

Gamasutra: Grhyll JDD's Blog - Post-Mortem Red Skies

 gamasutra.com/blogs/GrhyllJDD/20160518/272962/PostMortem_Red_Skies.php

 [Blogs](#)

by [Grhyll JDD](#) on 05/18/16 01:13:00 pm

Featured 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.

Red Skies

Post-mortem

A month and a half ago, I released [Red Skies](#), a nostalgic shoot-them-up for iOS and Android. Let's break the suspense: I'm not rich by now. However, it was far from being my goal. Some might find it odd, but it was merely an attempt to see if I could develop a complete game on my own (and with the help of an artist – and a sound designer for the last weeks), from concept to release and marketing, and see what I could learn in the process for future games. I have ambitious plans, but I didn't want to start right away with my Big Project; I just wanted to be aware of the most common pitfalls every game dev has to face at some point.

Note that I already have some experience, since I work as a programmer in a video game studio with a few released games; this project was an opportunity for me to have a look at the other aspects of the development, such as game design, project management, marketing...



Numbers

Don't lie, we all love numbers in post-mortems. Mine won't be very impressive, but at least you'll have an idea of the outcomes of this release. There was a premium version on iOS sold at about \$2 without in-app purchase, and a free version on iOS and Android, with ads, containing a single in-app purchase at \$2 that brings the game to the level of the paid version, removing the ads and doubling the currency earnings.

The paid version didn't get a lot of success, with 2625 App Store views resulting in 16 sold app, for a great total of \$23. Yup, not rich yet (although I don't even know where most of them come from).

The free version had a better conversion rate, with 856 view on the App Store, translating to 144 downloads on iOS. There were 126 downloads on Android, and an astonishing total amount of 1 iap bought (on Android).

About 500 ad views earned \$3.60.

And that's it for the numbers; I don't think it would be interesting to break that down further, since they are too low to have a real meaning.

The Big Bullet Points Lists

Now it's time to have the famous Good/Bad lists about how the project went!

What went right

- Project management

We managed to stay quite focused on the dev for most of the project's duration. Despite a few short periods with less efforts, there were pretty consistent progresses made, and the execution felt natural. We chose a very small scope for the game, which was a good thing in regard of the ambitions we had, and if obviously it evolved a bit while we were working on it, it didn't grow out of control. We managed to put in the game almost everything we wanted, and on most aspects I am satisfied by the polish level.

Initially, we saw this project as a 3-months work, but it would have needed a LOT more investment from us to reach that deadline. It ended taking about 7 months, which may be a bit long for a project of that size, but it's acceptable. The good part is that, at some point, we set a date for the release (two months before the said date), and we stuck to it. We had no imperative deadline, but I felt it was important to respect (as long as it was possible of course) our own decision.

- Unity & git

I use Unity and git in my work, and I can't say enough how much I love those. I won't list all of their assets, but we managed to have an efficient "team" process using them. It allowed the artist to easily test the game and tweak graphic settings, without me having to build new versions for each new element. We also had some git conflicts, because where would be the fun otherwise, but on a project of this scope, with only 2 people on it, they were quite easily resolved.

On a personal level, it gave me the opportunity to have a look at some aspects of the dev that I didn't have to consider on previous professional projects, such as optimization (and the game runs pretty well on not-that-powerful devices).

It also was very convenient for some less pleasing features: the social integration, be it posting on social medias or using Game Center and Google Play Game Services, was made extremely easy. When Unity didn't plain simply allow the use of it (Game Center works right out of the box), free plugins or tutos were available online and allowed me to implement all of those boring elements in a rather short time. The integration of Unity Ads was as well far easier than what I feared.

- Playtests

I never ran playtests before, but I found out it's a great thing to do! I didn't expect to learn so many things about the game just by watching people play it, it's extremely precious. Even if the game isn't the best game ever, you can feel that, because you're standing next to them, players enjoy quite a lot testing it. I made a first session in a local game devs group, with a lot of valuable feedbacks, and a second one at a nice event in my town over two days. A lot of people tried Red Skies then, and most of them signed to the newsletter, showing interest in the future of the game. Some kids (and one or two grown-ups) didn't even want to give up the device after a lot of tries, which was heart-warming.

In the end, it doesn't mean that much about the quality of the game, since being face-to-face with a player changes a lot his reception of the game, but it was very good for motivation!

The online playtests however were a lot more disappointing, with only a few people willing to test the game, and even fewer taking the time to fill in the feedback sheet. What's more, the most valuable part of the playtests is simply watching the player through the game, which obviously isn't possible online.

What went wrong

Most of the points listed below come from the fact that I didn't have really high hopes for this game. It was from the beginning a "small, training project", so why bother with the boring aspects? Obviously, one won't go very far with this mind-set, and I guess I should have try a bit harder, yet even by failing in those areas, I still feel I learned interesting things! It's easy to say "why bother, the project has no commercial viability anyway", but it's definitely something I won't tell myself for my next projects.

- Ads

It didn't *really* went wrong; as I said, Unity Ads integration was the easiest thing ever. The bad side is that we tried to offer a free-to-play, ad-monetized experience to players, without any marketing knowledge of the thing. There aren't a lot of forced ads in the game, especially if you choose sometimes to watch an optional one at the end of a run to double your income, so the model isn't probably the worst, but I feel there are a lot of details that could have been made in a better way, in the frequency or the presentation of the ads.

What's more, I'm completely lost when it comes to understanding how those ads bring us money. With Unity Ads, the eCPM varies freely, reaching \$7 in some countries for some days, and staying to a shiny \$0 in others. If this system is perfect for a noob like me, I'm not sure it's the best solution for someone really trying to make money with an ad-supported game.

- Analytics

I've read enough posts on Gamasutra to know that analytics are crucial for any project, yet my analytics implementation was very, very poor. I didn't take the time to really understand the important metrics and how to gather them, so I ended up with a bunch of events impossible to analyse in a way that make sense, and I only have a very vague idea of how players behaved in the game.

I didn't test seriously enough my implementation of events, just putting some here and there (yes, exactly the way most articles warn you not to do), and it probably would have been easy to improve that if I dedicated a bit more time to think about it.

- Audience targeting

We didn't spend a lot of time thinking about our audience, and I believe it's one of the core aspects of mobile development. The game is made to appeal to achievers, with a progress curve coming from the upgrades, the skill of the player and his knowledge of the enemy waves (which are always the same), but in the same time it has nothing really original. The mechanisms are very classic, with the small exception of the bombs you can drop on the boats, and the graphics aren't extremely compelling. All in all, our target players don't have any good reason to seriously look at Red Skies.

Another failure in that domain is the session duration. Mobile games without very deep mechanisms should probably have short sessions, something between 3 and 5 minutes. Those timers are respected in the training mode, but it isn't meant to be the core of the game; the infinite, main mode will require from you, once you're starting to be good at it, 15, 20 or even 30 minutes to complete a run.

In the end, except for some players that had the motivation to really dive into it (and made amazing scores, which means they must have been nicely hooked), I think we missed most of both casual and hardcore players through some poor choices and generic look.

- Numbers, marketing and visibility

Well if you read the Numbers section, I think it speaks for itself. On the commercial side, this project is a disaster, and I'm glad I never saw it as a real mean to make money (of course, one could say it happens the other way around: if you start to make a game without wanting to make money out of it, it won't happen). The good thing is that now, I have a personal better experience of the mobile market.

Red Skies doesn't really stand out visually, nor in its gameplay. It's classic, already seen, not bad in its kind (I hope), but today that's far from being enough to gain any visibility. I sent about 50 personalized and targeted e-mails to press, got two answers and 0 review. I hoped to get at least a small article on a shmup-specialized website or two, but nothing. The game has nothing to become viral, so it literally has 0 way to gain organic growth. Since I didn't want to invest money on marketing, there hasn't been a lot of players, beside family and friends (who overwhelmed me with joy by being a lot more to download and rate the game than what I expected).

If I'm still not too sure about what exactly needs to be done to obtain visibility, but at least I now know what isn't enough for that.

In the end, I'm rather happy with how everything went. I released a game I'm not totally ashamed of, which was my primary objective; I learned a lot of things, on various subjects, and I gained some confidence on my abilities. All of those "bad" sides aren't really so bad in my opinion, especially since I was aware of most of them, and chose not to extend the development for too long in order to fix them.

I still don't feel totally ready for my Big Project, but I'm a bit closer to it. I'll try to make one or two more games with similar scopes (a bit bigger, and better targeted – be it with a more stand-out game or with a niche game), in order to learn more things, and then, maybe, I'll try to go a bit further!

Want to talk with me? You can find me on Twitter: <https://twitter.com/Grhyll>

Post originally published on my blog: <http://3-50.net/post-mortem-red-skies/>