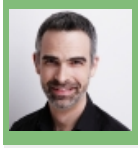
David Lightbown

My Account
My Profile
Blog Now
[LOGOUT]

PROGRAMMING
ART
AUDIO
DESIGN
PRODUCTION
BIZ/MARKETING

Latest Jobs

View All RSS

February 23, 2017

- Qualcomm
  Graphics Software Engineer
- Pixar Animation Studios
  Animation Tools Software Engineer
- Insomniac Games
  Gameplay Programmer
- Soulbound Studios
  Sr. Tools Programmer
- thatgamecompany
  Graphics Engineer
- thatgamecompany
  Senior Gameplay Engineer

Latest Blogs

View All Post RSS

February 23, 2017

- Maximizing the Impact of Procedural Personalities [2]
- How to fill your days as full time indie [4]
- 7 Lessons I Learned Making VR Games at Experiment 7
- On Cutscenes and Viewpoint Changes [1]
- Game design patterns for building friendships

Blogs

Classic Tools Retrospective: John Romero talks about creating TEd, the tile editor that shipped over 30 games

by David Lightbown on 02/23/17 05:31:00 am Featured Post

Post A Comment [Edit Post]

The following blog post, unless otherwise noted, was written by a member of Gamasutra's community.
The thoughts and opinions expressed are those of the writer and not Gamasutra or its parent company.

Introduction

If you were to ask a room of game developers about their first experience making games, many would tell you that it involved using a tool that came with the game that they were already playing. Most likely, they found it tucked away in the same folder as the executable, sporting an enticing name like "editor.exe".

Many years before Unity 3D, companies such as id Software, Epic, 3D Realms, Blizzard, and BioWare would release tools along with their games, in the hopes that enabling people to create new content would grow the community and extend the lifespan of the game. For many people, these tools were not only exciting to use, but they were also their gateway into the game industry. The content that they created would later become their resumes.

In recent years, retrospectives of classic games have been well received at GDC, but there have been very few stories about classic game tools. This series of articles will attempt to fill that gap, by interviewing key people who were instrumental in the development of those classic game tools.

For the first article of this series, I have the great pleasure of speaking with **John Romero** about **TEd**, the tile editor that he created at id Software, which went on to ship 33 games.

First encounter with a level editor

DL: Hi, John. Thanks for taking the time to talk to me!

JR: No problem!

DL: Before we talk about TEd, I want to take a step back: In the early 80s, there were very few level editors that came with the game. In your interview for the book Honoring the Code, you spoke about how the Terrain Generator in the game Pegasus][was one of the first level editors that you ever played with. What can you tell me about that experience?

JR: Well, the first time I saw Pegasus][was at Sierra College, in the Apple room. It was one of the very first things I saw on an Apple. While playing it, I saw that there was this option to do terrain generation. I thought that it was pretty cool that it has this tool built in to generate gameplay.

Press Releases

February 23, 2017

Games Press

- GUILTY GEAR Xrd REV 2 is coming to Europe in 2017!
- Popular Star Wars™ games now playable on...
- Albion Online: Official Launch date announced
- New MMORPG God Wars Making Closed Beta Debut
- Do you have what it takes to be the next oil baron...

View All **RSS**

About

- **Editor-In-Chief:** Kris Graft
- **Senior Contributing Editor:** Brandon Sheffield
- **News Editors:** Alex Wawro
- **Advertising/Recruitment/Education:** Courtney Blair

Contact Gamasutra

Report a Problem

Submit News

Comment Guidelines

Blogging Guidelines

How We Work

Advertise with Gamasutra

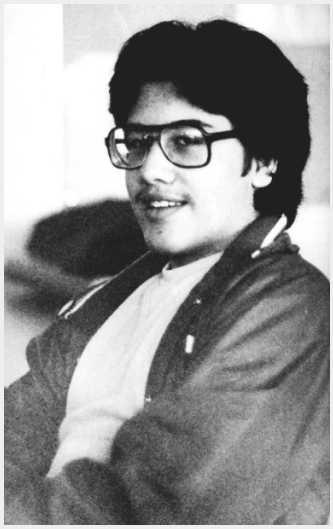
Gama Network

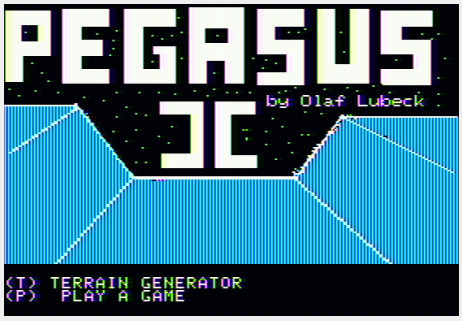
If you enjoy reading this site, you might also want to check out these UBM Tech sites:

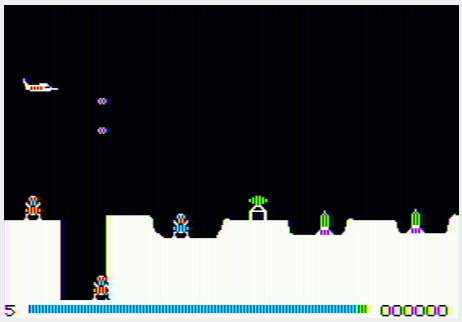
Game Career Guide

Indie Games

It's funny to think now that this is one of the earliest games that let you generate mods. This was the first time I saw anything like this. I thought it was really cool. It's funny, I never thought "why don't other games have that?"... I just never expected it!



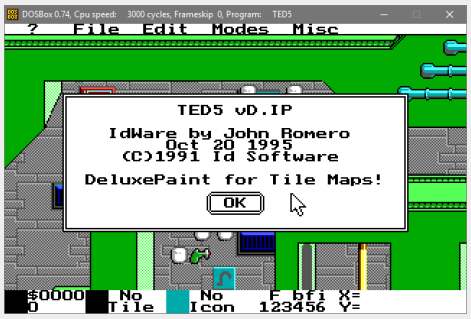


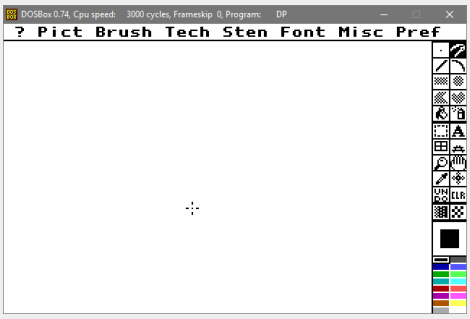


TEd, the Tile Editor

DL: OK, so, let's move on to talking about TEd.

[While we are talking, I launch two instances of DOSBox so we can look at TEd and Deluxe Paint, side-by-side]





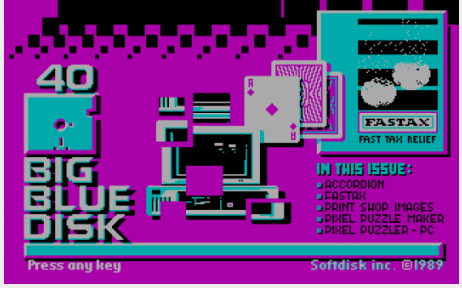
The first thing I want to talk about is the TEd splash screen. It says Deluxe Paint for Tile Maps. So, anyone who has read a bit about the history of id Software knows that Adrian Carmack used Deluxe Paint...

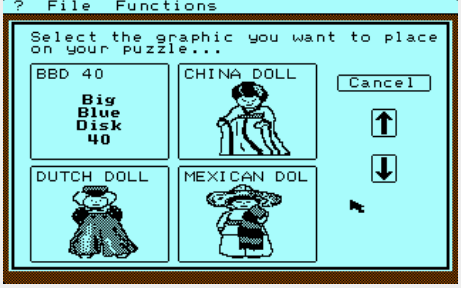
JR: The whole industry did.

DL: Yeah! Well, one of the first things that struck me was that the menu at the top of Deluxe Paint and TEd are very similar, in terms of look and feel. Were you trying to make something that was familiar to users of Deluxe Paint, or were you re-using something that was coming from Deluxe Paint?

JR: Before id Software, I had written some tools at SoftDisk and I was writing them on PC. I wrote a program called Pixel Puzzle Maker which was a tool used to create puzzles for a Softdisk game called Pixel Puzzler. So I wrote a pull-down menu system for Pixel Puzzle maker and then I did some modifications to it. I think I created some more tools for PC at Softdisk, then when I was doing the id stuff I think I created another one, because it was pretty easy to make this pull-down menu system.

It wasn't based on Deluxe Paint, because I never used Deluxe Paint, but I just thought having a pull-down menu it was much easier compared to dealing with a whole bunch of hotkeys... even though there are a lot of hotkeys!





DL: Speaking of hotkeys, there are other similarities that I noticed. These days, we're very familiar with the Control-Z hotkey being for undo, but back then - at least in the case of Deluxe Paint and TEd - it's just the "u" key on the keyboard.

JR: [Laughs] Wow. Is that what I did?

DL: Yes! Here, if I fill in a few squares and press "u", it goes back...

[I draw some tiles, and then press "u" to undo the last action]

JR: [Laughs] Nice.

Oh, by the way, you're using DOSBox, and when you scroll around the screen, that is super slow compared to how fast it actually was on a 386 back in the day. It was lightning fast.

DL: Oh yeah? No kidding?

JR: DOSBox is slowing it way down for some reason. EGA Scrolling using Latch mode was so fast back then, you had to tap the keys because it was just flying by. So, you're getting to see a weird super chuggy slow version.

"EGA Scrolling using Latch mode was so fast back then, you had to tap the keys quickly because it was just flying by."

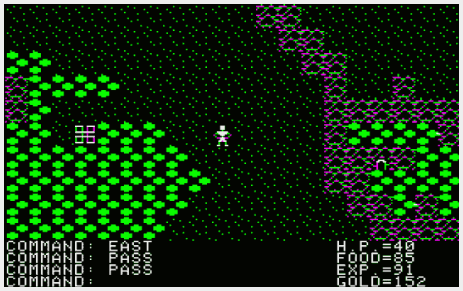
[Author's note: Here is John as "user42604" on StackExchange explaining EGA Scrolling using Latch mode: http://gamedev.stackexchange.com/questions/6488/how-did-they-make-the-screen-move-in-dangerous-dave]

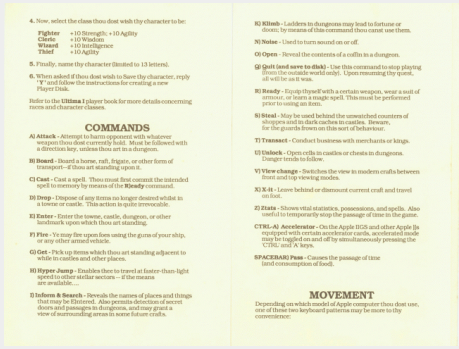
DL: OK, so, we were saying that TEd has a lot of hotkeys.
At the time that TEd was being used, you often had to read the help file to figure out how to use tools like these.

JR: Oh yeah, definitely, and I think some people were still at the tail-end of WordStar, which was like a nightmare of hotkeys.

DL: [Laughs]

JR: Back then, we were playing Ultima, and in Ultima, every letter on the keyboard was used, so it was not difficult to remember what things did when you were used to that, back then.



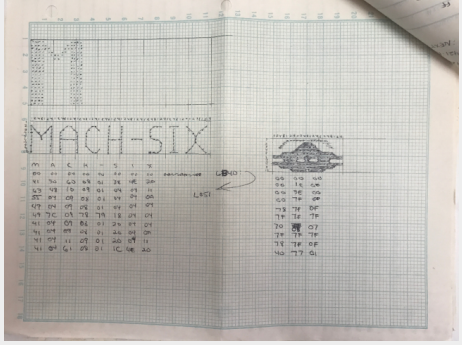

COMMANDS

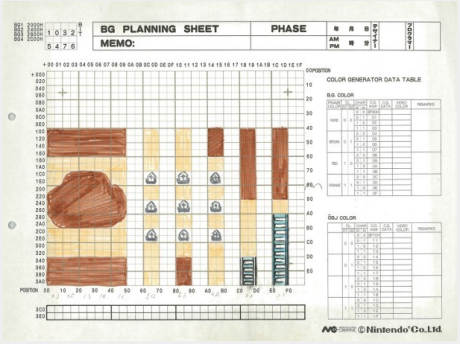
MOVEMENT

[Author's Note: I dug up the manual for Ultima 1, and John is absolutely right... almost every key on the keyboard was mapped to an action!]

DL: For sure! So, when it comes to designing a level at that time in the games industry, you might use a piece of graph paper to decide where the tiles would go, and then you would type them in one at a time.

JR: Yeah.



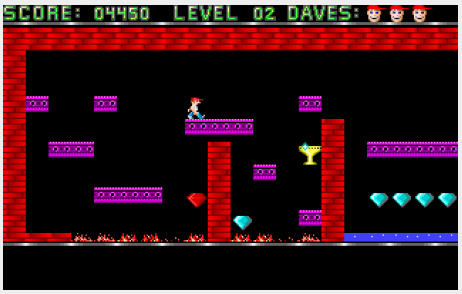


[Author's Note: After our interview, John sent me this graph paper drawing for a game he created called Mach Six. Also, click here to see an article on Gamasutra about how Nintendo used graph paper to design levels for the original Zelda on NES]

DL: So, at which point was it decided at id Software that it was necessary to create TEd?

JR: When I created Dangerous Dave in 1988, I needed to create levels for it, and I decided "well, why don't I just use the game itself, and just let my own game create levels, and save them out". So, that's what I did with Dangerous Dave.

So, when John Carmack saw that - because he was an Apple II guy - he was really impressed, because he was used to just using text to represent graphics. When we were first starting to make stuff together, I made TEd 1.0, and it evolved over the next six months to be TEd 5.0, which was used for 33 shipped games.

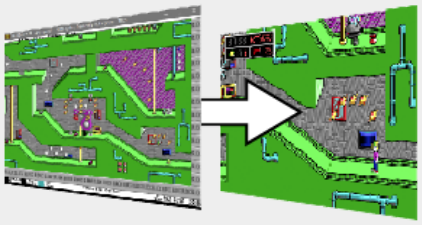


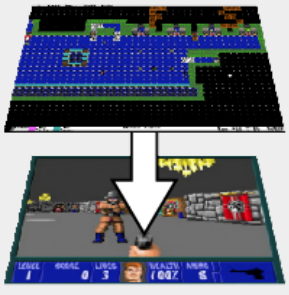
DL: Wow... now, when you say 33 shipped games, that's not only stuff that was within id - like the Commander Keen and Wolfenstein series - but also outside of id as well, like Rise of the Triad?

JR: Yeah! Also, Corridor 7, and stuff at Gamer's Edge, like Shadow Knights, Slordax, Dangerous Dave and the Haunted Mansion, Rescue Rover 1 and 2, BioMenace... a bunch of games!

DL: So, is it also true that TEd was not only used -- from a side perspective -- to create levels for two dimensional side scrolling games, but it was also used -- from a top perspective -- to create levels for three dimensional first person shooter games?

JR: Yeah. TEd 5.0 is the same tool that created all those games.



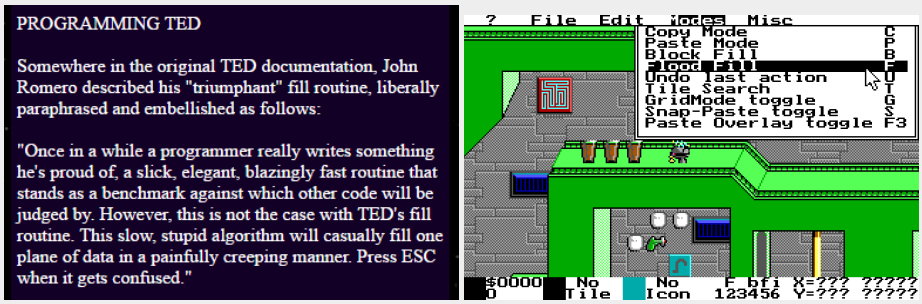


DL: OK, on that topic, I was wondering: At a certain point at id Software, you guys went from making two dimensional games to three dimensional games. Did you ever debate the idea of creating a new editor versus re-using TEd?

JR: Yeah, I think we talked pretty quickly about it. We were saying "we're going to have a 2D matrix to represent the level. Cool, we'll just use TEd for it." Yup, that was it! [Laughs] It was very simple.

DL: So, there's something else I wanted to ask you: In the Rise of the Triad help file – written by Tom Hall – he quotes you as saying this: "Once in a while a programmer really writes something he's proud of, a slick, elegant, blazingly fast routine that stands as a benchmark against which other code will be judged by. However, this is not the case with TEd's fill routine. This slow, stupid algorithm will casually fill one plane of data in a painfully creeping manner. Press ESC when it gets confused."

JR: [Laughs]



DL: [Laughs] So, why did you write that? Is there a story behind the fill routine?

JR: Well, the fill routine had some bugs in it. I didn't spend a lot of time writing it, and it *mostly* worked, but sometimes it wouldn't fully flood fill, so you had to keep on clicking to fill all the areas. It was easy to use.

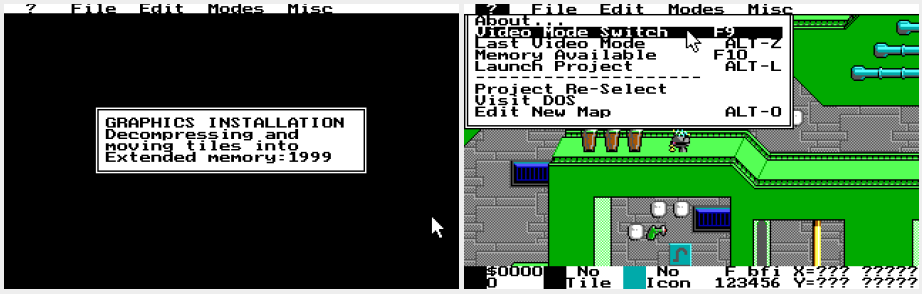
[As we are talking, I start using the Flood Fill routine, and TEd crashes inside DOSBox with a "Divide by zero" error]

JR: [Laughs]

DL: [Laughs] True to form! It's funny, because I was playing around with it before our interview, and I was saying to myself "I don't know why he was quoted as saying that, it works fine for me!"

JR: [Laughs] Well, it didn't ever divide by zero when we were using it... OK I think the problem is that you didn't have a tile selected. It will blow up if you try to flood fill with no tile.

[I restart TEd, which shows the graphics installation dialog]



See that graphics installation right here? What TEd would do is it would actually let you edit maps in CGA, EGA, and VGA and it would move all of the graphics data up into XMS memory, so it could pull it out of XMS when you wanted to switch graphics modes. So if you had a game that had all three graphic modes, then it would let you pull down those tiles. You could pull the VGA tiles down because you're switching to VGA mode, so they would all be loaded up and sitting in memory.

DL: Yes, I read that was a pretty revolutionary feature at that time.

"Totally! [That error message] was totally for Tom [Hall]!"

JR: Yeah. The other thing I did was that if you had a cursor somewhere in the level, you could press Alt-L, and it will launch the game and put you right at that location in the level. See if it will work.

[When I place my cursor in the level and press Alt-L, an error message appears]

DL: [Laughs] So, this is the other thing I wanted to talk to you about...

JR: [Laughs]

DL: I took a screenshot of this error message as I was playing around with TEd earlier, because I wanted to ask you about it. You wouldn't see this kind of error message in Photoshop or Maya these days... [Laughs]

JR: Nope, nope! This tool was only written for me and Tom [Hall].

DL: Of course! That's what I wanted to ask you: Did you write this error message specifically for Tom?

JR: Totally! It was totally for Tom. I would never make that error. [Laughs]

DL: [Laughs] I can imagine him reacting to it when he saw it for the first time.

JR: Oh yeah, he'd be laughing.

[John then shows me that you need to add the name of the executable of the game after the TEd executable so it knows what to launch when pressing Alt-L]

DL: Hey, that worked!

JR: Oh, but it put you at the start of the level. I think the reason why is that you're using a release version of the game, which ignores that command line parameter. The commercial version doesn't allow that, because they you could just cheat!

DL: Yeah, OK, that makes sense. But to me, cheating is one of the things that got people started in game development… you know, having the ability to modify levels in the game to mess with their friends...

JR: Oh yeah! [Laughs]

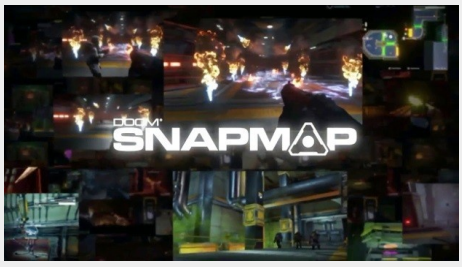
The future of game development tools

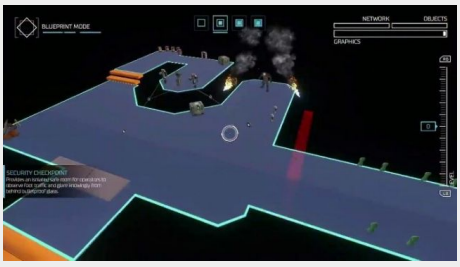
DL: So, I wanted to wrap up by talking about the future of game development tools.

In the early 90s, when I was getting started making games by using the tools made by you and other developers, it wasn't always easy: you had to read the help file – that is, if it existed. You needed a bit of technical knowledge, and you often had to learn some basic scripting or programming. However, at the same time, you could make some 8-bit pixel art and it could look half-decent. You also had the flexibility to make the levels look any way you wanted.

These days, there are tools like SnapMap which are a lot more accessible to everyone, but there is a lot less flexibility. There's visual scripting, so you don't really have to learn how to script or program. Making assets that look good enough to be in the game is out of reach for most people.

My question to you is: Do you think that some of these modern tools are not giving people who are getting into the industry enough flexibility, or do you think that this is just the evolution of the games development – in the same way that people went from coding in assembler to compilers – and this is just our industry is moving forward?





JR: This is where it's gotta go. The group who made SnapMap started their careers by playing Doom and using TEd. The company that was embedded inside id Software was called Escalation Studios. They did Doom Resurrection. I've known those people for over 20 years. They're very hardcore gamers, they come from the Doom world, and then the Quake world. They've followed the company since we made Doom. These were people whose lives were about making levels for years.

So, SnapMap was a natural evolution of where tools need to be for people, but on console. Obviously they didn't create Doom using SnapMap, but SnapMap is a really great way for people who have never done level design to – at least – see if they like it, and if they do like that, well there are more powerful things that they can do. Maybe not with this version of Doom, but with the Source engine, and many other engines out there, like Unreal. So this is to give them a taste of what it's like.

DL: OK, that makes sense. So it's not a replacement, it's a way to get people started on that path.

JR: Yeah, because making the pieces of SnapMap is the hard part. So, if that's something that people can do for the PC version of Doom, then they'll actually know what real level design is like, at least in those modular chunks.

John's advice for tools programmers

DL: So, for my last question, I wanted to loop back to where we started: looking back at classic game retrospectives.

"First, you need to write the tool for the user of the tool."

Sometimes there are tricks and tips that - over time - get lost. I think that one of the interesting things about the GDC classic game retrospectives is that people who are bringing back knowledge that has gotten lost over time. People like you, who have been in the industry for a while, can pass down this information to people who are just getting in to the industry now.

So, specifically from a tools development standpoint, what advice you would give to tools developers these days?

JR: Yeah, definitely. First, you need to write the tool for the user of the tool. So, if a level designer is going to be using your tool, that's the person that you need to make the tool for. Make it as easy as possible, with as much power as they are asking for. It's also important to take the time to really use the tool yourself, so you can experience what's annoying about it.

"A lot of times, tools programmers don't go far enough asking the questions about 'Why do you want that? What is it that you're actually trying to do?'"

DL: Definitely.

JR: Also, a lot of times, when designers ask for some power, a lot of times, tools programmers don't go far enough asking the questions about "Why do you want that? What is it that you're actually trying to do?" because we can wrap up a lot of functionality for you to make it easy when you want to put these things in the world... versus "here are all these components... have fun, level designers!" and then the programmers run away.

DL: Right! [Laughs]

JR: It's also great if the tools programmers and the users are in the same room, to make sure that the tool is being written with the best communication between the two.

DL: Agreed.

JR: Finally, don't make stuff in assembly. [Laughs] What I mean is, it's really hard to maintain a tool ten years later when it's using languages or features that are so hard-coded for the platform that it's on. So try to abstract the functionality so you can replace it later. I actually had that problem with TEd, where - in 2001 - I needed a tile editor to make levels for a new game I was going to make, and I went back and looked at the TEd 5.0 code, which was from 1991, and I was like "Wow, there's no way I can use this code in any way..." [Laughs]

DL: [Laughs]

JR: Remember, tools live longer than games do.

"Remember, tools live longer than games do."

DL: That's all great advice. Thank you so much, I really appreciate you taking the time to talk to me.

JR: Thanks!

David would also like to thank Kris Graft, Marc D'Amico, Shawn Guzzo, and Miles Holmes for their advice in writing this article.