

# Postmortem: zGames' Pong World



By Chris Howard, Tanya Aulachynskaya, Artyom Vorobyov

***Here's the story of zGames' [Pong World](#), the game that won Atari's Pong Indie Developer Challenge contest.***

In March of last year, one of our artists noticed Atari's *Pong* Indie Developer Challenge contest, and suggested we take a look. We had just spun off Softeq's zGames division into its own independent studio and, despite some of the [grumblings from the developer community](#) about the contest terms, we felt it was a great opportunity to get some exposure for our brand and get paid in the process.

Besides, it was the chance to reinvent *Pong* for mobile. We're talking about *Pong*... the iconic video game that launched the video game industry... and Atari, pioneers of the gaming industry. Who wouldn't want to be a part of that history?

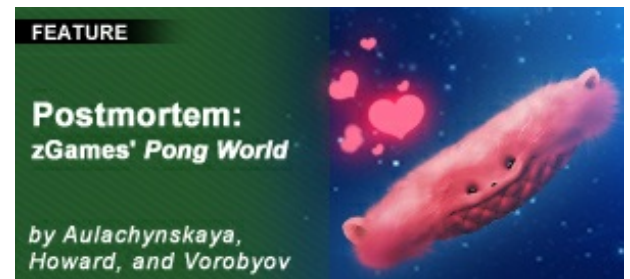
Who are we? Softeq is a 15-year old contract developer of technical software, firmware, embedded software, mobile apps, and games. We are headquartered in Houston, Texas with development offices in Minsk, Belarus.

We have been active in mobile development for a number of years, stretching back to the days of Windows Mobile, where we ran the iPAQ Developer Program for Compaq. We've been members of Apple's Developer Program since the iPhone launch, as well as active with Android and Blackberry.

Softeq's zGames division has produced some casual mobile titles for The War of the Worlds and Animal Alphabet, as well as contract game development for several studios.

The phased approach to the contest made it a low barrier to entry. At most, we'd spend a few weeks working on a GDD (Game Design Document) to see if we could progress to the next phase.

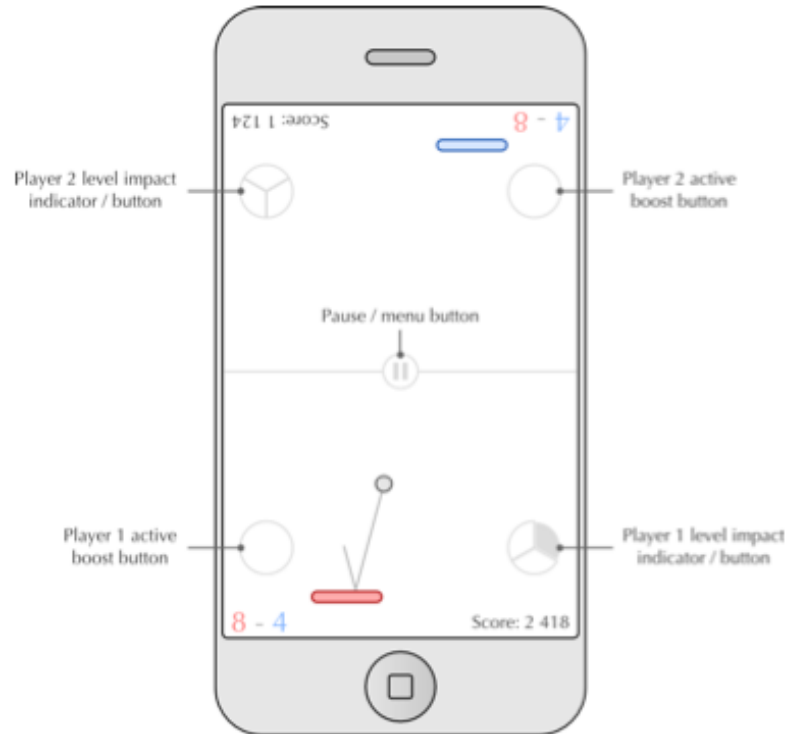
We have a really good GDD template and process, so after the team brainstormed some ideas, our game designer put together a draft, supported by art from our art team. There wasn't a lot of time for reviews and revisions, but after a few weeks we had a 37-page design doc for a game we called *Pong World*, and we submitted the GDD to enter the contest.



## HEADS-UP DISPLAY

The Heads-Up Display (HUD) provides all necessary information about the play flow and grants access to the ~~ingame~~ menu.

Multiplayer HUD mockup:



Active boost button appears only in case the paddle is upgraded with an active boost.

All sectors of the level impact indicator are empty at the beginning of each game. After the player fills all three of them (by collecting special bonuses with the ball) he can tap on it to start a level impact.

The button in the center of the field calls the ~~ingame~~ menu and pauses the game in case of local multiplayer or the game with AI (but not in the case of a networked multiplayer).

### Page from our Pong World GDD.

We took a traditional approach to designing our *Pong* version, deciding not to go too far in reinventing the basic gameplay. It should still be recognizably *Pong* but we wanted to give it a few twists. More about the specifics of the GDD are included in the "What Went Right" section below.

There were 90-plus submissions to the contest, and in May it was announced that zGames was one of 20 semifinalists. That meant a 1 in 20 chance of winning the \$100k top prize... not bad odds. Woo-hoo!

As expected, there was quite a bit of press coverage, and it was nice to see zGames getting mentioned by various sites. A little lost in the crowd of 20, but it was a start.

Developer Studio Name	
Marc Goudet	Caldera Studios
Pegpeg	Kaazing
Delusions	Island Officials
Rafael Sant	Andrew Sipotz
MadRuse Game	Fishbowl Productions, Inc.
zGames	Eppy Games
WorldHiveBrain	Producto Studios
Spantom Games	Autonomous Productions
Love at First Eye	SmugBee Games
Philosophy Major Studios	Meteo Studios

Woo-hoo!

### zGames in list of semi-finalists announcement.

After a brief celebration (that "woo-hoo!" about covered it), it was time to get to work. We had roughly a month to create a playable demo of the game, along with a short trailer. We created a mini-GDD that defined only the game demo... not something we'd normally do during the game development process, but something we needed to focus the team considering the short runway for the contest.

At this point, the *Pong World* team included a PM, a game designer, two developers, and two artists, along with support from our QA department. After a month of constant development, we had a full-featured playable demo consisting of gameplay of a single level. The game came together really well, mostly because of the clear vision contained in the demo GDD... one of the things that went right.

The trailer was put together very quickly at the end, since we needed the demo to be as complete as possible. Pavel, our game designer, roughed out a 30-second trailer using iMovie on the iPad in just a day, which actually looked pretty good. We used that to make a more custom version using Adobe After Effects, but having the initial rough cut helped us do that very quickly.

We submitted the demo app, the updated GDD, and the trailer to Atari at the end of May. We felt we had a strong shot at making it to the finals, since Atari could select 10 out of the 20 semifinalists to be finalists. We didn't know much about our competition other than the company names, and most weren't sharing much on their websites. There was a lot of creativity and ideas in what we could see, so kudos to everyone that entered the contest. Still, we had a 50/50 shot at advancing.

Atari announced zGames and *Pong World* as one of only seven finalists -- another woo-hoo! And again, we had roughly 50/50 odds at being in the top three, and in the real prize money. Those were good odds, and it was encouraging that most of the community reacted favorably to our concept via our trailer. There was a lot of contest coverage, and our trailer was featured on several websites. Although voting wasn't controlled (anyone could click on the voting buttons, up or down, as many times as they wanted to click), we consistently stayed in the top three for votes and rank.

Now the real work began. Another developer and another artist were added to the team, since we had roughly one more month to get the title to a finished state. This both helped and didn't help, for reasons explained later. We expanded the game from one to four levels, each with a unique look and level features, additional paddles were added and animated, the menu system fleshed out, and gameplay tested and tweaked.

Using Cocos2d as the platform, we supported both iPhone and iPad, in multiple resolutions. The GDD was continuously updated to match what we learned in gameplay tests, and by the deadline we had a nice game. We submitted and waited, and the weeks seemed to crawl by.



### Original *Pong World* paddle concept art.

Our third woo-hoo! came when we learned we had won first place, along with \$50,000 and a three-year publishing deal. It was a great feeling to win. After another three months of dedicated development, *Pong World* was released on November 29th to celebrate the 40th anniversary of *Pong*. As anticipated, we received a lot of coverage from this event, which was a primary reason for entering the contest.

---

## What Went Right

### 1. We Gained Respect and Recognition

A major draw for us was the potential publicity that would come from winning the contest. By winning the First Place Grand Prize, we've received a wide range of publicity, and our studio has started to receive new recognition and respect from the gaming community.



**zGames received broad press coverage after winning the contest.**

Though we can't name names, we've had several previous potential customers come back to us once they saw we had won the contest. Demonstrating beta builds of *Pong World* at the Casual Connect conference garnered us a lot of attention from publishers and companies looking for outsourcing partners as well. Initial feedback about the game was very positive.

## **2. We Were Very Focused**

We grew up playing *Pong* and its successors, and the knowledge that we were competing to develop an innovative new version of the iconic game gave us a clear direction to aim.

Moreover, the contest deadlines were really strict and we had to focus on the key features to deliver the game in time.

We had to quickly make decisions about game style, controls, and play modes and optimize our efforts to get the results we needed -- without the liberty of holding long discussions on features that do not have much influence on the game. The need for speed forced us to reuse a lot of architectural patterns and pieces of code from the games we made before, such as our *The War of the Worlds: Adventure* game.





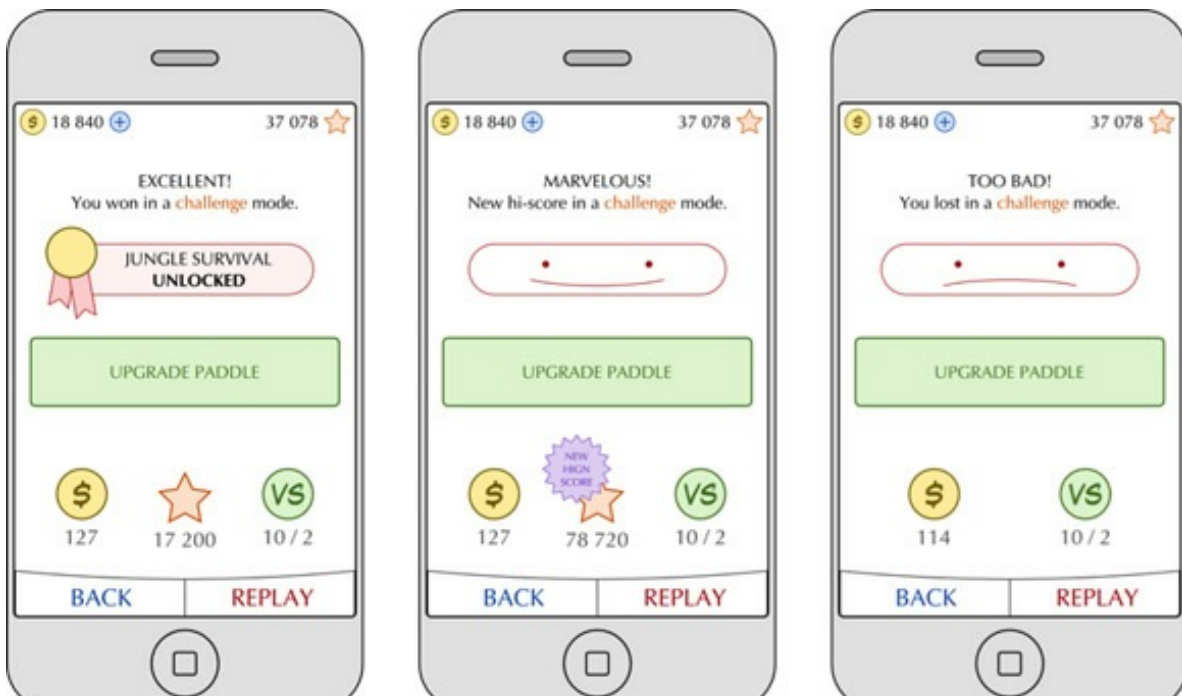
### ***Pong World* level gameplay.**

Finally, the decision to develop the game for one platform also helped us focus. The contest required only iOS, and we embraced that requirement -- no matter what they say about always developing cross-platform. We're very experienced in Cocos2d games for iOS, so it was much easier for us to write quickly for that platform. For example, it is half the effort of writing it in C++ for Cocos2d-x. We can always port *Pong World* to Android later if needed.

### **3. We Had Good Game Design**

We chose a classic approach to *Pong*, and it was a good decision because everyone understands how to play. It also made the game easier to design. We initially built the game around a single character that is cute and appealing, and it gave us a clear vision of what the game should look like.

Our game designer created a full game description in our *Pong World* Game Design Document before the game started development. We created a list with paddle abilities, estimated our effort and left complex features behind.



## ***Pong World* GDD Sketches.**

A stable design for the demo prototype allowed us to meet the short and fast initial contest submission deadline. We used a streamlined GDD with a reduced feature set for the demo, and kept that reduced feature set frozen. The full GDD was large and complete, with a lot of points, but for the prototype it was essential to have a focused design.

Pasha (our designer) noted all questions that arose during the demo creation phase, for later use in the full GDD, but he kept demo changes to an absolute minimum so that we could implement all the features within the time limitations. Processes are great, especially when the team is working hard and is motivated towards achieving the goal.

### **4. We Decoupled Gameplay Code From The UI Code**

Decoupling the gameplay code from the menu code was an important architectural decision. We had three developers working on *Pong World*: one for UI and two for gameplay. We decoupled the UI code from the gameplay code and connected them via a single small interface. As a result, we had one UI developer completely concentrating on his part of code, with no one interfering, which resulted in higher code quality, less bugs, and a reduction in development time.

## ***Pong World* menu interface.**

### **5. We Created a Cheat HUD**

Very often during game development you need to calibrate parameters or tweak a feature. It's a real time sink to re-compile and re-release every time you need to do that. To make this process easier, we developed an in-game Cheat HUD, enabled via a unique gesture.





## Pong World cheat HUD.

Our Cheat HUD is a list of buttons that enable different cheats: e.g. win this level, or add 100 coins, or change a paddle speed parameter. Whenever we added a new feature into the gameplay, we added a button to the Cheat HUD that allows us to change and test this feature. This helped us remove a lot of bugs, because we could simulate a lot of game situations that don't happen that easily.

Other examples of the usefulness of the Cheat HUD include the calibration of paddle movement speed, adding coins and points to the player's profile, triggering the winning or losing a level, enabling paddle bonuses and level bonuses, etc. All of this is very useful to both game design and gameplay testing, as well as testing and QA.

Including a Cheat HUD is one of our best practices for game development. Just make sure you don't include it into the App Store build by mistake!

## What Went Wrong

### 1. No Hablo Español

It's a long road between a winning contest demo and a shippable game. There are a lot of additional components that comprise a full-featured game besides gameplay. To win the contest we didn't need all of those other features, and in the development rush we didn't optimize our app for a worldwide release. After winning the contest, we had to quickly implement those additional components, and what's worse, these new features tended to require a line of code here, there and everywhere. One example was localization.

### Localization Mistakes

1. Text on image textures
2. Lack of space in text containers
3. Direct string references in code

In hindsight, we should have planned for localization from the beginning, but in the contest development rush it was a corner we cut. We had a lot of strings, level names, paddle names, and other text baked into our graphics and our code. To ship the final game, we had to remove text from all graphics and hard-coded text strings, and add localizable text support.

### Localization Best Practices

1. Never place text in a texture. Disable your artist's Photoshop font tool.



2. Design text containers so they can take 20 to 40 percent longer strings. For designers and artists, it's a good habit to draw an approximate text container size on UI sketches (with dashed lines, for example).
3. Use indirect string reference (using a macro, for example). Also include font size into localization files if your containers cannot select font size automatically.
4. Avoid localized sounds. For example, we used abstract paddle exclamations in most cases, but still have a "Tap Me" audio message that is strange and fearful for at least one Russian-speaking mom.

Localization required approximately 140 man-hours across the *Pong World* team, not including the actual translation and Language QA, both of which were done by Atari. *Pong World* now supports six languages.

## 2. We forgot the FTUE

One more feature that we didn't plan beforehand is a tutorial for first-time users, commonly referred to as the First Time User Experience, or FTUE. We like to pronounce it as "phtooey!"

This is a must-have feature for every modern game. When someone opens your game for the first time, he needs a guided tour of gameplay and features, and you need quite a lot of steps to show everything you want them to know.



### *Pong World* FTUE.

Our FTUE consisted of approximately 17 steps, and we needed to create a nice and unobtrusive way to show the tutorial texts. We created a stylized popup box and some arrows, but the hardest part was creating the scripting logic for all 17 steps. For example, one step included adding coins to the user's balance, and several steps needed to point at a specific button while all other buttons are disabled.

To properly implement a FTUE, you should keep in mind the following:

1. Don't insert FTUE code directly into your game logic. Encapsulate it into a separate subsystem and connect it with the game through a few clear interfaces.
2. Force your game designer to design the FTUE before you begin game development. If you know all the features that the FTUE requires, it will be easier to implement them. Too often, a team will wait to design the FTUE until after the game is completed, arguing the game must be finished first. It's a trap!

3. Make content lockable from the start (buttons, tables, game itself, etc.). At a minimum, UI features should support some kind of disabling/enabling property.
4. Some content items should be processed separately. For example, if you block all UI content on the level selection screen and allow only the "Unlock Level" button to be pressed. Design unique access methods for such separately processed items.

Adding and perfecting the *Pong World* FTUE resulted in over 200 man-hours of additional work.

### **3. Releasing Builds Too Often Can Lead to Mistakes**

One of the requirements by Atari was to release the builds often, especially when we came close to the App Store release. This is a problem, because the build process takes from two to eight man-hours to merge, build, and test the game, especially taking into account the amount of features that need testing.

Though frequent releases are a good practice, they can work against you when done too often. Our QA team was not able to regression test everything in every build, and started to miss some bugs. We had a very disappointing bug with the amount of coins shown on different UI screens, and people were obviously confused by it. Missing this bug resulted in a lot of negative reviews in the App Store.

For the future we plan to implement several practices that would help us speed up the build process. We currently use a build machine and building script, but we could set up a continuous integration server, and cover some of the core functionality with unit tests. While having a build schedule is good, out-of-turn builds should be avoided unless it's absolutely necessary.

### **4. There Was No Soft Launch, or "I've Got You Under My Thumb"**

It can be very useful to soft-launch the game in a secondary market a month prior to the worldwide launch. A soft launch can ensure there are no major issues with the store (in this case Apple), and it verifies you can get through the store QA and approval process. No one likes to get rejected days before launch over a guideline issue, and have the release set back by weeks.

A *Pong World* build was submitted to Apple more than a month before launch and was approved. However, the decision was made to hold off on the soft launch of the game due to the lack of the FTUE in that build. In hindsight, valuable information could still have been gathered.



### ***Pong World FTUE intro.***

A soft launch serves as a wide-area beta test, and can point out important problems with the game that aren't obvious to the core development team. For example, after the worldwide release many users complained they could not see the paddle under their thumb. It's an obvious and natural way to play, but most of the team tended to use their index finger instead, and this issue wasn't raised in our local play tests. A soft launch would have caught this easy to solve problem, along with a few others, and avoided some negative reviews for the worldwide launch.

### **5. We Were Surprised by the Number of Frameworks**



*Pong World* uses 14 (!) different framework SDKs. That's getting to be normal for any game, but even we were surprised by the number, and every framework adds its own set of problems. It can be hard to research and implement the various frameworks, because they don't tend to be very well documented and often do not work very well. It takes a lot of time, and you have to be ready for that.

Some examples of the trouble we faced with framework implementation:

**StackMob:** Generally provides really bad support. Our questions related to problems with push notifications weren't answered for weeks at a time.

**Flurry Video Ads:** We really wanted to add the Flurry vAds framework to show rewarded video for game coins, but the fact is, it doesn't work without server-side support and we didn't have time to add it, since our game is purely client-side. We went with AdColony instead.

**Facebook SDK:** This is a complete mess. Try to add sharing/authentication features to your game just once -- you will understand. The SDK has two primary APIs -- deprecated and new, and the documentation is only partially updated to comply with the new API. Have fun sorting that out!

**MoPub:** Sometimes the ads that come from the Chartboost server would not close themselves and have no Close button. An ad would get served and essentially lock the game up indefinitely. We had to add a hook that closes ads after some period of time.

Unfortunately, these were just a sampling of the problems.

## Conclusion

Entering the *Pong* Indie Developer Challenge and developing *Pong World* for Atari has been a very positive experience for us. Obviously it doesn't hurt that we won first place, but we also got the chance to show we can design and deliver a major title for a well-known publisher like Atari. We're continuing to update *Pong World* as well as develop new titles for iOS, Android, and Windows Phone, using Cocos2d, Cocos2d-x, Unity3D, and even PhoneGap.





## The zGames *Pong World* Team.

Our *Red Weed* game for Android was recently featured by Amazon as the Free App of the Day, giving our brand even more exposure. Ouya has our attention (we're a founding member) and we see a lot of opportunity out there for our own games as well as contract game development. Look for another reimagined classic title from us later this year from yet another icon of the game industry.

## Data Box

**Developer:** zGames, LLC.

**Publisher:** Atari, Inc.

**Start Date:** March 2, 2012

**Release Date:** Nov 29, 2012

**Platforms:** iOS

**Number of Developers:** 3 devs + 1 game designer

**Number of Artists:** 4

**Length of Development:** 5500+ man-hours, 7 months

**Lines of Code:** 67,000

**Development Tools:** cocos2d, Box2D, xCode

**Kicker (Foosball) matches played:** 240

**Nights spent at the office:** 9

**JIRA artifacts:** 590

**Number of intermediate builds:** 30

**GDD revisions:** 11

**Number of classes:** 250

**Number of play tests:** 3

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

Copyright © 2015 UBM Tech, All rights reserved