In the beginning...

Well, here I go attempting to do two things simultaneously; maintain a blog and build my first iPhone application in the process.

It seemed that simply embracing the idea of building an application was an epiphany. Having never studied C++ or Objective C or any other software code and never having designed a video game or application, it was a rather large leap in my mind to think "I want to build an application for the iPhone."

Mind you, at the time of this posting there are thousands of applications for iPhone and iTouch users. This idea isn't novel, but it is new for me. I'm uncertain how many people with my lack of experience embark on the process and I'm hoping my process will be of interest to others like me.

After embracing the ideas of building an application and blogging about my experience, it was time to do some serious, extensive research.

Apple provides open source software for iPhone developers. Their software development kit (SDK) provides developers with all types of documents, tutorials, and developer testimonials at their web site. http://developer.apple.com/iphone/

Initially, I found an excellent article "How to become an iPhone developer in eight easy steps" at http://www.guardian.co.uk/technology/gamesblog/2009/feb/10/gameculture-apple this gave me the leads to begin the process.

Once I was at Apple's SDK site, I quickly learned that one of the best articles to read BEFORE developing anything was the "iPhone Human Interface Guidelines".

Furthermore, I researched some of the best books on beginning the development process and I found "iPhone Application Development for Dummies" book received very high and favorable reviews. I've ordered the book and I am awaiting its arrival.

In the meantime, I am reading the copious materials on the Apple site, and searching video tutorials on the Web. One site offers a good tutorial at www.pragmaticstudio.com I will continue to search for more tutorials and update this blog with all the reference materials as I come across them.

The other research I have done is to begin to look through ALL the iPhone applications being offered in the Apple App Store in the genre relating to my application, which might fall under entertainment, lifestyle, health or education. Looking through these many apps helps me to understand what is currently out there and what apps may be similar to the one I'm creating. This research is daunting! There are thousands of applications, but knowledge is power and it's best to know what niche you are filling and what the competition is.

Hi Kanon,

Excellent! It's my sincere hope that this blog helps you. I'll cut to some conclusions so you know what the big lessons learned are:

1. We did choose to use command buttons to control actor actions in three of our levels. I believe this to be a less effective use of the iPhone environment. I would not recommend placing command buttons within a game for iPhone for several reasons. (a) Buttons eat into the screen real estate. The size of the buttons need to be large enough to accomodate fingertips of all sizes. (b) Buttons are a bit difficult to navigate on the device. (c) iPhone uses touch and accelerometer features that do work best on the iPhone, thus buttons seem antiquated. (d) The use of the buttons did cause in one case the iTouch device to turn off because of the location of the power button on the device being where the user places her hands. While holding the device, the player can accidentally click into a black screen saver mode, thus interfering with the game play. We never considered the possibility when we made the app and it didn't show up during the testing of the game. Fortunately, it is rare.

2. It is extremely important that all PNGs be "save for web" to reduce their size. Actors and elements should be less than 100kb with backgrounds no more than 300kb.

3. DO NOT build the game on several computers. We hade five team members each building a level on their own computer believing we could migrate to one computer and stitch the game together, but that does not work. GS provides original actor ID numbers for each element individually dedicated to one project. These do not copy over into a new project. We had to REBUILD each level and import all images into a new game to stitch them together.

4. GameSalad has its quirks and acts much like a beta, so do plan to experience some delays, crashes, and frustration. It takes a while to understand the game building psychology behind the program. Whereas, the GS forums do provide fairly quick responses to questions, the forums are not organized in a concise way by topic, issue, problem or error. It's time consuming navigating the forums for answers addressing specific problems. GS does have a number of tutorials (print and video), but there is no book on the software and many issues are difficult to find answers for. Be ware, some answers provided in the forums are obsolete due to GS upgrades and may no longer be relevant or accurate, but they are still listed among the entries.

5. Do test your game on a device before publishing it because the game will not necessarily work according to the computer simulator generated version. It may work perfectly on the simulator, but not on the external device during testing. Furthermore, we experienced tremendous difficulty getting our game to play on an external device. Initially, we thought it was due to the Mac OS and Xcode versions not agreeing with each other, however, after considerable time installing Snow Leopard and the latest

version of Xcode we were still unable to get the game to play. As a work-around we uninstalled Xcode and reinstalled Xcode several times. Eventually, it worked, but we never understood how it corrected itself or what was truly the problem.

Hope the blog helps.

Day 2

Yesterday was a long one. I received my "Building iPhone Applications for Dummies" and spent a great deal of time perusing it way late into the night. Upon first glance, it does appear like a good book. But here's the problem, there are so many books and materials to study. Where is my time best placed for the greatest and fastest return?

I'm unsure whether I should read the books first or dive into building. I suppose I could do a combo thing where I read, watch tutorials and create an example app. Initially, I thought reverse engineering might prove viable. Jailbreak into my iTouch, after identifying an app that most closely reflects the one I want to create, and work backwards on the code by substituting the action code that would drive my app. This is not as easy as it sounds.

There is an article that discusses this approach for those of you who want to do it that way. http://dvlabs.tippingpoint.com/blog/2009/03/06/reverse-engineering-iphone-appstore-binaries

However, I've decided that this is not a practical approach and I've chosen the origination-to-creation process.

The continuation of my research at the Apple App Store proved enlightening. Seriously, two years ago when I bought my iTouch and became hooked on apps, I spent hours for weeks solidly reviewing and downloading apps for my device. (Dare I admit that I spent almost as much on apps as my iTouch cost! Thus, the app appeal and its economy summed up.) At that time, I DO NOT remember seeing any pornography (T&A) apps.

However, my most recent inspection has me concluding that at a minimum, 1/4 of all apps in the entertainment genre are porn! Why! I guess technology brings with it the good and the bad, including the seedy underbelly of the world's oldest profession always lurking to exploit new means of commerce.

Equally, there is burgeoning fetish market with such apps as armpit pics for those who want to get their fix. The cornucopia of apps available is mind-boggling and there seems to be plenty of room for all kinds. Many apps overlap with slight differences, but the vetting criteria at Apple seems to allow a plethora of apps to reach inventory status. And why not?

Apple has an vested interest to keep the production lines going. It receives 30% of sales for apps that are sold and the great seduction of the iTouch and iPhone is the wide range of apps to choose from. It is the crux of the business paradigm shift in the ether economy. "Create great open source (free) software and

they will come" - to use a takeoff of the "Field of Dreams" great one-liner. Google has also created open source software for the same reasons. I'm sure there are others. Pranav Mistry is making Sixth Sense open source, including instructions on how to dissemble a cellular phone to build a Sixth Sense device for free! (check out TED.com at http://www.ted.com/talks/pranav_mistry_the_thrilling_potential_of_sixthsense_technology.html and watch his presentation. Expect to be blown away.)

I recently read that the strongest, more robust and stable OS is Snow Leopard. For $29, I ordered the latest upgrade for my computer. Why? Because the recommendation is that Xcode 3.2 in Snow Leopard is the most user intuitive, user friendly Objective-C coding version out there. I currently have the Mac Leopard OS, so the upgrade was fairly inexpensive to get.

It is important to remember that you need to have Mac Leopard OS to build apps. It is further encouraged to have and be familiar with the iTouch or iPhone.

http://labs.adobe.com/technologies/flashcs5/appsfor_iphone/

Day 3

There is much to write, to study, to create. This app development process is intimidating and overwhelming. I have found so much material on the subject I feel like I'm swimming in information overload. As of last night, a reader left yet another document to read called "Building Photokast" found at www.scribd.com/doc/12684298/Building-PhotoKast-Creating-an-iPhone-app-in-one- month

Thanks, Christian, for this link. This doc is really, really good. It deals less with code and more with the principles behind User Interface (UI), application motivation, process development expectations, and customer service. It's an excellent document that I highly recommend.

I have continued to cull through materials, books, and videos. Ironically, as I begin this process, I am starting at the same time as Stanford University's class on iPhone App Development. (the link is listed below in iPhone Resources). Classes are on iTunes for free. Admittedly, students are required to have taken Objective-C before entering the class, so auditors like me who have no experience are at a slight disadvantage. Furthermore, these videos aren't permanent on iTunes. Stanford warns that they will be removing the videos after a couple of months.

But Stanford is just one of many, many tutorials out there. Furthermore, there are many blogs on the subject of building apps. Therefore, it begs the question, "Why create this blog?" Well, of the blogs, articles, tutorials, books, and web sites that I have seen, I have yet to stumble across one that really starts with someone as unfamiliar and wet-behind-the-ears on development, C++, Objective-C, and iPhone app creation as I am. Undoubtedly, there must be blogs out there with newbies like me writing about their journeys, but I haven't found them.

I still feel that all the information I come across has someone with some type of coding experience, game development or app creation background working on the app. Even the "iPhone Application Development for Dummies" book that states you do not need any iPhone or Mac development experience requires some coding experience. The author, Neal Goldstein, assumes the reader has some programming knowledge and familiarity with C++ and Objective-C.

To which I respond, "No, no I don't." That's the crux of the problem. Most blogs and sites don't really begin at the very start and that's where I am.

Literally, I feel like a beginner on every level and facet of app building. If the repetitive statement so readily found in these materials is that anyone can build an app, then I'm a good test for such a qualifier. I figure if I am able to accomplish this app thing, then quite literally anyone can.

It is the reason I'm spending so much of my time reading, researching and getting my hands on anything and everything to do with app building, rather than just diving into Xcode. Yesterday was spent watching videos on the subject and continuing to collate resources for the blog.

As I wrap this entry up, I will restate my enthusiasm again for the "Building PhotoKast" document. It provides a comprehensive overview of their development process with tips, advice and lessons learned. It's short and sweet, but I think it is an excellent document to READ FIRST before all other things. With so much information to get through, this is saying something.

Ultimately, it seems that at its core an app must fill a niche and engage the user to the point of addiction, while "telling a story" of some sort. Perhaps, that is the one area I'm not so callow in. Story telling is something I love to do and the idea that an app tells a story feels, at least to me, intuitive. This is why games immerse users.

Furthermore, it is advised to build apps for one of the seven deadly sins. (read "Building PhotoKast") This statement is an intriguing way to generalize a key design rule that acknowledges the importance of building apps that fulfill and create solutions that deal with the different areas of human needs and behaviors.

As soon as I receive my Snow Leopard upgrade, I'll download the SDK for Snow Leopard and begin the simple app tutorial. In the meantime, I continue to read, watch, and research information.

(I have added a significant number of new links in the resources section. Check them out.)

Day 4

Read. Read. Read. That has been my main objective and I've discussed this extensively. I emailed author, Neal Goldstein (neal@nealgoldstein.com) about the Objective C issue. The issue of my not having any

experience, and he suggested I read "Programming C for Dummies" another book he wrote. He was exceptionally fast at responding to my email, which I greatly appreciated. I have ordered the book and I will add it to my growing list of materials.

Along with understanding the iPhone application development process, I have begun to identify and shape my application idea, core functions and scripting. This touches on an area I'm familiar with in story telling.

Designing the look of the interface and story-boarding the script for the application are exciting processes. Writing something down makes it so much more concrete and real. Psychologically, it is encouraging to feel that at this moment I can be productive. Rather than having to wait to acquire the skills to design physically the application, designing the idea out on paper generates that sense of forward motion and it's a prudent move.

By making a paper version of the application, quirks and problems are discovered. The process should also streamline coding when that time comes. I am a true believer in visualization. As an extension of visualizing, the paper process helps to create in my mind the look and feel of the application so that my actions lead me to the app's creation.

By researching and reading in tandem with designing, such issues as screen real estate, pixel allocation accommodating for "fat fingers", graphic simplicity and aesthetics are all kept prominently at the forefront of my mind.

Furthermore, by creating a statement of purpose for my application, I focus the idea that influences the app's core functions and abilities. Many people suggest to simplify apps to one core function and not to try and design an app with a slue of bells and whistles.

This sounds self-evident, but I believe people over-complicate apps because their idea for an app is weak and they are trying to mask this problem. But in the end, users will play with apps because they are immersed in the apps functions and services. To that end, beautifying an app that ultimately lacks substance will not prove successful. However, this formula for success is elusive. If it were th at easy to create a true immersion experience then everyone would be doing it and all apps would be a hit.

Statistically, the majority of apps do not do well. Outside of the risk of financial recovery of development costs and the ability to make profits upon release, there are other risks associated to building apps.

You could build an app that Apple chooses NOT to publish in their app store. If that happens, you're dead in the water before you've begun. I haven't found a criteria list from Apple identifying the do's and don'ts for apps to ensure my app doesn't get squashed. If I find one I will most certainly post it here. This is probably one of the worse risks to app building.

Then there is the possibility that your app is never really seen or advertised and it gets lost among the

thousands of apps out there. Presently, Apple doesn't sub-list in any general genre. For instance, in the entertainment section of the app store (the largest of all the listings) repetitive apps for hot babes and fart noises drown out other entertaining gaming apps. There are so many of each of these types of apps that it warrants Apple's attention to diversify their classifications further so that new apps don't disappear in the huge numbers of apps all thrown in together.

Also, there is the risk that your app receives revenge reviews from angry users or simply dissatisfied ones. Your app's rating is at the mercy of users. The only real way to beat this is to make an engaging and satisfying app from the start.

Well, that is easier said than done isn't it. Ultimately, creativity and ingenuity are the skills needed to be honed to a fine point in this industry

Day 5

It seems I have had a break in the case so to speak. I decided to check out Lynda.com (I'm a long time subscriber to the tutorial hot spot). I was looking for Objective C tutorials, but what I found was something so amazing. A full course on iPhone App building in video tutorials. Gotta love Lynda.com for putting this out, but more importantly the guy whose teaches it.

Simon Alladice, a software developer with more than 30 years experience, is incredibly easy to understand and he breaks down everything so clearly. The tutorials are engaging and utilize a large touch screen viewing monitor where Simon interacts with the device to illustrate key points. It is easy on the eyes and ears by reducing app building step-by-step from the ground up. Although, he expects viewers to have some Objective C, C++, C# (sharp), Java, PHP or coding experience (again this issue of previous experience) he actually does take you through code with fluid ease. Whereas, I was expecting to have to muddle through, I find I can follow along and (here it comes) RETAIN the information. Wow! Kudos to Simon.

I was discussing the point that there is this idea that anyone wanting to create an app must seemingly have had coding experience. The person I was speaking to said that maybe it's because people who don't have coding experience would not think or want to build an app. There may be some truth to that. I suddenly decided and locked onto the idea to create my own app and at the time it seemed like a big jump mentally for me. However, was it? There are thousands of apps out there and people keep saying anyone can build one, perhaps I just finally took them literally.

Honestly, I have accepted that I will go at this app building business without any programming background and just dive in. I figure with enough exposure, repetitive directions and copious notes I would"get it" eventually. Only a few days in and yes, I would say I am beginning to get it.

I have also found a place that will build for you and/or consult with you on your app. I was actually contacted through twitter by AppLiya Studio in Japan. They offer app building for hire services at

www.appliya-studio.com So should I become disillusioned, I can default to someone else to build the app I've dreamt of. Of course, that would defeat the purpose of this exercise and it would admit defeat. So I shall trudge on.

As of today, I have yet to receive my Mac OS up-grade and I keep peering out to see if the mailman has come. I don't want to download the SDK for Leopard, when Xcode 3.2 for Snow Leopard is thought to be so much better.

Day 6

Since my last entry, I did receive and install the Snow Leopard OS up-grade and download the SDK program, however it took two days, primarily my evenings, to complete these operations.

Initially, when I installed Snow Leopard OS X, it immediately threw me to the Apple website for the recent up-grade of Snow Leopard 10.6.2. I then had to download the upgrade (which took more than an hour). When I tried to install the up-grade my computer wouldn't allow it to be up-graded. After much haranguing, I called Apple.

I was instructed to try Software updates under the apple icon. I checked up-dates but my system didn't display details, which struck the support tech as wrong. So, she then instructed me to go to my system utilities and run the repair program. This did isolate an issue and repaired it.

After which, I was able to download the software up-grade 10.6.2 and install. The next step was to install the SDK from the developer center at http://developer.apple.com/iphone/

Remember, that before you can download the SDK you have to register as a developer and create an Apple ID. I had done this days earlier. So I logged on to my account and downloaded the Xcode for Snow Leopard (3.2 version).

Finally, the technical stuff was out of the way and I immediately began new project.

Day 7 - App building and the Business Formula for Success

I have made it through several hurdles. I did finally up-grade my OS, download SDK, and begin the "Hello World" tutorial with some success. I am still watching iPhone App Development tutorials on Lynda.com. Although the information does seem overwhelming on the tutorials, it also appears manageable.

I completed researching the iTunes app store and I found that my idea for an app is NOT represented, even once, within the genre I'm interested in. That's excellent news; no competition. And this segues nicely into my topic for today on the formula for success.

As I pursue my app building journey, I do so working full time and going to school, where I am concurrently obtaining two master degrees; MBA in Marketing and MFA in creative writing. I seek these

degrees as a means of formalizing my 17 years of work in marketing and writing.

Presently, I'm studying entrepreneurship. It has brought to the forefront some significant aspects of creating a business that are important to consider.

Often businesses are assessed based upon their management teams, the product, the opportunity, the risks and the start-up position of the proposed business. As the area of management team goes, I am a seasoned professional with entrepreneurship experience. I started two companies and working on my third. I've learned some excellent lessons along the way. However, to my disadvantage, I do lack experience in app building and software programming, so I need to seek out partnerships or hire people with those strengths. That's a topic for another day.

My interest today concerns the area of opportunity in relation to risk. This is where I believe there is real potential for my business model. The market is growing and the niche I plan to fill is unsaturated. That's a good thing. Critical to my success is the rate to which I can build and release my app. It's just a matter of time before someone else does release a similar app that I am working on, so time is of the essence. Strike while it's hot and empty.

Furthermore, my start-up costs have been relatively low, thanks to the open-source SDK and the plethora of free information or inexpensively priced tutorials that are readily available. My primary cost is sweat equity; learning the programming side and developing my app's hook and interface. I justify the learning curve as one of on-going education and skill development. It's a win-win situation for me.

In fact, I have found that my new hobby has impacted the way I think about my job at work. As I embrace the concept of user interface applications and their impact on organization and presentation of information, I realize that there are many areas of the University where I work that could benefit from applications that integrate users and information in a specialized and engaging way.

Entrepreneurship is never without risk, but the degree of risk varies. Knowledge is power and it is extremely advisable to research the industry you plan to enter. With iTunes App Store, it's relatively simple to comb through the competition all centralized in one place. It is also prudent to read the iPhone Human Interface Guidelines provided by Apple and other materials. Currently, I have been working with such books as, "iPhone Application Development for Dummies", "Objective-C for Dummies", "Cocoa Programming for Dummies", "Sams Teach Yourself iPhone Application Development", and "Game Design Workshop" (Fullerton). All of these materials have proven effective.

So, I go about my project with a lighter heart knowing that the area I'm interested in isn't crowded and there is plenty of room for one more player.

Day 8 - Zombies and a New SDK to handle iPad

I'm still reviewing the iPhone Application Development tutorials on Lynda.com. I'm on section seven that deals with User Interface. However, the previous section, Debugging and Troubleshooting was a real brain drain! It's incredibly important to remember the issue of memory. Experts constantly remind users that Apple is exceptionally stringent on use of memory. It is shocking how little memory is allocated in the iPhone to applications. So, objects must constantly be released back to the pool after each alloc and init. To help with the release of memory, but without losing track of the code, there is a tool within the debugging process called by the cool name "Zombie" that can be used.

(*NOTE: I'm interrupting here to briefly mention what I find intriguing about programming jargon. Common, everyday terms are used in specialized argotic syntax that plays on a word's original meaning and extends its purpose further to define unique properties in code. An example of this would be "heap", which in C++ means available memory. "An area of memory used for dynamic memory allocation where blocks of memory are allocated and freed in an arbitrary order and the pattern of allocation and size of blocks is not known until run time. Typically, a program has one heap which it may use for several different purposes." Heap in common, everyday speak means a pile. It is easy to see the extension made here. Then there is the use of the word Zombie. - Zombie objects, whose only purpose is to report an error when someone calls them, acts as the husk of the previous object. It's fascinating to see these uses of words.)*

Using Zombies as a tool to figure out problems is a good way to check code. It's important to "enable Zombie" BEFORE releasing an object during debugging. A Zombie acts like a placeholder or shell of the object that was released back to the pool. Therefore, when checking code, Zombies help you identify where the problems are in code after an object has been released. However, if you don't release the Zombie afterwards it continues to hold on to memory which you don't want. In addition, you must enable Zombies in all caps YES or NO for the BOOl. (In computer science, the Boolean or logical data type is a primitive data type having one of two values: true or false, intended to represent the truth values of logic and Boolean algebra. - Wiki)

I realize this all seems complicated, and yes, to a degree it is. But Simon, Lynda.com instructor, does his level best to keep the information to a reasonable degree of difficulty.

On a side note, the iPAD was release this week and with the release comes a new SDK! Although all the apps currently in existence will work on the iPAD, the dimensions remain the same. There is a feature where you can pixel up, but quality is lost of course. Apple recommends that developers work with the new real estate and the new SDK. At the rate that things change I hope I'm able to get above the flailing and into the creating of product.

Day 9 - The Human Interface Guidelines a must read

I finally finished the Lynda.com iPhone Development Application tutorials. What can I say; I need to go through them again. Sigh. :(There is simply too much information to understand on the first run. Mind you, I often had to stop and restart a section more than once to stay focused during the first time

through.

In section 13, I found the answer I was looking for regarding the guidelines. Apple's Human Interface Guidelines document found in the Apple developer center ARE the specifications for building the application. This 136 page document is an absolute MUST READ and it will guide developers on what Apple will expect and except when reviewing an app for the app store. It's recommended that the guidelines be read in their entirety. Quick glances through it are not recommended.

Furthermore, another very important document that needs to be read is the iPhone Application Programming Guide, also found in the Apple Developer center at http//:developer.apple.com/iphone/library/navigation/Guides.html

Such specifics, as creating an interface app icon that MUST represent the app and match the 512 X 512 pixel app icon and its description that is sent to apple upon your release, are described here. Other specifications such as 57 X 57 pixel icon for the app, or that the iPhone only allocates 128MB for apps, which is shared with the iPhone OS! And such information as the size of screen real estate at 320 X 480 pixels, which is actually 320 X 460 because the status bar eats up 20 pixels, is all provided in the guidelines.

Memory is a critical aspect and issue when building an app. Objective C is centered on the allocation, initiation, and release codes that efficiently handle memory. Everything you have your app do will be centered on the memory it uses to do it.

Now, I must take the time to read all the way through both sets of guidelines and I'm still watching the Stanford iPhone Development course found on iTunes for free. I project that all of this initial research will be largely done by the end of this month.

Day 10 - What I know about my app

After finishing the Lynda.com tutorials, I have returned to the Stanford podcast on SDK. It provides interesting stuff on memory allocation and management and it reinforced what Simon stressed in the Lynda.com tutorials about memory issues. However, the two tutorials don't use the same templates. The Stanford people use the Windows-based template, while Simon uses the View Based template. Simon deliberately sidesteps the Game-based template, which is the one that I need.

That leads me to the focus of today's blog entry; describing what I know about my app. Apple identifies three different application formats that all apps fall under; productivity, utility, and immersive. It's important that you identify the type of app that you are building from the beginning stage.

For me, I am building an immersive app. It will only be an iPhone app and not a web based or a hybrid of iPhone app with web based content. My target audience is health conscious and seeks health related information. I'll have some features like customizable scenarios, music choices, and text prompts within

the app.

I will be incorporating the accelerometer for my app, which I believe adds additional complexity. Furthermore, I plan to bring the user into a virtual world, which means I will need to create a world outside the immediate space they manipulate. I need to create the horizon.

I think of this similar to the sound stage of "The Wizard of Oz", where the stage was the real space the actors performed on, but the backdrop set created the illusion of Z space out to the horizon where the City of Oz was located, but it was just a wall. This principle seems most fitting for how I envision my app and its design. But that's just it, I don't know how to work with the Game-based template, accelerometer, or the inter-workings of the graphics.

The process of building an app isn't restricted to the learning curve of coding using C++, Objective C and Cocoa. It also deals with the graphic construction and Flash animation. But most importantly, building an app is about problem solving. How do you build what you can see in your mind's eye? I can see it perfectly, but how do I get there?

Furthermore, I have to decide how the program saves when the users cuts out instantly. I want my app to automatically save where the user left off and not reset itself. A consolidation of some of the things discussed so far can be found in the tips from Apple.

"As you design the flow of your application and its user interface, follow these guidelines to build in simplicity and ease of use:

- *Make it obvious how to use your application.*
- *Concentrate frequently used, high-level information near the top of screen.*
- *Minimize text input. (typing)*
- *Express essential information succinctly.*
- *Provide a fingertip-size target area for all tappable elements." – Apple Human interface Guildelines*

Day 11 - iPad SDK download, Cocoa for Dummies, and Android SDK in Java

As of last night, I downloaded (still installing) the new SDK beta for iPad. I figure I should have the latest version. Apple seems insistent on using the iPad title, and it will likely have adopters for the device.

I continue to read and work through tutorials in the books I've purchased. I made the mistake of buying used the 2003 version of "Cocoa Programming for Dummies". I find the book obsolete because the pictures in the book don't match the current software interface. You can't follow along. It's best NOT to work with earlier versions of books dealing with software because of the mismatch that occurs due to the rapid change in the software. This was a significant reason why I decided to go ahead and install the iPad SDK software because it is the latest version and I want to work with relevant, current materials.

As I have been spending significant time learning the facets of iPhone SDK 3.1.3, I began to have my doubts about developing for the iPhone over that of the Android.

Android is a newly released phone by Google. I'm a huge fan of both, Google and Apple. My loyalties reside with both. However, the advantage Android has over that of the iPhone for developers is the Android has far, far less apps built for it than the iPhone, which is oversaturated with a jumbled mess of apps, especially in the entertainment section.

Strategically, going into a market with little to no competition is a safe bet. You would stand out, receive more hits, have less competition, be found more easily and hit the ground floor in the newly evolving environment before others, which would give the app more prestige. So, why don't I just switch from iPhone SDK (C++ & Objective-C based) to the Android SDK (HTML & Java based)? It's not as if I have more experience in one over the other.

First and foremost, the programming language is not the same. Android is build with Java and requires prior knowledge in HTML. Though, I'm new to programming and I have no other language knowledge, I have spent weeks and some coin on books for C++, Objective C, and Coca programming. "By Jove, I believe I'm getting it." It feels a little late turning back.

iPhone and iPad apps can only be made on Macs using the Xcode SDK. To switch now would feel like giving up. By the time I would have mastered Java and built a releasable app for the Android, there may be a significant number of apps already on the market for Android. Ultimately, I'm committed to doing this app build for iPhone and iPad. There will of course be areas of doubt, but staying the path seems reasonable. I could continue to switch over and over again as new versions of things are released, but that feels like ADD (attention deficit disorder).

The Apple SDK does permit a developer to create computer games for Macs working with the Mac OS X.6. There are by far fewer games for Mac computers than PC. So, there's hope of entering an emerging market within Mac.

Read SDK Shoot Out Android vs iPhone http://www.infoworld.com/d/developer-world/sdk-shoot-out-android-vs-iphone-074

On another note, I have discovered there is an order in which to read the SDK documentation. I suggest reading the HIG (Human Interface Guidelines) first. I didn't find much information on games creation in the HIG, but it still proves informative. Remember, the HIG does set the standards that Apple will judge and review your app. Then I would read "Cocoa Fundamentals" before reading "iPhone Programming Guidelines".

I do find the "Objective-C for Dummies" book extremely useful in breaking down the code terminology, explaining the interworking of the code. I'm actually reading the "Dummies" series over the Apple Dev

Center documents because the “Dummies” books are easier on the eyes and friendlier writing. They should be they’re written for beginners.

Day 12 - What format should video be in for the iPhone

This week I spent a good deal of time reading the Cocoa Fundamental guide and Objective-C. The opening lines says "*readers should be proficient C programmers and should be familiar with the capabilities and technologies of each platform.*" This is a problem because I don't know any programming language thoroughly and I have only scratched the surface of Objective-C.

I do understand the tiered diagrams that explain the differences between Mac OS X and iPhone OS. It is obvious that some support code is not necessary for the iPhone such as printing capabilities, multitasking etc.

What I found curious was that Quicktime is not a component of the iPhone as a core service, but it is for the Mac. Also, Flash can’t played in iPhone, but .mov files (Quicktime) can be, so I’m confused. Perhaps, it comes down to compression. It's an area I feel less familiar with and I have some design questions about.

I was surprised to see a utility called Shark, which is used for memory allocation or memory mapping. I didn't know about this application, but it appears similar to the task manager in Windows.

At this point, I am still reading the Cocoa Fundamentals Guide, I am in the section dealing with Foundation Classes. I did pick up the Human Interface Guidelines, or what is lovingly called the HIG. Most of it was actually intuitive. Lots of the content reinforced things such as simplicity, and attractiveness.

What I found very interesting was the description of "*Web-only content, including web applications, which are websites that behave like built-in iPhone applications and Web pages that provide a focused solution to a task and conform to certain display guidelines are known as web applications, because they behave similarly to the built-in iPhone OS applications. A web application, like all web-only content, runs in Safari on iPhone; users do not install it on their devices, instead they go to the web application’s URL*."

I haven’t determined that a website has to be structured or designed specifically to fit within the iPhone. This doesn’t make sense, since the web browser in iPhone will allow users to go to any website in their search and access that website. The website may work, but the video may not because of the format it is in on the website. This issue is of some concern and I’m researching it as I write this.

Day 13-Tossing GameSalad around as an app patch over Objective C

This blog was started because of my desire to document my attemtps to build an iPhone app. Initially, I struggled with Objective C and all the coding it requires. And although I have learned about it and I did scratch the surface, I have recently gotten off track because I have discovered GameSalad.

"GameSalad is the world's most advanced game creation tool for non-programmers. With GameSalad, game makers of all levels can bring their ideas to life without programming a single line of code. Build games visually using a drag-and-drop interface along with a robust behavior system. Publish your games quickly and easy to the iPhone, iPod Touch, Mac Desktop, and the Web."

http://www.gamesalad.com/ It's a 2D graphic game builder.

There are quite a few video and written tutorials listed under the Support tab in GameSalad to help beginners. It seems pretty easy to use, but I have to qualify that statement. It is easy to use relative to Objective C.

Presently, I am reverse engineering a basic app from Crayon Physics called "Cannonball". It's one of the templeates that opens up under the new project window. But as easy as it is, I am having difficulty with the behaviors and attributes section learning how to create new rules for my actors within the game I'm building. In my opinion, I believe there could be better videos done for this program.

There are two videos found on YouTube for building a simple space ship war game app, it comes in two parts called "GameSalad Create a Game From Scratch Part 1" http://www.youtube.com/watch?v=P-0-mZqMUbY&feature=related and "GameSalad Create a Game From Scratch Part 2" http://www.youtube.com/watch?v=fWET2sLvwb4&feature=related Both are pretty good. In fact, YouTube carries videos of GameSalad development that aren't found at GameSalad.com

These tutorials are a great place to start, but I'm still finding it difficult understanding how rules can be created and placed upon my actors. But just like Objective C, you will want to build all your graphics as PNGs in photoshop.

Also, to publish your app from GameSalad, you will have to pay $99. This IS NOT the amount you then have to pay Apple to release your app to iTunes. That is an additional $99. What is crazy about GameSalad is the two price offerings. For $99, you can get your one year liscense and publish your games using the GameSalad program, but for nearly $2,000 ($1,999) you can pay for unlimited customer service help for developing your game. That's a huge jump and it strikes me as strange because of the enormous price difference. It might also explain why there is such a limited number of videos and tutorials available on GameSalad. I believe the company hopes to make money on offering customer service help. When I last checked there was NO book available on Amazon dealing with GameSalad.

I will say that I am keeping my eyes and ears open for other free game building programs. A new program due out this spring is Atmosphir.http://atmosphir.com/ It builds for both PC and Macs and it's a 3D animation game builder that doesn't require coding knowledge. I had to sign up to be a beta tester

with the site. I have been recieving emails with questions like, "How I would like to see a blast gun work and what would it look like." Clearly, Atmosphir wants user feedback and will incoproate suggestions into its gaming program. It looks awesome and I can't wait to see it when it's released.

One last thing, I recently investigated Nintendo's gaming institue called DigiPen. They offer online programs over the summer in two areas: animation for beginners and game design for beginners. The workshop sessions are held for two weeks for 2 1/2 hours each day. I ordered the material at https://www.digipen.edu/

I am really excited about this workshop possiblity for the summer. I like the "begi nner" without prior coding experience approach. The fact that it is only a couple of weeks and online equally compel me to want to register. However, there are some fairly strict non-refundable policies regarding registering. So, read the information carefully.

All-in-all, once I became more involved in the gaming area I have learned about so many opportunities. I recently came across some fascinating information. *"Gaming: Not just for kids anymore. I think the very fact that the largest player base of passive online games is women flies in the face of the typical view that games are for kids. According to Nielsen Entertainment in August 2009, of the 117 million active gamers in the U.S., 56 percent play games online and 64 percent of those online gamers are female. And the revenues generated from online games is enormous and growing. Do not underestimate the power of games and gaming — and not just the marketing and revenue opportunities, but also the learning opportunities as well in the form of fun quizzes and polls. Have you used gaming yet in a social media marketing campaign?" (8 Significant Developments in Social Media You Should Watch by Aliza Sherman)*

It would appear that I'm just riding the wave http://www.thinkingworlds.com/

Day 14 - GameSalad struggles, frustrations and let downs; Unity 3D overview

(To start with, I must say that when I started this blog it was to journal my experience with building an iPhone app. I thought I would write on most days, not dissimilar to "Julie & Julia", but that hasn't proven effective. I'm not cooking daily recipes. I'm building a game where the progress is slow and the learning curve sharp, very sharp. So as this entry states “Day 14” it is not accurate. I’ve been at this many more days then 14 of them. I now use the "Day" reference as a marker to address the number of entries. However, I do at times place entries that are a re-posting of other people's articles that I think are important and or related to this project. Those entries aren't numbered, which helps to identify them separately from the journal entries.)

As I write this I have fully embraced and embarked upon actually building my app with graphics (PNG) and game play within GameSalad. I have formed an opinion about this program.

GameSalad has some weaknesses. Firstly, I can guarantee you are going to crash the program if you

have an actor spawn itself. That is a repeatable error that happens consistently. However, more damning are the non-repeatable errors that crop up randomly. For instance, I might employ a rule, behavior and attribute and copy the line-up, but for some reason not be able to recreate the same action. I don't know why. In fact, not knowing how things work together is immensely frustrating.

GameSalad is a beta and acts very much like one. I find it unstable with aberrant behavior that isn't repeated consistently. It reminds me of the horrible days, some 10 years ago, when I inadvertently became a beta tester for Incite non-linear editing program. I was driven mad at times because the system would just crash or behave randomly in some unexpected way.

Presently, I'm having extreme difficulty spawning an actor based upon another actor's behavior in the scene. For instance, if Actor one throws a ball then Actor 2 (the ball) must be spawned at the point of release and do something that I want it to do. Well, it's certainly easier said than done. First, I must figure my X & Y coordinates at the point of release from Actor 1 on the screen. The question I had was, HOW? How do I know where those coordinates are on my 480 by 320 screen? Here is where the good news comes in.

Game Salad lacks some things, but one of its strong areas is the Forums on its site. I found the response rate for answers extremely fast, helpful and user friendly. My thanks go out to firemaplegames, Mike Quinn, and scitunes. The GameSalad members were helpful and sweet with their advice to my questions. Mike Quinn suggested that I download a free iDevice from iTunes onto my iTouch called VML, an awesome app by vmlweb built on GameSalad. **http://vmlweb.co.uk/page.php**? It will help me to isolate the coordinates of my X and Y.

There is a video on this link that shows you how easy it is to use the app when determining anywhere on the screen a point is and it works for the accelerometer as well.
Firemaplegames suggested that I watch the tutorial
http://gamesalad.com/wiki/tutorials:video_tutorials addressing the spawning issue. I received both pieces of advice within minutes of posting. That is an excellent response time.

GameSalad has some issues, no doubt, but let's not forget why I opted for this route . When I began this journey I was learning XCode and reading the HIG and combing through copious material before even beginning to design my game. That was consuming enormous amounts of time and proving daunting for someone who has never coded or programmed anything.

GameSalad is very much a cut and paste program that requires some math, but not coding. It expedites the learning curve and allows the user to jump right in and begin building. Cheers to them for that. I would recommend saving, saving and saving some more when working with the program because it is notorious for crashing. Also, as intuitive as it is, it does lack in areas. It requires playing around and reading, watching and asking for advice when dealing with issues. And be prepared that it will behave strangely for no apparent reason. Non-repeatable errors happen. It is a beta.

In my constant quest to find programs that build apps without programming knowledge, I have stumbled across numerous programs. In an earlier post, I posted an article that listed 13 app building tools that make building apps and games easier. I also came across Unity **http://unity3d.com/** "Unity is a multiplatform game development tool, designed from the start to ease creation. A fully integrated professional application, Unity just happens to contain the most powerful engine this side of a million dollars." This is an exciting game builder program for sure.

In the Unity 2009 Conference keynote address **http://download.unity3d.com/support/resources/files/Unite09_Keynote.mov** , David Helgason, CEO and co-founder of Unity, stated that Unity Indie would no longer be available, but that "Unity" would replace the Indie version and be free to users. In fact, Unity is open source and doesn't require a license fee to post games on the web.
There are two more versions of Unity that do require fees; Unity Pro and Unity iPhone Pro both cost $1,200. More importantly, EA, Lego, Disney and Cartoon Network all work with Unity Pro, so it has some pretty heavy hitters using its program.

An additional advantage for using Unity, it has good video tutorials for the program for free. There are 7 hours of tutorials found at **http://www.learnmesilly.com/** produced by Will Goldstone. He also authored the book on using Unity called "Unity Game Development Essentials". (I've ordered this.) Unity has a lot of positives. It is open source, has strong tutorials, has a manual for purchase, provides strong support and creates 3D worlds.

I will continue to work with GameSalad for the completion of this first game, but I have every intention of exploring and hopefully creating in Unity. I haven't worked with it yet, so I'm not absolutely sure how much coding is necessary. In an older article (referencing Unity Indie) they state, "First of all, it is possible to buy a source code license for Unity (contact us for details), but the vast majority of games don't need it (code) and can be made using vanilla Unity Indie or Unity Pro." That's pretty exciting.

Day 15 - GameSalad trials and tribulations

Where to start? Immediately after my last posting, within minutes, a team mate's level he'd been working on crashed. What's worse was that upon re-opening the project there was nothing, I mean nothing, left of the project. It had been completely erased. And yes, he had been saving his project. Not a scratch of the file was left. It was a completely white screen.

I experienced something less serious a couple days earlier. When the program crashed on me and I went to re-open the project it opened to an earlier stage and I lost a day's worth of work. Beta!

Here's the first bit of advice for this entry, save, but more than save, save different versions so that you can recover something even if it's a few steps back from where you were.

I did download and play with the VML iDevices app from iTunes. I'm excited to have that handy tool with regards to determining X & Y coordinates on my screen. Kudos to VML, out of the UK, for creating this handy device.

I also came across an app called PositionApp that helps to chart an app's popularity and reveal new apps that join the iTunes app store. It looks at all genres. "PositionApp is the world's most in-depth, intuitive app discovery and performance tracking tool available today...it offers instant remote access to all chart positions of the top 300 apps." It looks like a pretty cool app.

Both apps are free and found at iTunes app store.

As for my own issues with GameSalad, I have yet to solve my spawning issue. By every account, my actor should be able to spawn another actor, but alas no. I've thought of all the time I've worked on this and I feel that I've lost some of the advantage I thought I gained in working with GameSalad to that of learning Xcode. I've spent a ridiculous amount of time on this one issue. But, I want to solve this issue. I chalk up my difficulties to my lack of experience, regardless of the quirkiness of the program itself.

Mind you, I've yet to create the mirrored levels I need, by positioning the actor in different places on the screen and performing the same actions. I've yet to discover how to have my actor launch itself to a designated area of the screen, have that area receive an action and change the HUD (heads-up display). All this stands in front of me, while I lose time on the action of spawning.

The cannon ball template most closely resembles my game (a little bit), but I'm having a miserable time reverse engineering it to figure out how it works. I don't need a camera following my actor or particles, so I think there are less detailed things about my game. I return to the grind.

This entry is brief because not much has changed. I trudge on with my building.

Day 16 - Switching Gears in Game Strategy with GameSalad & Project Crash Recovery

Well, little progress on getting my actor to shoot a basket with any random activity involved. I get him to make a basket, but every single time. I have all my colleagues reviewing the problem with no success. After my last posting, I did get a response directly to my email from Jonathan Samn, (CodeMonkey), Lead QA from the makers of GameSalad. You have to hand it to the guys at GameSalad for their customer service. They must scan the Net for anything referencing GameSalad and they came across my little blog. They have been very, very good with responding to questions on the forum or simply addressing a distress signal on some blog. Thanks GameSalad.

My last entry started off as an extremely annoyed diatribe discussing crashing issues and the lack to getting my actors to behave as instructed. Samn contacted me that same day with some possible tips.

He also addressed the issue of recovery for crashed files. His response below:

1. Make a copy of that project file that has broken.

2. Right click or Option+Click the copy of the file and 'Show Package Contents'. This should show you the files and folders in your project.

3. Look in the scenes directory and see if your scenes are still there. (named #.xml) If the y are still there, hope is not lost.

4. In your original project(or a second copy of it), open it up in GameSalad and add a number of scenes equal to the total number you saw in step #3 above.

5. Save that project and close it.

6. Now Right click or Option+Click the project to 'Show Package Contents' and navigate to the scenes folder.

7. Copy all the #.xml from the folder in step #3(first project copy), to the folder in step #6(original or second copy project), overwriting the files.

8. If need be, also do the above steps similarly for the actors folder.

9. If all works, you should now have your project back in a working state.

Unfortunately, my colleague began again his project using the same file name, so that nothing he had worked on could be recovered, but anyone else who reads this may be helped.

Before I write further about my experience with building the game, I think it's a good time to discuss my motivations FOR building a game. Why at my professional stage in life get involved with game bui lding without any prior experience? There are many reasons.

Firstly, my son enjoys them very much and I'm fascinated by them. I want to share in the experience with my son and encourage him NOW to begin developing and designing games because I believe firmly that it is a good career choice. Many of these programs tout how easy it is to build games using their programs. I'm simply putting the idea to test. I believe I'm an excellent example of someone who lacks all knowledge of the process, and that if I can do it than anyone can.

Secondly, I have no illusions that I'll be a fantastic game developer, but I'm not doing this because I strive for that result. No doubt, I'll get better with more practice and I really enjoy it so it doesn't feel like work. No, it's because I'm building a team of people who do know what they are doing. I don't need to be the best; I need to hire the best. The only way I can assess the competency of the people I hire is

to know what they are talking about, to be able to follow along and make educated decisions.

I use the example of a good film director. She will be a better director, if she has worked in every phase of production. If she has produced, edited, and shot film, she will be able to identify issues and integrate ideas accordingly to improve the outcome. This I do know firsthand.

Thirdly, it may come as a surprise that I have no intention of selling the game once we release it. This is not a for-profit product or project. Actually, its primary intent is as a marketing tool. The game is riddled with product placement signage. The game building is being used as a learning curve for my team and marketing tool for the university that I work for. It seems like a lot of work for that expected outcome, but a necessary one in my opinion.

When I finally had my fill of the issues of shooting a basket at varying degrees of release, I decided to experiment with a new tactic. Because I can't seem to get the basketball to randomly move based upon the timed release of the user, perhaps I can have the basket itself move.

The first thing I did was to change and eliminate all the rules, and behaviors and attributes of the basket actor. I cleaned its slate. I placed a simple "move" behavior and nothing happened. I played around with "accelerate", "move to" etc. Nothing. It occurred to me that I could remove the actor from the scene and create a new one. So I literally added a new actor, brought in the same image, renamed it, and dragged it to the scene. Then I went into the actor attributes and added the behavior and it worked immediately.

This is what I proposed happened. The original actor, even with everything set back to zero, somehow clung onto some of the properties of the behaviors it originally had. Yes, I did go into and check that it was movable and cleared any coordinates in the actor's attributes. It seemed if I wanted to work with an actor I needed an entirely new one from which to make something happen.

This goes back to an earlier complaint about the way that behaviors, attributes, and/or rules don't work the same way consistently or as expected. It's arbitrary and inconsistent. I can't be absolutely sure about this because I didn't take the time to do it over and over again. It's not like the crashing issue that remains a constant problem or having an actor spawns itself repetitively and crashing the system, but there is something definitely there.

By the end of the day I was getting tired. I decided to change tactics and rather than having the actor hit a non-moving basket I decided I might be able to create game play by moving the basket back-and-forth.

Yes, it's not quite the same, but desperate times require desperate measure. I knew this had to be easy enough and I thought I remembered the tutorial somewhere. I spent some time looking, but caved in and went to the forum for answers. Immediately, CodeMonkey (a.k.a. Samn) answered with the link to the very tutorial.

“How to Make an Actor Pace Back and Forth?” **http://gamesalad.com/wiki/how_tos:gsc_pacing**

I’ll come full circle and end with, “Strong customer service.”

Day 17 - GameSalad new release: Quirks, Crashes and New Features; iDevices comes through

As I start today's entry, I acknowledge it's been a while since I updated my app building progress. My app is a game for marketing purposes, but I actually like playing my level. When I do play it my heart races as I try against the clock to make points. I love it!!!!!

When I started this journey I was working with the Apple SDK - Xcode point-of-entry into game designing that ate up at least half the time it's taken the entire journey to get here and the only thing I had accomplished with Xcode was reading the HIG, learning a little Objective C, and how to build the infamous "Hello World!" program. Oh, how far I've come with tech help and GameSalad software.

That leads me to discuss the newly released version of GameSalad. I haven't identified much in it that has improved. I believe that the touch rule can be applied and previewed by using the mouse to see if it will work properly. That's a nice change because I had to use a rule like “spacebar” before without knowing if it will work using the touch feature for the iPhone.

I did discover a repeatable crash in the program. If you add a "Play Sound" behavior to two different actors for two different scenes and you go to preview the game the game will crash as it transitions to the next scene. I found that I had to add a sound cue to one actor and one scene, then save a new version and add the sound cue to the next actor and scene and save that new version. Now in the new version the game transitions and cues sounds without crashing.

My team also experienced something strange. One member couldn't get anything to show up that she imported into a new project in either the old or new version of GameSalad. Yes, she brought in PNGs and she placed them into the scene, but when she went to preview nothing would appear. No one had experienced this and we couldn't isolate what was different with her work from our own. Ultimately, she open the Canon template cleared all the information, actors, rules and such to create a shell in which she could work. And it did work. I have no idea what caused this.

As aesthetics are concerned, I found that simple, bright, and clean graphics work well when building a level. At first, I tried to create a crowd scene that initially darkened and cluttered up the scene. I removed the crowd and completely overhauled my set. It's so much nicer to have simple sets with clean objects when building for a 480 x 320 screen. Sound FXs add to the ambience and help to create the illusion of a crowd without having to see one. I believe the psychology behind designing for small screens includes the art of illusion with use of color and sound. It's rather fascinating really. The

ergonomic design of the virtual space is compelling. Simplicity is best. When identifying where actors fit with spawners nothing beats iDevices iPhone app.

Recently, I was contacted by Victor Leach, administrator for Vlmweb, the producers of iDevices. He'd seen my blog and earlier reference to iDevices and just wanted to say hello. Great guy! I'm a big fan of their work with iDevices. I love that app because it really helps orient the designer with the coordinates of X, Y, and Zed (a Brit term) both for the screen and the accelerometer. I highly recommend it. Victor offered his help and assistance should I need it.

I've written this before and I'll write it again. Game designers are a nice bunch of people. Full stop. I have been provided a lot of generous advice as I have fumbled through this process. Vlmweb admin, Victor Leach, is no exception. There seems to be a general sense that people are learning, most far ahead of me, and we're in this together because we share something in common, namely GameSalad and the desire to build games for iPhone. There is a community out there of nice, intelligent, and helpful people. It's wonderful.

I'm getting seriously close to having to stitch the five levels together into one game. My team of five (including me) built our own individual levels on different computers so we are getting to that stage where we need to put it together. Victor generously offered to help us with that next step and he's in the UK! Gotta love technology. I'll likely take him up on his offer, but the team feels we need to learn the steps for ourselves. Game building is such an intriguing problem solving activity that not doing the work feels like you cheat yourself of the learning the process. No doubt we'll search out the information on how to bring it all together.

It's exciting to be at this stage of the journey. We've come exceedingly far in a relatively short period of time. I don't want to jinx the project, but it feels good. I still hold firmly to the opinion that GameSalad, for all its issues, did help expedite our process. And yes, it gets easier the more you use it. I'm still focusing my sights on Unity. I want to wade in its waters and see what I can soak up. Game designing and building is addictive. Seeing and playing with something you built has so much intrinsic worth that exceeds the sole goal of making the almighty dollar.

Game Salad - Repeatable errors & Glitches; Stitching Levels - iDevices creator comes to the rescue

It has been a month since my last entry about the game app process using Game Salad. We have been busy promoting our WEBBY People's Voice Award nomination and I'm super happy to report we won! Hard work pays off. During that time, Game Salad has released two updates to its program. Clearly, they are diligently working on fixing the problems with their software. But to that end, I'll do as I have done before and comment on some problems we have had. I know people don't enjoy criticism, but this isn't that. We're reporting our issues as beta testers in the hopes the Game Salad developers can identify the

problems. It's a feedback thing.

We have found:

- That if you update to the latest version of Game Salad, actors lose some of their settings. Restitution to be exact. Be sure to check it if you discover a problem.
- Sometimes when opening the game, it would regress to the previous session before the game was saved, which means it's not opening up on the latest saved version of your game.
- Some of the attributes within an actor will not engage until the actor is deleted from the scene and dragged back in, and yet some attributes do engage. Example: An actor with a display text rule that has already been put into the scene will still display "Hello World" even after the new text has been entered. So the actor had to be deleted from the scene and brought back in for the new text to show up.
- (Here's a big problem) Some actors who have a collide rule, say with a ceiling or platform actor, will ignore it and jump right through it and either (1) continue to zoom off screen into the ether or (2) come back down and land on top of the actor they were to be colliding with.
    - We suspect it has something to do with the PNG. - More specifically, if an actor box is created and a PNG is placed in it (a picture of a building) and that building actor is dragged to the scene and a collide rule is added to that actor, THEN another actor CAN NOT move through the newly built actor with the PNG.
    - However, if you create an actor box and DON'T place a PNG into that actor box, and you drag the actor box into the scene and give it a collide rule, the other actors CAN pass through the newly created actor with no PNG. Perhaps, the issue is centered around the PNG aspect of an actor.
        - We just tested this and it doesn't seem to hold. PNGs may make no difference to the actor moving through a collide rule.
- Resizing an actor in a scene causes glitches to the game play in preview.
- The "touch" rule didn't work until shutting down the system completely, once I quit out of Game Salad and relaunched it, the program reset and my "touch" command started working in preview mode.
- (Another big problem) After we updated to the latest version of Game Salad, the program restarts and the latest version of GS opens - BUT the sound it is gone. Before we discovered that it is best to shut down the system AGAIN and relaunch to try and reset GS, we lost 3 hours trying to fix the sound. We when relaunched GB the sound came back on but we don't know why.

- IMPORTANT NOTE: When things aren't working correctly close down Game Salad and relaunch it. It seems to clear up some of the disfunction.

There are problems with Game Salad, but with time we have gotten better at working around them and to expect issues. On a positive note, Game Salad seems far more stable than ever before. I haven't experienced one crash. That's an enormous improvement.

Now we're ready to go to the next phase of building our game . Four of our five levels have been completed and we are ready to begin "stitching" (my term) the levels together.

I'd like to send a shout-out of thanks to my new friend Victor Leach of iDevices. I've said how much I appreciated his app iDevices because it helps you find coordinates on the screen of the iPhone. Well, he put together a little video to help us stitch, and as of this week it was still on SendSpace at

http://www.sendspace.com/file/ax4nqe

Stitching looks a little tricky, so I'm especially grateful to Victor for putting the video together. We keep playing it over and over at our end. I highly recommend people to check it out if they too are having difficulty combining levels that have been built on multiple computers.

Dare I say it, this next phase will prove interesting and I will likely have much to write about migrating levels onto one computer and putting the levels together in my next entry. "Onward and upward."

Game Salad: Can't merge scenes from different projects, BEWARE the program crashes!

My last entry dealt with migrating and merging. The team was about to migrate different projects (levels) from several computers into one computer in one main project, but we were having difficulty. We reviewed many times the video clip produced by Victor Leach (iDevices), where he explains the process, however in the end it proved ineffective because Game Salad DOES NOT merge different projects together well.

Why? For starters every project creates each actor with it's own ID tag number. You can't copy the actor into another project with its original tag number. The ID appears to be the number that links the actor to the rules and behaviors within its project and everything in that project. Thus, we've found that the only way to merge the projects is quite literally from scratch, starting over. Here are the steps we took to merge:

- We ultimately opened the two projects up side by side (the original next to the new main project).
- We created the number of scenes for each original project in the main project.

- We entered into the new scene in the main project and imported all the images we needed for the original project including audio clips. (HINT: Think Identical - Everything must be identical between the original project and its copy in the main project)
- We created each new actor in the main project and named it exactly as it was named in the original file. NOTE: It's very important that all actors be specifically named in relation to the particular project. By this I mean, say every level has a spawner. You can't just use the generic name "Spawner" if each level spawner does something different. So we named one Spawner - Baseball Spawner, while another spawner was named Basketball Spawner etc. What this means is that there cannot be the same names used in different scenes, UNLESS it acts the same way for every scene.
- We dragged each actor into the scene according to the way it was laid out in the original project.
- In each actor (not its prototype we didn't work with prototypes) we were able to copy and paste the rules and behaviors from one project into the main project. REMEMBER: You're building from scratch so you have to check all the actor attributes and make sure they are identical.
- Your project needs to work just like the original so don't leave anything to chance. You'll have to add all the game attributes and check your layers as well. MAKE IT IDENTICAL.
- Each scene in the original must also be in proper order in the main project.

The things we have left to do after we merge all the scenes together:

1. Create intro page to the game
2. Create the touch to start level over on the loser page.
3. Touch to save when leaving the game.

We have five levels in our game. Each one follows according to the other, so we are rebuilding the levels in the order that they would appear in the game. When I realized that merging was going to be a significant problem, I questioned how it was that people were able to do this. The advice I was given was to build a project with multiple levels on the same computer working each level after the other in succession.

I watched the various tutorials numerous times. I mentioned months ago that we were building levels of the game on various computers, yet nowhere was it mentioned that merging projects is next to impossible and that Game Salad does not recommend it. I cannot believe we are the first to encounter this problem or to suggest the idea of merging. Yet, nowhere is it explicitly mentioned that merging is not compatible in Game Salad. This is a huge oversight on the part of the Game Salad makers.

I would think people creating multi-level games would in fact work on different computers or on different projects with intention to merge later because it's efficient. Perhaps, a majority of designers

build as one individual on one project at a time. We didn't and now we're learning the hard way.

Clearly, we weren't completely at a disadvantage because the original project does act as a map for the rebuilding process in the main project file. The time was obviously shorter building the level in the main project, but the merging issue wasn't expected thus we did not realize we would be dealing with the time to go through this process.

Another thing we discovered was that the ability to hyperlink comes with the pro version of Game Salad. I can live with putting the Web address at the end without hyperlinking the page, but I felt it should be mentioned here.

Also, since updating the latest version of Game Salad we have had numerous crashes. I had said that the program seemed more stable. Well, I have to eat my words. My first day merging I lost six hours of work when the program just wiped out all the work and left me looking at a white screen. It was so bad in fact that there was no way to recover the project and I had to start over. That day the program unexpectedly quite five times. We did start over and we made 12 renamed versions of the file so that if there was another hard crash we could recover something.

We are making progress merging the files. Jonathan Samn (from Game Salad) has been generous in offering to help us. We are closing in on finishing the game, but I have a small voice going off in my head. The size of our game. What is the size of the game ultimately going to be and is this going to kill our chances as an iPhone game app? We shall see.

GameSalad: Merging Levels and Quirky Nuances to Remember

Well, things are moving smoothly enough in the final stage of building "Bucky's Challenge", our multi-level sports game and first-time app building experience. Our journey began nearly six months ago, taking twists and turns along the way. Ultimately, we are nearing the end using GameSalad to reach our goal.

In the last blog entry I shared the team's surprise and frustration at learning that GameSalad doesn't encourage merging projects because of the original Actor IDs generated specifically for each project that cannot be transferred over into a new project. This is an important fact because it means that each level built in a different project was rebuilt from scratch inside the new Merged Project Folder.

In the process of merging we have encountered some other issues worth mentioning here. First and foremost, we have chosen NOT to update GameSalad to the newest released version. We have found in the past that attributes lose their settings in the process of updating and we must go into the settings and reset them, by first finding those that reset. One example comes to mind, Image settings seem to get dropped. Gravity or density seem to be affected. We can not say that such things are happening with the latest version because we aren't updating to find out. I believe that as a general rule it's wise not to

update during a project build, unless the new version positively addresses a major function of your game.

We have discovered as well that the random spawner rule isn't all that random. For instance the game chooses a number at random initially when the game or level starts, but it sticks with that number consistently so that the spawned object has the same time and space between each spawned object.

However, the most challenging issue we have dealt with has been starting over a level when a person loses and wants to replay that level. The player would lose the level and be redirected back to the game menu. All well and good so far.
The problem was encountered when the player tried to restart their level and the game would take them straight to a win or lose screen. We discovered this happened because scores and lives weren't being reset when the level would start over. HINT: A saving level rule becomes very important here.

As a solution, we created a global (game) attribute integer called "level" set at 0. In addition, we inserted within each level's *game controller actor* a:

- Change Attribute Rule that changed the integer level to whatever number that level was. Example: Bucky Basketball Level 3 - we would change that integer to 3.
    - Right underneath that rule we have a Timer Rule that saves the *game.level* attribute after 0.1 seconds from the start of the level.

To add to the solution, we created a "Continue" button at the menu scene, which is at the very beginning of the game. Within the "Continue" button we added the following:

- Added a Touch Rule to the "Continue" button
- When button is touched **and** game.level = 1
    - Then change scene to level 1
    - and, reset the scores and lives for that level accordingly.

This solved the problem of the game throwing us to a "win" or "lose" screen. NOTE: This was done for each of our five levels.

As a last note, don't forget to address your sound. We enabled "Run to Completion" for each of our levels for the "win" and "lose" screens. However, this option caused the sound to bleed over into the other scenes when scene changed. Simple solution: *uncheck* that option.

Publishing GameSalad Game into Apple iTunes Connect - The Learning Curve

It's been a while since I last blogged because the process of publishing our game has been monumentally difficult. When we were ready to publish we ran into issues in buying the GameSalad license. Apparently, the GS server wasn't working properly at the time and I was sent around in circles trying to purchase the license directly from GS or from Amazon. Eventually I gifted it to myself because it wouldn't work any other way. That proved a mess because I couldn't find straight answers on how to redeem the gift and was charged again when I accessed the gift account on Amazon. It was a mess. Things just aren't straight forward. There seems to be a lot of unnecessary hunting on GS.

Even within the forum, you can not simply plug in a phrase or word and call up entries that deal with the specific issue you are searching. I hope over time Gendai Games will work to improve user navigation and word search on the GameSalad site.

I did mange to get the license and move past that part, only to encounter another frustration when we tried to publish the game to GS.

Initially, we received an error message "No provisioning profiles found..."

Of course, we needed to go to the Apple Developer Center and download our Developer Certificate, Distribution Certificate and our Provisioning -> Distribution tab. Aslo, we needed a WWDR (intermediate certificate) from Apple. Once we had filled out and downloaded these certifications, we put in them into the KEYCHAIN (on the computer ->applications->utilities->Keychain Access).

Then we went back into the GameSalad Program, into our actual game and clicked the orange Publish arrow to upload our game to GameSalad.

When we first tried this (and many subsequent times before speaking with GS) we received an error message. Eventually, we discovered with the help from the guys at GS that our project file was enormous because our images were huge.

HINT: When you build your game save the images you will use as "Save for Web and devices" from Photoshop. It will bring down the image size greatly. It is suggested that actors are below 100KB and backgrounds at 300KB or lower. We knew this going in, but by the time we published the game we had forgotten this golden rule. We believe the size of the file was creating the error message.

We had to go back and re-save ALL our images for Web, which reduced the project size from 198MB to 27MB! That was a huge savings. When we uploaded the file to GS it worked!

Initially, we had problems getting the file sent to Apple because of the file was rejected with an error message saying "The binary being uploaded does not contain an .app bundle". This ate our lunch for about five days because we didn't know what it meant and where to get the .app bundle it wanted.

Apparently, after we published in GS, we received a pop up window with three choices, "View in folder",

"Manage portfolio", "What now?". We clicked on "Manage portfolio" and "downloaded" the published .gameproj file GS provided from the "Manage pportfolio" window, we zipped this file and this was the file we kept trying to publish to the Apple iTunes Connect site.

BIG MISTAKE! Mind you the window "Manage Portfolio" has a big "Download here" sign, which has you believing that's what should happen. There aren't more straightforward directions on GS telling you that this isn't the window you want.

You do not want this *.gameproj* file when uploading to Apple. This *.gameproj* file DOES NOT contain any of the .app bundle documents needed to publish to Apple, thus the reason for the error message from Apple when we tried to publish it.

Instead, we needed to click on the "View in Folder" window, which directed us to our designated folder - we had selected - to save the GS binary file when we clicked the "Publish" button. In that folder, we found a .app file that we had to *compress* into a .zip file. It is this .zip file we uploaded to Apple iTunes Connect. It went through.

Just remember: You do not need to use Xcode in any way to publish your GS.app file. Gendai Games has provided everything you would need to publish to Apple iTunes Connect in that *.app* file. Now we are just waiting for Apple to approve our game and send us the notification that it is available in the iTunes App Store!

At the time of this writing, Google just released its new App Inventor Software for free. It too boasts no coding required when building apps for the Android. I see the possibility of a new project just ahead. Thanks Gendai Games for the GameSalad software that helped us build our first app. It was truly an adventurous journey.

"Bucky Challenge" is in the iTunes App Store!

The Pan American
by Victor Ituarte

After recently winning a Webby, an Emmy, and a Telly, Reel to Red Productions continues to raise the bar with its release of a UTPA-themed mobile game for the iPhone and iPod Touch.

The app is a sports-themed game called “Bucky Challenge” in which the user guides Bucky, the UTPA mascot, through a series of athletic challenges by using the touch screen.

Chelse Benham, the Director of Reel to Red and the game’s Creative Director, led five Reel to Red team members along the journey of creating the app.

“It’s actually a marketing tool to expose the youth to the University,” says Benham. “The idea of doing a game app became interesting because The University of Texas at Austin had a utility app that was popular. It was number seven on the most popular free apps in the iTunes App store. I wanted an iTunes game that could be used to market UTPA.”

The idea behind the app is to expose the public to the university behind the mask of a fun game that is free to download.

What is novel about “Bucky Challenge” is the entire game takes place on the UTPA campus.

“’Bucky Challenge” is a multi-level sports game where we literally took photos of the campus and incorporated those into the game. We figure that if it takes someone anywhere from 30 minutes to an hour to play the game, what they’re seeing are the real University buildings and signage all that time. At the end of the game, the University’s website is displayed on the win page. It’s a new way to market to the youth on their terms,” Benham said.

Not only are users exposed to images of the campus, a few familiar faces will be present in the game as well.

“The game has humor. It’s meant to be funny. I and our other graphic designer, Hilda Del Rio, made the Reel to Red team into bubble-head characters that can be seen in the game. We also arranged a photo shoot with Ricardo Gonzalez who plays UTPA’s mascot, Bucky ,” says Alexis Carranza, graphic designer and assistant director for Reel to Red. “We did a one-hour photo shoot with him dressed in costume and posing in different sport poses. That was a lot of fun because Ricardo was patient.”

Designing the game came as no easy task. The group had no coding or app-building knowledge, and learning the software development kit from Apple to create an app that functions with the iPhone was an enormous undertaking.

“It required learning everything from the human interface guidelines (HIG) for Apple to trying to learn Xcode, which is C++ and Objective-C-based coding. That’s almost impossible trying to learn overnight and still put a game together,” Benham explained.

After about six weeks of trying to figure out Apple’s Xcode, Benham’s son assisted the group by informing them of a program called GameSalad. GameSalad is a program created by Gendai Games, a company based in Austin. **http://www.gamesalad.com/** It is a beta 2D game design program with software glitches that were difficult to solve.

“GameSalad is freeware, but it’s a beta and it’s constantly having problems. It can be hard to work with. There are quirks about it you have to figure out what is causing the problem,” explained James Hernandez, the game’s technical engineer. “I’m sure it’s much easier on a second game. Mind you, our

game is one of the very few being built with GameSalad that's multi-level."

Despite the trouble brought on by working with a beta, Game Salad turned out to be a simpler program to work with than Xcode.

"We were kinda discouraged at first working with Xcode, but once GameSalad came along the game building went from some impossible idea to being a real product in a short period of time," says Hilda Del Rio.

Ultimately, the project was started as a learning curve for the members of Reel to Red.

"There was a lot of trial and error on trying to fix the movements of our characters. Especially in the editing portion when we put the levels together, and how those levels would interact with each other. It was challenging," says Hernandez who initially underestimated the amount of physics that went into the creation of a game.

"It's just another creative product from Reel to Red, where we expand the skill set of the team and showcase the University in a unique way," says Benham.

People can see the screenshots of the "Bucky Challenge" at **http://www.reeltored.com/** or download it for free from iTunes.

GameSalad - Testing game on external device/Pull app from iTunes

Since I last wrote, we did get our app up, but not running, in iTunes App Store under - Games>Sports>Free> "Bucky Challenge". It was so exciting seeing it listed. I was excited until I read the two comments under the game that rated it poorly because it was too hard to play. Mind you, this app wasn't in the store 12 hours before the reviews were in, so I suspect they came from people at iTunes who tested it. Honestly, that's pure conjecture because I have no way of knowing, but I was shocked to see two reviews first thing in the morning when the app got posted some time in the middle of the night.

Needless to say, it was disheartening to see the app get the low ratings. I immediately tested the game on my iTouch and had each team member try to play the game. Sure enough the first level wasn't working properly. I immediately pulled the game from the iTunes App Store and brought it back into GameSalad to review the rules.

**I need to mention here that we did not test our game on an external device *before* we published it. Of course, it seems completely reasonable and due diligence to test an application BEFORE releasing it, but it ran well in the program so we thought nothing of it. However, let me say that it is critical to test the game because the functions in GS don't necessarily work according to plan in the application once

published and running on a device.

In fact, our game uses directional buttons on at least three of our five levels. Two of the levels require the actor to jump over other actors. Our jumping behavior was too extreme, thus not allowing the actor to move over other actors before falling back down on the actor it was jumping. The player lost a life for each actor it hit. With only three lives, the player would lose the level fairly quickly. The jumping action wasn't moving directionally left or right, just up and down. We eventually fixed this, but for the purposes of this blog, here is a tutorial on jumping **http://gamesalad.com/wiki/how_tos:gsc_jump**. **http://gamesalad.com/wiki/faqs?force_rev=1**

We did trace the rules and discovered something missing. We added another rule that allowed the jumping function to work, while the directional keys are pressed and we lessened gravity so that the actor wouldn't fall so quickly. We hope this works. The optional word here is *hope*.

We still didn't test our game. Why you might ask? Because all of our iTouch devices are updated to the latest version of iTunes, while our Mac OS is Leopard, not Snow Leopard. The newest version of XCode works with Mac OS 4 and the external device needs to have OS 3.1.3 or higher to work with Mac OS4.

Because we don't have the newest version of the MacOS we could not connect with the device to download the test version of our game to the device to test the game. Yep, that's right. Though we know it to be extremely bad practice to release a game without testing, we did republish the game to iTunes again without testing it.

Here's another bit of information you need to know. We tried to republish our game to GS in order to upload it to iTunes, but we encountered an error message. We found out that GS doesn't allow earlier versions of the program (before 0.8.8) to be published. We had refused to upgrade to the newest version of GS because we were nervous that our game would lose some settings and not function properly. It has happened before when we upgraded. Fortunately, that was not the case this time.

**Remember, work with the latest version of GS when you go to publish, otherwise you might run into an error message preventing you from publishing.

I am posting a link here to some tutorials that might help you to publish. The following link shows a video tutorial by Tshirtbooth on how to get your game onto an external device.

http://gamesalad.com/wiki/developing_for_iphone:building_gamesalad_viewer#how_do_i_install_the_gamesalad_viewer_on_my_iphone

http://gamesalad.com/wiki/developing_for_iphone:preparing_for_build

We shall see what the future holds for us. If again our game doesn't function properly we will be upgrading to Snow Leopard and hooking up our device to test. Wish us luck.

UTPA media group\'s mantle growing heavy with awards

The Monitor.com

by

Travis M. Whitehead

Click on the editing doors at Reel to Red Productions' Website **www.reeltored.com**, and you'll find seven animated commercial spots for the university.

Click on the screening room at the reeltored.com website, and you'll find trailers for the productions that have won awards for Reel to Red Productions. The program has come a long way since it started in 2004.

"It was an internship program in the office of University relations," said Chelse Benham, director of Reel to Red Productions and TV specialist V for University Relations at the University of Texas-Pan American.

"It was sort of a student production company inside of the office of university relations," Benham said. "Really it was an internship program that I kind of set up to give students applicable work experience in TV production. But we have expanded way beyond that. In the last seven months we won a Lone Star Emmy from the state of Texas, a Webby People's Voice Award, and a Telly."

The tellyawards.com website says that the Telly Awards "honor the very best local, regional, and cable television commercials and programs, as well as the finest video and film productions, and work created for the Web."

Reel to Red Productions won its Telly this year for seven animated commercial spots collectively titled the "Who Knew" campaign behind the editing doors on the organization's website.

"That's the one that won in the professional category. We weren't (competing) against students," Benham said. "That's saying a lot because we had zero budget. We didn't work with any budget. We weren't competing with other students, so we're very pleased with the result."

James Hernandez, 28, worked on the "Who Knew" campaign that won the Telly award.

"All seven campaigns follow certain characters, and there's a male and female character," said Hernandez, a public relations and advertising senior at UTPA. He played the male character in the piece Location, Location, Location, which he developed. .

"It's actually to bring to light a lot of things that maybe some students or potential students knew nothing about the university," he said. "Hence, Who Knew?"

In the “Location”, students and potential students were shown what was available near campus, such as Mexico and South Padre Island.

“Of course, we have another commercial that was advertising the affordability of Pan-Am compared to other universities,” he said. “Well, Who Knew it was that affordable? And so on and so on. They all follow the same formula, we just emphasize different benefits.”

Numerous students worked on different components of the Who Knew campaign.

“I did the audio,” said Oscar Garza, 28, who just finished his master’s degree in English. He plans to teach but also do free-lance work on the side for video production and media. He’s very proud of his work on the “Who Knew” campaign.

“My feeling toward all that is just an actual sense of accomplishment, kind of like hard does pay off,” Garza said. “It feels like it’s not work. That’s probably the best way I can describe it is, working on these things, it’s not working at all.”

While the Telly award was for the seven commercial spots, the Webby People’s Voice Award this year was for the entire website, Benham said.

“It was global,” she said. “It is the Oscars of the Internet. It awards the best Internet content sites, websites, blogs, globally. And we were in competition with Germany and California and North Carolina and beat them out for the People’s Voice Award and we just returned from New York City on that. And then of course we’ve won two Lone Star Emmy’s for two productions.”

Viewers who click on the screening room will find trailers for the productions that have won Emmys for the organization. The trailer for Dead Letter won an Emmy in 2007.

“It was a trigger film,” Benham said. “A trigger film is what you call a short film dealing with some heavy issues that triggers dialogue with teenagers, or at-risk youth. It’s used as a way of creating conversation and our film was 20 minutes long.”

To make this film, she and some students went to a detention center in Hidalgo County and spoke with 18- and 19-year-old detainees who were fresh in prison or waiting to go into the prison system. There were five issues in the film: teen pregnancy, going to college, substance abuse, teen relationship violence, and drinking and driving.

“Those were the five issues of the film and then with the testimonials we kind of show you what consequences there are if you really make the wrong decision,” Benham said. “And then we created an animated comic book.”

Reel to Red also won an Emmy in 2009 for a documentary called Heart of Experience. In that film, the team from Reel to Red followed a study abroad group that went to Europe.

“That Heart of Experience won a competition in Germany because social media and education was the conference,” Benham said. “It had won because of the shock that our students encountered with things

in Europe, especially the Holocaust. And so that documentary is an hour long, and it won the Lone Star Emmy for that. We are very very busy. We do a lot of things."

The group's latest project is the Bucky Challenge game that will soon be released on Apple iTunes.

"It's a 2D five level sports game," Benham said. "It's free. It'll be able to be played by anybody who downloads it. Bucky's our mascot here at the university, and so we used him and the buildings and various places on campus as our images and created the animation and game aspect in this software program for the iPhone or iTouch, so that once we publish on iTunes, anybody who wants to can download it and they're gonna play a game that has our mascot and our university and our signage all over it. We're very excited about that. "

New iPhone, iPod Touch game features mascot

http://www.panamericanonline.com/new-iphone-ipod-touch-game-features-mascot-1.2283873**http://www.panamericanonline.com/new-iphone-ipod-touch-game-features-mascot-1.2283873**

Ever wonder what it would be like to suit up as a school mascot? Now, anyone with access to an iPhone or iPod Touch can take on the role of Bucky, The University of Texas-Pan American's mascot, when they download the UTPA-themed application from the iTunes app store.

The application, which was created by Reel to Red Productions, is a sports-themed game called "Bucky Challenge," in which the user guides Bucky through a series of athletic challenges by using the touch screen.

Chelse Benham, the director of Reel to Red and the game's creative director, led five Reel to Red team members along the journey of creating the game.

"It's actually a marketing tool to expose the youth to the University," Benham said. "The idea of doing a game app became interesting because The University of Texas at Austin had a utility app that was

popular. It was number seven on the most popular free apps in the iTunes App store. I wanted an iTunes game that could be used to market UTPA."

The idea behind the app is to expose the public to the University behind the mask of a fun game free to download.

What is novel about "Bucky Challenge" is the entire game takes place on the UTPA campus.

"'Bucky Challenge' is a multi-level sports game where we literally took photos of the campus and incorporated those into the game," Benham explained. "We figure that if it takes someone anywhere from 30 minutes to an hour to play the game, what they're seeing are the real University buildings and signage all that time. At the end of the game, the University's website is displayed. It's a new way to market to the youth on their terms."

Not only are users exposed to images of the campus. A few familiar faces will be present in the game as well.

"The game has humor. It's meant to be funny. I and our other graphic designer, Hilda Del Rio, made the Reel to Red team into bubblehead characters that can be seen in the game," said Alexis Carranza, graphic designer and assistant director for Reel to Red. "We also arranged a photo shoot with Ricardo Gonzalez of the Visitors Center who plays Bucky. We did a one-hour photo shoot with him dressed in costume and posing in different sport poses. That was a lot of fun."

Designing the game was no easy task. The group had no coding or app-building knowledge, and learning the software development kit from Apple to create an app that functions with the iPhone was an enormous undertaking.

"It required learning everything from the human interface guidelines for Apple to trying to learn Xcode, which is C++ and Objective-C-based coding. That's almost impossible trying to learn overnight and still put a game together," Benham explained.

After about six weeks of trying to figure out Xcode, Benham's son assisted the group by informing them of a program called GameSalad, created by Gendai Games. The Austin-based company's freeware seemed perfect for the team, but it came with software glitches that were almost impossible to solve.

"GameSalad is freeware, but it's a beta and it's constantly having problems. It's hard to work with. There are quirks about it you have to figure out what is causing the problem," explained James Hernandez, the game's technical engineer. A beta is software that is in the second stage of testing and is used as a prototype, or preview for potential buyers. It is often free or at a discounted price because it typically contains glitches. "I'm sure it's much easier on a second game. Mind you, our game is one of the very few being built with GameSalad that's multi-level."

Despite the trouble brought on by working with a beta, GameSalad turned out to be a simpler program to work with than Xcode.

“We were kind of discouraged at first working with Xcode, but once GameSalad came along, the game building went from some impossible idea to being a real product in a short period of time,” Del Rio said.

Ultimately, the project was started as a learning curve for the members of Reel to Red.

“There was a lot of trial and error on trying to fix the movements of our characters, especially in the editing portion when we put the levels together and how those levels would interact with each other. It was challenging,” said Hernandez, who initially underestimated the amount of physics that went into the creation of a game.

“It’s just another creative product from Reel to Red where we expand the skill set of the team and showcase the University in a unique way,” Benham said. The group recently won state-wide and international awards for exhibiting the University in a documentary, website, and commercial campaign including a Lone Star Emmy, a Webby People’s Voice Award, and a Telly Award.

Benham documented the entire process of creating the game app in a blog called “The Novice App Builder,” with the hope of aiding others who would like to create a game using GameSalad. The blog can be found at noviceappbuilder.blogspot.com. Screenshots of “Bucky Challenge” are available at www.reeltored.com.**Reel to Red Productions website** The app is free to download from iTunes.

http://toucharcade.com/games/bucky-challenge **Touch Arcade**

Impossible for us to test our GameSalad game before releasing it

We are still in throes of publishing our game without errors. In my last entry, I fully acknowledged the need to test a game before its release to avoid issues, but despite our best efforts we have not successfully tested our game before releasing it. Below is our account of our efforts and the surprising conclusion we arrived at for why we have been unsuccessful. The following post was written by our Technical Engineer, James Hernandez, recounting the trials and tribulations of our efforts to test the game before its release…

After our second publish to the App store, and after Apple's OS 4.0 migration, our Gamesalad game once again had issues. Some of our images were replaced by white blocks (later we attributed the problem to an image sizing issue). We knew we had to get our game tested on a device and not just the simulator, but Xcode Base SDK did not support our iPod Touch version 3.1.3.

To solve this problem we purchased the Snow Leopard upgrade and installed it along with Xcode 3.2.3 and IOS SDK 4.0.1. After all this was completed and our iPod Touch was connected, Xcode did recognize the device and had no options for version 3.1.3 device support. We made sure that Xcode was properly installed and was located in the MAC HD/Developer/Applications.

We then knew that we needed to get a hold of an iPhone with OS 4.0 installed. Once an iPhone was obtained, we registered it in our iPhone Provisioning Portal under DEVICES. Once this was completed, we then navigated to PROVISIONING and clicked on the Development tab. We modified the provisioning profile we created for testing to include the new device. The profile was then downloaded onto our computer. We double-clicked the downloaded profile which opened in Xcode.

Once Xcode opened, our new iPhone device was recognized and displayed in the Xcode Organizer window. We clicked on the button asking us if we wanted to use the device for developing and then selected the provisioning profile we just downloaded. We closed the Organizer window and opened the GamesaladViewer.xcodeproj file we downloaded from Gamesalad.com. The GS viewer opened within Xcode where we then had to make a few changes in the Info.plist category listed in the Gamesalad Viewer folder.

Note: The Bundle identifier within Info.plist needs to match the Bundle identifier that was created in the provisioning profile in the iPhone Provisioning Portal.

We right-clicked Gamesalad Viewer listed in the left column of Xcode and selected Get Info. Once in the project info window, we made sure that the Base SDK listed matched that of the device or simulator we were testing our game with. We scrolled down to Code Signing/Code Signing Identity/Any iPhone OS Device and selected the developer profile we created in the iPhone Provisioning Portal.

Once back into the main window of Xcode, we made sure that the Overview drop down menu showed that Device - 4.0 was selected and clicked Build and Run. The GS viewer was then successfully installed on our device and was waiting to connect with the GS creator.

However, after al of this preparation to connect the device to the computer for testing, our game couldn't be tested. It opened within the GS creator, but our device never connected with it. Our device stayed at "Waiting for Connection" for a while then displayed the following error: "Gamesalad Creator not located on network." WHA?!

We were so close yet so far from getting our game tested on a device. After upgrading the Mac OS to Snow Leopard, after locating an iPhone with 4.0 installed, after connecting the device and setting up the requirements to "Build and Go", we achieved nothing.

We searched the net and GS forums for help, but ultimately we concluded that the issue might be with our University's wireless network. Our computer and device were both connected to the same wireless connection, but somehow the GS viewer did not recognize the GS creator on the network. The device did not have phone service, but could still access the internet through Wi-Fi.

We tried a numerous options. We created a network from our computer, which the device recognized, but could not connect to the internet. We tried to connect the device through the Bluetooth function, but the device refused to be "discovered." After reading a few more GS forum posts, we concluded that

the issue might be with the University's router and/or firewall settings.

In the end, our game is awaiting its third review for the App Store without it being tested on a device despite our best efforts.

Boom! It works! GameSalad success.

Our previous entry addresses the difficulty we had, and our conclusion that we would not be able to test our game on an external device. It was so frustrating. We had installed and reinstalled Xcode, but to no avail. James Hernandez, our technical engineer, picks up the thread from here...

With some help from Jonathan Samm at GS, today was a new day and another attempt at getting Gamesalad's viewer to install on our iPhone device and get it to talk with the GS creator.

I consulted with John about our issues. He wanted to make sure all our software and devices were updated. I assured him that Xcode had been reinstalled numerous times, new networks created, and no luck. After confirming that everything was updated, the next step was to once again build and go to see if anything had changed.

The GS viewer installed onto our device but, once again, the device could not locate the GS creator on the network. We confirmed that our device and computer were recieving Wi-Fi from the same campus network, but the two machines would not talk to each other. We tried to get the simulator to work, but some errors were found. NOTE: The simulator had been working fine the day before, so these errors were odd.

Jonathan then suggested we install Xcode again. The next step was to delete all Xcode and the GamesaladViewer.xcodeproj files and re-install them. This is important, I can't stress it enough to reinstall Xcode when the device reads, "Waiting for Connection" for a while then displays the following error: "Gamesalad Creator not located on network."

After this was completed, our provisioning profile was re-downloaded and installed into Xcode. I clicked "build and go" and the simulator was up and running. Now it was time to try the device again. After the Xcode drop-down menu was changed to device from simulator I pressed build and go and the viewer was installing onto our device. After the viewer was "waiting for connection" for a few seconds the "GS creator was not located on the network" error displayed again!

John then asked if I tried creating a network within the computer for the device to connect to. As you all recall from yesterday's post, we tried unsuccessfully to create a network which I let John know. I tried once again to create a new network. BOOM! the network connection worked! A new day brings new

surprises.

NOTE: It is important to note that nothing had changed about the process from one day to the next. These were all the same steps taken previously, but for some reason they worked today whereas they had not before.

I pressed build and go and the viewer on our device located the GS creator and we were off playing our game on the device.

Our game played through with minor issues that we could not see in the simulator, but showed up on the device. This is important: The game you build in GS will not behave perfectly in the device. There are subtle differences that affect the game in the device.

The lessons of the story would be, if all else fails clean the slate and start fresh.

Boom! GameSalad device success at **http://ping.fm/A6oRY**

GameSalad iPhone game released - what we learned along the way

Well, I believe I am officially at the end of my first mobile game building experience. What a journey it has been. I started the project in January and now nearing the end of August, *Bucky Challenge* version 1.4 is out and ready for download in the iTunes App Store.

You may have noticed that I wrote version 1.4. It is true we did have corrections and it lead to four versions of our app being released. Much of it because we had great difficulty loading the game in an external device for testing. I have made the point of testing in the previous entry and I do know better than to release a game before testing it. However, we were under a time crunch because we had a news article about the game approaching publication.

Now as I look back over the eight months, I have some final thoughts regarding the things we learned while building the game.

1. We did choose to use command buttons to control actor actions in three of our levels. I believe this to be a less effective use of the iPhone environment. I would not recommend placing command buttons within a game for iPhone for several reasons. (a) Buttons eat into the screen real estate. The size of the buttons need to be large enough to accomodate fingertips of all sizes. (b) Buttons are a bit difficult to navigate on the device. (c) iPhone uses touch and accelerometer features that do work best on the iPhone, thus buttons seem antiquated. (d) The use of the buttons did cause in one case the iTouch device to turn off because of the location of the power button on the device being where the user

places her hands. While holding the device, the player can accidentally click into a black screen saver mode, thus interfering with the game play. We never considered the possibility when we made the app and it didn't show up during the testing of the game. Fortunately, it is rare.

2. It is extremely important that all PNGs be "save for web" to reduce their size. Actors and elements should be less than 100kb with backgrounds no more than 300kb.

3. DO NOT build the game on several computers. We hade five team members each building a level on their own computer believing we could migrate to one computer and stitch the game together, but that does not work. GS provides original actor ID numbers for each element individually dedicated to one project. These do not copy over into a new project. We had to REBUILD each level and import all images into a new game to stitch them together.

4. GameSalad has its quirks and acts much like a beta, so do plan to experience some delays, crashes, and frustration. It takes a while to understand the game building psychology behind the program. Whereas, the GS forums do provide fairly quick responses to questions, the forums are not organized in a concise way by topic, issue, problem or error. It's time consuming navigating the forums for answers addressing specific problems. GS does have a number of tutorials (print and video) , but there is no book on the software and many issues are difficult to find answers for. Beware, some answers provided in the forums are obsolete due to GS upgrades and may no longer be relevant or accurate, but they are still listed among the entries.

5. Do test your game on a device before publishing it because the game will not necessarily work according to the computer simulator generated version. It may work perfectly on the simulator, but not on the external device during testing. Furthermore, we experienced tremendous difficulty getting our game to play on an external device. Initially, we thought it was due to the Mac OS and Xcode versions not agreeing with each other, however, after considerable time installing Snow Leopard and the latest version of Xcode we were still unable to get the game to play. As a work-around we uninstalled Xcode and reinstalled Xcode several times. Eventually, it worked, but we never understood how it corrected itself or what was truly the problem.

Ultimately, we are very pleased with our game. It is the first of it's kind (sports game)for our University. Each team member can take pride in the knowledge they have contributed to the great iTunes App Store. There is a true sense of accomplishment seeing the game on display in the store, watching other people playing it, and learning from people we don't know that they downloaded it. That's so exciting. I believe it was a successful project, well worth the effort, and GameSalad was a good program choice.

In addition, we want to thank the people at GamesSalad, especially Jonanthan Samm, for helping us through some tricky issues.

We encourage and do hope that you, the reader, will download our game and provide feedback at iTunes or here on this blog. Just plug in B*ucky Challenge* in the search bar and Viola!. Please, cull through

this blog's past entries when building your GS game for some troubleshooting advice and examples of challenges we faced. I truly hope this blog is of use to others.

As I turn my attention to the future, I have decided to build another app. I am looking at building a social network game for Facebook. Yep, you read that right. Initially, I spent some time looking for a SDK program specifically designed for or by Facebook, however, it is woefully sparse. That came as a shocker. Let's not forget, I am not a programmer by trade and the primary point of this blog is to document my experience as a *Novice App Builder*, using turnkey programs to build apps.

I have found one program. The Appainter program at **http://www.appainter.com** looks promising. If I choose to use that program it will become the next focus of this blog. I believe Facebook apps are extremely interesting to people and there doesn't seem to be a lot of straightforward information. (I could be wrong.)

So, onward and upward.