective departments, but the product as a whole.

## PIPELINE AND SYSTEMS ARCHITECT

Generally speaking, the technical artist should be able to design and develop all art pipelines necessary for the game. In this sense, part of the technical artist's role is to be a pipeline and systems architect. At Volition, this works on a few different levels.

Depending on the system, we work with the programming and art departments to determine what works best for both parties

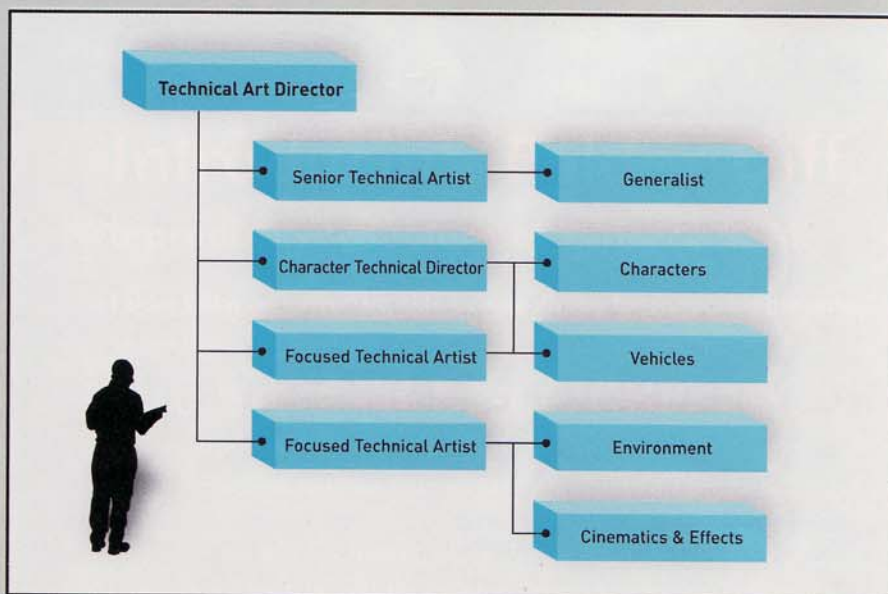


FIGURE 1 Volition's typical technical art team structure is shown.

and try to reach common ground. At this level, you could say that we act as negotiators between the technical- and content-oriented disciplines.

However, designing critical game systems requires technical artists who have intimate knowledge of both the game engine and development hardware, such as the Xbox 360 or PlayStation 3. The degree of knowledge necessary is such that if the technical artist isn't experienced enough or hasn't made due diligence a priority, pipelines can quickly take a turn for the



worse. More often than not, early mistakes are felt in the middle of production, or even later in the development cycle.

Not only do technical artists design and spec-out these systems (in coordination with other disciplines), they are the ones driving and championing the changes. Because of their intimacy with such systems, it makes them the primary source of information when it comes to how things work and fixing bugs in tools written to support the pipeline.

It is important to note that at Volition our technical artists do not design code structure for the programmers. This level of granularity is neither our job nor our area of expertise. In contrast, we work with them at a higher level to develop the best way to get requested features from Point A (content creation) to Point B (the game) and everything in between.

## CONTENT BUDGETS AND GUIDELINES

The technical artist is the driving force behind setting budgets and developing technical guidelines for creating art content. It's important for the technical artist to remember that while these requirements help to serve game performance, they must also be balanced such that they allow for a high degree of visual quality.

Art should be authored in a way that doesn't hinder performance of the game. Instead, it should be created to take advantage of the game engine and hardware strengths. To meet these goals, the technical artist works closely with the team's rendering programmers.

## TOOLS PROGRAMMER

A technical artist should be flexible enough that she or he can develop tools for the art pipeline without any assistance. Typically, programmers do not wish to write these tools and this is where a technical artist can fill that void.

The level at which tools are developed varies. Most technical artists, by today's standards, come from an artistic background and favor the use of dynamic scripting languages such as MaxScript or Mel.

As their interest and experience level grows, some technical artists grow into more lower-level languages such as C# and C++. This gives them greater ability to write compiled plug-ins, such as exporters, and tools outside the content creation packages.

## SPECIALIZED DISCIPLINES

If there ever were a difficult position to fill, it's that of the technical artist. Technical artists were once content artists who taught themselves to write scripts out of necessity and had a natural interest in the technical side of the craft. At Volition, we break down these roles into core areas of game development. See Figure 1 for an example of the typical structure for our technical art teams.

**Technical art director.** The technical art director is at the same level as the art director in our typical team structure. This person is responsible for coordinating the technical art team, prioritizing features, identifying and assessing project risks, and scheduling and designing critical tools. In addition, the

technical art director also designs and implements game systems and pipelines, creates guidelines and budgets for art content creation, and makes sure that the game's rendering performance is running optimally, while working with the art director to maintain a high degree of visual quality.

**Generalists.** What we call generalists are typically the senior technical artists who can drive any system in the game. They have a wide range of experience to pull from and are typically the critical go-to guys.

**Character technical director.** The character technical director is responsible for setting up the character skeletons, rigging, identifying and assessing motion capture and animation needs, scheduling and coordinating the animators, and developing or designing tools and pipelines to support the game's character systems.

**Senior technical artist.** The senior technical artist is primarily responsible for the design and implementation of larger and more critical game systems and pipelines. She or he is also partly responsible for ensuring that content is being created in an optimal fashion for not only rendering performance, but high visual quality as well.

**Focused technical artist.** Focused technical artists are typically entry- to mid-level technical artists. They focus primarily on specific areas of the game, such as environment art or character art. These focused technical artists take the point position for their particular art department and get approvals through the generalist or technical art director types. They provide direct support and develop any needed tools and pipelines necessary for their respective department(s).

## TECHNICAL SUPPORT

Another major area of responsibility for technical artists is to provide technical support to the art team. This includes tasks such as diagnosing a problem an artist is dealing with in the content creation package (3ds Max, Maya) or other in-house proprietary tools.

Filling in the support needs of artists can be a time-consuming process. To speed up the process for technical artists, we require that all requests be sent through email in the form of a descriptive explanation of the error or problem, along with attached screenshots.

We do this for several reasons. At Volition, technical artists support a vast number of artists, including any who are outsourced. It's critical that the response be focused to reduce the amount of time spent on the request and get the artists back up and running so they can continue to work with as little interruption as possible.

## SCHEDULING

A lot of what a technical art team does (depending on how far the team is in the development process) affects scheduling, both their own and the project's. As individuals, technical artists frequently switch between developing tools and fighting fires, often at a moment's notice. And as multi-disciplined team

# THE CODE/ART DIVIDE

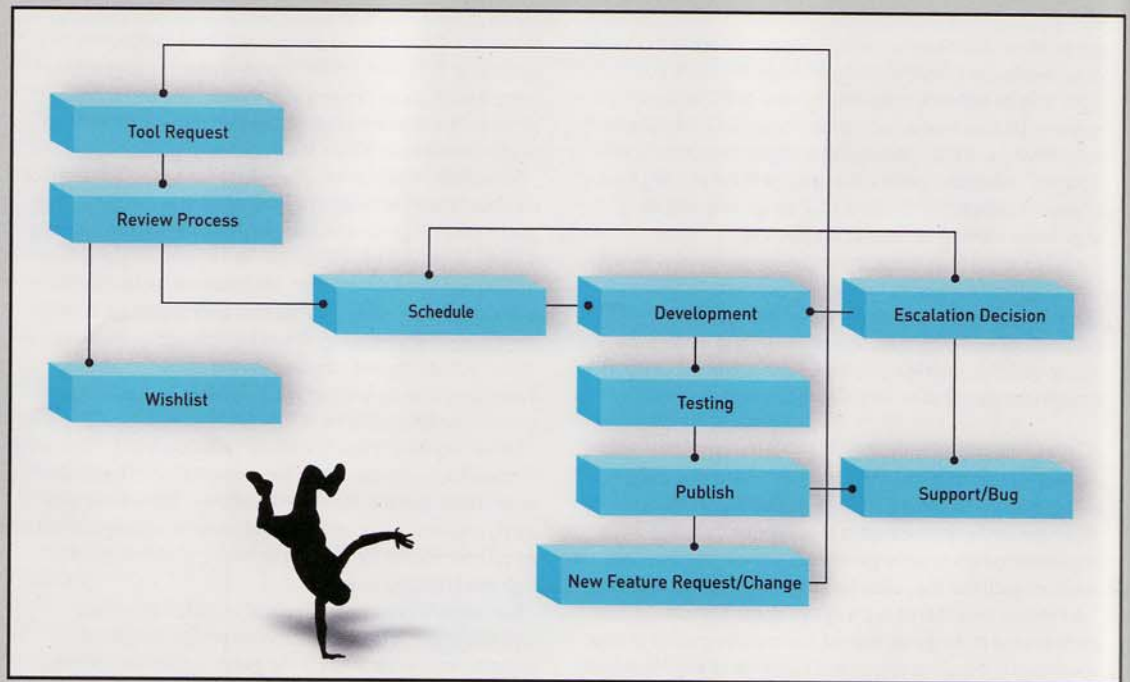


FIGURE 2 Volition's flowchart accounts for feature requests as well as a change management plan.

members, they are required to be in many other departmental meetings, some of which come up unexpectedly. If management accepts that the discipline is difficult to schedule and maintains an open mind, scheduling technical artists can be manageable.

Here are a few approaches we've found that work well in creating technical artists' schedules.

*Early identification of needs.* At Volition, we allow anyone in the studio to submit ideas for a feature or new tool to the project via email aliases. These items are reviewed by all project leads and dependencies are identified at this stage.

When evaluating a request, we use a five-point rating scale, with 1 signifying "must have." We make it a priority to fill all level 1 and 2 requests, while items with priority levels 3 through 5 are reviewed later to fill out empty spots in schedules.

*Due diligence.* Every tool or feature request goes through a three-phase process: investigation, implementation and documentation.

Investigation is used to identify risks of the request. Too often, risks are not accounted for in the schedule; by identifying them, you impress the importance of building solid tools.

Documentation is often glossed over. Good documentation ensures that anyone using the tool, no matter their technical ability, will use it properly.

*Schedule support time.* This is to accommodate the roller coaster frequency of support calls. From our experience, we have found support time typically ramps up quickly the closer

you get to the end of a milestone or deadline.

*Schedule buffer time.* Even with the scheduled support time, things inevitably crop up that can't be foreseen.

*Change management.* Implementing a solid change management plan for tools and feature requests is essential to keeping to the schedule. Too often, features are requested for existing tools that may seem minor to implement, but added all together they further complicate the already difficult problem of scheduling tech art. See Figure 2 for a flowchart that illustrates how we have implemented our feature request and change management plan.

## IMPLEMENTATION

How many technical artists should a company have? Over the past few years at Volition, we've found a need for three or four per project with a team size of roughly 80 to 90 people.

We structure the team such that there is the lead, who is the technical art director, then at least one senior technical artist, and a character technical director. The others are more focused technical artists who are assigned to specifically dial in on certain areas of the game.

While finding the right person to fill this role is difficult, it should not be overlooked in today's competitive and high cost environment. If your studio has no technical artists at all, or has some that aren't being used to their full potential, I encourage you to take another look. You'll be glad you did. ❖