

# appville

# CONTENTS

05

## [From 0 - 12,000 Downloads: The Story of the Appville Newsstand Magazine](#)

TOE ABAYOMI, FOUNDER, APP DESIGN  
VAULT

06-07

## [Developing a Mac App Without App Store Distribution](#)

OISIN PRENDIVILLE AND PADRAIG  
KENNEDY, TOKEN APP

08

## [A Developer's Journey: From Idea To App Store](#)

PAUL SURETTE, MY CYCLE CLASS

09-10

## [Video App Marketing: Outsourcing the Production of your Marketing Videos](#)

MANUEL ZAMORA, FOUNDER,  
DOWNSHIFT INTERACTIVE

# A DEVELOPER'S JOURNEY: FROM IDEA TO APP STORE

- PAUL SURETTE

**Editor's Note: This story is not about how a developer made millions on the App store. I am sure you see a lot of that elsewhere. This is about a Father, Husband who is also a**

I soon realized that “Problem A” can be resolved with the proper View Controller layout and graphics. It was amazing, like a light bulb went off. I can solve a lot of these little issues I’m having by re-designing my user interface. I finally got to the point of no return and finished my app. I liked it, I didn’t love it. I wanted to unleash it to the masses to see if there is any interest in it. Once the app hit the App Store, I always

**cyclist, who had an idea for an app and how he went about developing it and getting it live on the App store.**

**I hope this inspires you to take your ideas and execute on them. Nothing will happen unless you do.**

Paul: I'm an avid cyclist. That is, I try to get out on the road when I can, but I'm fair weathered as well. I don't like the cold when riding. Living in Ontario, Canada, my riding season ends around October. I'm also a certified indoor cycle instructor. When I would teach my classes, I would use my iPhone and create a set list for each class. In the lyrics section, I would make notes of motivational cues, tips, etc. There they would be, when a song played, the information I entered was right in front of me.

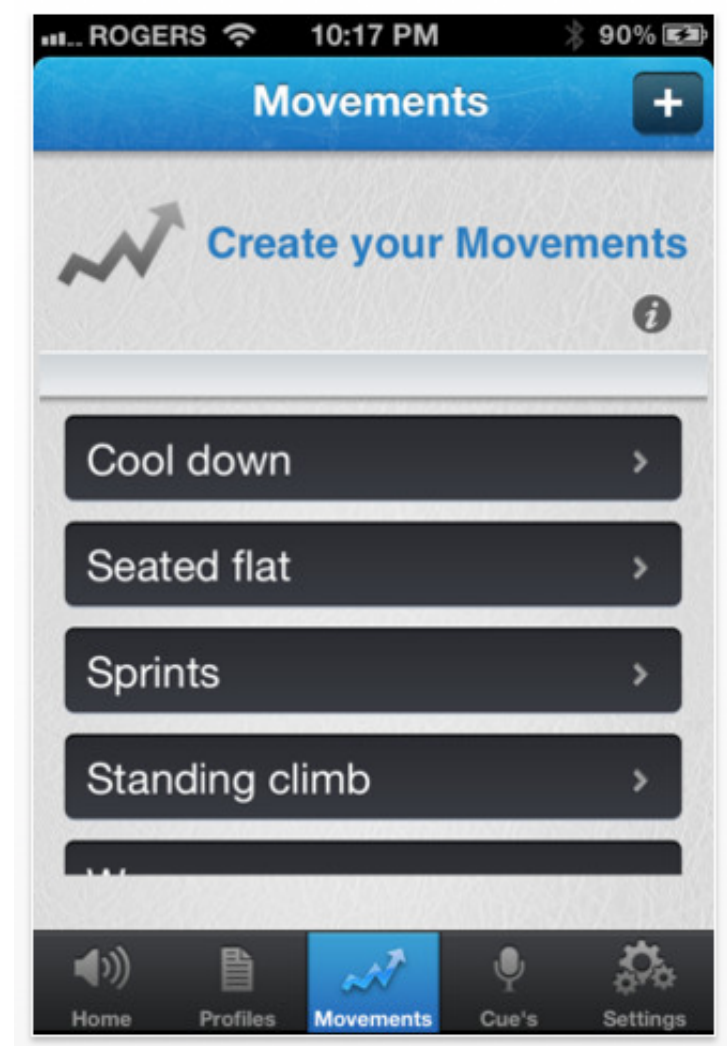
However, I also needed a piece of paper that had my class profile. That is, a list of movements and intervals that I would instruct in my class. I was also never good at remembering my timing on the analog clock in the room. I would time certain interval's using the second hand, but as soon as I looked elsewhere, I'd lose my spot and my time. I thought, there had to be a better way.

I searched the App Store and didn't find much, the ones I saw didn't fit my needs,

told my friends, I'll be happy if I sell one copy, to a customer, not friends or family. And I did. I sold one, then ten, and retirement here I come.

Seriously, I was very happy to come from knowing nothing about developing apps for iOS to have an app for sale. It was a very long journey, and I learned a lot.

Here is the GUI for v1.0 of myCycleClass that hit the App Store.



*Screenshot of v1.0*

This is the main screen. There are five “screens” on the bottom, that you swipe, left to right to get to them. However, in practice this proved to be a challenge,

and to be quite frank, they were ugly and very confusing to use. This began the birth of myCycleClass.

I never, ever, programmed an app for iOS, let alone anything in Objective-C. I didn't even know what that was. I heard of C, C++, C#, etc. My background is a computer programmer, but it's not what my real job entails. I worked on Visual Basic, Cobol and very little C. With a strong programming foundation, I set out to try and create this app.

Once I signed up, paid my \$99 to Apple for a developer "license" I began and downloaded XCode. The intimidation level set in as soon as I opened XCode and began to wonder how to do something as simply and the obligatory "Hello World" program. It was clear I was in over my head, and that maybe this wasn't a good idea. But I didn't give up. I purchased a few books, beginners' guides to iOS 5, and set out working through them. It wasn't long for my love of programming to come back and the passion re-unite.

I read and read, watched as many tutorials as I could, and set out to begin this task of myCycleClass. It was clear that I needed a cool interface. I'm also a professional photographer, and know Photoshop pretty well when it comes to editing pictures, creating buttons, backgrounds, etc. The list of things that I needed to do kept piling up and I'm only one person. Principal programmer, graphic designer, and marketer were the roles that began to show themselves during this process.

I decided I would leave the graphics to the

especially if your iPhone/iPod has drops of sweat on it, or your hands are sweaty. Touch screen doesn't work very well when wet. So I had to re-design.

Like I said earlier, I like the above design, it's clean and simple, but it's not practical. I know I could do better. I needed to love the design to truly be proud of it. I also needed to rework the bottom "Swiping" section.

MyCycleClass v2 was born. At first I thought it would be a minor overhaul, but it wasn't. Every piece of code was looked at again, and after doing a major re-design, a lot of code was no longer needed, as good design replaced code. App Design Vault kept releasing amazing templates that all seemed to work for me. But it wasn't until [Radiojive](#) was released, that I finally found the missing piece. Since myCycleClass is primarily a music player, I needed amazing media controls. Radiojive finally had them. I began to re-work and re-think the UI. It had to be better.

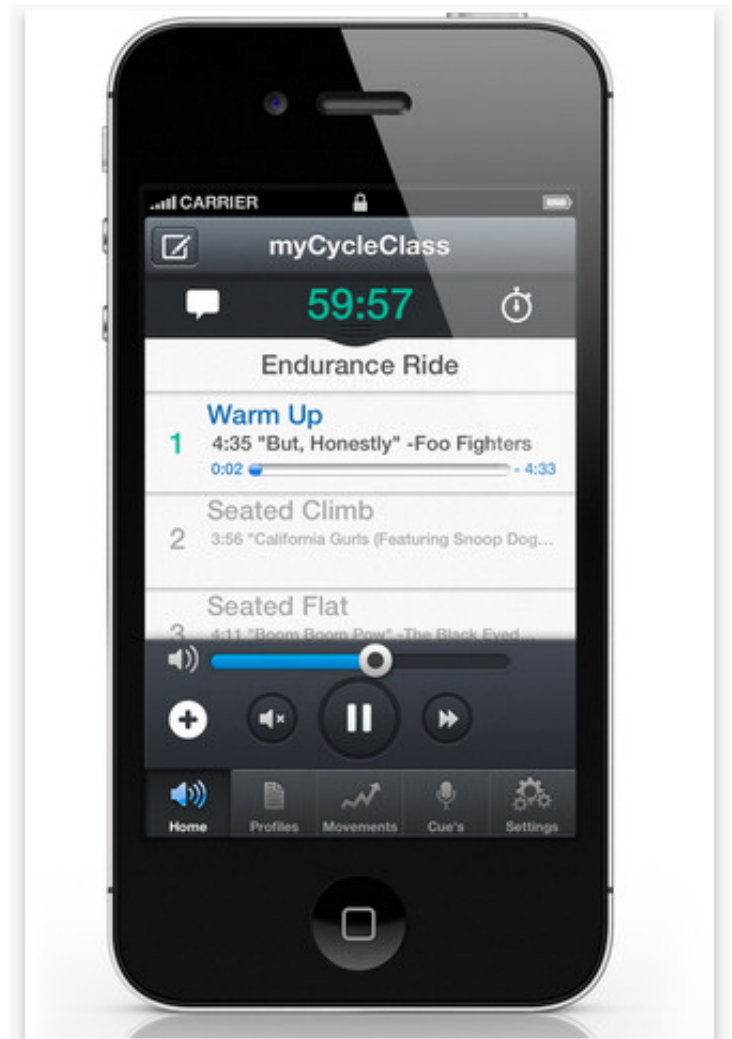
After another few months of coding and resigning, I finally released v2 of myCycleClass. This design I love, this is what I'm proud of.

end. Worry about the code, was my mission. I created my app using the default Table Views and Buttons and I would work on one piece of the puzzle at a time. As I did more work, it was clear, I needed to store “things” so now I had to learn CoreData. Again graphics were pushed to the bottom of the pile.

Once I buttoned down CoreData, TableViews, ViewControllers, etc. I began to work on the graphics. I tried to use Photoshop to create something simple, but it was very frustrating, and off I went to research more. I found a few websites that offered “free” graphics, but they were often too small, or wasn't the look I was looking for, and also felt like a compromise. I wanted this to be a professional looking app, I wanted it to stand out in the App Store as best as it could.

Especially for a single developer with limited time to code and do graphic design. As a father of two young boys, and a very supportive wife, my time spent coding was on my commute to work, and the odd time at home after the kiddies are in bed, for an hour or two. It's hard to gain any momentum only working on your app a few hours a day, and not contiguous hours at that.

I would say [myCycleClass](#) was in development for close to a year from the time I signed up to the time it launched in the App Store. As the time piled up and more and more code was written, I would always stop and look into the graphics. I finally found App Design Vault. At first it appeared to be my savior, my saving grace, the final piece of the puzzle. I bought a



*Screenshot of v2.0*

Radiojive has truly opened up this app, makes it easier to us, it's more fun to use, and it's nice to look at.

When I designed my second app myGasTank, I did the GUI and layout first, before anything was written, and it made the process so much better.

The lesson I learned from a beginner are simple: Keep the UI simple, and clean. Resolving coding issues is possible with good UI. Never, ever use a UITableViewController in your app, using UITableView in a View Controller offers much better flexibility. Pay attention to your graphics to make your app stand out.

template that I thought I wanted my app to look like.

Once I bought my first template, the level of intimidation rose again. What do I do now with these? Where do they go, how do I make my app pretty like the sample? More and more work went into figure this out. How hard can it be? Thankfully, there was sample code supplied, and this helped. Finally I have graphics. What a relief! Then it dawned on me; the code wasn't working with the graphics. I was trying to program everything and I didn't think how something can be accomplished with a proper flow and proper graphics.

[You can download MyCycleClass on the App Store here](#)

# D EVELOPING A MAC APP WITHOUT APP STORE DISTRIBUTION

- OISIN PRENDIVILLE & PADRAIG KENNEDY

## Where The Idea Came From

**TOPE:** How did you come about the idea? Was it through personal frustration on generating and sharing tokens? Complaints from other developers etc.

**OISIN:** Padraig and I had worked on a few web projects together in the past and were keen to collaborate on a native Mac or iOS app. Earlier in the year we started brainstorming project ideas and one of them was to build a native app for iTunes Connect since we were both sick of dealing with the web interface.

At first we thought scraping iTunes Connect would be too much of a pain so we brushed it aside but the



Editor's Note: The Tokens App was launched recently and there was so much buzz, I had to get in touch with Padraig and Oisin to come and tell their story.

This is a classic case of an app solving a painful problem and I believe that contributed to its launch success. If you are an app developer, you know how painful it is to distribute tokens and manage which ones have ben redeemed.

Best of all, Tokens is not available on the Mac App store. (Read the article to find out why). These mavericks are tackling the marketing, payment processing and everything a startup has to deal with!

We developers have a lot to learn from them and I'm sure this is a tip of the iceberg in their journey. I will definitely be watching them.

the idea kept coming back up in discussions. Eventually we decided the idea had legs, but that we should focus on one piece of core functionality rather than trying to do everything in one app.

Promo codes seemed like a natural fit since we both had experienced frustration when sending them out before. So we built Tokens to take all the hassle out of promo codes.

## **H**ow to Get Traction

**TOPE:** You seem to have received a lot of buzz during the launch of Tokens. How did you go about marketing the app and can you give us some tips on how to get the same buzz for our apps?

**OISIN :** We've been blown away by the amount of positive feedback we've gotten over the past couple of weeks. It's definitely reassuring; after months of development, launch time is both exciting and nerve-racking.

We knew we had focussed on the product and making sure we smoothed out all the edges but when it comes time for other people to start using your app, you're never quite sure how it's going to be received.

## How to Get Traction

In the weeks leading up to launch we started contacting press people and offering them a sneak peak at what was coming, we also contacted lots of our favourite developers in the hope that they'd be interested in Tokens, and nearly all of them were.

That was really reassuring. When we launched, posts started appearing the same day, and there was a good bit of buzz among developers on Twitter, which helped bring traffic to our site.

First impressions count, so we put a lot of work into a nice website which we continue to tweak. And after reading [Pitch Perfect](#) by Erica Sadun and Steven Sande we realised we really needed to put together a short demonstration video.

We spent a day putting together a demo in Screenflow. This helped give people a feel for how the app works and the benefits it offers.

One thing that really helped is that early on we clearly outlined the problem the app would solve and we focussed all development around that problem. From day one we said it had to be super easy to generate, share, and track promo codes. As well as keeping us focused during development, this also helped us pitch the app to bloggers and developers because we had a very clear statement of what Tokens does.

## Pricing

**TOPE: You are using a freemium pricing model. How did you decide the pricing model and what to include in each tier?**

## App Store Distribution

**TOPE: Is there a reason why you are not making the app available in the App store?**

OISIN: We scrape iTunes Connect, which is explicitly prohibited in the App Store review guidelines, so Tokens would never be accepted there. We always knew we would have to distribute this app ourselves and we're comfortable with that.

We do sign the app using Gatekeeper, which offers much of the same security for users as the App Store. The main disadvantage of not being on the App Store is that payment is more awkward. The main advantage is that we can issue updates immediately.

Padraig previously built some other non-app store apps so we were able to re-use that payment and licensing system.

## The inner workings of Tokens

**TOPE: How did you figure out how to use the API to generate and redeem tokens in iTunes? Is there documentation available somewhere?**

OISIN: For redeeming promo codes, the App Stores have a URL for the redeem page; you can pass a promo code to that URL to redeem it automatically. I'm not sure if Apple have ever publicly documented this URL. I think TapTapTap may have been the first to [document it on their blog](#) 3 years ago.

There is no official API for generating promo codes unfortunately, so we have to scrape the iTunes Connect web interface to generate the codes. This means pulling down the HTML in the background and searching it for the

OISIN: We always planned on offering some kind of trial mode and early on we started discussing what exactly the limits of the trial should be. We considered a more traditional time-based trial or the idea of limiting the number of promo codes you could generate for free.

In the end, we reasoned that if we let developers use Tokens for free to share codes for one app (of their choice), many of them would upgrade when they release their next one. There are some drawbacks to this approach too; we have all the usual costs in servers, support and maintenance.

Giving away a fully functioning version like this means that there's not much incentive to upgrade until you release your next app and that could be months or years away. Will they remember to come back to Tokens then? So we're working on new paid-only features to reward people who buy sooner.

It's surprisingly rewarding to watch the notifications come in telling you that your shared codes have been redeemed. I'd hate to go back to not knowing which codes are still valid and which are gone, so we hope that we'll spoil developers into needing the app.

data we want. It's not the most elegant approach but without an API, there's no other way.

The scraper is written in Objective C and all this happens locally on the user's machine so that account details never have to get passed to our server.

We also make some use of the [iTunes Search API](#) to get the URLs of app icons.

**Tokens for Mac is free to generate, share, and track unlimited promo codes for one app of your choice, after that it's \$29 to unlock all of your apps. Special for Appville readers, [use this link](#) to get \$10 off your purchase before 31st January 2013. [Download it here](#)**

**We're on Twitter as [@usetokens](#), [@padraig](#) and [@prendio2](#)**

**Version 1.1 launched this week and includes the ability to**

**import any previously generated promo codes so that you can get started using Tokens right away.**

---

# FROM 0 - 12,000 DOWNLOADS THE STORY OF THE APPVILLE NEWSSTAND MAGAZINE

- TOPE ABAYOMI

*In October 2012, I launched the Appville Magazine as a way to give developers tips on how to market their apps and understand the business*

***of the App store.***

***Fast forward to 3 months later, I believe the app has been successful (at least for my standards). Here is the story from beginning to the current day and how that came about.***

The idea is not to brag, hell no. But to show you what is possible and hopefully give you some inspiration for your own apps. The app is free and there are no in-app purchases so there no revenue numbers to report. Nonetheless, I have seen many free apps that don't manage to get a 100 downloads so the tips in this article will also help you if you had a paid app.

## **How I got the idea**

The idea came from wanting to have an app for iOS developers. An app that explains the marketing of an app. Today, there are a lot of blogs and publications out there that talk about development, tutorials and tools that developers can use to make apps. But you will not find a lot of sites where people talk about the marketing end of things, how to get your apps in the App Store, how to promote your app, how to get reviews and how to get your app downloaded a lot more.

## **Why Make a Newsstand App And not post it on a blog?**

I could have actually written a blog and posted these articles on it but I'm a developer, we don't make things easy :-). In the mantra of not making things easy, I decided to make a Newsstand app and actually have a magazine on the App Store! Moreover, I wanted to learn how to publish Newsstand apps anyway.

It is something that I've always wanted to work on because it is a very interesting technology. A lot of print publications are [packing up their print editions](#) and moving to digital editions. I wanted to find out how to use the technology that is supposed to save the magazine industry.

The Newsstand category is also a separate one in the App store so you are not competing with tons of other apps out there. So why not take advantage of that?

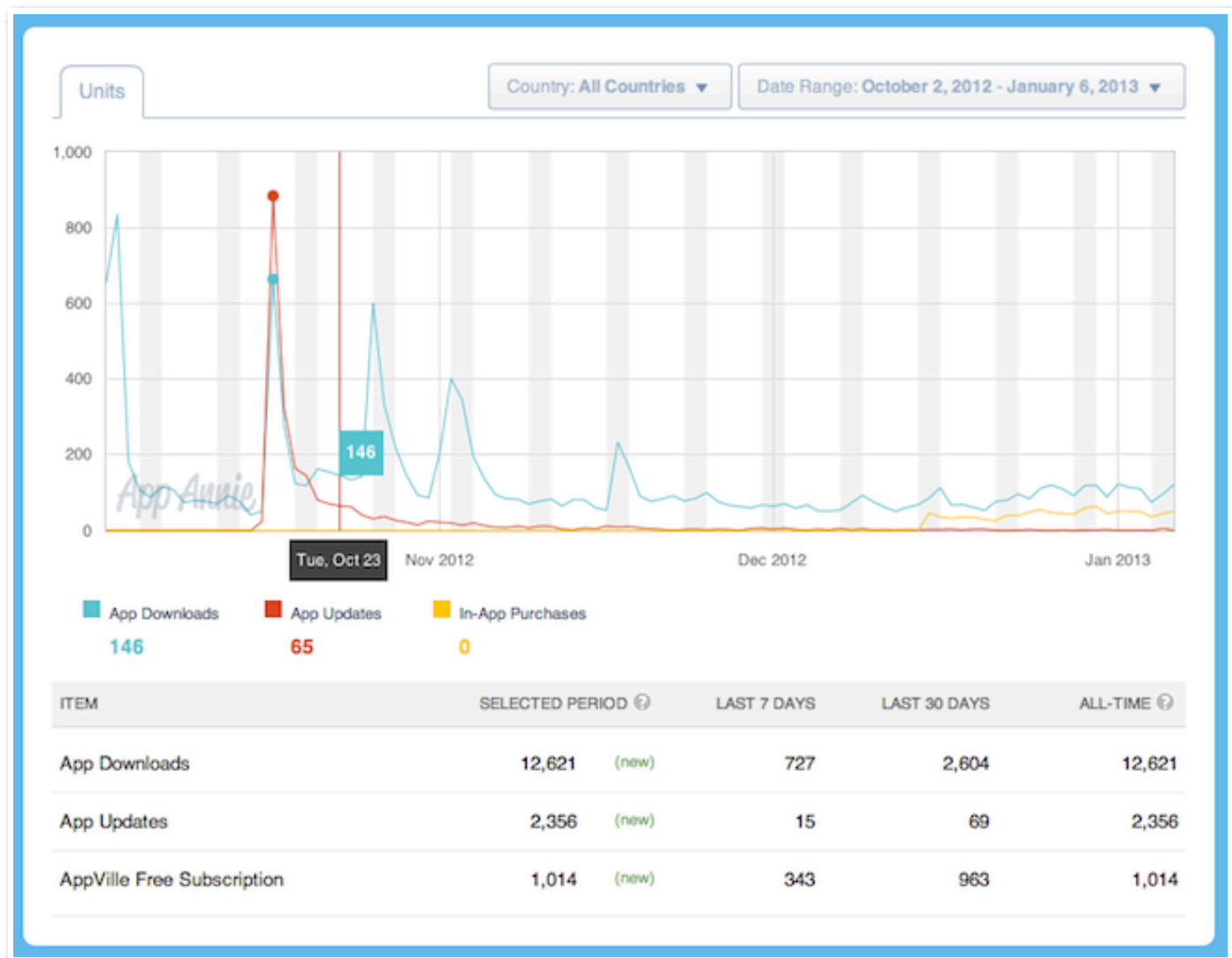
## **How many downloads has Appville had?**

Before I go ahead, let me talk about the download numbers and the results right now. I am sure you are interested in these numbers, so I won't hold anything back. It has been

featured in the “What’s hot” category in over 140 countries. That’s actually something which I’m very proud of.

## Number of Downloads

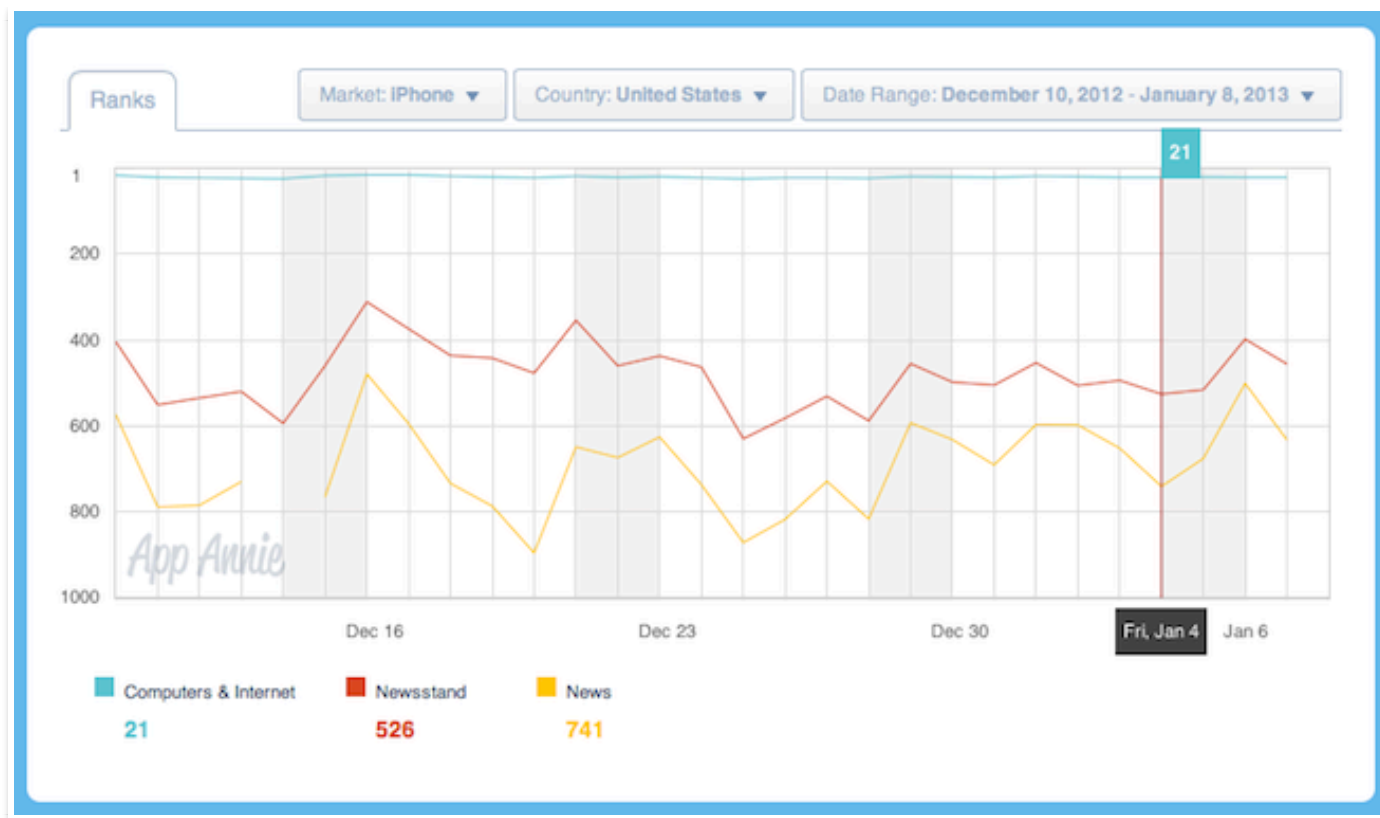
Up until now, Appville has had got over 12000 downloads and I am really happy about that. In the graph below, you will see a lot of peaks. These peaks actually coincide with when I have done some marketing.



I will go more in depth into the marketing later and then explain to you what I did to get the word out.

## Where Does Appville Rank?

Appville has been ranking in the top 50 apps in the Computers and Internet category in Newsstand. It hasn’t done so well in the total Newsstand category but in there I am competing with the likes of Vogue, GQ and Newsweek so I have my work cut out for me.



## How Many Reviews Has It Had?

Appville also has 81 reviews at the moment. A majority of them are 5 stars and I thank you if you are one of the people that left a review.



We need more though so please [tap here](#) to go the App store and leave a review. Thanks, I will appreciate it.

## The Technology



Appville, as you know by now, is built off of the Newsstand platform and it's kind of a complicated platform to develop on. It has its complicated things you have to deal with like issue management, background downloading, auto-renewing subscriptions. The most frustrating part is the submission process. It can be quite annoying the first time you do it.

These are the issues you can expect to encounter when developing Newsstand apps but for the same reasons, it is really a good platform. For example, no other app category can have background downloads. You also have the advantage of auto-renewing subscriptions.

Appville app is free right now but in essence it could have people paying \$1.99 a month to subscribe. This actually is a game changer when it comes to revenue models for your app.

The technology platform which I use to display the magazines pages is called the [Baker framework](#). I love the platform. When you get a hang of it, it is really easy to use.

I could have gone with the other platforms like the Urban Airship Newsstand publishing platform, Adobe Publishing platform, MagCast or MagPlus, but I decided to use the Baker framework because first of all, it's free and then it allows you to have your issues as HTML files instead of PDFs. The flexibility was important for me. It's as easy as publishing web pages (well-designed web pages) and having those in a magazine format.

## The Design

I practice what I preach, so of course I was going to use a Vault template for Appville. The app's frame that holds the list of issues was designed using the [Gridlocked design template](#).



Gridlocked was originally designed for a shopping or fashion app. But, like all the other templates on App Design Vault, you are only limited by your imagination. I decided it would be a good idea to adapt it for the Appville app.

The next thing I had to do was design the pages for the magazine. I decided to go look for HTML magazine template. I could have had our in-house designer design some pages for it but I decided to practice what I preach again.

I went to [GraphicRiver](#) and I got some magazine design templates from the site. The templates on Graphic River are done in Photoshop or Adobe InDesign. I needed HTML files though so I hired a freelancer on Odesk.com to convert the InDesign files to HTML/CSS that I could use for the app.

So now I was able to create a magazine content using the HTML files. All I had to do was substitute my content into the HTML file.

## How Issues Are Managed in the Magazine

The Baker framework gives you a framework to display pages for one magazine issue. But when you have more than one issue, you have to roll your own issue management.

I had to learn how to create Newsstand apps with subscriptions, how to get users to subscribe to the magazine. For that, I found a couple of tutorials useful. What got me

started was the [Newsstand chapter in the iOS 5 By Tutorials book](#) by Ray Wenderlich and the tutorial team.

The second one is a tutorial written a while back but is still very relevant today. [How to Create Newsstand Apps By Viggiosoft](#).

These really helped me a lot to get up and running with Newsstand subscriptions.

## Marketing Strategies And Tips

For the marketing, I followed a three-pronged approach. Having Contributors promoting the app, informing App Design Vault subscribers and contacting a couple of friends and acquaintances in the iOS development space

### Contributors Promoting the App

The first one is having contributors to the magazine. I kind of built marketing into the product in the sense that I'm inviting people with audiences to come out and contribute content to the magazine. In the first issue, I had David Barnard on who has a blog at AppCubby.com and is very active on Twitter.

When contributors share the magazine with their audiences, they will naturally come and download the app to get the content. Interviewing friends and experts in the iOS world, and then in-turn having them promote their interview, is the first tactic I used in spreading the word about the app

### App Design Vault Subscribers

I had 5000 people subscribed to App Design Vault's newsletter and I had been teasing about the app for a while on the newsletter. I wrote a series of four email messages explaining the content that was going to be in the first issue. These were not just promotional emails but content-filled emails with a lot of value.

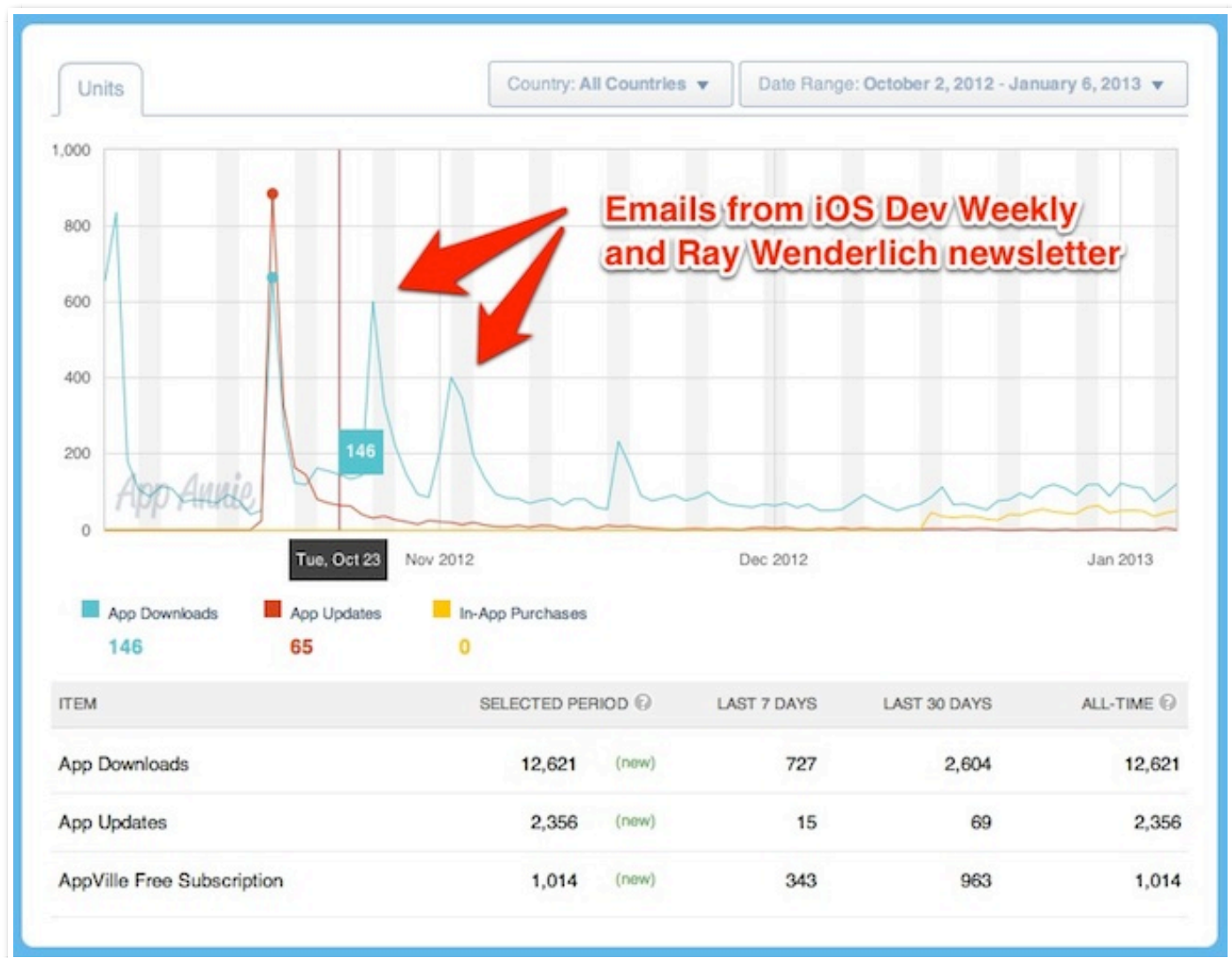
But I also left open loops and was always teasing about the content that was in the magazine. To close these loops and get the full picture, you had to get the magazine.

I sent those emails to everybody on the list and about 2000 people actually clicked on the link to see the app. This contributed to the c.a 800 downloads Appville got on launch day.

### Pitching Influencers in the Industry

If you take a look at the graph below you will see three peaks in the download count. The first peak was on launch day and I have already explained how I got those downloads. The 3rd and 4th peaks (will explain the second later) came from a couple of mentions in

email newsletters. I sent an email to Dave Verwer who runs the [iOS Dev Weekly Newsletter](#) explaining what the app was about and asked if he could include it in the next issue. He agreed. The mention of Appville in the iOS Dev Weekly was the reason behind the third peak.



I also pitched Ray Wenderlich who runs [an amazing site on iOS tutorials](#) and asked if he could include a link to the app in his monthly newsletter. He did include it and that accounts for the fourth peak you see in the graph above.

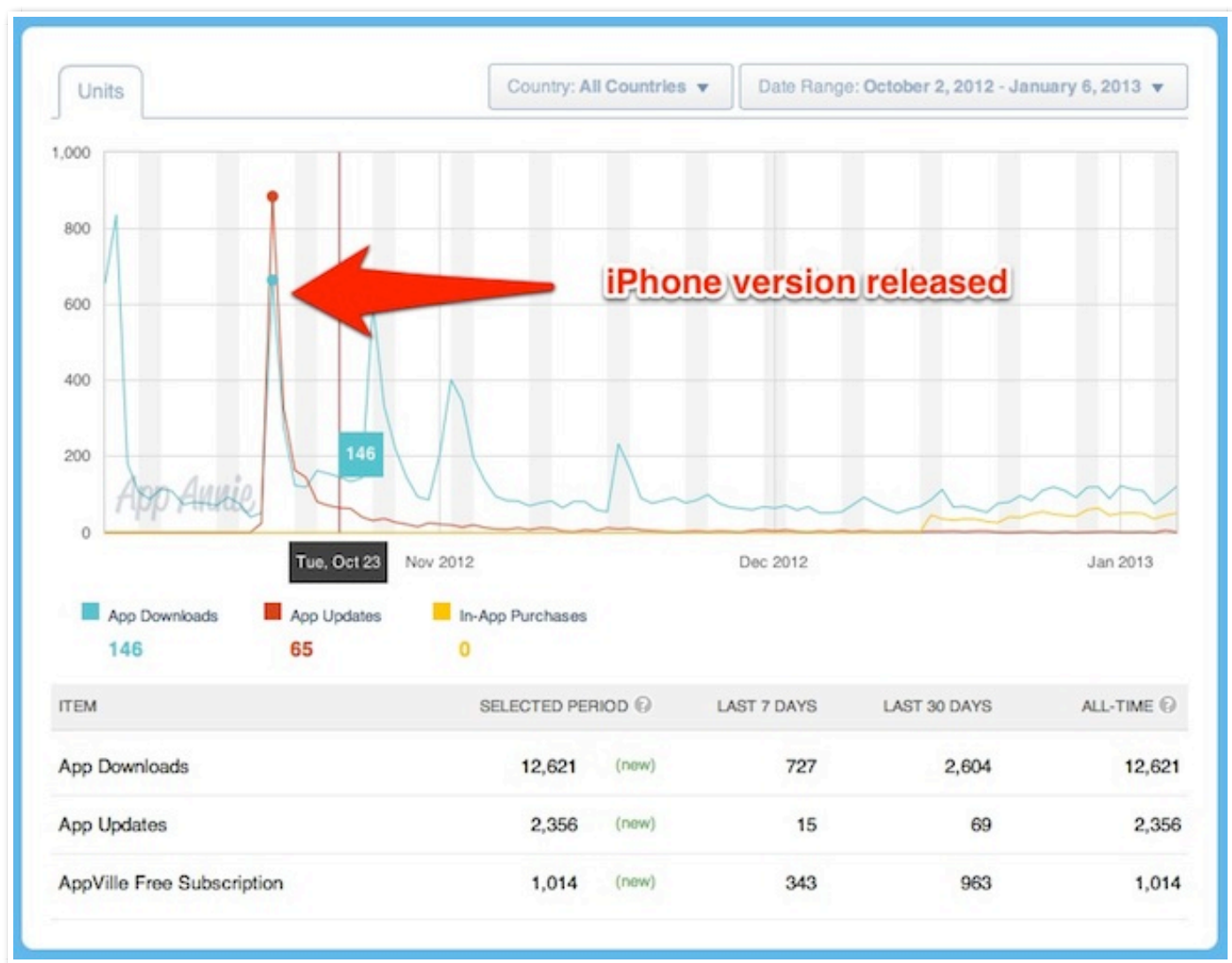
I haven't pitched large blogs and review sites yet. I still plan to do this but frankly I have been too busy (or lazy) doing other stuff for App Design Vault. But from the results above, you can already see that you don't need to include the major blogs and publications to get a reasonable amount of traction for your app. A good strategy is to contact people with audiences that your app targets. In my case, I contacted people that have iOS developers as their audience.

## Mistakes and Failures: Things to learn from

## Ignoring iPhone Users

One mistake I made was that I did not release an iPhone version from the get go. When I released the app and I sent that email to the App Design Vault subscribers, a lot of people were asking me if there was an iPhone version. I assumed a lot of people had iPads and would want to read the magazine on the iPad.

I could have easily made a universal app and used a responsive HTML template for the magazine's pages. I did release an iPhone version later on October 17th and you will see that there was a spike when the iPhone version was released. Up until now, about half of all downloads of the Appville magazine have come from the iPhone. So with over 12000 downloads at the time, this means I would have lost 6000 users by not having an iPhone version at this time.



## Underwhelming Virality

I included the [Socialize plugin](#) which was meant to generate some viral sharing on Twitter, or maybe some comments, but that hasn't really worked well for some reason. I

think the reason is because it's kind of hidden down at the bottom of the app when you tap on it and it's not initially visible when you open the app. I have to work more on that and find out how I can generate some more viral sharing of the content.

## Conclusion and Final Thoughts

That is all I am able to share about the journey of developing and launching Appville. My advice is if you have an idea, go for it. I mean there's no reason why you shouldn't. Just make sure that you just don't think about the development of the app. The fun part for you maybe having elegant code. But also be sure to also plan your marketing before you even start.

You should always make sure that your app is solving a problem. If your app is solving a problem, your marketing would definitely be easier because it is easy to tell a story and convince others to spread the word. So if possible, integrate your marketing into the app as I did, and then find a way to get people to share information about your app.



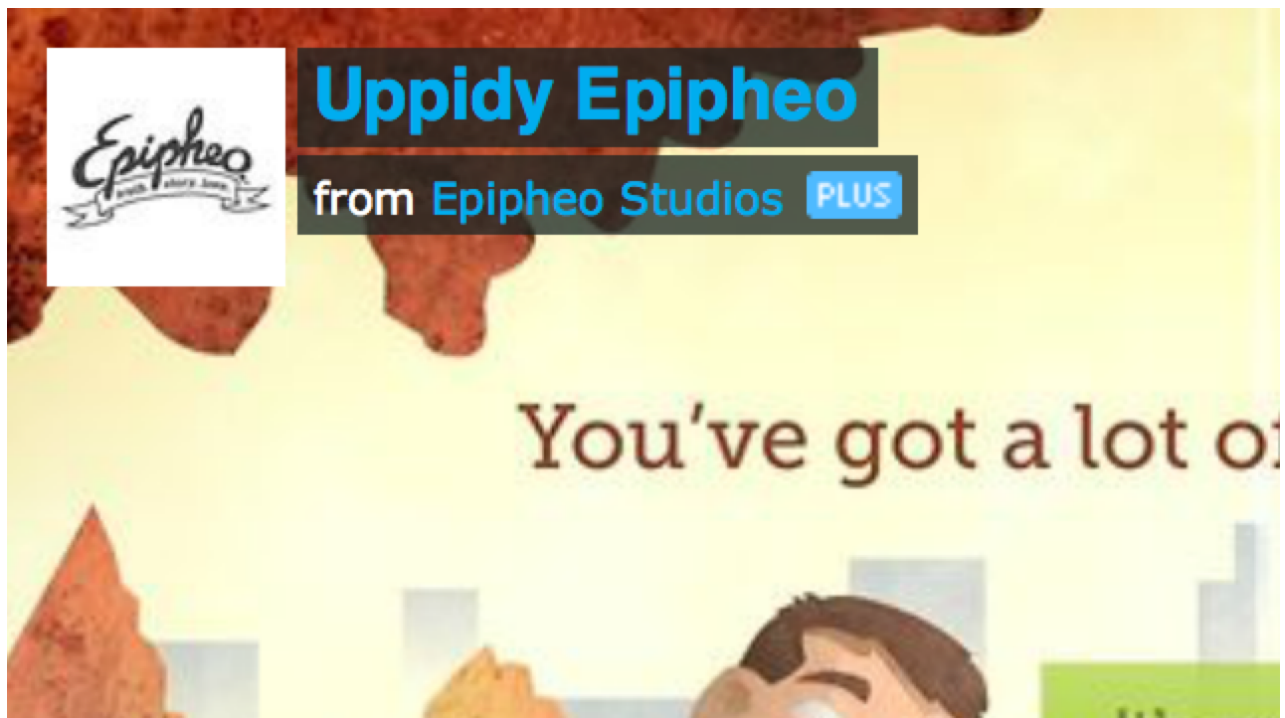
---

# VIDEO APP MARKETING OUTSOURCING THE PRODUCTION OF YOUR MARKETING VIDEOS

- MANUEL ZAMORA: FOUNDER, DOWNSHIFT INTERACTIVE

*Nothing beats a video when it comes to strategies for getting your customer's attention. A great demo video will not only showcase your application but convey your values, personality, and uniqueness. A great promotional video should definitely be at the top of your marketing to-do list.*

Professional video production, however, is expensive. As iOS developers we've all seen the work of some of the more successful video marketing companies (like Grumo and Epipheo). The work you get from services like Grumo Media is impressive and you definitely get what you pay for. Depending on the kind of video you want, 1 minute of professional animation can cost between \$3,000 and \$10,000. If that's not in your budget range, you might want create your own promotional video.



*Make sure your iPad mute button is not activated so you can hear the soundtrack*

## Animated Videos Can be Expensive

Making an animated video is no easy task. Get ready to purchase expensive software and plugins. Video production software is far from cheap. Adobe After Effects 6 can run you \$999! Also, if you don't already have the necessary expertise to get the most out of this software, it's going to cost you a LOT of precious time.

As an app developer you know you need a great marketing video. But with the options of making your own, hiring top of the line production companies, and everything in between where should you start? I suggest you start by considering the advantages and disadvantages of your basic options.

## To Animate or Not To Animate. That Is the Question

When I started making applications in 2008, the app store was in its infancy. There weren't many businesses looking to help mobile developers. It wasn't a proven market yet. Early efforts at marketing my applications included submitting to Mac/app review sites, press releases,



social media, etc. With just these efforts I saw some success, but something was missing in my marketing, the piece de resistance, if you will.

So I looked into making an animated video. Let's just say it's not noob friendly. It takes a fair amount of time, effort, and skill to produce one.

You will need to:

- Write a killer script.

- Turn your script into a fully designed storyboard.

- Use Adobe illustrator to create the characters and scenery.

- Use Adobe After Effects to rig and animate your illustrations

- Get a great voice over.

- Find Sound effects for your video or make your own.

- Put it all together.

Your completed product would hopefully end up something similar to:



*Make sure your iPad mute button is not activated so you can hear the soundtrack*

Sound like a lot of work? It is. Luckily, there are many less sophisticated and comparably effective video production options.

## Are You Good At DIY?

You can start by looking for free video templates and learn some basic After Effects or you can reference video creation sites such as [Animoto](#). While these sites don't offer many custom options, with some intelligent use of screenshots and creativity you can make very compelling videos.

## Pros

Cheapest – Using freeware, rather than Adobe products, can save thousands.

Learning – Knowledge, as they say, is power, and by the end of it you will have learned very useful, relevant skills.

Efficient – Using free sites you can get some inexpensive videos out and about in no time.

## Cons

Quality – Everyone starts somewhere and if you're new to design, your video's quality, and therefore its effectiveness, will likely suffer.

Not Time Efficient – Often you will be better off spending a little money to save a lot of time.

Results - Using free sites will get you a video that will resemble a LOT of other videos and they won't be as customizable as you would hope.

In the end you always get what you pay for, and for DIYers this will be a good route. You can make a great video with basic resources, and your success will likely depend on your ingenuity. If you have some marketing money to spend and are looking to step it up a notch you can always consider other options.