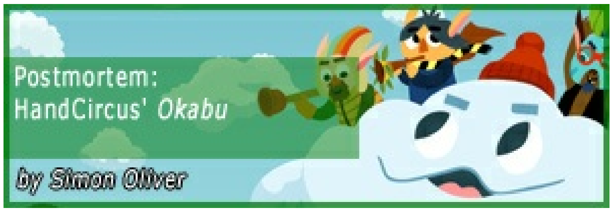
Postmortem: HandCircus' Okabu

PRINT

By Simon Oliver

[*What happens when a developer that was born in the iOS space transitions to making games for consoles? In this postmortem, Simon Oliver, founder of HandCircus (Rolando series) shares the experience of moving to PlayStation 3 development with Okabu.*]



We started HandCircus in 2008, and got off to a flying start with our first title, *Rolando,* which launched on iPhone in December 2008, in the very early days of the App Store. We partnered with Ngmoco, which was just starting up at the same time and shared a passion for the opportunities presented by the iPhone, and its potential to transform the gaming landscape. We followed up with a successful sequel to *Rolando* in summer 2009, taking everything we'd learned in the development of the original and building on mechanics and visual style.

Once *Rolando 2* was wrapped up, we began to think about our next steps as a studio. We were riding high of the successes of our first couple of games, and decided we wanted to think a little bigger for our next title.

The world and storyline of the original *Rolando* had evolved in quite an organic way, and *Rolando*'s characters and world proved to be very popular with the iOS audience.

This led us to greatly appreciate the importance of characters and design of the environments, and when we began discussing the next game, one of the earliest objectives was to create a world that had the capacity to grow into a multi-game, multiplatform entity.

In autumn 2009, we started the planning, prototyping and mood board assembling for the early stages of our next title, codenamed "Project 3".

Working with ngmoco had been an amazing opportunity, and provided a wealth of experience in many areas of the development and publishing process that I was completely unversed in at the time. The commercial success of *Rolando* and *Rolando 2* put us in a great position -- we were now able to fund the development of our next title ourselves, and potentially self-publish. Although it would mean taking on a considerable workload and risk, this would have some major benefits, particularly the full ownership of our own IP.

Following the early stages of pre-production, we spent early 2010 speaking to potential partners. Keeping ownership of the IP was a key requirement to any partnership, but our previous experiences had demonstrated how beneficial a strong partner can be.

We were looking to do something a little different with *Okabu*, so it was important to find a partner that really shared our vision for the game and was really passionate about the project. Of all the potential partners we spoke to, Sony really stood out from the competition. The people there really got the game and it fitted in well with their objectives to build a rich and distinctive portfolio for PSN.

After presenting the game and explaining our vision for the title, we were fortunate to be accepted into the Pub Fund program. This gave us the full support of Sony while allowing us to keep full ownership of the IP -- the perfect arrangement for us.

We had a pre-production phase of nine months, following by a production phase of 12 months and final launch in October 2011 on PSN. It's been an invaluable experience, and we've learned a lot along the way!

What Went Right

1. Co-op and the Casual Audience

While we may have not catered fully for the core gamer (see "What Went Wrong"), the feedback and reviews that we had from casual players was fantastic. One of the key targets was casual co-op players, particularly those that might be slightly asymmetrical in their abilities -- perhaps a father with his son, or a couple playing together (with one player possessing a greater ability than the other).

We really wanted to provide a variety of activities and play styles, so that if one player is more experienced they can take care of the more challenging parts of each level -- leaving the second player to collect pickups or look for the various secrets hidden throughout each level.

From the very early stages, we spent a considerable amount of time prototyping and user testing to make sure that the game was as pick-up-and-play as possible (particularly with non-gamers). The very first prototypes were strikingly different -- co-op was absent, and the game had a more advanced twin stick control system and used a crazy amount of buttons.

As the weeks passed, we ironed out the kinks in the controls and continued to simplify and build in many assist systems (such as snapping and camera guidance) to make the game more approachable. This really paid off, and the problems our early testers encountered began to fade away.

It's been a joy demoing *Okabu* at various shows and festivals (from GDC and E3 to GameCity in Nottingham) and watching such a diverse audience enjoying the game. We've had players as young as three able to pick up the controller and start guiding our cloud-whale heroes around the level.

Asymmetrical play has been a particularly fun phenomenon to observe, as we see the more skilled player guiding the other around, passing on instructions, and creating a trail for the other to follow. A lot of the emails that we've received from players since launch have been from couples playing together, and it's great that we managed to create something for that audience.

2. Development and Execution of a Visual Style

Coming from the iPhone to the PlayStation 3, we were obviously targeting a completely new audience, but we hoped that there would be some crossover with iPhone gamers and some awareness of the *Rolando* series amongst the community. With that in mind, one of our objectives was to create a visual style that was fresh and distinctive, but that still shared some common ground with *Rolando* in order to bridge the gap between the two audiences and hopefully bring fans of our previous games along for the PS3 ride.

The early days of pre-production were spent exploring mechanics and concepts that would provide a solid foundation for our new game, and we settled on the idea of cloud characters fairly early on. Using clouds as our heroes seemed to present some great opportunities, from the ability to absorb liquids and rain down upon the ground, to sucking up small objects and building up and firing electrical charges. Their appearance and properties also seemed to reflect the environment that they occupy, which drew us towards the themes of nature and industrialization.

Around this time, the BBC Natural History show Planet Earth was being repeated on TV, and a section of one episode proved to be a major inspiration to the project. Set in the Okavango Delta in Botswana (from which *Okabu* takes its name), it began with images of the bone-dry delta, starved of water and completely lifeless.

Once the waters began to arrive from the distant mountains, the various seeds that had long sat lifeless under the ground began to germinate, and small plants and grasses began to grow. This in turn led to a growth in the insect population, which in turn drew birds from all around.

Fish began to arrive from further upstream, and mammals journeyed to this newly formed oasis. In the space of weeks, this dry barren land had been transformed into lush wetlands teeming with life, and it was this vision of the power of nature that served as a bedrock for the development of *Okabu*.

We started to explore different sources of inspiration for the art direction, and following on from the wetlands of Botswana, we were soon drawn to the landscapes and cultures of Africa. Piecing together mood boards of images of the Souks and Bedouin encampments or North Africa and the vibrant villages of Ghana, it really struck us how rich a seam of inspiration there was in front of us, and how underused it has been in games.

Digging around in bookstores, we found a wonderful book by the German photographer Michael Poliza called Eyes Over Africa. He'd hired a helicopter and flown over 19 African countries, documenting the dramatic landscapes and striking habitats he'd encountered. This soon became our bible and served as inspiration for the characters and four worlds of the game.

Another key objective for the art direction was to imbue the game with a real, illustrative storybook aesthetic. While this is a more straightforward task in a 2D game, we were now making our first forays into creating a full 3D title, and so early on we experimented with getting the look right and maintaining the vibrant, block color look that illustrator Mikko Walamies' concept art shared with *Rolando*.

Presented by the relatively beefcake-like performance of the PS3 -- compared with the modest power of the first gen iPhone we had previously been targeting -- it was tempting to dive into elaborate rendering techniques and to amp up the detail of the assets and environment. The more detail we added, however, the more it felt like it was detracting from the game and the art style. Simple, bold visuals felt much more effective, and we ended up going for a straightforward, two step cel-shader that subtly hinted at the shape.

As an indie developer, it's obviously important to differentiate your title from bigger-budget releases and to take advantage of your diminutive size. Differentiating our art style was a big part of that (as other indies do so well, such as *Pid*, *Eufloria*, *Sword & Sworcery*, *Papo & Yo*). The bright, cel-shaded world of *Okabu* is worlds apart from most mainstream releases (even those targeting kids) and really help set us apart. We've had so many compliments about the visual style. The team did such a great job building this charming world to dive into.

---

3. Finding the Right Music Partner

Music was a big part of the *Rolando* games. After much amazement at how complicated music licensing can be (and hitting a ridiculous number of dead ends while trying to license tracks we wanted), we ended up going direct to label Ninja Tune and licensing tracks by the UK breakbeat artist Mr Scruff. Mr Scruff's music helped create a really distinctive feel for the game, complimenting Mikko's art wonderfully and greatly contributing it its success. This was something we really wanted to do again for *Okabu*, and from the early days of development we made music a core consideration.

The nature theme of the game drew us towards an acoustic flavor. We were keen to have something that sounded very authentic and natural. This drew us down the world music route, with a particular focus on Africa, given the direction we were headed with the art style.

We began piecing together a few tracks that we felt represented the desired feel of the game and built up a Frankenstein playlist, featuring everything from Bollywood anthems and Malcolm McLaren to Paul Simon and MIA.

We began looking for potential partners to either curate a soundtrack or create something entirely from scratch. It was our first time meeting game audio folks and it was a real pleasure to meet such passionate, talented people. We met people from various parts of the globe, but in the end we settled on a studio based almost in our backyard here in East London: Resonate Music.

I was introduced to Liam Paton from Resonate through a mutual friend, and popped round there one afternoon to give them a demo of the WIP game and to talk through our goals for the project. They really got what we were going for -- i felt like there was a great chemistry there, and we began working together in the Autumn of 2010. I'll let Liam go into a little more detail in his own words:

"With the *Okabu* soundtrack, we have really tried to capture the social aspect of African music, recording all the music live and recording as many of the musicians together in the same room as was possible. All the music and recordings are completely original. We didn't want to rely on any sample libraries for this project, and have tried to retain a real human feel to all elements of the tracks.

"We have worked with a number of very talented musicians and performers over the past six months who have not only brought great playing skills to the project but who also consulted for us, making sure we were creating something with real credibility. With such an original idea and concept for the *Okabu* game, we wanted to create something that was unique and that helps bring this vivid world to life."

Andy, Liam, and the team at Resonate completely blew us away with the *Okabu* soundtrack. It brings so much to the experience and sounds totally unlike any other game out there. From the moment I first heard the *Okabu* theme that they created for our launch trailer, it brought an enormous smile to my face, and the trailer comments mirrored this, with so many people saying how much they loved the music.

They've created a sound for the world of *Okabu* that will be an amazing foundation going forward, and their great work was recognized with a nomination for Best Original Composition in the Music+Sound awards, going up against the King's Speech, Tinker Tailer Soldier Spy, Frozen Planet, and My Week with Marilyn!

4. Building an IP

One of the core objectives with *Okabu* was to create a long-lasting, comprehensive IP -- a building block for future games, future platforms, and future endeavors, and this is definitely something that we've succeeded at.

We've a developed a huge *Okabu* canon, with backstories, characters, continents, fauna, and flora. We have an enormous war chest of assets from each of the game's four worlds -- all manner of buildings, creatures, vegetation and animated avatars, not to mention the outstanding soundtrack that Resonate Music created for the game.

From the beginning of the project, we wanted *Okabu* to be a multi-platform IP, and so the assets have been created efficiently and are highly suitable for lower power platforms, such as mobile and tablet.

This huge collection of assets essentially represents a very flexible toolkit that will allow us to make the transition to our next *Okabu* project smoothly and swiftly. After wrapping up the initial PS3 game, we spent several weeks brainstorming ideas for subsequent games for the *Okabu* IP. With our roots in iOS, we're particularly excited about what we'll be able to bring to the iOS and Android space. We've already been hard at work building prototypes for mobile, and the *Okabu* IP feels like such a good fit.

In addition, the fact that we've retained full ownership of the IP gives us free rein to explore any and all opportunities, no matter how peculiar a tangent they might take and offers us considerable security for the future.

5. Partnering with Sony

Since our very first conversations, Sony has been a pillar of support for *Okabu* and HandCircus as a studio. It's been such a positive experience having them on the project -- the Pub Fund program is a great fit for us, supporting developers looking to self-publish while also allowing retention of IP.

I really admire the objectives that Sony have set out with the Pub Fund program, to build a distinctive and diverse portfolio of games for PSN -- it allows developers like us to spread our wings and set our sights a little higher on more ambitious projects.

The marketing support from Sony was also a huge benefit -- our own marketing budget was relatively modest, but Sony's channels are obviously substantial, with enormous reach. In addition to some great support online, we were given the opportunity to demonstrate *Okabu* at GDC and E3 as part of Sony's program -- something that we would never have been able to do ourselves.

What Went Wrong

1. Planning

The *Okabu* plan required us to make some major changes to our development process, many of which were completely new to us. These included:

- Moving from iPhone to a much more demanding platform: PlayStation 3
- Moving from a simple 2D design to an expansive narrative-led 3D world
- Moving from a publisher partnership to self-publishing
- Self-funding for the first time

Making one major change would have presented a challenge. Trying to do all four simultaneously was extremely ambitious. Riding high off the successes of *Rolando*, we felt like any challenge could be overcome, and that each major change to the process could be dealt with once we got started. Confidence is a wonderful thing; naivety, less so.

Not having worked on a console game previously put us at a major disadvantage during the planning stage, as although we could make estimates as to the resources needed to create *Okabu* based on our previous experiences (and build a schedule accordingly), the accuracy of these estimates was questionable. Working without a publisher also denied us the capacity to verify the estimates and schedule, as well as denying us a safety net if things went wrong. There are obviously fewer options to source additional capital if you are self-funding.

Every area of work ended up being much more demanding than originally envisaged. On the publishing side, this proved to be a huge amount of work -- we were aware how much Ngmoco contributed to the project on the publishing side, but it was even more work in the console space.

On the art and design side, building out a full 3D world is obviously a great deal of work. We were being very ambitious with the size of the game (four distinct worlds, six playable characters, four minigames, controllable vehicles and 12 to 16 hours of playtime). As development progressed, it became evident that we'd been optimistic in our art and design schedule, which led to a very pressured period towards the end of development.

Looking back, our biggest takeaway on the planning side is that we should have really examined the scope of the game more at an earlier stage to better balance the ambitions we had with the resources that were available. For a downloadable title, 12 to 16 hours is on the large side, and it still would have felt meaty enough if we'd cut some of the content. This would have allowed us to reallocate resources to polish the core game and given us the breathing room we needed.

2. Engine Tech

The game started life as a PC/Mac prototype, leveraging the power of great open-source libraries such as Ogre and Bullet Physics. This gave us a real head start, and allowed us to get a playable prototype up and running quickly, but as the project progressed (and we moved from preproduction to full production) we hit some major problems.

The architecture of the PS3 is obviously a world apart from the PC, and while this wouldn't have been too much of a problem if we were porting a simple 2D game, we were dealing with some quite expansive 3D environments and were targeting 60 frames per second. While Ogre had been great on PS3 (and we had some successes porting it to PS3), it became evident that we would need to leverage the SPUs to deliver the kind of experience that we were looking for.

As a result, we began exploring our options -- either optimizing Ogre for PS3 or using some additional PS3-specific tech. Fortunately Sony provides an extremely high-performance engine on PS3 in the form of PhyreEngine, and after much deliberation, we decided that our best option was to leverage the PS3-oriented power of PhyreEngine to get the performance we needed.

This shift to PhyreEngine was obviously a major undertaking, and was not something that we'd factored in during the planning stages (we'd assumed that our PC-centric engine would have been sufficient). Developing a stable, performant engine that delivered all the required features on PS3 was a lot of work.

Towards the end of the project, this really led us to reflect on the effectiveness of our initial tech plan -- most of the elements and features that we'd required and built for *Okabu* (physics, rendering, scripting, audio, save game support, leaderboard support, trophies) would have been very capably provided by some off-the-shelf engines.

The type of game that we were making was a pretty good fit for an existing engine (we weren't using any exotic rendering or physics techniques), and using an existing engine would have allowed us to benefit from some desired rendering and toolset features that we hadn't had the resources to implement ourselves.

While we had learned a great deal from rolling our own engine, the resources that the engine development process consumed may well have been better spent on other areas of development.

Our major takeaway on the tech side is to put more thought into the benefits of building your own tech, particularly for small teams like ours. Does rolling your own engine really makes sense, or are there existing engines that would be a good fit? For some genres and some programmer-heavy teams, building your own tech may make perfect sense, especially if you are trying to innovate with non-standard rendering (voxels, etc.) or have quite a minimal feature set.

In other cases, if you require quite a large number of systems and features that are commonplace in major engines (as we did with *Okabu*) or if your team is less programmer-heavy, existing engines might be a better choice. Mature, stable tools are obviously another major consideration here (which I'll go into in the tools and pipeline section).

The widespread adoption of flexible engines like Unity has opened up some really interesting opportunities, not only in the ability to reduce the tech burden on small studios, but also building a standard toolset that programmers, designers, and artists are familiar with.

---

3. Excessive Multi-Hatting

One of the principle joys of indie development is the involvement that you end up having with a great many aspects of the development process. The wearing of many hats is something that I've had a deep love for on previous projects, but there is definitely a point beyond which it becomes unwieldy. This is a point that we not only crossed, but continued to sprint past in Forrest Gump fashion, without pausing to consider the implications.

As the *Okabu* project was considerably more ambitious than our previous games, and as we were also taking on the publishing role for the game, this meant that there were a whole lot more hats to wear. However, the team size hadn't adjusted proportionally to accommodate all this new headgear.

For a project of this size, it's important to have a natural tension between key decision-makers from different disciplines. The game director might suggest a feature, the lead programmer might consider the implications from a tech standpoint and the resources it would take to add to the game, the producer might consider what impact this would have on the schedule, and the studio head work out the feasibility from a financial standpoint based upon any required changes in schedule and resource requirements.

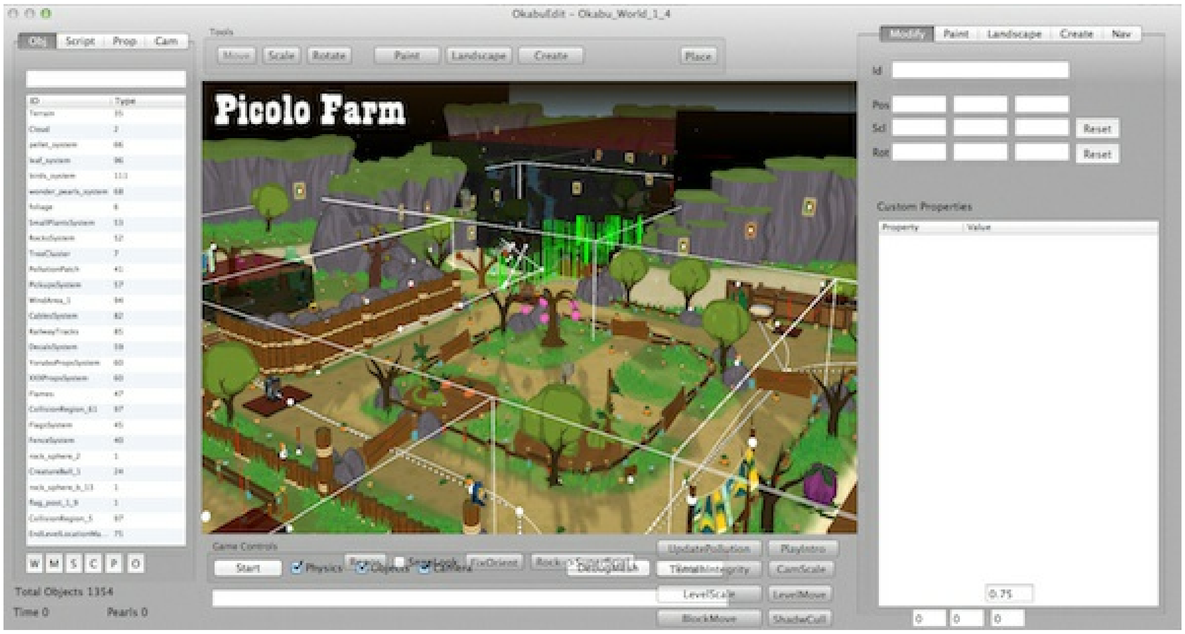
When these roles start to be inhabited by the same individual, it's easy to lose perspective on the implications on all disciplines, and the decision-making process becomes infinitely more challenging. Decisions can frequently be made based only on internal deliberation. This is considerably less effective than the Mel Gibson / Danny Glover-style discussion that you get when two team members without cross-vested involvement can openly explore the implications of a decision.

Our biggest takeaway here is that our structure had become unwieldy from too much multi-hatting of roles and we'd have really benefited from at least one more team member to split those roles up. In particular, another member taking on production responsibilities would have been a real boost to the team.

4. Tools and Art Pipeline

As mentioned in the tech section, we decided to roll our own engine for *Okabu*, and as part of that we ended up rolling our own tools. We spent a fair bit of time delving into the level editing tools for a number of other engines, making a list of the features that we'd like to have, and those that we felt we really needed to craft levels around our desired feature set. Top of that list was the ability to rapidly switch between "Play" and "Edit" modes, as capably demonstrated by CryEngine and Unity (not to mention the amazing in-game tools for user generated content-focused titles like *LittleBigPlanet*).

We wanted to maintain a very playful, flexible development process and making level creation as dynamic as possible was key to that. Being able to pause the game, make a couple of tweaks and try out the results immediately has some great benefits -- it encourages rapid iteration and can potentially make the creation process much more efficient. Being a small team, we really wanted to ensure that we were maximizing the resources that we have, and removing any time-consuming steps in the level building process (compiling/building assets) really helped us achieve this.



ID	Type
Terrain	35
Cloud	2
pellet_system	66
leaf_system	96
birds_system	113
wonder_pearls_system	68
foliage	6
SmallPlantsSystem	53
RocksSystem	52
TreeCluster	7
PollutionPatch	41
PickupsSystem	57
WindArea_1	94
CablesSystem	82
RailwayTracks	85
DecalsSystem	59
YorubaPropsSystem	60
XXXPropsSystem	60
Flames	47
CollisionRegion_61	87
FlagsSystem	45
FenceSystem	40
rock_sphere_2	1
CreatureBall_1	24
rock_sphere_b_13	1
flag_post_1_9	1
CollisionRegion_5	87
EndLevelLocationMa...	75

Total Objects 1354

Time 0 Pearls 0

(Click for full size)

While the tools started off working pretty well for sculpting terrain, placing and manipulating objects, and writing simple scripts, we started to encounter issues as the scale and complexity of the levels increased. As we started to add additional features and systems to the game, the simple editor began to creak somewhat, and this definitely cost us in terms of efficiency due to an increase in stability issues and some exotic workflows that we had to develop for some of the game's more complex features.

Our main takeaway here is similar to our takeaway from tech -- to consider if the benefits of rolling your own tools outweigh the resource cost for your project. In retrospect, there would have been some real advantages in going for an engine/toolset like Unity. While it would have incurred a considerable up-front cost, the amount of time it took us to develop the tools and engine were very significant and we would have seen a major boost in productivity if we'd not rolled our own.

As mentioned previously, an additional advantage of using a popular engine/toolset is familiarity when people join the team. Whereas our tools and engine were bespoke and somewhat eccentric (and required several days to get comfortable with them), by using an engine like Unity we could have hired designers/programmers on short contracts towards the end of the project that could have been effective almost immediately.

5. Core Audience

Coming from an iPhone/casual background with our previous games, our focus has been developing games that can be enjoyed by a really broad audience. This focus is something that we really wanted to bring to our first console project and from the very beginning of *Okabu*'s development, we designed the controls to be as pick-up-and-play as possible.

My brother has been playing games with my nephew since he was four years old and this was a big inspiration during preproduction. Right from the start, we wanted to create a game that could be enjoyed by gamers with a wide range of abilities and experience. We also wanted to build upon the art style employed in our previous games, creating a bright, warm, charming world begging to be explored and appealing to all.

While we feel we catered well to the casual audience, we definitely underestimated the proportion of core gamers on downloadable console platforms (overwhelmingly male 18 to 25 and primarily interested in core experiences). The pace of the early sections of the game didn't cater well to core players. In retrospect, we definitely should have offered an alternative play mode that provided additional challenge and less hand-holding during the early sections of the game.

Conclusion

Although there are still some parts of the game that we'd love to have had more time to refine, we're amazingly proud of what we managed to create with our small team. While there were some significant challenges along the way, *Okabu* gave us a wealth of experience. From the publishing side of the business, to the production process for an expansive narrative-led 3D game, there's so much that we've learned during the game's development.

That experience is going to be invaluable to us -- whether targeting mobile and tablet gaming or creating a follow-up to *Okabu* on PS3, the lessons learned and processes adopted put us in a really strong position to move forward in an exciting and ever changing marketplace.

The work that we've done to build the *Okabu* IP will also serve as a solid foundation for our future titles. The war chest of assets, characters, stories and landscapes will allow us to create new properties in the *Okabu* universe quickly and effectively. The hard work of the past couple of years has really paid off.

Return to the full version of this article