What went right

gamasutra.com/view/news/237657/Postmortem_Failbetter_Games_Sunless_Sea.php

Alexis Kennedy is co-founder and Chief Narrative Officer of Failbetter Games. He was creative director and lead writer on* Sunless Sea*, and mostly remembered to water the office plants.

Sunless Sea was born out of both creative restlessness and financial need.

We had a much-loved but unprofitable browser game with a cult franchise that had never quite made enough money to support the studio. So we leveraged our browser game fan-base into a Kickstarter; leveraged the Kickstarter into a successful Early Access campaign; graduated from Early Access after seven months.

As each stage went well, our budgets and ambition grew a little. Soup to nuts, the game took about fifteen months. The core team was three - writer, artist, Unity dev - but around a dozen people worked on the game at one point or another.

It's a very Failbetter game: distinctive, untidy, dark, daft, meticulously detailed, with an awful lot of carefully written words. We've been around for five years, and we've established our voice and our vision. But *Sunless Sea* was also our first actual videogame, after years of making only browser games.

So although it wasn't our first rodeo, it was… um… our first fish rodeo. I don't really know how rodeos work. We don't have them in the UK. This metaphor's not working out. So let's just say it was at the same time both well-explored territory, and a gigantic learning experience.

These two things - our patchwork of experience and inexperience, and our careful, eat-only-what-you-kill, gradually growing approach to scope - were at the heart of everything that went both right and wrong during development.

1. Thematic and mechanical focus from Day 1

In the Kickstarter pitch, I described *Sunless Sea* as a game of 'exploration, survival and loneliness; a game of light and dark.'

From that point on, we wrote, drew, recorded or made everything with that in mind. The resource management aspect emphasised hunger and terror. We iterated over the art direction, trying to find ways to convey the sense of vast subterranean space with islands of light, and feed that into the systems which rewarded the player for discovering new land.

We filled every scrap of land with stories - so that you were exploring destinations, not just rock formations. We chose the theme of home, and nostalgia and homesickness (saudade, hiraeth), as a way to emphasise loneliness. We found ways to surface that theme again and again in our writing. We shaped the game to encourage you to return again and again to Fallen London - your home port - and to feel relief when you saw London's lights in the distance. We found a (very talented) composer who'd worked on IMAX documentaries about the deep sea and the Hubble telescope.

Every single review of *Sunless Sea* has picked up on this. Every thread we see in a new forum emphasises the atmosphere, the mood, the sense of peril. Even when some of our creative decisions have made the game a more niche experience, we feel that we've achieved what we set out to do, and the players who click with the game really click with the game.

Ship speed is a good example. *Sunless Sea* is a stately game. You could reasonably call it a slow game. But we've resisted speeding up the ship, because it would reduce the tension, the sense of space and distance, and the menace of the dark. I think it's quite possible that if the ship was 50% faster, the game would be more fun and less grindy - but I also think there's an invisible line we'd cross, somewhere before that 50%, where the atmosphere was diminished without anyone quite knowing why. If we hadn't had that iron creative focus from the beginning, I don't think we'd have held our nerve, and *Sunless Sea* would have ended up a zippier, slighter experience.

2. Transparency

We have a really good relationship with our very lovely community. Kickstarter, Greenlight, and an Early Access launch were all great for us, in terms of early feedback, buzz and cash-flow. Lots of indie projects use these approaches, and you've probably read retrospectives talking about how wonderful it is to develop in the open, how useful the community is. It's all true! But you've heard it all before - so I'll focus on a couple of things that worked out for us specifically.

The worst outcome for a Kickstarter is not that you might fail to reach the goal. It's to reach the goal, start building the game, and then run out of money. As I described here in our Kickstarter retrospective, the original design for the game needed us to raise about 90K (plus our war chest and other revenue). We thought we could expect to raise about 60K. So we cut about 30% out of the game.

In the end, we did manage to raise 90K. So we slipped our release date; and we also went for Early Access, which brought more money in and allowed us to increase scope further. In the end, our dev time went from 6 months to 15 months, but we had a regularly updated public roadmap at every step of the way. It mutated as we went, but we made sure that our public roadmap was the same one we were using internally. Everyone could see that we'd regularly released milestone updates, even when they'd run a bit late. Again and again we saw people ask on the Steam forum about whether it was worth buying into EA, and again and again we saw players respond that we seemed pretty reliable, and hey, here's the roadmap.

YOUR COMATOSE FERRET

It lies across your desk like a lumpy scarf.

Play with your Ferret

Attract its attention.

A weary roll of the eye

Play, that eye-roll says. Play. I played once, in the happy warrens of London, before you installed me in this floating coffin and made me eat dried food. Damn your play. It is an eloquent eye-roll.

- September 23rd, 1893: FULL POWER (Beware: your engines will occasionally explode.)
- September 23rd, 1893: FULL POWER (Beware: your engines will occasionally explode.)

So it helped with sales, it helped keep us realistic, and we now have the enviable reputation of having done Early Access right. That reputation will follow us into our next project (although Valve now discourages EA devs from providing a roadmap, in case those expectations aren't met). The key to this was keeping scope and expectations tight, and making sure that the player base could see our plans.

It did cost time and effort, and we did drop the ball occasionally. We had separate communities on Kickstarter, on our own site and on Steam - it was extra effort to keep them all in the loop, and on one occasion in particular we irritated our Kickstarter backers by neglecting to warn them of a major system change on Kickstarter (it'd been on the roadmap and other channels, but of course, why would they go looking?) Once you've set good communication expectations, people get understandably upset if you don't meet them. Early Access gave us enough funds to hire a good comms manager to make sure this angle was 100% covered.

That's irrelevant advice for AAA studios, and for indies who aren't fortunate enough to be able to afford a comms person, but it was a really big help for us, and I'm glad we did it. Extroverts: they're not just for marketing!

What could have gone better

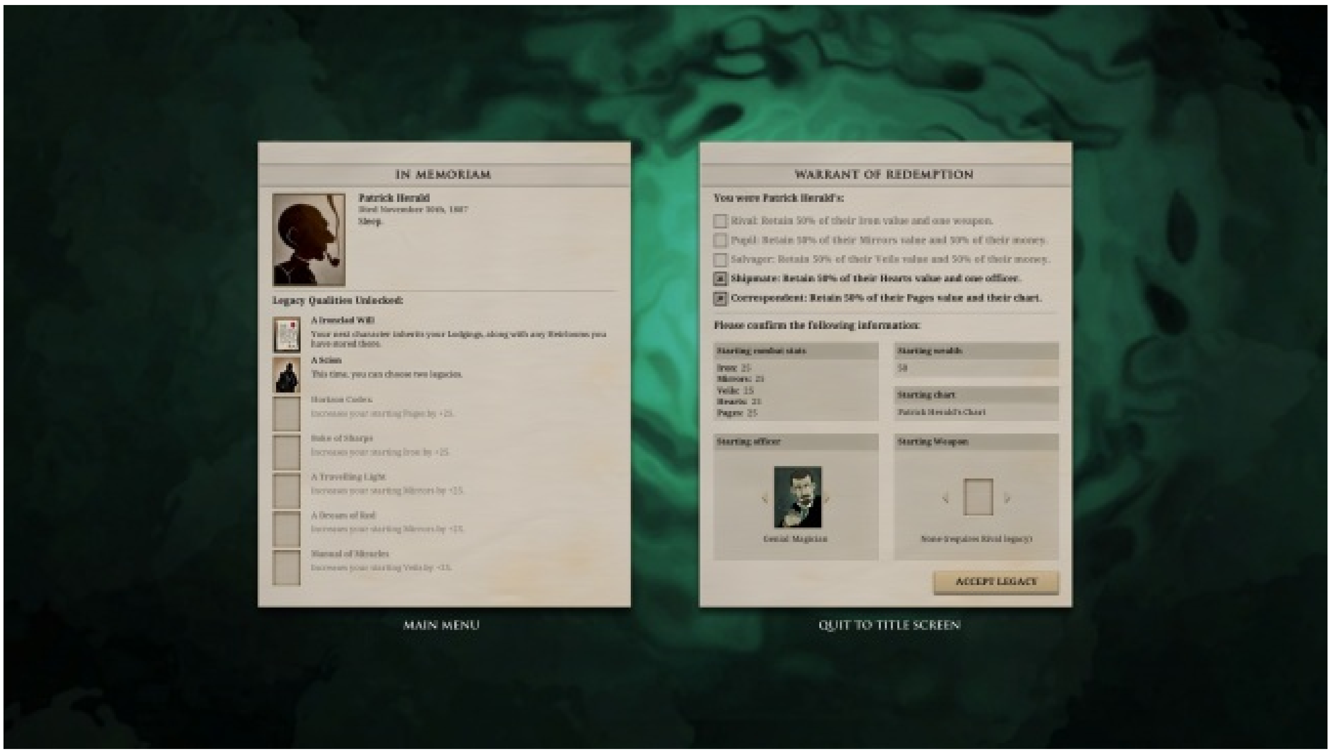
gamasutra.com/view/news/237657/Postmortem_Failbetter_Games_Sunless_Sea.php

3. Creative collaboration

Writers are egotists. We get to change a world with every sentence. *Fallen London*, the game which spawned *Sunless Sea*, had great illustrations, and had benefited from the work of a half-dozen other writers, but it was my baby first and foremost, and the text was the most important part of the game. When we started *Sunless Sea*, I was the sole writer, the sole designer and the project manager, and the writing led everything else.

Fortunately, as I started to realise how fallible I was, and how specialised my skills were, I surrendered creative autocracy. It became belatedly apparent that moving pictures and noises and game feel were, you know, important. Paul's art informed the writing more and more, and I began to realise that it was okay to change the lore, especially if it was something I'd invented the previous week.

The music and the audio fed back into the mood of the game, and stimulated other ideas. I spent less time adding tasks to a spreadsheet, and more time prototyping with our Unity devs and co-ordinating design sessions round a whiteboard. We brought in talented freelance writers - and Chris, our other Failbetter staff writer - to do guest spots. The structure of the world, with islands that could be stylistically or mechanically quite different one from another, lent itself unusually well to a kind of anthology structure.



IN MEMORIAM

Patrick Herald
Died November 30th, 1887
Sleep.

Legacy Qualities Unlocked:

A Ironclad Will
Your next character inherits your Lodgings, along with any Heirlooms you have stored there.

A Scion
This time, you can choose two legacies.

Horizon Codex
Increases your starting Pages by +25.

Bake of Sharps
Increases your starting Iron by +25.

A Travelling Light
Increases your starting Mirrors by +25.

A Dream of Red
Increases your starting Mirrors by +25.

Manual of Miracles
Increases your starting Veils by +25.

MAIN MENU

WARRANT OF REDEMPTION

You were Patrick Herald's:

☐ Rival: Retain 50% of their Iron value and one weapon.
☐ Pupil: Retain 50% of their Mirrors value and 50% of their money.
☐ Salvager: Retain 50% of their Veils value and 50% of their money.
☒ **Shipmate: Retain 50% of their Hearts value and one officer.**
☒ **Correspondent: Retain 50% of their Pages value and their chart.**

Please confirm the following information:

Starting combat stats
Iron: 25
Mirrors: 25
Veils: 25
Hearts: 25
Pages: 25

Starting wealth
58

Starting chart
Patrick Herald's Chart

Starting officer
Genial Magician

Starting Weapon
None (requires Rival legacy)

ACCEPT LEGACY

QUIT TO TITLE SCREEN

I was properly astonished to find out how much fun this was, and how invigorating it was to surrender some control to people who could do things I was incapable of. Failbetter has become a genuinely collaborative studio, and *Sunless Sea* unmistakably carries everyone's fingerprints. In hindsight it sounds obvious - but it was a distinct change in our evolving company culture. And it wouldn't have been possible without #1, the strong shared vision that we all bought into.

4. Use of existing story tech

We've been building story-centric browser games for five years, and we used the toolset that we'd built for those - StoryNexus, a web-based CMS / story engine. We ported some of the components into the Unity code which could interpret a human-ish-readable JSON download from the CMS. This meant that I - and other writers who'd used StoryNexus before - could work quickly to produce quite complex content and iterate on it; and because it was data that the local game client pulled from the web, we could share work and iterate quickly.

The lesson for us here was twofold. First, even when a tool's imperfect, there's a big win from using it when the content creators are comfortable with it and the tech team understand it well enough to tune it. Secondly, we're already thinking about the possible projects-after-next, and gently biasing our decisions on how to develop the toolkit in the direction of what might be useful in 2016.

Spoilers: there were some downsides, too, but we'll get to that.

5. No crunch

As we came closer to launch, it became obvious that we were running hot, and people were getting tired. This is what I posted on the company Slack:

"*Sunless* is in a good place. The hay is in the barn. Essential content is nearly done, essential art is done, essential tech is done. We have more than a week to go [before we lock the code], and the risk of f*cking up because we're tired outweighs the benefit of making the game 2% better. Overtime is for emergencies. We'll be around on Friday night and Saturday in launch week. Work your 7 hours with a proper lunch, go home, sleep 7 hours. If you have more work than you can do in the time, talk to me and we will agree what to cut."

I know that not everyone is lucky enough to be able to say this. We're a small studio with low burn and no third parties to answer to - I know that many intelligent and compassionate developers are under very different pressures. But I've been in tech for more than fifteen years, and again and again I've seen that the productivity gains from overtime dissipate very quickly. Of course people who are heart-and-soul committed to a shared project will put in extra hours even without management pressure - they love their jobs! - but managers and producers can help them keep a sense of perspective.

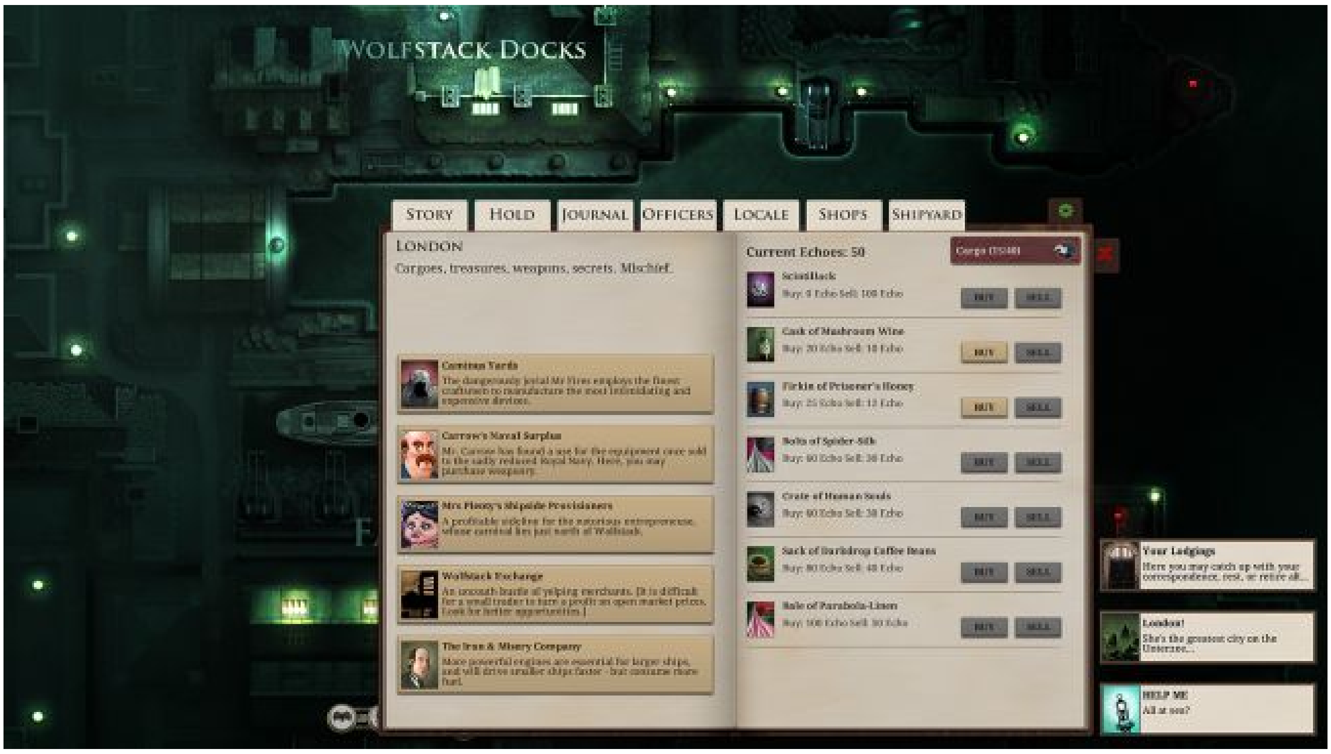
So I'm dropping my vote in the jar that says crunch is bad. We ran hot in the last few months of *Sunless Sea*, but we didn't crunch, and we made a good game. It's not a perfect game, but it wouldn't be perfect if we'd crunched, either. And our lead Unity dev told me yesterday, a little surprised, that - less than two weeks after launch - he was really enthusiastic about working on our first expansion pack.

1. Roguelike or CRPG?

I said above that we nailed the thematic and mechanical focus of the game. That's true, but there was one area where we vacillated badly. Was *Sunless Sea* a CRPG or a roguelike? My answer to this, for most of the development time, was 'Yes!' and that was our biggest mistake - specifically, I have to say, my mistake.

Sunless Sea has the punishing early difficulty, permadeath, and some of the randomness of a roguelike. It has the rich story, predictable early-game, slow pace, and flat late-game difficulty arc of a CRPG. We even have a default roguelike permadeath save system and an optional more CRPG-like save system.

This is the source of very nearly everything that's wrong with the game. It's left us with an uneven difficulty curve. It's irritated players who don't relish repeating early story. It exacerbates other issues of player confusion - *Sunless Sea* is already an unusual game that requires captains to make chancy journeys of exploration to survive, and teaching players to do that rather than sticking to home and grinding is already difficult. We've seen a number of reviews that say the game is 'almost a classic' and falls short on exactly this point.



LONDON
Cargoes, treasures, weapons, secrets. Mischief.

Cantigua Yards
The dangerously jovial Mr Fires employs the finest craftsmen to manufacture the most intimidating and expensive devices.

Carrow's Naval Surplus
Mr. Carrow has found a use for the equipment once sold to the sadly reduced Royal Navy. Here, you may purchase weaponry.

Mrs Plenty's Shipside Provisioners
A profitable sideline for the notorious entrepreneuse, whose carnival lies just north of Wolfstack.

Wolfstack Exchange
An uncouth bustle of yelping merchants. [It is difficult for a small trader to turn a profit on open market prices. Look for better opportunities.]

The Iron & Misery Company
More powerful engines are essential for larger ships, and will drive smaller ships faster - but consume more fuel.

Current Echoes: 50

Cargo (7/240)

Scintillack	Buy: 4 Echo Sell: 100 Echo
Cask of Mushroom Wine	Buy: 20 Echo Sell: 10 Echo
Firkin of Prisoner's Honey	Buy: 25 Echo Sell: 12 Echo
Bolts of Spider-Silk	Buy: 60 Echo Sell: 30 Echo
Crate of Human Souls	Buy: 60 Echo Sell: 30 Echo
Sack of Darkdrop Coffee Beans	Buy: 80 Echo Sell: 40 Echo
Bale of Parabola-Linen	Buy: 100 Echo Sell: 50 Echo

Your Lodgings
Here you may catch up with your correspondence, rest, or retire all...

London!
She's the greatest city on the Unterzee...

HELP ME
All at sea?

I was aware it was an issue from the start, but I didn't realise for too long how much of an issue it was. Our solutions - in particular, we have a flavourful legacy system that helps fuse the two together - help a lot, but you can see the joins. And the next issue made it worse...

2. Commitment to backwards compatible saved-games

We made this commitment very early on. Every saved game from the version immediately before Early Access would be playable right up until final release. And it is! - even if you have a super-early saved-game map with island names like GENERIC TILE! It seemed an appropriately ethical promise to make to early purchasers, and also a sane one. We've got players with years-old game state in Fallen London, and we're used to changing things without breaking their experience. How hard could it be?

Well, *Fallen London* has no permadeath. So this unwise decision caromed directly into the uneasy seam between CRPG and roguelike - our 'Wrong #1' - and nearly tore the game in half lengthways. Many of our veteran players had saves which were months old - long chains of captains who'd inherited previous legacies, well-equipped ships with stacks of cash. The most vociferous and enthusiastic forum regulars were veterans who rarely saw or cared about the early-game experience… and so the early-game experience stayed rawer and grindier than it should have been.

We're not daft - we knew this was the case, and we tried to correct for it - but it's very hard to correct for skewed feedback by blind reckoning. The things that players say, day in, day out, bubble to the top of the priority list. By allowing our players to stay in the mid-game, we ensured that we balanced for the mid-game.

On top of that… backwards compatibility was just much, much more work than we'd expected. And the work always popped up at the worst possible times, as a last-minute bug pre-patch or a support burden post-patch; and it popped up inexplicably with weird legacy issues from players who'd been out of the loop for three months and didn't think to tell us they were playing a three-month-old version with a six-month-old save.

You industry veterans who are shaking your heads fondly and mouthing 'I coulda told you so'? Yes, FAIR ENOUGH. Give us a break! It was our first vidjagame.

3. Combat

Originally, when players encountered enemies at sea, we switched from a top-down sailing screen to an abstract, semi-turn-based, queued-action system in a separate interface. This was because:

(i) We knew a lot of our customers would be Fallen London browser-game players, and we didn't want to make the gameplay too twitchy.

(ii) We had built a good dozen browser-based projects, but this was our first PC game. We were afraid that adding combat physics and AI to the mix would take us way out of scope.

Unfortunately, the original implementation just wasn't very compelling. We worked on balance issues, and we had something interesting in the core mechanic - where you had to light up your enemy while avoiding being lit up in turn - but the fundamental problem was that the separate battle arena yanked the player out of our carefully layered atmosphere into something aggressively ludic.

We'd been inspired by *FTL*, but of course *FTL* has a deliberately minimal travel interface, so no clash (and Subset are a hard act to follow for game design and balance, anyway!) The majority of players disliked combat or were unmoved by it, and every Early Access review highlighted the combat as an issue.

We listened to feedback and announced that we were pushing back release to rework the combat completely. We did a bunch of design exercises and built a prototype that would have been more fun and more atmospheric, but the fundamental mismatch between sailing and combat remained.

So we had a Serious Team Meeting. We noted that we'd added a Unity dev with AI and animation skills to the team; that even the hardcore *Fallen London* players didn't love original combat; that we needed a fundamental change. We threw out the original mechanic completely, and took the extra dev time to make it as good as we could afford.

It was a huge improvement. Suddenly the survival and resource management mechanics, and the layout of the ocean and its hazards, fed directly into the combat. It was a big improvement, although…

- A minority of our players were happy with the original combat, or at least saw the potential in it, and complained. This included a few of our original Kickstarter backers - as I mentioned above, we didn't communicate the upcoming change to them specifically.
- The real-time combat wasn't nearly as polished as if we'd done it from the beginning. It still comes in for criticism in reviews, although the criticism is 'could be better' rather than 'dull and disruptive'.
- This played into problem #1, the roguelike / CRPG identity crisis. A lot of CRPGs have quite ordinary combat systems, because the changing power levels keep things interesting and often they're just intervals between the story. But in roguelikes, players expect more variety and challenge.

So should we have gone with real-time combat from the beginning? Eh, difficult counterfactuals. It was probably still the least worst decision to make at the start, when we thought we only had 6+ months dev time and no experienced Unity resources.

It became a less good decision as time passed and the scope expanded. If we'd changed our minds and switched to real-time combat earlier, we might have had a better treatment of combat in the final game - but we would have pushed out the release date further or meant we'd have done a more half-baked job without the experienced resource.

The ideal solution would have been to start with a more experienced team, but of course that wasn't an option. I'm confident, in any case, that changing course - though frightening and expensive - saved us from a much worse outcome.

Postmortem: Failbetter Games' Sunless Sea

4. Insufficient time building tools

This was a product of two things: our relative inexperience, and the way that scope gradually increased. The game relied a lot on JSON configuration files. Some of those were output from the CMS, some of them were just edited by hand. This was fine when we were experimenting, or aiming for an early launch, but as time wore on and the game grew, it became (to quote Liam, our lead dev) ‘like chopping down trees with your hands’. That only works in *Minecraft*.

We found some limited automation solutions to speed up these issues at the end, but we all agreed that we wished we’d spent more time on initial tooling up front. Again, a hard call to make when we thought we were going to be done in six months - our slowly expanding scope kept us honest, but it was a double-edged sword.

5. Use of existing story tech

I listed this as a ‘Right’, too. On balance, it was a big win! But we could have been more judicious.

First, we tried to repurpose several mechanisms from StoryNexus when really we should have just given up and changed them. The first version of combat relied on code that had originally been designed to deal with data-driven player-player social interactions. It made a lot of sense when we talked it through - entities affecting entities - but the number of special cases and limitations on design grew rapidly. By the time we reworked combat, we were ready just to hardcode most of this stuff. We’d have done much better

Second, we inherited a bunch of non-ideal UI issues. Our story engine was designed to work with a web UI, and has in any case remained largely unchanged for some years. It’s flexible and effective, but it can also look clunky and ugly. We imported that clunkiness directly in our game.

So for future games, we still want to use StoryNexus. But we’re going to take some time to think about how best it will work, and we’re going to look at more carefully partitioned architectural approaches. Of course, this bleeds over into Wrong #4 - if we’d spent more time on the tooling, we’d have suffered less later.

Journey’s End

It’s hard to avoid maritime metaphors when you’ve just made a game about the sea. You think in terms of unexplored water; setting course; unexpected rocks; helpful crew; charted waters; safe harbour. But let me try to fish something concrete out.

I said at the beginning that our patchwork of experience and inexperience, and the careful but flexible project scope, were the deciding factors. The thing that went best was our shared thematic vision - when we’d successfully internalised what we wanted to do, it solved a lot of trivial confusions and disagreements, and helped us find the real problems. But sometimes we lost sight of the genre conventions that we needed to respect - and then we couldn’t tell the temporary problems from the fundamental ones. So next time, we’re going to make sure that we count all the stars we’re steering by.

Data Box

- **Developer:** Failbetter Games
- **Publisher:** Self-Published
- **Release Date:** 6th February 2015
- **Platforms:** Windows, Mac, unsupported Linux beta

- **Number of Developers:** 3-6 at different project stages
- **Length of Development:** 15 months
- **Budget:** 250k GBP
- **Lines of Code:** 20K +
- **Development Tools:** Unity, StoryNexus
- **Total words in game:** 250,000
- **Total illustrations:** 415
- **Word: Picture ratio:** 600
- **Tattoos Accrued:** 2
- **Venetian Carnival Masks in Office:** 10
- **Bat Skeletons in Office:** 1