There Isn't an App for That ™

Justin Nightingale

I've lost faith in a particular type of app; I'll call them "content you add" apps. I was in denial for many years but after progressing gracefully through denial, anger, bargaining and depression on numerous occasions I now find myself in a state of acceptance that data created or purchased within an app (that is unique to that app) is a consumable with a finite lifespan.

The cycle goes like this: find a great app, create/download content and then something happens that inhibits usability or worse, prevents me from ever seeing my data again often because a major OS upgrade from Apple renders the app buggy and unstable.

Apple's Bento database was a good example. Great software, easy to use, perfect for what I needed. After spending countless hours making detailed databases Apple suddenly said they were halting future development. In other words, your hard work just evaporated before your very eyes. it reminded of Marty McFly in Back to the Future as he is being erased from time.

Book and magazine purchasing apps are another example. As much as I enjoy them (or devices like the Kindle), what am I actually purchasing? Can I be confident that I'll be able to read that (often encrypted) content when that app or device no longer exists? This is a valid concern in all areas of modern digital consumption: films, music, audio books, games and software in general.

Not being prepared to make the same gamble with my data again, I decided to get back to basics by avoiding proprietary standards and relying rather on field proven, future proof data storage formats that are device and platform independent: text files, basic spreadsheets and the ubiquitous PDF.

Books

In a world where schools are full of textbooks, EIKEN and iPad based learning, I feel I almost have to lower my voice when I say that I'm coming around to reading aloud good old story books in the class more and more. More than anything, books don't care about ability or test scores. They're a leveller.

Seeing as I have to use the overhead projector to share content I started looking into apps that allowed me to buy books as an IAP (In App Purchase) but for reasons mentioned I wanted a more future proof way to build a library – something between real books and IAP content. Time to write an app.

The Problem

Yes, I know there are countless PDF viewing apps out there but have you ever tried showing a PDF on an overhead display to a class? You would think this would be a perfectly reasonable usage scenario for an app to address but in my experience with every app I've tried they are fine for the user but not for the 3rd person. If you're just swiping left to right there's no problem but trying to draw attention to something or eliciting.. a long stick maybe?

So, I decided to make my own app with the following features:

1. Smooth Zooming

40 children watching one screen means zooming. Pinch and zoom will do the job but the slight jerkiness you may experience when you view the iPad screen is jarringly magnified exponentially on a big screen. I want to be able to touch a point on the screen and see the software gently zoom in for me while a long touch would zoom me out. It makes a big difference as my finger dexterity is no longer a factor in the viewing enjoyment. The medium that content is viewed with should always be invisible.

2. 赤シート

Covering words with a clear "Red sheet" is a popular way to memorise textbook content. In the same vane, I want a "white sheet" I can drag around the screen hiding key content in order to elicit discourse from the students before the page as a whole is revealed. Of particular interest to me is displaying story books with pictures. If I am able to hide the picture first, I can encourage everyone to focus on the text – impossible when the "rabbit in headlights" picture is showing.

3. Book Cover Browser

Instead of just a list of filenames with an icon, I want to have my screen full of book covers only. In other words, like a real library. Touch a book cover and it opens.

4. Temporary Writing

I have no interest in editing PDFs but writing on the screen to bring attention to a letter, a word or a picture is a must. A laser pointer like red dot that follows my finger movements on the iPad screen would be half way there but it is too transient. I need to be able to write a word or underline something and let it "hang" before disappearing after a predetermined amount of time – perhaps 5 seconds, just enough time for me to adequately comment or elicit a response. And anyway, Laser dots make me think of funny cat videos on Youtube.

5. Audio Syncing

This one is very specific to my current needs. I want to be able to stop/start background audio while browsing a PDF. Many of the story books I use have accompanying audio giving 2D PDFs an extra dimension.

Not a Document

PDFs are an app not a document. When you click on the icon and open a PDF you are in reality starting an app replete with its own interpretive programming language that instructs the viewing software to place textual and graphical elements in a certain way on the screen. There is even loop and condition program flow ("if page size changes, move this graphic to there").

Based on the Postscript language that allows printers to print exactly the same page regardless of hardware and very similar to the HTML premise, pages are not necessarily stored in a human readable format but in reams of vector co–ordinates and instructions with image data and fonts thrown into the mix to create one, self–contained presentation app.

Building

My first step was to scan hundreds of story books over 4 months of lunchtimes into PDFs. As I was unable to cannibalise the books and throw them into our scanning machine, I had to use the photocopier scanning function and turn page by page. After a few weeks, I approached scanning zen. It was something to behold. At one point I wasn't even there.

The next stage was to start building the app using Xcode. Adding the aforementioned features was just a question of putting in the hours but to actually display a PDF page was surprisingly very, very tricky and took many weeks of research and trial and error to make an engine that could quickly

scroll large PDFs (>30Mb, 600dpi). I've been programming 35 years but that ranked as one of the toughest challenges I've come across. In reality, there are already excellent PDF engines freely available for download from talented and generous developers that most sensible people implement into their software instead of building there own but I had difficulties adding my wishlist of features to them.

In short, the easiest way to make it work well was to convert each page to a JPG file in the background. Deep zoom operations would continue to reference the PDF (vector) to maintain clarity but all other operations would reference the lightweight JPG file (bitmap) instead with the transition between the two rendered invisible to the user. Whilst reading a page the app will continue to background convert all remaining pages or as much as the device's memory will allow. The app is kept constantly busy in the background to give the user the illusion of a swift and lag–free user experience in the foreground.

In Use

At the start of this project I contacted the UK publishers of the books I use to double–check the legality of scanning books and displaying them in a class. To be honest I was surprised at the laissez–faire response which could be summarised as, "Yes, you can show your own scans as long as the students don't have their own copy." Unsurprisingly, Japanese publishers were a little more Japanese. So, UK books only.

It is still early days using the app in the class. As I spend more time with it, adding features and making it a more effective and invisible medium, I hope to be releasing "PDF Sensei" in the App Store spring, 2019.

(ジャスティン・ナイティンゲール　聖学院大学総合研究所特任講師)