



# Going Online

**Moving from boxed title  
game development to F2P**

*Ralf C. Adam*

## DISCLAIMER

*This lecture was first held in August 2010 at the European GDC in Cologne/GERMANY. The presentation may not include all original materials shown during this speech (pictures, videos etc.) and is solely meant as a summary and handout. All copyrights of presented materials belong to their respective owners.*

## ABOUT MYSELF

- Ralf C. Adam
- Vice President Publishing / Gameforge
- In the games industry since 1993
- Track record includes work for Atari/Infogrames, Ubisoft, Disney Interactive, Take 2, Ascaron, Sunflowers, Spellbound and other Publisher & Studios

# ABOUT MYSELF



## ABOUT GAMEFORGE

- One of the leading European provider for Online Games (Developer + Publisher)
- Founded in 2003
- More than 400 employees
- Product portfolio of more than 20 Games
- MMOs, Browser Games, Facebook
- Fully localized in over 50 countries
- More than 200 million registered players
- Europe's biggest MMOG: Metin2



# ABOUT GAMEFORGE



# EXTERNAL DEVELOPMENT GAMEFORGE

- Ticking Bomb Games (Hamburg)



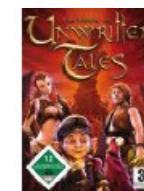
- Rough Sea Games (Mannheim)



- Steroid (Mainz)



- KingArt (Bremen)



- Keen Games (Frankfurt)



## Welcome to the Final Frontier

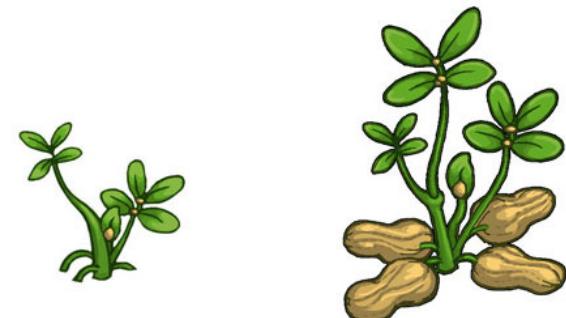
- Target Audience: Studios working in the traditional games industry, thinking about switching to development of online F2P titles
- Identifying typical misconceptions and pitfalls
- Outlining differences (and similarities) of both worlds
- Best Practices/Lessons learned
- Focusing on Browser Based and Social Network Games



Source: Star Trek Infinite Space / © Gameforge/CBS

## Online-Publisher Perspective on External Studios

- Getting the best game developers possible
- Using synergies between own online experience and know-how of traditional game developers, including game design, proven workflows and production methods
- Broadening of development capacities
- Scalability of resources
- Collaborating with studios outside your home territory



## Traditional Games Market – The sky is falling

- Exploding budgets & team sizes & development cycles...
- ...while the gap between the Top 5 titles and the rest is getting bigger every day
- Success depends on the first 4 – 6 weeks after release
- Pirating vs. Copy Protection (BattleNet, Cloud Copy Protection etc.)
- Sequels vs. Innovation
- F2P is there – and it's going to stay



## Looks like an easy call – but be aware...

- You might be dealing with smaller budgets
- You might be doing different kind of games (FarmVille vs. Crysis)...
- ...and therefore very likely won't be seen as the technological trailblazer of the gaming industry
- You might need other developer with different skill sets in your team
- It will have a massive impact on your production workflows as well as your game design approach



## Questions you should ask yourself as a developer

- Where do I want to be with my studio in 2 years from now? What's my long-term strategy?
- Maybe I'm a great game developer – but am I a good service provider? And do I want to be one?
- Do I have the business skills (and people) that it takes to be successful?
- Only make your choice with a full knowledge of the impact of your actions!

## Publisher REALLY has to be your Partner

- Vital Online-Publisher Services:
  - ✓ Producing (Development)
  - ✓ Live-Producing/Product Management
  - ✓ QA
  - ✓ PR
  - ✓ Localization
  - ✓ Payment
  - ✓ Business Intelligence
  - ✓ Customer Service
  - ✓ Community Management
  - ✓ System-Administration
  - ✓ Tech-Department
  - ✓ Marketing



## Best Practice

- Choose your partner wisely
- Make a Publisher Due Diligence
- Your key contact persons: Producer (during development) and later Product Manager (aka Product Director or Live-Producer)
- Keep a close relationship to your publisher
- Establish direct lines of communication between experts (including regular meetings and workshops)
- Always communicate personally and directly: Meeting wins over Phone/Skype wins over ICQ wins over Email

## Traditional Business Model:

- Developer gets advance against royalties
- Developer has to recoup the advance
- Percentage of gross profits or net receipts or a flat fee per product sold
- Only few games earn back their advances
- Therefore developer is “forced” to build profit into his advances



## Important questions

- Who will pay for development?
- Does the budget include general operating expenses?
- Is it an "advance against royalties" model or not?
- Who will pay for the Live-Team?
- Does the studio have to recoup the Live-Team?
- Which Share Model (Profit, Revenue etc.)?
- Does the studio have to recoup Marketing?

## Best Practice

- Learn & understand the vocabulary (ARPU, Churn, CLV, Conversion Rate, Retention etc.)
- Try to do as much research upfront as possible and build your studio plan and business case on solid numbers and expectations
- Keep in mind:
  - Development Budget is just one piece of the Puzzle
  - You won't break even after one month or two
  - The costs that really matter are Live-Team costs and Marketing spends

## »Unlearn what you have learned you must!«

- Asynchronous Gameplay (Browser & Facebook)
- Each player spends different amount of time
- Many players (>1,000) on one Server
- Unique Difference in Facebook compared to Browser Games: No Game Rounds!
- The player's experience in the first five minutes is essential
- F2P also means: no strong commitment of player in the beginning as he had no upfront investment



## How to earn money in F2P

- Players don't buy your game – they pay for goods and services while playing
- You want as many players as possible
- And you want players to spend money
- But players that never buy anything at all are also of great importance
- It's very dangerous to not annoy them (otherwise your worlds will be pretty empty)



## Items & Services Player will pay for

- Vanity Items
- Customizing Character
- Decorative Items
- Functional Items
- Consumable Items
- Temporary available Items
- Time saving features
- Convenient features
- Additional Content (e.g. Maps)



## Monetization vs. Game Design

- Monetization needs to be woven into the design right from the start – not added later
- But always ask yourself first:
  - Is that a great feature and is it fun?
  - Not: Will the player pay money for it?
- Biggest Challenge: Balancing for payment features
- You do not want to have a pay-to-win strategy
- But payment features must be meaningful to the player on the other side

## Best Practice

- Early prototyping of core gameplay is a must
- Full Pre-Production is vital
- Do a lot of focus group testing – especially for UI
- Two (or more) game designers working collaborative –  
One with focus on gameplay and fun and one  
concentrating on monetization
- Make the entry into your game as easy and as  
accessible as possible
- The player is never wrong
- In the end is still about having a great game!

## What does Scope mean?

- ***Product Scope:*** All features, functions and assets that are supposed to be in the game
- ***Project Scope:*** All work that must be done to deliver the game
- Scope needs to be clear and to the point
- Scope needs to be shared



## Small is beautiful

- Don't make it overcomplicated
- Keep the scope as small as possible in the beginning
- You do not need to deliver all content at day #1
- If you fail...
  - ...fail early
  - ...fail fast
  - ...iterate fast



## It's an ever-changing World!

- Online-Games are supposed to be “live” for years
- New DirectX-Version: roughly every 4 years
- Internet: Constant changes!
- Problems with Flash Update 10
- Facebook changes with heavy impact:
  - Limitations on User Messaging
  - Facebook-Credits
- Worst Case: From one day to the next  
your game is not working anymore!
- Your publisher might also do changes  
on his own API that could effect you



## Mayday! Mayday!

- 1<sup>st</sup> Rule: There's no »quick & dirty« in Online-Games!!!
- Never ever produce a single line of “hard-coded” stuff
- Classic Mistake: »We will not change this feature later, let's do a quick hack here.«
- Think about it: Once your game has taken off, there's no landing anymore
- Any update or code change is like repairing the engine or the wing of an airplane during flight



## One in a million

- Coming up with a great idea for a name is already tough enough for a boxed title these days
- But at least you don't have the URL problem
- If someone already owns the domain, you can just add „-game“: [www.torchlightgame.com](http://www.torchlightgame.com)
- Longer title might work as well:
  - *Call of Juarez – Bound in Blood*
  - *Operation Flashpoint: Dragon Rising*
  - *Star Wars Clone Wars Republic Heroes*



## Lessons learned

- Define the name already during pre-production
- Find a name that works everywhere in the World
- If you're not native and have an English name: Check with UK/US natives!
- Make sure, that all relevant domains (.com, .de, .es etc.) are available and buy them asap
- Name needs to be short (ideally one word)
- Spelling must be evident – even when communicated only verbally
- Website is equivalent to shelf positioning

## Your Problems have just been multiplied...

- No “text hacks” or hard coded stuff possible
- Higher requirements for Game Designer/Texter
- All texts in localization tool from day #1
- If you screw up only one text key when “live” – Dante’s Inferno awaits
- Each text change automatically means a change in all languages
- Availability of update in all languages at the same time is required
- Means also localization test of all languages for each & every update



## »What's my line?«

- A close collaboration with Localization Department throughout the whole production and beyond is vital
- Make sure your game is properly connected into the publisher localization tool already during Pre-Production
- Never do any quick & dirty fixes with hard coded text (aka »I will attach the loca key later«)
- Always keep your Localization Manager in the loop for every text change, no matter how minor it seems

### Lessons learned the very hard way

- Some Bugs do not occur, when only 200 people are playing
- Different story with 200,000 people though
- Therefore: Never skip a Beta no matter how good the Alpha looks!
- Fatal: Bugs in features the player has paid for
- Fixing problems when “live” takes 5x times longer
- Always use test servers in the first place
- Automated testing is a must
- Remember: If it’s not tested, it doesn’t work!

»Ah, we do that at the very end. Won't take long!«

- Player Support
  - Emails notifications, Guild-Support functions etc...
- Management of user generated content:
  - Editing of content, delete functions etc...
- Observing game rules:
  - Suspending players, Chatlog access etc...
- Payment Support:
  - Player Inventory, Statistics etc...
- User Management
  - Different kinds of access level etc...



## »No Battle Plan survives contact with the Enemy!«

- During Production already prepare a post release plan for the first 3 – 6 months...
- ...and be ready to just throw it away
- Even a good Alpha, a long closed Beta and a well organized open Beta can not nearly prepare you for what is about to hit you
- Players will report bugs you'd never have imagined
- Player will find exploits you'd never have thought of
- Player will give you feedback and demand for features you'd never have dreamt of



Helmuth Karl Bernhard  
Graf von Moltke  
(26.10.1800 – 24.04.1891)

## »Drums... Drums in the deep! We cannot get out!«

- For Browser Games: first 6 months after release  
usually full development team size for Live-Team
- Especially on Facebook even more common: Two  
(or more) Live-Teams in parallel
- FB: Release Cycles of 1 – 2 weeks
- Each version tested & in all languages
- Heavy workload for Project Manager
- Strong collaboration with Marketing  
is vital
- Once you're “live” – it's all about metrics!

## You never ship!

- In the traditional games business you embrace a new game idea as soon as the current title has been shipped
- Now you're not making "products" anymore
- The more successful the game gets the longer the team will work on it – maybe for many, many years!
- This means an increasing risk, that your team members will get unsatisfied and demotivated

## Protect your team from burning out

- 1<sup>st</sup> rule: Only make the kind of game you always wanted to do – as you might end up doing it for a quite long time!
- Rotate team members in the different teams (Live-Teams, AddOn/Update Teams, Dev Teams)
- Live-Teams are an excellent way to train junior people
- Though: Problem for small studios with only one game live at a time

