Chapter 11

Task Tracking

Production Begins—Now What?

Congratulations! You have made it through preproduction, your project is approved and funded, now all you have to do is follow your plans and make your killer game! This is a short chapter on how to track the completion of tasks and how to get the most productivity out of your team.

Task Visibility

You cannot just print out copies of your Gantt chart then surf the web for a year while your people make the game. *This will not work*. Even if you made the most professional Gantt chart ever, printed out in color and spiral bound. Passing out these project binders to everyone is an excellent idea, but if that is all you do to make your developers aware of their tasks and their team's tasks, then you will fail to get anywhere near your team's full production potential. I am not saying people are inherently slothful, no, quite the opposite—almost everyone I have met in the industry prides himself on his ability to work hard under a crunch to produce a hit game. It is just that left to their own devices, your folks will probably work on what tasks are most interesting to them unless they are reminded of where they are on the schedule and where everyone else is on the schedule.

The key is to make the tasks visible. Team members need to know in detail what they should be doing, and they need to know how the work they are doing correlates with others on the team. They need to feel a part of the team and share a sense of urgency to get the job done. As tasks are completed it should be communicated as quickly as possible to the rest of the team to give them a sense of the pulse of the project. I have some specific techniques to share with you to achieve strong task visibility.

The Wall

I have an effective, low-tech way of getting task visibility out to the team members: I print out the Gantt chart and/or task lists and pin them up on a central wall in our workspace. Software solutions such as Microsoft Team Manager and intranets to publish your schedules and tasks are distinctly unsatisfactory for two reasons: One, your developers need to remember to even open up the document or visit the site, and two, monitors are too small to show a whole Gantt chart, denying your team the appreciation of the project progress as a whole.

It is easy to print out your schedule and pin it up. I recommend just displaying task name and ID, start time, end time, who is assigned to it, and any predecessor tasks on the left-hand side and the Gantt chart on the right-hand side. You should use the widest time setting you have wall space for; when a schedule is scrunched up into just displaying quarters or months on the Gantt window, you are not getting any real-time information.

Now I make a requirement to my developers that they come out to the Gantt chart and mark the tasks off themselves. I do not mark them off even if I know they have been completed. This is to get the developers to come out and find their place in the schedule, mark off with a bit of pride what they have finished, and then look ahead to see what is coming up. Developers will almost always take the time to then look over the whole schedule to gauge how are they doing compared to other team members.

When I first started using this method of task tracking it was considered somewhat controversial. Some people asked me privately if this was a good idea. If someone were not accomplishing his tasks on time, would it not be demoralizing for him if this were made public knowledge? Would not that developer feel more comfortable staying in his office and explaining privately why he is behind in the schedule? Bah! My first assumption is that everyone on my team is a professional, and even on an off day all would want to be treated as professionals. Why would protecting their comfort be of higher importance than getting our tasks done in a timely manner? If people are tasking late, they must have a reason. Was it illness? Jury duty? Task underestimation? Were they distracted helping another team member on another problem? All of these are legitimate reasons for being late and certainly nothing to cause embarrassment or discomfort. On the other hand, if they are late because they were just goofing off, then I feel comfortable making them squirm in front of their other team members and letting them know they have let the team down. Knowing that the whole team is aware of what they are and are not getting done goes a long way to inhibit goofing off.

A healthy bit of competition develops with a good wall. Assuming your schedule was a sane schedule and manifestly fair in the time allocated to the tasks to be completed, your team will be in a high morale state to begin with. I use brightly colored highlighting markers to mark off the tasks. Your developers will come out at the end of the day to mark off what they got done then look ahead for something simple to do before they go home—bam! Another task is taken care of! This competition effect will give extra momentum to your whole project. It will give your developers a meta-game to push themselves, and they will enjoy it.

Another benefit of the wall is that it makes a great piece of visual feedback to the executive management team. They look over the wall and see all the marked-off tasks spanning 25 square feet of wall space and nod to themselves and move along. Do not underestimate the importance of reassuring your management that you are respecting their time and money and are making measurable, steady progress. If you are working in a large studio or in a publishing house, the other teams will see what you are doing and think you are obviously trying to get attention. So what—you are trying to grab management's attention. There is no glory in obscurity.

Encourage your team members to go ahead and write any unanticipated tasks they had to complete onto the wall's task lists. This will help team members who might be falling behind in tracking due to being sidetracked by tasks that were not originally on the schedule. While it may seem crude to scrawl new tasks on the list, it is legitimate. You are after the maximum visibility for all tasks, not just the ones you were smart enough to think of earlier.

When the time comes to update the schedule, the wall charts with the new tasks written on it and the completed tasks marked off will come in handy. Just tear it off the wall and bring it to your workstation where you have Microsoft Project.

Journals

I have a background in engineering, and while in school we were introduced to the value of a journal to record actions, observations, and data from the lab. The idea is that no effort you make should be unworthy of record. While I admit that when we make a game we are not building a skyscraper or a transorbital spaceship, we are still creating something important and we should take every care we can on the execution of our game projects.

The Cult of the Yellow Notebook

For the last seven years I have been using yellow notebooks that are about 5" by 8" inches and feature lined paper on one side and quad-ruled paper on the reverse. This format allows me to track micro-tasks and thoughts on the lined side, and use the graph paper for game designs, user interface layouts, and technical designs. I have this notebook open as I work, taking notes whether I am working at my workstation in Photoshop, MS VC++, Project, Visio, Excel, or simply Word. I also take my journal with me to every meeting to record what I need to do and what I need to follow up with. On a shelf in my office are the 40 or so notebooks I have filled so far in my career. These yellow notebooks are a staple that we purchase for all of the employees at Taldren, and we have an ample stock for when people fill theirs up.

I am passionate about these notebooks because I have seen countless small tasks fall through the cracks in our overburdened minds—such a waste that the simple act of note taking can fix! About once every two to four weeks I go back through my pages to search for tasks I might have failed to address, and I pull them forward into a new checklist.

Walk Around

There is no older and simpler method of task tracking than simply walking around and seeing how people are doing. I try to carve out an hour or two every day to walk around and meet with the individual team members to see how things are going. At this pace I would visit everyone in the company two to four times a month. This lets people know their work is important, and the human connection really shows you care about getting a great game done. When the project hits a tough spot you will find that you want to stay in your office and focus on the burning fires. But it is when the times are smoky that you should make the extra special effort of visiting with your team members. Also be aware that no matter how much you like everyone on your team, there will naturally be some personalities that you enjoy spending more time with than others. Some people might feel slighted so be sure to visit all of your team members, not just the ones you like to talk to.

Often it is by walking around that you discover that tasks you thought were the clear responsibility of one developer have been conveniently relegated to the no-man's land between two developers and have dropped to the floor. This is a great time to clear up such misunderstandings and get these tasks properly assigned. If you ask the right questions and remain approachable, these walkabouts will also turn up the deeper concerns your team members might have felt too uncomfortable bringing up in some other forum or method. Keep your ears and eyes open and *talk* to your team members.

Milestone Orientation Meetings

Another useful technique I have found is to kick off each milestone iteration with a milestone meeting to review what everyone is tasked with and what the associated expectations are for their work. I did not start this ceremony until just this year; however, each time I run the meeting I am amazed at how many misunderstandings we are actually carrying around, and this is on a project that has received our most detailed preproduction to date!

At these meetings I simply keep everyone in the room as I go through the features and tasks one by one and get a verbal discourse back from the responsible developer to be sure they understand what they need to do and to give them a chance to request clarification. They will also get full visibility for what they need to accomplish in front of the whole team; this goes a long way to fight the impression that so and so does not have much work to do.

Praise People Publicly

I also take the time to praise individual team members at each of these meetings—not necessarily everyone—but I do try to keep a running tab of who is due for some recognition. While no one I know would admit it, I think receiving praise and recognition from your team and supervisor is a great morale boost, and the lack of praise and recognition can be a significant drain on morale.

There are good books devoted to how best to reward your employees with all sorts of clever ideas from silver nameplates to holiday turkeys, but I think the best is a public thank you.

Maintain the Gantt Chart

By far the least fun part of project management is updating the Gantt chart. As you sit in front of Microsoft Project, none of the tasks will seem to have been completed on the days you planned. And so if you simply check off tasks as they are completed, you will be left with a schedule that is full of hard-to-move completed tasks that indicate they were completed on the wrong day. These blocked off dates will not be used in subsequent leveling operations, and soon your schedule will look like a mess.

No matter how tedious it is, do not put off maintaining your schedule for longer than a month. I have slacked myself and have regretted it every time. It just takes too much time to repair a badly out-of-date schedule. When a schedule is really in bad shape I sometimes just start a fresh Project file.

The latest version of Microsoft Project does have one simple new feature: It lets you move completed tasks back and forth in time! A minor miracle, I tell you. In the older versions you would have to unmark a completed task, move it to the time it was completed (or at least out of the way of the current task leveling concern), and then remark it as complete. This only made a tedious job twice as hard as it needed to be.

Take the time to enter in completely new tasks that your developers have taken the time to write on the wall. Also take a close look at any open tasks that are refusing to complete despite one of your developers working hard on the task. My bet is that if you look under the hood of that task, you will discover it is composed of multiple tasks, some of which have been completed. Take the time to break up this task into its component parts and give your developer credit for what has been accomplished. Quite often your developers will tell you that this or that feature is 90 percent done and that they clearly had to move on to more pressing tasks for fear of causing stalls in the project. Their judgment is almost always correct in that there was little profit in having them polish up some feature to true shipping quality when there are others waiting for them to finish something else. This is the same as a task that is really composed of subtasks. In this case the subtask is that final 10 percent of polish on the radar, which is unimportant to solve now. Take that 10 percent polish task and enter it into the schedule; just put it further down in time to when you really will take care of the polish task.

For larger projects I strongly suggest you delegate to your section leads the input and maintenance of their part of the schedule. This will help them grow a valuable task, and it will help you keep your job sane. To facilitate this I favor using a tree of inserted Microsoft Project files so that each developer can work on his section of the schedule. I discuss this in detail in Chapter 20.

Update the Risks Chart

Rounding out the task tracking set of duties is to update the risks chart: Take the time to review your Gantt chart; is it indicating a new problem down the road? Are the artificial intelligence tasks tracking? How is the mission editor? How are the art assets coming along? How is the testing of the multiplayer code coming along? Ask yourself these types of questions as you review the Gantt chart to see if a new risk has developed or perhaps an older risk has risen in priority. Also take a look at the old risks; have some of then lessened in importance or have they faded away altogether? Some new risks may be introduced from your walks around the team or from a daily journal type mechanism or simple email from your team members. Also take the time to review what you are expecting from your third-party vendors. Are they on time? The true impact of a risks document only comes into play when it is maintained like the project schedule. Be sure to visit with your executive management and apprise them of the latest risks. Post these risks in a public place so that all of your team can review them and have an opportunity to respond to them. After all, taking the time to discover your risks is a good idea, but sharing your risks with the rest of the team and management is key to getting focus on the problems. Of course occasionally you may develop a risk that is personal in nature and is not fit for wide dissemination. Use your common sense and discretion when choosing what to post on a wall.