

## Indie Postmortem: Large Animal's *RocketBow!*

By wade tinney

*RocketBow!* was developed by Large Animal Games, a small, independent developer based in New York City. Large Animal was founded in January 2001 and has been developing web-based promotional games and downloadable games for the casual game market for the past four years. We currently have a full-time staff of eight and a number of regular contractors.

*RocketBow!* was Large Animal's fourth original self-funded title and by far our most ambitious. Production started in June of 2003 and the game launched in November of 2004. The core production team consisted of five people, including one who came on board five months before the game shipped. I was the lead game designer and producer on *RocketBow!*, and the following is my perspective on how things went.

### What Went Right

**1. Clear Vision for the Game.** The *RocketBow!* concept is a simple one: bowling over contoured terrain. Having a clear and compact core mechanic gave us a point of focus even as great ideas for game features proliferated and threatened to overwhelm the project. We were pretty aggressive about cutting or limiting features, such as moving platforms, multiplayer, and more intricate level designs. We did this not only to contain the scope of the project, but also out of sensitivity to our audience - a broad demographic consisting of 25-50 year-old casual players who don't necessarily have a lot of gaming experience. For each feature, we would ask ourselves the hard questions: Is this critical? Will our players miss it? Would they "get it?" It was a constant struggle to separate our desires as experienced players of traditional console and PC games from our instincts as creators of games for more casual players. We knew we needed to stay focused on the core mechanic, and keep the game simple and easy to learn.

**2. Torque and ODE.** After investigating a number of 3D engine possibilities, we settled on Garage Game's Torque engine. This option was attractive for a number of reasons: Garage Games was very vocal in its support for independent game development, the price was within reach (\$500 for a commercial license), and it had fostered an active community of developers around their engine. This community proved to be an incredible resource for us.

Early in the development process, we made the decision to integrate the Open Dynamics Engine, an open-source physics engine, into Torque. Since the whole mechanic of the game was based around objects rolling and colliding in interesting ways, a realistic physics simulation was crucial. Though others had previously had difficulties integrated ODE with Torque, we managed to avoid the problems they had due to the fact that *RocketBow!* was a single-player game. We went with ODE because it offers a feature set similar to more expensive physics middleware packages, but at an open-source cost.

**3. The Independent Games Festival.** The IGF was positive for *RocketBow!* in a number of ways. First off, it gave us a concrete external deadline to meet. Actually it did this twice, since we submitted versions of the game to the competition in both 2004 and 2005. The additional year's worth of development paid off: the game was selected as a finalist in the 2005 IGF and went on to win the award for Technical Excellence. Being an IGF finalist resulted in some great exposure for *RocketBow!*. Media outlets picked up on the IGF story, and we were invited to show off the game at the IGF pavilion at this year's GDC. We try to take the whole team out to GDC every year and those free conference passes are important for a small shop like ours.

**4. Eighteen Months of Development.** As a small, self-funded studio, there were many times when most of the production team had to step away from the development of *RocketBow!* and work on a client game in order to pay the bills. Though there was at least one full-time programmer who was able to keep working on the game during those



*RocketBow!*, a 2004 IGF Finalist.



periods, the net result was nonetheless a long and drawn out production schedule. However, such an arrangement gave us plenty of opportunities to come back to development with fresh eyes, which helped tremendously when iterating on the gameplay, levels, assets, interface, and sound. Furthermore, we had plenty of time to tune the physics and tweak difficulty settings. If a publisher had funded the game, we wouldn't have had the luxury of spending as much time polishing as we did.

**5. A Great Team.** Through *RocketBowl*, we found a fantastic programmer and an outstanding associate producer and level designer. Both started out as interns, contributed to the game in a huge way, evolved into contractors, and are now full-time employees. Yossi Horowitz started working with us in the Summer of 2003. Having a programmer dedicated to this project was absolutely critical, given our other obligations and the fact that we were working with technology that was new to us. Yossi climbed the learning curve steadily and made sure that the rest of us we were at least *thinking* about *RocketBowl*, even when we didn't have time to actually work on it.

About five months before we launched the game, we brought in Coray Seifert as an associate producer and level designer. Coray earned our respect right away with his impressive *Halo* skills. He also quickly got his head around the Torque level design tools, dove into writing the remaining game copy, and helped playtest and tune the physics. Most importantly, Coray brought enthusiasm and a fresh perspective, which really helped to re-energize the project in the final stretch.

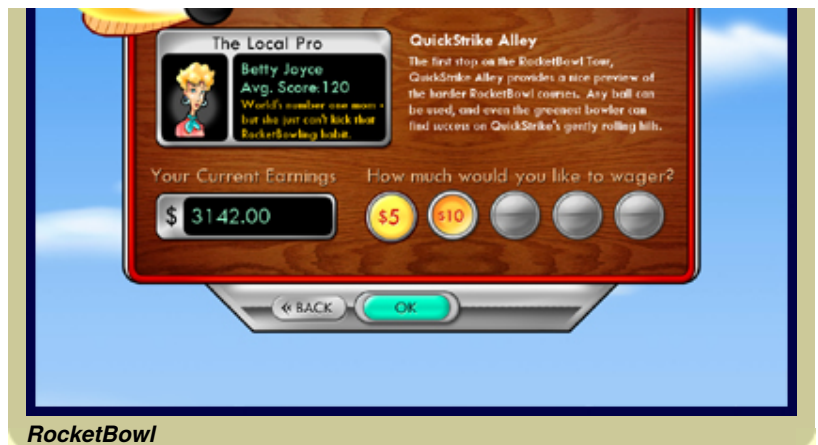
I think we may take it for granted sometimes, but our team worked (and continues to work!) together extraordinarily well. Everyone on the team was committed to making a great game and impacted *RocketBowl* in a fundamental way. As a result of everyone's hard work, we managed to maintain a quality of life that, by all accounts, defies the industry standard. We only crunched for a handful of days and even when we did, it felt more like a barn-raising than a death march. I feel incredibly lucky to work with such a smart, creative, and dedicated bunch of people.

## What Went Wrong

**1. Money.** Large Animal does not have deep pockets; in order to build our original titles, we've had to stash away the profits made from building web games for clients on a work-for-hire basis. As a result, it was nearly impossible for us to give the amount of sustained focus to *RocketBowl* that the game really needed. Our efforts were deeply fragmented, as most of the team would need to go off and build a game for LEGO or Cartoon Network to pay the bills. Despite what the number at the end of this article would suggest, we never had a "real" budget for *RocketBowl* since, officially speaking, we couldn't actually afford to make the game at all. We simply didn't have the money to bring in additional programmers, modelers, or other specialists.

**2. Too little planning.** We jumped into production on *RocketBowl* too quickly, without a pre-production and planning phase. Prior to this project, no one at Large Animal had developed a game of this scope using a 3D engine. There was a lot of learning to do. Rather than setting aside an explicit pre-production phase in which to explore the technology, identify areas of risk, and figure out solutions, we just forged ahead and started making the game, learning as we went. As a result, we didn't always know the toolset as intimately as one would like. While working on the recently released expansion pack for *RocketBowl*, we uncovered a useful feature of the Torque level editor that we'd never found before. And we realized late in the schedule that we should have centralized all of the editable data in the game so that the designers could make changes without consulting the programming lead. If we'd had a dedicated exploration phase, we would have saved ourselves time in the long run.

In addition to the very real limitation of not having the financial resources to completely focus on *RocketBowl*, there were other obstacles that we created for ourselves by not approaching this project in as structured a way as we do our client work. For example, whereas all of our client



projects begin with a design doc, a code plan, a timeline, and a set of deadlines, *RocketBow!* had only the most minimal documentation. Also, we never created a full prototype of the application interfaces, so that we could see how the experience flowed (or not!) for the user. As a result, we ended up making adjustments once the "real" interface was in place - adjustments that were more expensive as a result.

## RocketBow!

*RocketBow!* never had a clearly defined budget. Consequently, the entire project seemed to lack definition; it was hard to know if we were on track because there were no tracks to speak of. It wasn't until the last four months of development, when we were finally able to bring on an associate producer, that the project started to look more like a "real" game project, with an actively updated task list and a roadmap to release.

**3. A Completely New Technology.** The production time on *RocketBow!* was at least doubled as a result of our lack of experience with the Torque engine, or for that matter, with any 3D technology. We'd built dozens of 2D games using Flash and Shockwave, and even some 3D games using Director, but had never worked with a C++ 3D game engine. When we started the project in the summer of 2003, Torque had some rough edges (for example, some of the documentation was lacking at the time) which, because we were so new to this area of technology to start with, had a big impact on our work.

We were also naïve about how much additional QA attention a 3D game requires. Even though we applied more QA energy to this project than any other project we'd previously worked on, it still wasn't enough. Video and sound card issues that we hadn't encountered or foreseen reared their ugly heads post-launch, necessitating patches and new builds for all of our distribution partners. We didn't have a full-time QA person on *RocketBow!* until the last few weeks before launch and that wasn't quite enough.

**4. Trimming the File Size.** In the downloadable world, it's important to keep your games small, lest your risk intimidating potential customers with a lengthy download time. After all, there are plenty of other free trials to download if yours is taking too long. We knew this, but in the midst of an ad hoc project, it fell through the cracks. It wasn't until a few months before we shipped that we got serious about file size and began scrutinizing every file to see where we could trim the fat. When it came to making assets smaller, a little utility called PNG Gauntlet came through in a big way.

**5. Gameplay.** There were a number of areas in which the gameplay could have been improved as well. For example, we've always felt, even from the very first prototype, that each *RocketBow!* level is highly replayable. Each "frame" of each course is different, and there are a wide variety of approaches to any given frame. In real world bowling the "level" is always exactly the same and holds an endless appeal for countless players. What we failed to consider is how the progression of level unlocking would affect the player's *impression* of the game's replayability. Since the levels in the game are unlocked according to performance on the previous level (or in the case of tournaments, require a cash entry fee), players would play the game long enough to unlock all the levels, consider the game "beaten", and email us complaining that it was too short. The free expansion pack that we just released is in part a response to those complaints.

We didn't really dig into the fictional context of the game until the eleventh hour, and it shows. In the end, our presentation of the backstory of the game was boiled down to a single title screen. There are also great NPC characters in the game, created by our art director Brad MacDonald, but they were introduced late in the process, when we were reluctant to change the flow of the application too drastically. As a result, the characters don't occupy as prominent position in the game as they deserve.

There are other aspects of the game design that I feel could still use some improvement. For example, in the game you can aim away from the pins for the frame you are on and knock down pins that are off in the distance (a "Wild Shot"). Effectively, Wild Shots earn you a free throw. However, that payoff isn't really enough to justify the use of Wild Shots as an on-going strategy. Our scoring system is pretty faithful to real-world bowling. We attempted to integrated Wild Shots into the scoring system in a variety of different ways, but I feel that we never quite struck upon the right approach. At the very least, we should have rewarded the player with some game cash.

## Conclusion

Working on *RocketBow!* really helped us grow as a company. Not only did we tackle a big technology challenge, but we saw firsthand what happens when a production process is not clearly defined.

Given the exposure the game has gotten, the technology expertise it gained us, and the myriad lessons it taught us about game production, *RocketBow!* has been a huge success for Large Animal. And while it hasn't made us rich beyond our dreams, it has been a commercial success as well. It's outsold all of our previous titles combined, has been downloaded over 750,000 times, and has found a loyal group of fans.

## Game Data



**Developer:** Large Animal Games

**URL:** <http://www.rocketbowl.net>

**Target platform:** Pentium 733 with 128MB, Windows based OS.

**Length of development cycle:** 18 Months

**Budget:** \$150,000

**Internal team members:** 4

**External contractors:** 2

**Development workstations:** 1.5GHz to 2.7GHz PC's running Windows XP Pro

**Development tools:** Torque, Visual C++ 6, Hammer, MilkShape, PNG Gauntlet

**Number of Ball Bag Jokes Made:** 1,439

**Size of project source materials:** 539MB (uncompressed)

**Size of final product:** 13.9MB

**Release Date:** November 9th 2004

---

[Return to the full version of this article](#)

Copyright © 2016 UBM Tech, All rights reserved