

# Getting Out Of The Garage: Managing Teams, Legal Fees, Cloud Servers And 97 Other Things To Manage Growth

**Iman Mostafavi**  
Co-Founder & COO  
**Limbic**



SMARTPHONE & TABLET GAMES  
SUMMIT

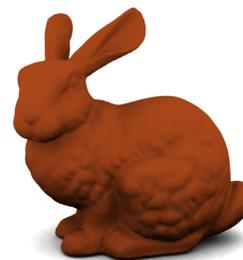
GAME DEVELOPERS CONFERENCE

SAN FRANCISCO, CA  
MARCH 5-9, 2012  
EXPO DATES: MARCH 7-9

**2012**

# A Brief History of Limbic

- **3 Co-Founders**
  - Technical Backgrounds (Computer Graphics)
- **Entrepreneurial Spirit**
  - Startups, Side Projects



Founding team is all technical with backgrounds directly applicable to game development.

No “business guy” founders, but each founder has developed business sense and has previous entrepreneurial attempts.

First worked together closely in an entrepreneurship course w/Arash

Met our CTO who was a visiting scholar (Volker) and got to know him by taking a surfing class together in sunny SD.

We had no formal “business plan,” just went straight to making a prototype

Incorporated online in February of 09 once we realized we were definitely going to give working together a shot.

# Overview

- Getting Started
- "Success"
- The Next Threshold

What to expect from this talk:

I'll be running through some of the challenges we faced as we went through the various stages of growth.

Some of you in the audience will likely have been through some of the same experiences we've, and some of you will have been through completely different ones.

# Operating Virtually



Company presence: concentrated in California, but team members around the world (Germany, the Netherlands, and New Zealand).

Advantages: Environmental benefits, zero commute time, no need to relocate ever, access to a global talent pool. Employees can work in their own ideal environment, they can control their own distractions. You call them pants, I call them oppression!

Disadvantages: Need to get away from distractions (kids, significant other, errands). Coordinating meetings, communicating ideas electronically can be more cumbersome. Line between work and personal time is blurred which can be very difficult for some.

We have made this setup work really well for us, but it may not work for everyone.

# Getting Started

- Learning the Tools
  - Xcode, Objective-C
- Working together as a team
- Ambitious Goal
  - Build a complete game in 8 weekends



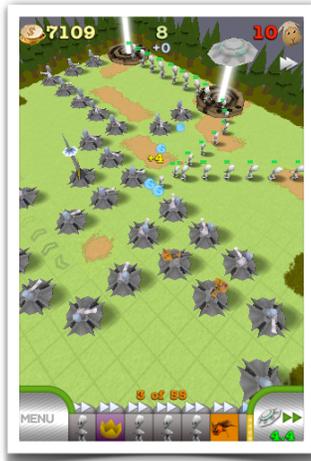
TowerMadness Prototype

First we had to get up to speed with the development tools, neither one of us had developed much on the Mac before. Learned the tools and how to work together as a team.

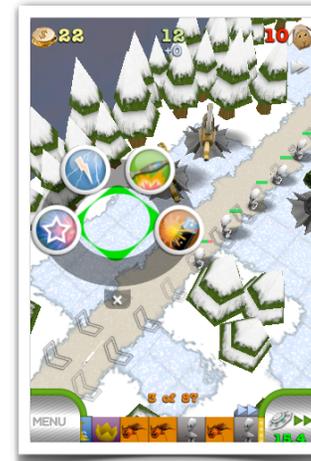
TowerMadness was inspired by flash games such as Desktop Tower Defense, and the Warcraft Tower Defense custom games we had all played. At the time there were only 2 really bad TD games on the App Store.

We set off and developed fully functional TD game with 'programmer art' in about 8 weekends

# Differentiator



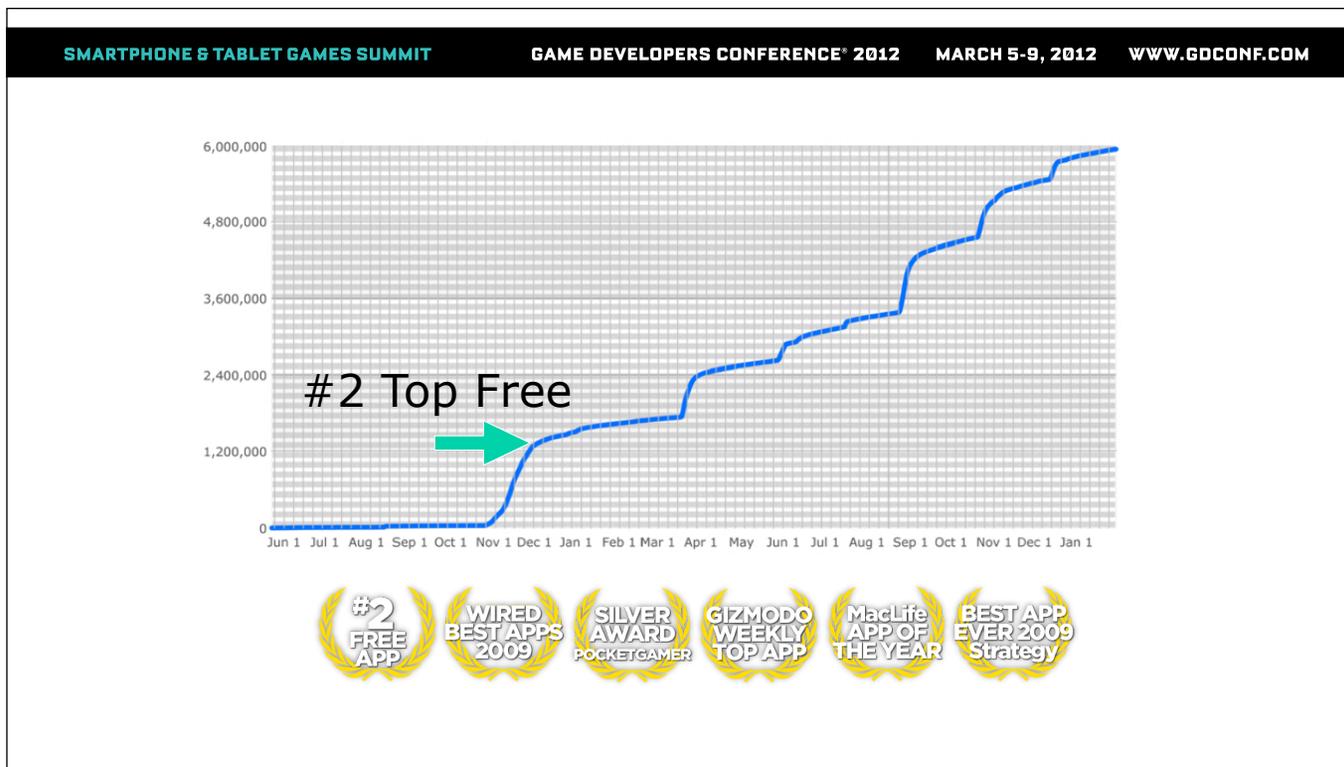
TowerMadness



What would be our differentiator in the marketplace: Let's do it in 3D!

6 months part-time development, released May 2009

3 guys with no external funding, \$0 marketing budget



First 5 months - Strong initial launch, but plateaued quickly to a less than sufficient level to sustain any of us full time.

Updates Followed

Free Full Ad-Supported Version (Nov 2009) - Reaches #2

Revenues hit levels supporting full time operation.

A keep point to end with here (and a key underlying theme for the whole presentation) is at this point business became about MANAGING not RELEASING.

# 1,000,000 Downloads

First million users are the hardest to get. Besides being a nice round number, it's a point at which you can be pretty clear that a market exists for your app and you can build a business around it.

Once you pass this threshold you will encounter several new 'problems' you didn't have to worry about before.

Server scalability: Servers need to scale with the user demand.

We had Leaderboards before Game Center (now iCloud + GC is all you need)

Optimizing Ads: Ad networks can't handle your requests. Have to use multiple networks for best monetization. We had to solve this problem before "mediation" services (AdWhirl) and the like existed.

Handling customer feedback: Rare game issues not rare anymore, lots of support inquiries would result if there were any usability issues or bugs.

# Server Scalability



Game Center



iCloud



Overview of the most popular options for scalable server solutions.

Game Center + iCloud (iOS)

Google App-Engine, Amazon, Heroku, etc. Lots of options.

We were early adopters of App Engine, but were burned by the price hikes last November.

# Optimizing Ads

GREYSTRIPE



Google™

admob



For ad-supported games, it becomes worthwhile to properly manage your ad inventory. Experiment to find the right ad-units for your app

Interstitials generally worked better than banners for TowerMadness

You may exceed the fill capacity for any one ad network

Managing the irregular fluctuations in ad network performance (mediation services now available)

When you have a large enough install base, it's like running a side business (sell your ad space with Burstly like some apps like Angry Birds or Words with Friends do)

If you have more than one product in your portfolio, house ads for your own products can generate more returns than third party ads

Big point: you are in the success phase of your company, managing your install base is not a side business IT IS your business. And managing ad revenue is part of that.

# News & Alerts



Congratulations, you now have enough users to do lots of new cool things! Do you have a way to communicate with your players directly so you can make the most of the user base?

In game news and alerts help you mobilize your users to:

Do things for you ("Like" on Facebook)

Take surveys to find out more about what they want

Alert them about promotions for other apps you might think they would enjoy

Every link to App Store should be an affiliate program link. Don't underestimate how effective it can be.

Inform players about any important issues, direct them to find solutions faster and with less work on your side.

Well integrated news feeds will generate higher click through rate than generic banners if done right. Trick here is to not abuse these feeds otherwise the player will eventually treat it like any other ad (especially important to be considerate in a paid app).

# Customer Feedback



Zendesk



Tender Support



## Knowledge Base



Search

## Limbic Support

### Welcome to Limbic Support!

Limbic Support Dec-20 2010 • Limbic / Announcements

Which of our games can we help you with?



Nuts!™



TowerMadness®  
TowerMadness® HD  
TowerMadness® Zero



Zombie Gunship™

**How you deal with your customer feedback is another important way to differentiate yourself from the competition.** Where others might outright ignore emails, or outsource them to a 3rd party, we handled all customer feedback directly and often by one of the co-founders & engineers. If there's a usability issue or a bug, it's already at the highest level and can sometimes be fixed by the person who reads the request.

**Developers need to make handling customer feedback as efficient as possible,** especially if they have free apps.

Use Help Desk services to organize and track tickets

Auto-responder with a link to the Knowledge Base upon receipt of a request to confirm and help as quickly as possible

Macros to speed up response entry when human intervention is required

These tools can even integrate Facebook and Twitter posts, email, and centralize all of your inquiries

API allows custom automation of advanced filtering

Customer satisfaction rates can be measured so you can quantify how satisfied your customers are with your support level.

Quick overview of what's out there: Mention Zendesk & Tender, as well as Get Satisfaction - A neat service which makes it easy to let your most involved users help each other to reduce your support load

# Surveys



Demographics



Bugs

Getting proactive about customer feedback for updates

Do you really know who plays your games or are you just guessing?

How about how often your games are crashing?

Find out which new features players want

Don't waste time with a generic question like "Do you want more free content?" ...everyone will say yes and the results are not particularly actionable. Try to find out what it is they want. More maps, more levels, different input schemes?

Don't rely only on people writing in with complaints or reviews on the App Store, only a tiny fraction of players do that. Most will just stop using your app.

SurveyMonkey is pretty good for creating surveys to find out this information. Design your surveys carefully to avoid bias in the questions. They have template questions you can modify for your own purposes.

# Crash Reporting



# Times Occurred	# Users Affected
435	168

The iTunes Connect provides crash logs but they're they're delayed and also require the user to sync their device with their computer in order to collect the logs.

Use a LIVE crash reporter. We tried a few different ones, they're largely the same. Crittercism works well so far.

These services can 'symbolicate' your crashes automatically and tell you which line the crashes occurred.

Detailed stats show you how many users were effected, crash frequency, so you can narrow in on the most critical bugs to fix first

Taking this extra step to be aggressive about crash reports contributes to the management of your install base. You want keeping them happy and playing.

# Distributing Builds



TestFlight

Testing is very important. Having good testers is essential and getting the most out of them starts with distributing your builds in a hassle-free way.

Emailing builds to testers to sync to their devices with iTunes is the old fashioned way to do it. It takes way too many steps to install. Half of the testers probably won't even install it.

Use a service like TestFlight if you're not already

It's free, plus it lets you track exactly which one of your testers are actually installing the builds, so you can kick them out if they're useless in your next beta!

# Quality Assurance



With TestFlight everyone you invite will register every iOS device they own, but only accept to 1 or 2 devices per person so you can actually get enough testers on your developer account before hitting the 100 device limit. Yes you can create more developer accounts but that's a hassle in its own way.

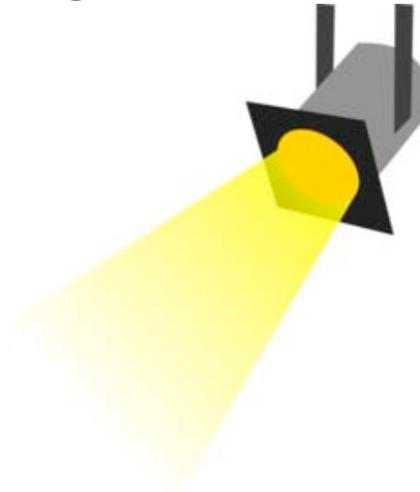
Not all testers are created equal. Unfortunately most of them will not play through your game. They are easy to "burn out" with too many test builds. Unless they have some incentive to keep testing, by the 5th build you would be lucky if more than a handful install it, let alone play it seriously.

Developers should do a LOT of the playtesting. You should be able to find all of the major problems internally, don't rely on volunteer testers to do this. Release to testers to uncover rare issues.

Outsourced QA services available, we've never used them but have heard good things.

# Living in the Spotlight

- Acquisition
- Lawsuits
- Opportunists



Overview slide for the ways becoming successful makes you a target for both potentially good things like acquisition, and bad things like lawsuits, copycats, etc.

# Suitors



Suitors will come knocking once you have success. They will ask lots of questions about your revenue, userbase, various statistics, your “exit strategy”

Decide between your founders what your policy is on revealing any such confidential information

Once you release confidential info you can't take it back

No point in releasing your numbers if you're not going to go through with deals

Unless you took a lot of VC funding and the investors are breathing down your neck, you probably don't need to have an exit strategy in mobile. It's still growing at breakneck speed

Side point - don't take investment if you don't need it.

Potential acquirers may make vague promises about how they will help you push your apps. Publishers may also suggest you let them “publish” your app under their brand for the same reasons. They may even insinuate that if you don't sell to them sooner or later you'll get crushed by them.

# Legal Challenges

- Intellectual Property



TM



Obviously, don't use existing trademarks

Check sources for all of your code and art

Adhere to open source software licenses

Fonts are software too, don't just embed them!

Check any app names, and if you're just starting, your company name on USPTO

Keep an eye on your contractors

Ask actual lawyers don't pretend you are one and rely on just internet advice

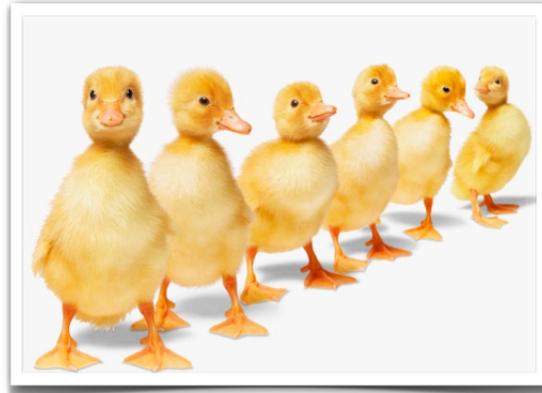
You have a lot more to lose now.

# Legal Challenges Continued



- Acknowledging Unsolicited Suggestions
  - Fan feedback can be dangerous...
- Contractors
  - Ensure they're not doing work for your projects using another company's resources
  - Are not bound by other agreements
  - Be clear and explicit about ownership and compensation

## Get Your Ducks In a Row



Before you go any further...

Have you set up the right corporate structure?

Do you have founders agreements in place between all of the founders and any other shareholders?

Have you had everyone you've worked with sign something that transfers the ownership of their work to the company? This can be a major impediment to acquisition if that is your company goal.

Make sure your license agreements are in order, if you're licensing any technology or software.

Are you doing your accounting and payroll properly? One of the first things you can outsource to focus on the core business activities.

# Opportunists



Temple Run



Temple Jump



Temple Gun

Trademark your game names and takedown those who infringe immediately

Apps will come out that capitalize on the success of your app name or likeness  
Shady people will squat the domains you didn't register

They will set up unofficial fan pages, twitter accounts

Sometimes they will even unofficially "port" or clone your apps to other platforms for you

We've had to have no fewer than three fake Zombie Gunship apps pulled from the Android Market.

Fortunately the official Android Market and Apple are pretty good about this.  
Legitimate takedown requests acted upon pretty quickly.

# Clone Wars



If they can't beat you...they'll clone you.

The sincerest form of flattery?

Figure out how you will deal with it. It's more likely to happen if you have a successful game.

Are you going to sue? Appeal to the community?

Unfortunately this trend is going to force indie developers to be more secretive about their upcoming projects. You can't stop the clones, but at least you can avoid giving them a head start.

## Next Steps



Many directions your company can take after initial success  
Continue with supporting one app? Have you maximized your return on the existing title?  
Creating more products? Should they be derivatives or spin-offs of the first one, or something completely new.  
It all goes back to goals and what type of business you want to be running.

# Working with Established Brands



Once you have success on the app store, you may receive offers to work with established IP. We gave it a shot with Grinchmas, developed by Limbic and published by Oceanhouse Media in 2009

Pros: You get to work on something that has a lot of name recognition. If pulled off successfully, can lead to even more contracts to work on big IP. Can build your reputation as a developer, and add predictability to your short term revenue stream, help you meet new people in the industry.

Cons: You may not end up building your own brand, it may purely for monetary benefit. Short-sighted. You may not have the freedom to implement your vision. You may have to constrain your creativity to stay within the brand's guidelines. Know why you are doing it and judge it on those standards. Know what you expect to get out of it.

## Work For Hire



You could run a highly profitable shop just doing projects for brands, but you have to decide if that's what you really want to do.

It may be worth it for you to work with an IP you've always loved, but otherwise it's a slippery slope that can become addictive and distract you from bigger opportunities.

You are choosing to be a **service business** at that point and you give up a certain amount of control of your destiny.

# Publishing & Investing



If you're successful the opportunity may arise to invest in others' ideas.

Star Farm, released in 2010, developed by another firm. One of the first 3D farming games on iOS. Proved to be a misstep for the company because we didn't factor in the difficulty applying our experience to other peoples' projects.

Working in a genre where nobody in the company has a real passion (farming games)

Lack of control, limited by your developer's resources which may not be fully transparent

The moving target of social games resulted in the game being released with not enough content, too many bugs. The only app we had to stop supporting, and it quickly faded away.

Lesson learned: If you're not an experienced publisher or an investor proceed with caution. They are different businesses altogether. It may be smarter to simply focus on what you've been good at so far. If it has been building games, focus on that and keep doing a great job.

## Back to Basics



Following the failed publishing attempt, we went back to basics and focused on developing another original franchise.

We wanted to do this to grow our userbase while expanding the demographics of our players.

We saw a gap in the market for this type of game.

We stuck to the following basic design criteria: casual game, something we would want to play, non-violent, and appeals to everyone.

# Nuts!



Decided we'd try to create an action-arcade style game with a tilt input mechanic

Simple game, nice nature theme

Launched into the Top 50 as a paid game, eventually settled lower down in the charts but still has been pretty successful with about 5 million App Store downloads to date.

# Virtual Currency



Nuts! was our first game with virtual currency

It features one time unlockable items and consumable items

Seems to monetize better than fixed price purchases from our experience

Easier to deal with from customer service point of view. People seem to always have trouble restoring specific purchases, but if it's coins you just give them coins and problem solved.

Flexibility to do things (that we don't do) but many social games do including daily bonuses for coming back. Incentives for doing things (maybe taking a survey), Liking on FB, etc.

Gives you the ability to manage the game (more dials to turn) to drive behavior.

## Incentivized Installs



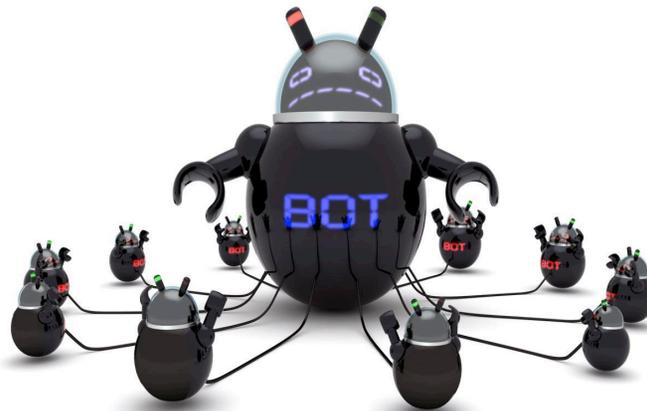
We actually had a paywall integrated in the Nuts! but had to rip it out just before ship as it was no longer permitted.

Apps have been pulled for incentivizing users to download other apps.

Just say no to shady ideas. It's not worth it in the long run and it's bad for the long term health of the ecosystem.

Not allowing incentivized installs was bad in the short term as it prevented some apps from generating some extra revenue from customers who otherwise never would have paid. However, all of that money was coming from big companies who would ultimately use it to influence the rankings at the detriment of indies with no marketing budget.

# 3rd Party Marketing



Some firms promise guaranteed Top 25 placement for cash.

Rumored to use bot-nets.

Apple has warned against using them, and can result in termination of your developer account if they find out.

Do your homework. If it sounds too good to be true, it probably is.

It's your reputation on the line, not theirs.

## Legitimate Marketing Tactics



Going FREE, it's a simple and powerful strategy and it works better than ever

Give your whole game away for free. If nobody is willing to download it when it's free, you also know that it's time to move on to the next project

If you have in-app purchases you might even monetize better



Temple Run



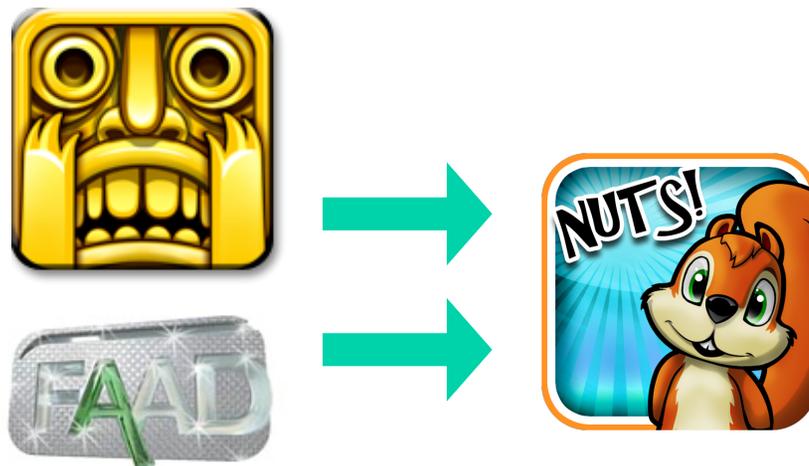
Jetpack Joyride

Released August as a Paid game  
Switched to Free in September  
Updates along the way  
Climbed all the way to the #1 spot  
Over 40 million downloads

Launched September as a Paid game  
Went Free in December  
Still FREE

Both doing well in the Top Grossing

## Cross Promotions



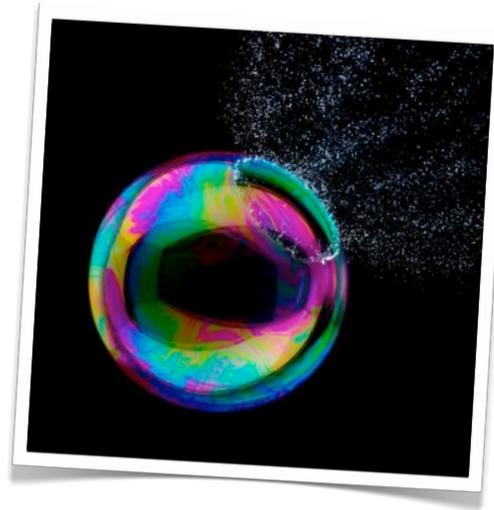
Espouse the benefits of cross promo-ing with other indies. It's not losing customers to them, it's mutually beneficial.

Promotions from other games. Traffic trading. Traffic is a form of currency.

Promotions from apps that are just for promoting apps such as FAAD can be quite effective. Essentially it's a cross promotion that you can pay for if you don't have the personal connections.

Choose good partners. Peers are best because you're more likely to be able to have a balanced exchange.

# Expanding Safely



We believe that as a startup, you should avoid hiring people for as long as possible. Paul Graham essay - "hiring people is the worst thing a company can do."

"People are a recurring expense, which is the worst kind. They also tend to cause you to grow out of your space, and perhaps even move to the sort of uncool office building that will make your software worse. But worst of all, they slow you down: instead of sticking your head in someone's office and checking out an idea with them, eight people have to have a meeting about it. So the fewer people you can hire, the better."

It should only be done when you need to do something but you can't do it.

Beginning of 2011 we wanted to do another new game, but we were going to stretch the founding team too thin developing a third franchise while supporting the existing two games.

Top notch developers are actually pretty rare, and they tend to either already be employed or have their own companies

How to attract and work with talented people?

You could try to offer them equity and give them a salary, but this is really too safe for a really early stage startup. You still want founder grade talent.

Artists are easier to hire contractually by the hour or asset. Programmers not so much.

## New Game, New Pie



Solution: give everyone involved skin in the game

That means revenue share

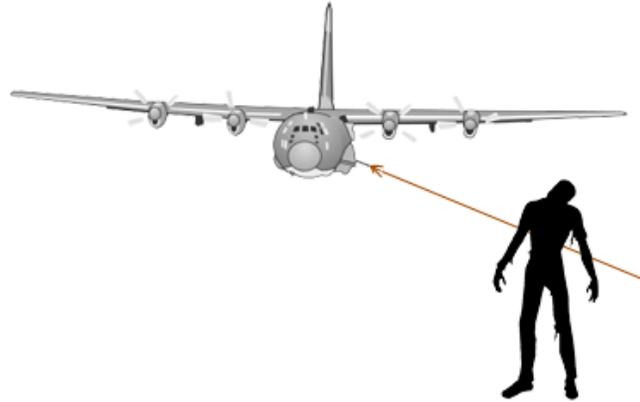
Everyone gets a “piece of the pie”

High up-side potential. Everyone’s interests are aligned to make the game as successful as possible.

Weeds out people who in it for the safe paycheck. This is not safe at all. You can get hot-shots.

People are never as committed as when they have skin in the game

# Zombie Gunship



We decided to build a game for the first time with new engineering partners on this revenue sharing model.

Game itself was inspired by a Call of Duty mission, adding zombies to make it more awesome.

Goal to bring the best part of a “hardcore” game to the masses

Distinctive visuals – B&W – FLIR

Highly satisfying – explosions and gibs

Easy to use controls



Came out in July

Was the #1 Top Grossing iPhone game for several weeks, still hangs out at the top of the charts

Despite only being out half of the year, it was the #39th Top Grossing App for 2011

Order of magnitude more successful financially than our previous games

Why do we think we had the success? We were part of our own demographic, we invested in things that mattered, art, sound, polishing the gameplay. We didn't get greedy with the monetization.

# First Update: Going Universal



Small team allowed us to move very quickly

We received lots of requests for iPad version

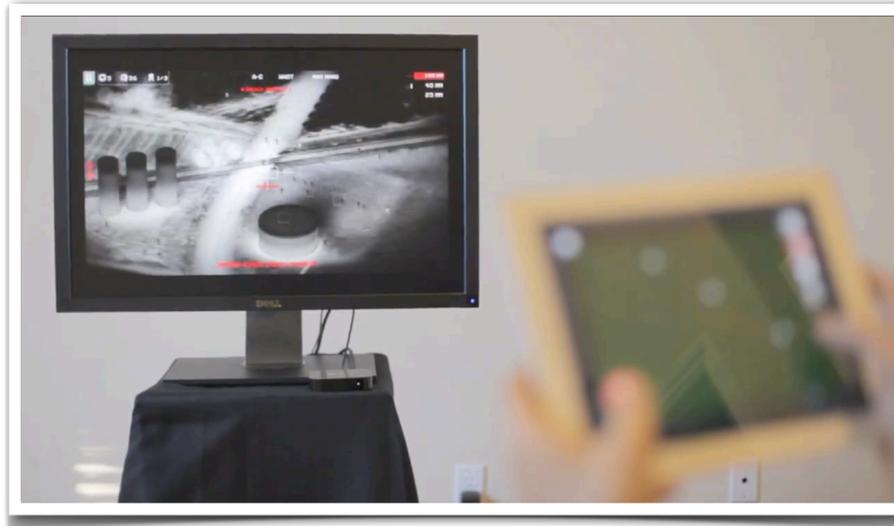
We had a choice to make: do we release a separate HD version and do what is known as “double dipping” to maximize short term profit, or do what the players would want the most, a Universal build. Do we also raise the price now that it’s universal?

Challenges with Universal means getting all the assets to fit in 20 MB, and inability to price differently for iPhone and iPad.

We went with the most generous approach. Free Universal update. It’s a decision to manage and take care of the customer base.

Fans loved it, it shot right to the top of the iPad charts. It’s just another way to differentiate yourself from the competition. At some point everyone will do it and if you don’t, you’ll stand out in a bad way.

# Leveraging New Platform Features



We started leveraging fancy new iOS features.

AirPlay Wireless Streaming was added despite the fact that we knew very few users would be able to use it, at least initially.

In line with our goal to maximize the awesomeness of the game, stand out above the noise! Turned out to work really well with the game actually.

# A New Threshold

1,000,000+  
Downloads



2009

5,000,000+  
Downloads



2010

20,000,000+  
Downloads



2011

Our first year we broke the million download mark which established us as a real game studio on the app store.

We expanded our first franchise with the free ad-supported version, and iPad enhanced versions in 2010 which grew the userbase manifold.

In 2011 we stepped on the gas and tripled the number of franchises, now at 20 times the install base of 2009.

We're beginning to face new business questions and opportunities at this new threshold. For example, are any of our games popular enough for merchandising to make sense? Licensing for other media? We're facing an exciting new era where a mobile game can become more popular than a prime time television show or movie.

# Now what?



Create a sequel?

Keep updating the original hit?

Both?

What about our other games?

What about localization?

Maybe it's time to expand to another platform? But does it make sense?

# Localization



China is a huge emerging market.

Not everyone speaks English, and many countries prefer localized versions

In general, it's a good idea to have localization but it's overhead to translate all of your marketing materials and in-game text. A lot of extra work for an indie.

You'll likely need to contract translation out as you probably won't have native speakers of all languages in your company.

As an indie, it's still perfectly acceptable to stick with English for your launch version (perhaps unless your game is a text heavy RPG).

There is no point in wasting extra resources localizing in the event that your game isn't as successful as you thought it would be, plus it's a relatively easy thing to add later if your game is a huge hit. Focus on the big revenue generators first, this is a relatively minor optimization.

Pro Tip: design your games so that you don't even need to be able to read to figure out how to play. This can be achieved for many types of games.

# Platforms



Is it safe to put all of your eggs in one platform basket?

Conventional business sense says no, but we still think it's fine if it's clearly the best basket by a large margin.

For now Android finally has some market share, but iOS still has the lion's share of the profits. Android still has a fragmented hardware base, increasing development complexity and customer support issues, and the hassle of multiple stores to deal with.

Right now there is a lot of complexity to manage and uncertainty in the market. Can be a tough place for indies, unless you are prepared to handle them.

It can make sense to port if your game will run on the top few handsets without too much custom work. Might even be easy if you used a game engine like Unity.

In general though, for a small development team with successful iOS apps, at the moment effort is best re-invested into iOS.

Desktop? If your controls are designed for smartphones, it may not map well. Other platforms, different beast altogether. Closed systems compared to smartphones. More contracts and other hassles.

# Horizon Research



Don't get complacent if you have some early success with one game. There is always something cooler coming out every day. Reduce your risk by starting development on another game. When thinking about a new title, view the landscape as if you were starting from scratch with no prior success. What kind of game would you make now? Where is there obvious room for innovation? What type of game can maximize your team's strengths? (2D, 3D? Lots of content, procedural, server side features, multiplayer?)

# Continue Support



19+ Updates



5+ Updates



4+ Updates

BUT...as long as players keep playing, you should keep updating your existing titles. Updates are expensive. Factor in not just new assets, but quality assurance, and opportunity cost. For a small team it's less overhead to keep evolving one unified product into a better one than to create a sequel and deal with maintaining yet another app.

Plus it's more advantageous to have one really popular app vs several less popular apps.

You get many more organic downloads the higher you are in the charts.

It's all about managing your install base, giving value to the players and a reason to keep playing.

# Conclusion



Apple Computer, 1977

Unless your goal is to be acquired by another company or grow into a multi-billion dollar company, there is nothing wrong with never getting out of the garage!

At the end of the day, you need to decide what is important for your company. Is it the success of your games and the revenue they generate?

Remember the benefits of smaller teams and their agility. Even if you "grow up" it's important to keep the garage mentality.

You have a choice between seeming impressive, and being impressive.

For us, the company that maintains creative control and achieves good revenue with the fewest people is the most impressive.

# Questions?



iman@limbic.com



@imost

**LIMBIC**