GAMASUTRA
The Art & Business of Making Games

PRINT

Postmortem: Disney Online's Toontown

By Mike Goslin

Disney's *Toontown Online* is a massively multiplayer online role-playing game (MMORPG) created for kids of all ages. *Toontown* is under siege from an evil band of business robots called the Cogs. An unsuspecting Scrooge McDuck accidentally unleashed the evil robots and they are attempting to turn the colorful world of *Toontown* into a black and white metropolis of skyscrapers and businesses. The player's job, as a Toon, is to join forces with other Toons and use gags such as cream pies and squirting flowers to defeat the Cogs and rescue *Toontown*.

Toontown is published by Disney Online and was developed by the VR Studio, which is a group of animators, modeler-painters, and programmers that were originally brought together to develop virtual reality attractions for the Disney theme parks. Our first project was *Aladdin's Magic Carpet VR Adventure*, which was deployed at Epcot in 1994, and then two years later at Disneyland. Between 1996 and 2001 the VR Studio developed three attractions for DisneyQuest, which was a business created to build themed arcades filled with virtual reality attractions and modified arcade games.

While we were working on *Pirates of the Caribbean: Battle for Buccaneer Gold*, our third attraction for DisneyQuest, the first 3D massively multiplayer online games started to emerge, and this really caught our attention. We realized that massively multiplayer online games represented a tremendous opportunity for a company like Disney, and believed that our experience in creating interactive spaces for theme parks could be used to create compelling entertainment for the home. We also believed that the tools and techniques that we had created for developing theme park attractions could also be used to build games for PCs.

Our objective was to create an MMORPG for the Disney audience of kids and families. We decided to use our own software engine and development environment, called Panda3D, and began work designing a server infrastructure that could support tens of thousands of simultaneous players.

Although the term "fight" is used to describe the meetings of the Toons and Cogs, the violence is quite tame--this pie-throwing mob is about as rough as it gets.

What Went Right

1. We got kids, but we also picked up adults along the way. We also hit both males and females. This is always a goal when building theme park attractions, because usually you end up with family groups riding together. We tried to take the same approach when designing *Toontown Online*.

One way to appeal on multiple levels is to use humor. Generally speaking, visual or slapstick comedy works well with kids, while verbal humor and puns tend to appeal more to adults. *Toontown* has plenty of each, because it features all the classic cartoon gags such as hitting bad-guys in the face with pies, but also includes references to office humor, such as the enemy attack that literally wraps a Toon up in "red tape". Interestingly, many adults report that they really enjoy the silliness of role-playing as a Toon, and kids seem to appreciate being "in" on some of the grown-up humor.

Another way to have broader appeal is to make the game easy to learn but difficult to master. In order to achieve this, we spent a great deal of time on the first 30 minutes of the game experience and on refining the in-game tutorial. The ease of the initial experience is critical for attracting and keeping both younger children and non-gamer adults. The game must become challenging relatively quickly in order to engage older children and adults who are gamers, however.

Battling the Cogs, for example, is very easy to learn but difficult to master. The combat system is turn-based, and the player is initially presented with only two possible attacks: throwing a cupcake at the enemy or spraying them with a squirting flower, both of which are similar in terms of the strategy of their use,

In *Toontown*, the fight rules are a bit different than in other games. In this universe, any player that sees a battle through to the end gets credit for it--no "kill stealing" here.

accuracy, and resulting damage. Battles become much more complicated as players advance in the game because they gain access to other gags and learn that much of the strategy involves coordinating attacks with other players. For example, damage bonuses are awarded when multiple Toons hit the same Cog with pies in the same round. Another example is that dropping an anvil on a Cog's head is more likely to hit when the Cog is stunned from being hit by a pie in that same round. To win some of the more challenging confrontations in the game, a player will need to communicate with team-mates, know their strengths and weaknesses, watch what they do, and make choices accordingly.

Another result worth mentioning is that not only does the game appeal to kids and adults alike, but we also ended up with an audience that is at least 50 percent female. We believe this is fairly unique for an MMORPG. We think *Toontown* appeals to females because of the cooperative nature of the game play, the social interactions that come from being online, the turn-based combat system, and the colorful palette and Toon themes of the game setting.

2. The game is safe. A vexing problem for us was how to build an MMORPG that was safe for children, without giving up the essential communication features that are required to support a community. We focused much of our energy on safe communication.

In *Toontown*, there are two ways to communicate with other players. "Speedchat" is a hierarchical, menu-based chat system that allows a player to say everything they need to say to be able to play the game, but since there is a finite set of possible sentences, it is impossible to communicate any personal information. Alternately, the "Secret Friends" system allows players to exchange a secret code outside of the game that will allow two friends to chat with each other inside the game.

<

Another safety issue that we worked hard to solve was making it difficult for one player to "grief", or ruin, the experience for another player. The worst thing a Toon can say to another Toon using Speedchat is "You stink!" and because Toons can teleport freely between different "districts", or copies of the world, it is easy to avoid another player who is trying to bother you.

A common practice in other online games, often referred to as "kill-stealing", is for a more powerful player to be able to join a battle that is underway and finish off the enemy, thereby earning experience and even treasure at the expense of the players who started the battle. In *Toontown*, anyone who participates in a battle that is still there when it concludes is awarded experience and quest items independently, so weaker Toons are usually happy to have a more powerful Toon join their battle because everyone benefits.

Speedchat provides a quick way for Toons to talk to each other--and one that prevents kids from sharing personal information of any kind.

3. Cooperation made easy. Cooperative game play is essential to maintaining a thriving in-game community. *Toontown* facilitates cooperation by making it very easy to form groups. To join a battle on the street, simply walk up and "bump" into it. To join a group playing mini-games, just join them on the mini-game trolley. To take on a building, join a group by walking into the elevator with them. In our recently released Cog Headquarters building, up to eight Toons can join together to take on one of the boss Cogs.

Additionally, a Toon may always teleport to another Toon on his or her friends list. This creates a social space that emphasizes friendship and teamwork rather than walking around looking for other people. The ability to teleport to a friend extends to all copies of the world, so there is no concept of being isolated on a particular game server. In the cartoon world of *Toontown*, it feels natural for a Toon to pull a portable hole out of its pocket, toss it on the ground, and jump into it as a way of getting around quickly.

4. Online distribution. *Toontown* is downloadable. The entire client is currently less than 30 MB compressed, and uses a staged download so you can start playing the game after the first couple of megabytes reach your PC. We originally designed the download this way so that kids would not have to wait long before they could start playing. It is possible to download and play the game on a narrowband Internet connection as slow as 28Kb/s.

Making the game easy to download and install also allowed us to pursue a viral marketing strategy. Even a skeptical player can check out our game without making a major time or financial commitment. This feature is particularly valuable to us because Disney is new to the MMORPG genre and we would like as many people as possible to be able to try the game easily

5. Inexpensive to operate. When we began development, we were unsure how much a kid would be willing to pay for an MMORPG, so we designed *Toontown* to be as low cost to operate as possible. No in-game support is required, which eliminates a significant component of traditional customer service costs for this genre of game. In addition, we consume a fraction of the bandwidth of other MMORPGs. Both of these costs scale with the number of players, so they are important ones to minimize.

We also spent a considerable amount of effort on the in-game tutorial system. Since many of our players are new to the MMORPG genre, and possibly even new to 3D games, there is a lot to learn just to get started. We have had people contact us whose only other game experience was playing "Minesweeper". A well-designed tutorial really helps our players begin enjoying the game sooner, and results in fewer customer service calls about how to play the game.

We discovered that another large expense for us was customer service contacts related to graphics driver issues. Many of our customers are unable to identify a driver problem and simply assume that something is wrong with the game. This confusion is magnified in the common case where someone has a brand new PC, since consumer PCs often ship with beta drivers for the graphics hardware that need to be updated by the time the PC is powered up on someone's desktop. We eventually developed a comprehensive system to proactively detect driver problems and to provide the latest information and even actual links to the approved drivers. Providing this service is logistically difficult because of the frequency with which various manufacturers update their drivers, but to us the effort has more than paid for itself.

What Went Wrong

1. Making an MMORPG is hard! Anyone who has worked on one of these will tell you this. We had to develop several core competencies from scratch, rewrite our development software, work for years, and are even still revising our operations plan on a regular basis. The number of skill sets necessary to produce and run an MMORPG surpasses anything we had experienced before.

A good illustration of the difficulty in programming a massively multiplayer game is the *Toontown* battle system. We designed a turn-based combat system where players could choose their actions and then a round of the battle would play out as a short scripted sequence where the Toons do their attacks and the Cogs retaliate. These battles take place on the streets of *Toontown*, which are a public place. It was important for us to allow passersby to be able to observe battles and even discuss what was happening in them as a way for new players to learn how to play this part of the game. This required us to design the battle system so that players coming around a corner and observing a battle for the first time would see essentially the same thing that players in the battle were seeing at roughly the same time. In other words, the sequences generated for each round of the battle had to be distributed to everyone in sight of the battle and synchronized. In addition,

these sequences needed to support arbitrary starting points in case someone arrived late. Our solution was to represent every animation, sound, transformation, and effect for a given sequence as a hierarchical set of "tracks" that had defined state throughout the duration of the movie. We could then distribute this set of tracks and every player in view of the battle could synchronize by setting the network time value. As a result, a player that rounds a corner and sees a battle for the first time can actually see a pie in mid-throw.

2. We had to change scripting languages. Changing scripting languages is the last thing you want to contemplate more than six months into a project like this, but by that point we had begun to realize our scripting language was not going to get the job done. The language we originally chose was terrific for rapid prototyping and we were able to make a lot of progress. As the code base grew, however, we began to experience both performance problems and code management problems. We finally made the painful, but necessary, decision to switch, and ported the game to Python. Things have been great ever since.

This was the starting point for the feel of *Toontown*--the houses were modeled after the feel of this concept art.

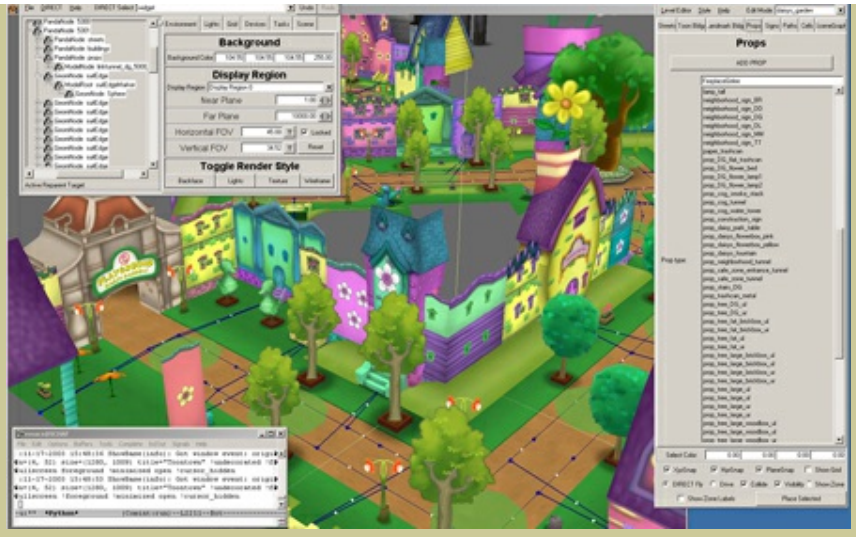
It is worth mentioning that our development environment consists of C++ libraries that are loaded by an interpreted scripting language. This allows us to have the performance of compiled code where it matters, but also gives us the flexibility to make changes to the game without having to recompile. We like being able to iterate quickly and even occasionally rewrite methods while the game is running.

Python is working out for us because it is relatively efficient, lightweight, and scales well. It has a syntax that is easy to learn and debug, and the documentation is excellent.

3. Be careful when you let players create their own names. We assumed correctly that we would have to filter name choices to eliminate offensive words or phrases. We underestimated the ingenuity with which some players would approach the challenge of trying to defeat our filters, however. After refining our filter and accumulating an impressively long list of bad words, euphemisms, and other exotic uses of language, we finally resigned ourselves to the fact that we would either need to revoke the privilege of naming characters, or require a human reader to check every submission. Currently when you create your own name in *Toontown*, you must wait for approval from the "Toon Council" before you can use it.

Having a human look at every name submission can be fairly expensive, however, so we developed a system intended to make name-picking fun. Our name generator can produce a wide range of Toon themed names such as "Professor Skimpy Googlemuddle" or "Dippy Funnycorn". The name choices available to new players make it easier to begin role-playing as a Toon because they convey the humor and sensibilities of the Toon world. Also, for new players, and especially younger players, it is often a relief to not have to come up with an appropriate name from scratch.

4. The Internet can be a challenging platform. The Internet is very good at delivering web content. Web browsers are very forgiving when packets fail to arrive quickly (or at all). Unfortunately, an MMORPG requires a persistent, relatively low-latency connection that is reliable and sustainable for up to hours at a time. This is extremely difficult to deliver using today's Internet. We have suffered from a variety of packet loss or corruption problems that happen between our servers and client applications. These types of problems are extremely difficult to reproduce and isolate. Even when you manage to identify them, they are often impossible to fix because they don't happen in your code. For example, we eventually decided to use SSL to send game data back and forth from the clients, not only for the obvious security benefits but to prevent occasional misinterpretation of our data by various pieces of networking hardware between our servers and our players on the Internet.



The levels in *ToonTown* resemble a Hollywood backlot set--only the necessary details of the streets and buildings are modeled.

Latency is another major obstacle that we had to contend with when we designed *Toontown*. Since latency is extremely variable by geographical distance, Internet service provider, and congestion at any point in time, it is difficult to take anything for granted. We chose turn-based combat rather than real-time combat in order to minimize the effect of latency, for example, and generally designed all of our game systems to work smoothly with up to two seconds of lag. As part of our testing process, we exercised most game systems under simulated latencies of up to twelve seconds to make sure things would not degenerate completely.

5. Online distribution. Online distribution turns out to be both a blessing and a curse. As mentioned in the "What Went Right" section, online distribution helped us to market *Toontown* virally. Unfortunately, it is difficult to market an online-only game using means other than viral ones. As it turns out, many people today are still generally uncomfortable with the idea of buying a game that doesn't come in a box that they can hold. They also worry about what happens to their purchase if their hard-drive crashes, for example. Other people are unwilling to use their credit cards online at all. Finally, it is awkward to give an online-only subscription as a gift.

As a result, we chose to distribute a CD for *Toontown* that can be purchased at retail. This gives us a more traditional game product around which to rally our marketing campaign, and provides a faster way for narrowband users to load the game on their computer.

Conclusion

It is difficult to build and operate an MMORPG, and probably even harder to do this for a broad audience. You are confronted with the increase in design complexity to handle massively multiplayer but at the same time must keep things simple to learn for the mass audience. You need to keep things safe without smothering your community and taking away fun and freedom. You need to find new ways to reach players and get them to experience and understand this genre of game, on a delivery platform that can be tricky and unreliable.

Despite the numerous and often painful lessons we had to learn about building and operating an MMORPG, Disney's *Toontown Online* was finally launched online in June of 2003. In case you are wondering if we have been deterred by our experience, you should know that we have recently begun development on another MMORPG.

Game Data

Toontown

http://www.toontown.com

Publisher: Disney Online
Developer: VR Studio
Number of full-time developers: n/a
Number of part-time developers: n/a
Contractors: Level editor, artists.
Length of development: 2 1/2 years
Release Date: Online launch June 2003, retail launch September 2003
Target Platform: Windows PC.
Development Hardware: Typical development platform was a high-end Windows or Linux PC with 1.5GHz CPU, 500MB RAM, and 32MB graphics card.
Development Software: Microsoft Visual C++ v.6-7, Python, Panda3D (internal rendering engine), DIRECT (internal level design and interface tool), Softimage, Maya, Multigen, Photoshop, DeepPaint, Miles, CVS.
Notable Technologies: Staged online distribution system, safe but effective communication (SpeedChat and Secret Friends), scalable servers.

Key Staff:

Mike Goslin, Director
Daniel Aasheim, Production
Felipe Lara-Garcia, Art
Mike Goslin, Director
Roger Hughston, Servers
Mark Mine, R&D
Joe Shochet, Game Design
Bruce Woodside, Animation

Return to the full version of this article