fatshark

Fatshark and Bitsquid

Game engines, game development and design

Rikard Blomberg

-

CTO and Founder of Fatshark AB

-

Board Member and Founder of Bitsquid AB

-

rikard@fatshark.se

Let’s Start with a survey so I get to know you…

Q: What are you studying – what program etc…

Q: Do you plan or would you like to work with computer games?

A good idea if you like plenty of hardware on your desk!

How often do you play computer games?

a) Several times or hours a day
b) Once a day
c) A couple of times a week
d) Less than that, but sometimes
e) Games are not for me – games are for kids, boys, nerds, etc…

fatshark

What is your preferred platform?

a) PC or (Mac) or (Linux)
b) Web based
c) Current Gen Consoles (Xbox360, Ps3)
d) Anything from Nintendo
e) Tablet
f) Phone
g) Other… (*I bet no one of you even owns a PsVita – am I wrong?*)

Do you know how to program?

a) I am awesome! (*John Carmack, Linus Torvalds*)

b) I could make a living on it.

c) I know the basics.

d) No, but I know how to use MS Office…

Have you ever made a game?

a) Yes, and it has been released and people actually pay me for it.
b) Yes, but it is not commercial, I am not a capitalist pig!
c) I just need to fix those last bugs…
d) I started on a project but then….
e) No – isn't that what we are supposed to do in this course?

END OF SURVEY

Thanks

So how did I end up here?

Spent a lot of time playing computer games and pen-and-paper RPGs

Have always loved games of all sorts.

Also loves “world-building” and simulations

Did I do a lot of Programming?

I started programming when my father brought this home in 1986.

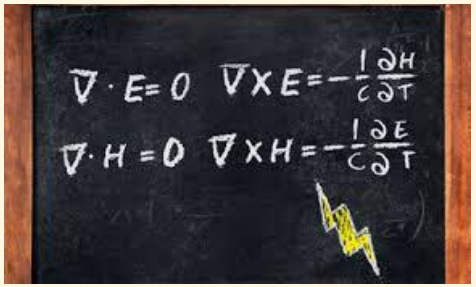
What is the name of the computer?

Considered myself a good programmer (even fooled my friends that I was) until I actually started coding large complex systems i.e. games…

Actually I became a rather good programmer around 2006 or so

Now I am back being a n00b… cause programming is a **craftsmanship**.

So, how did I end up here?



- Wanted to become a physicist
- started at F on KTH.

- Started at HHS by a fluke.

- Met *Martin Wahlund* who was to become my business partner. He is now CEO of Fatshark.

Do you remember this company logo? →

How did we start making games?

First game we made was for KTH Game Awards 2003.

Project page still alive at:
http://excitera.nu/kthga0203/boundless.php

After the competition we started a company to commercialize the game, but we never got there…

Things got in the way

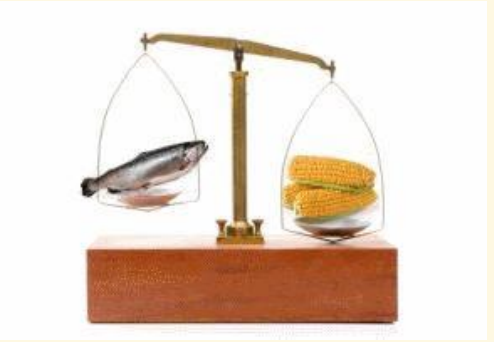
To make actually make games that we wanted to we found out that we needed:

- Financing
- A balanced team
- Experience

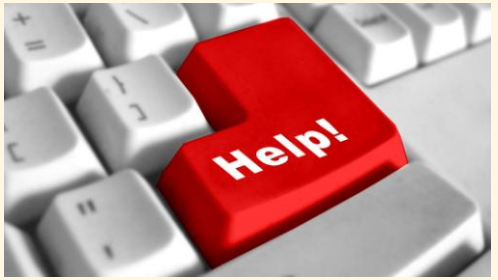
So what happened was:

Consulting, sub-contracting

More common in the game industry than what you might believe.



Game projects usually have unbalanced resource requirements over time.



More programmers are always needed as you get close to launch.

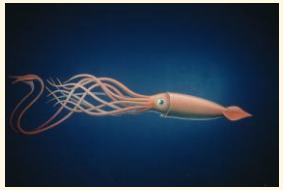
Grin and Fatshark

Fatshark was started in collaboration with Grin to work as a subcontractor on parts of games or on whole games.

But eventually bad things happened:

But Fatshark survived…

(and a new creature emerged)



Bitsquid founded 2009

Goal: To create a new high-end game engine, built from the ground to focus on performance, flexibility and productivity.

Tobias Persson

Niklas Frykholm

Old logo (showing it will get me killed)

fatshark

A Skunkworks Project

"It is an especially enriched environment that is intended to help a small group of individuals design a new idea by escaping routine organizational procedures."

Where are we now?

40 man strong studio doing:

- Self published projects
- Publisher projects (work for hire)
- Supporting, continuing development of released games


~40 Experienced Developers (and one inflatable shark) based in central Stockholm
THE BLACK
El Camino
KEYS
HELL

And we have some really interesting stuff in the pipe…

(but I can’t tell you about that)

What is a game engine?

A game engine is a system or framework for creating computer games.

Consist of two parts:

- Tools
- A runtime component

Typically encapsulates platform differences (Hardware Abstraction)

Why use a game engine? Why not?

- Decrease risk
- Don’t re-invent the wheel
- Increase platform reach
- Decrease cost
- Decrease development time

- I get back to the “not” part…

What game engines are out there?

(it's a jungle)

- Commercial
- Publisher owned
- Strictly In-house

Unreal (my totally biased description)

- *Unreal Engine* grew out of *Unreal*, a first person shooter.
- Focusing on larger development teams, console titles and AAA games.
- Has made inroads into the mobile space, targeting Android and iOS.
- Long list of licensees, and many best-selling AAA games have been made in the engine.
- Unreal also has a “free” version targeting smaller developers: *UDK*

Unity(my totally biased description)

- The *Unity Engine* is used by over 500 000 game developers.

- Unity supports console development, it is mostly used for simple 2D games for mobile phones, PCs and the web.

- Unity has a free version with limited functionality and a Pro version needed for serious development which costs $1500 per seat.

- Has a hugely successful asset store where developers can trade systems and assets.

Some other engines

Roll your own? Why?

1. You are a technology freak

2. You have some very special needs

In either case: *Expect Pain*

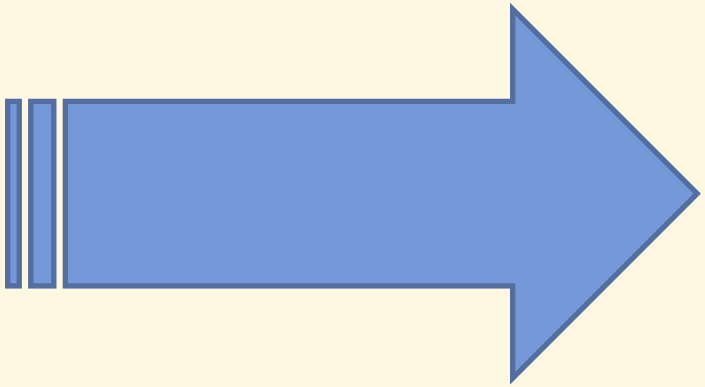
fatshark

bitsquid

- Is both the name of the company (a subsidiary to Fatshark)
- … and the technology (game engine) being licensed to a steadily increasing number of game developers.
- Company currently employs a CEO and 7 developers

The Bitsquid Philosophy

- Simplicity
- Job parallelism
- Openness
- Data driven
- Flat simple data structures

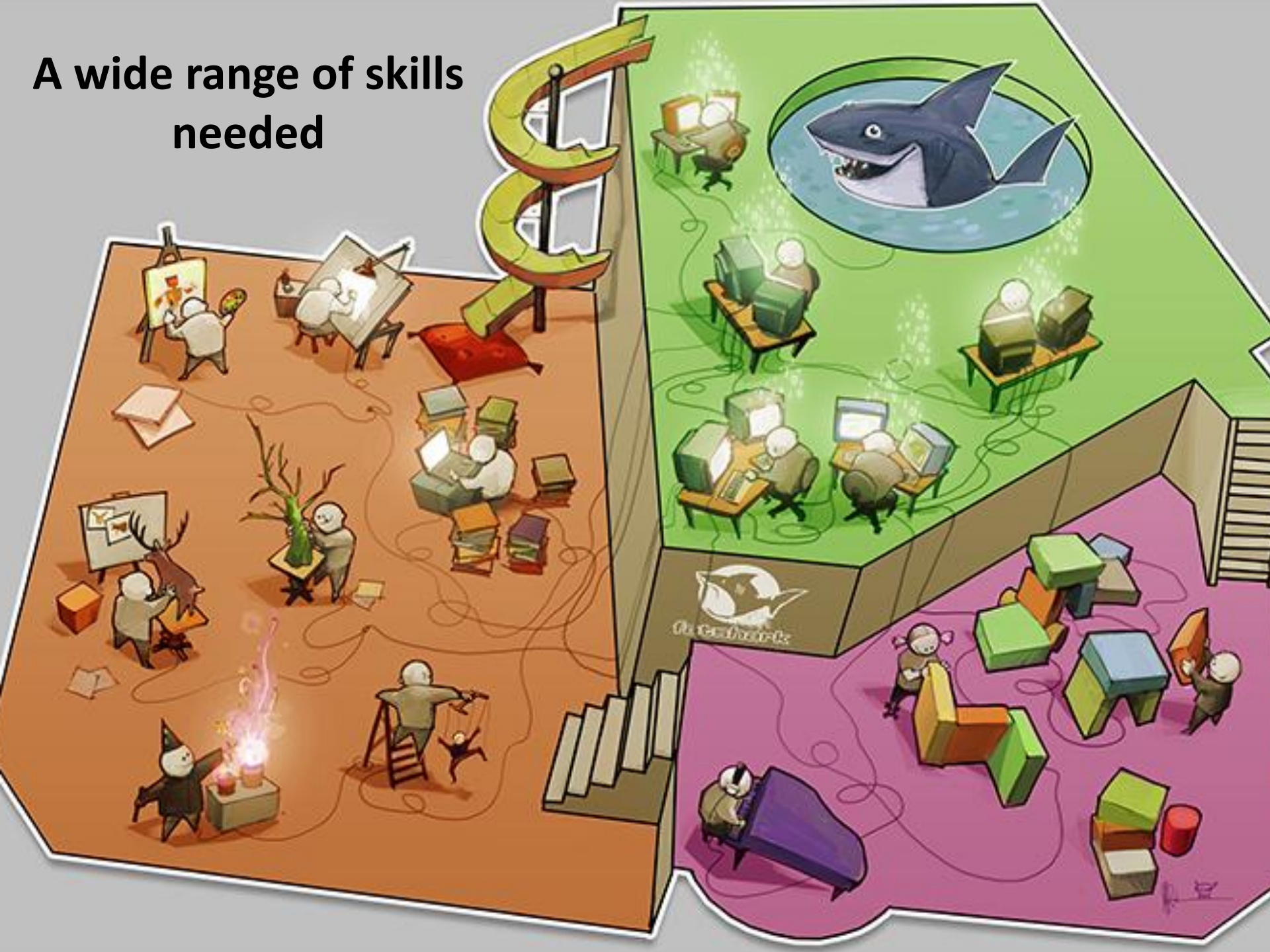


- Fast
- Dynamic
- Flexible
- Lightweight

fatshark

How do we make games?

A wide range of skills needed



The Fatshark way of making games

Usually fast – for good or bad

Design is done collectively – for good or bad

Ideas are either customer driven *or* taken from a backlog of ideas created at certain *creative days*.

We try to reuse things to decrease risk. It can be reusing a setting and the art or reusing a gameplay framework.

Game Design

I am not a “Game Designer”.

My current work mostly concerns running and managing a company.

But I have some *opinions* (and they should be regarded as such).

Ideas

If I got a penny for every unique and fantastic game idea being pitched…

contact@fatshark.se <xyz@aol.com>
Oct 3

to contact
Your Email - xyz@aol.com
Regarding - general
Your message - Do you guys take game idea
general public? I'm asking because I have so
I would like to send you guys.

contact@fatshark.se <xyz@gmail.com>
Apr 8

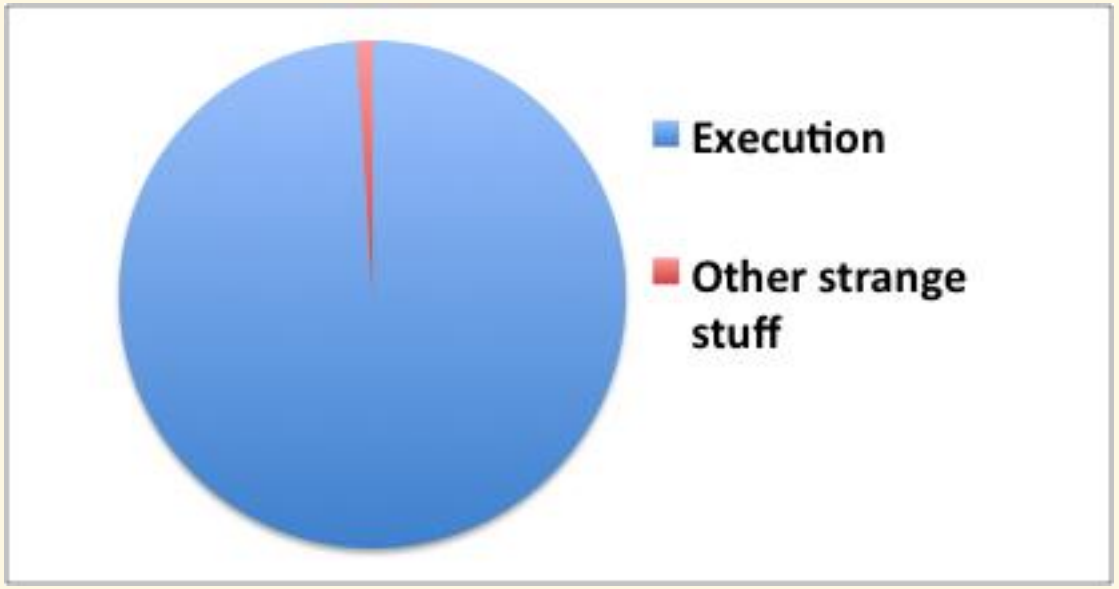
to contact
Your Email - xyz@gmail.com
Regarding - jobs
Your message - Hi, My name is Robin xyz and I want to contact you becouse I have an really special game idea. pls reply if you are intrested in hearing more about it. .

So ideas doesn't matter???

To make a successful game

You need a good idea and and way to get it to market.

From there - It is 99% about execution!



However – to make an insanely successful game (…think Minecraft)

You need a brilliant idea.. and some crazy streak of luck

From there - It is 99% about execution!

No one will be able to tell if an you have a brilliant idea and luck – so I advice focusing on what you can influence.

Where to start

Start somewhere at least…

It could be:

- Defining an audience.
- Having that brilliant (oh well … at least good) idea!
- Having a technology that shines in a certain area.
- Imitating that favorite game with just a small twist.
- Getting a high-level-vision from that crazy publisher

Set a vision! Agree on the vision.

Start coding… write that *main.cpp* (or whatever it is called) now!

Please - make choices – narrow down

Decide on features: – **to skip!**

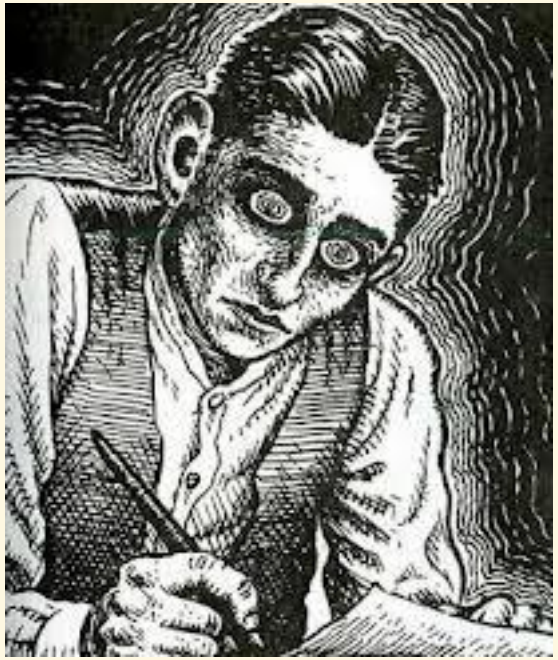
Decide on platforms: – **to skip!**

Decide on content: – **to skip!**

Decide on preferred distribution and monetization

/* Documentation */

Document decisions – if you are more than one individual!



For other purposes – skip documentation – you are making a game not an article or a book!

If you like writing – make a blog about your development instead!

When it comes to code: *The main source of the information about what the code does should be the code itself.*

I have never used any single line of documentation written for our productions. But I have one big regret…

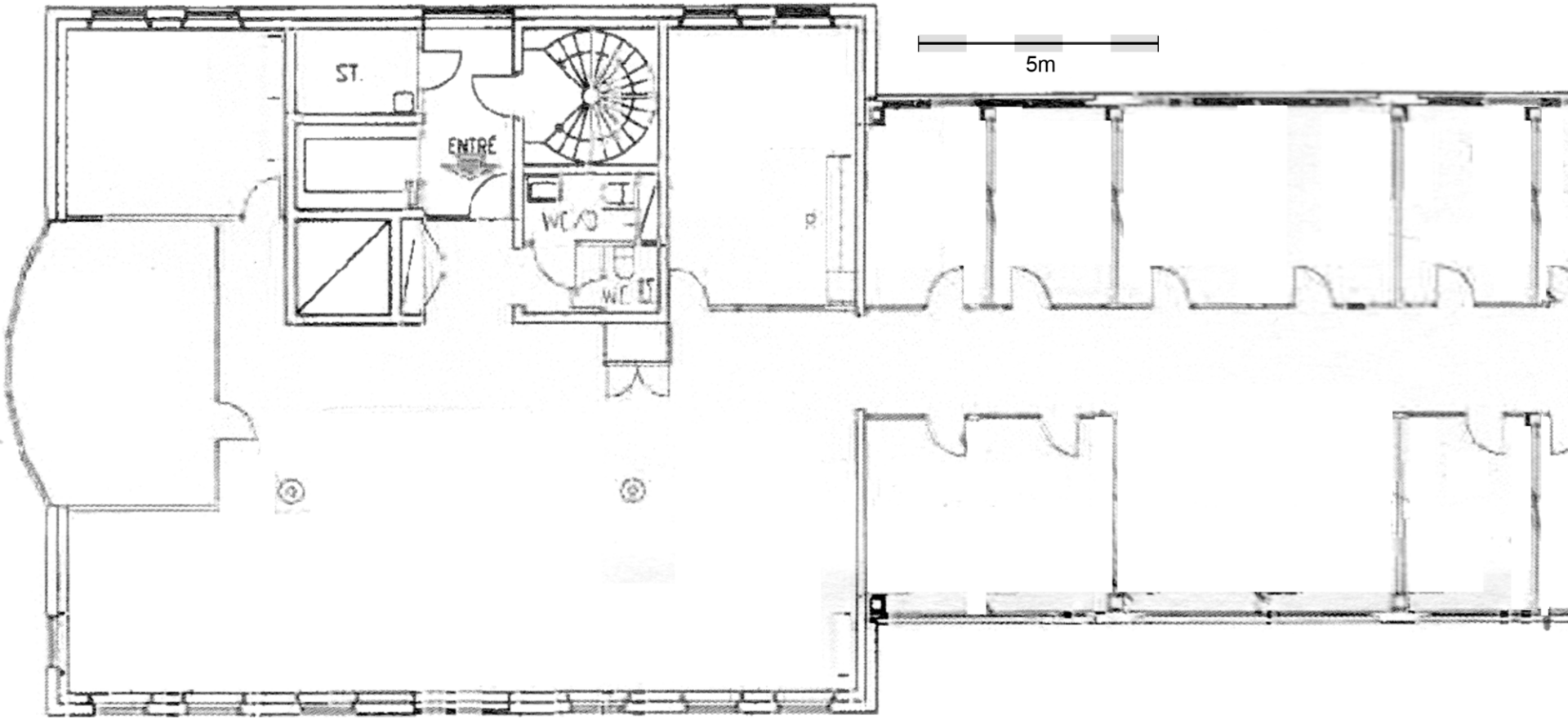
…I regret that I didn't take more pictures along the way, including especially screenshots of all funny bloopers and bugs created.


“OFFICE” 1


OFFICE 2
KYUSS

OFFICE 3

OFFICE 4



OFFICE 5


Content r

MULTIPLAYER

PROFILE EDITOR
COAT OF ARMS
SETTINGS
CREDITS
EXIT GAME

Make it playable – NOW!

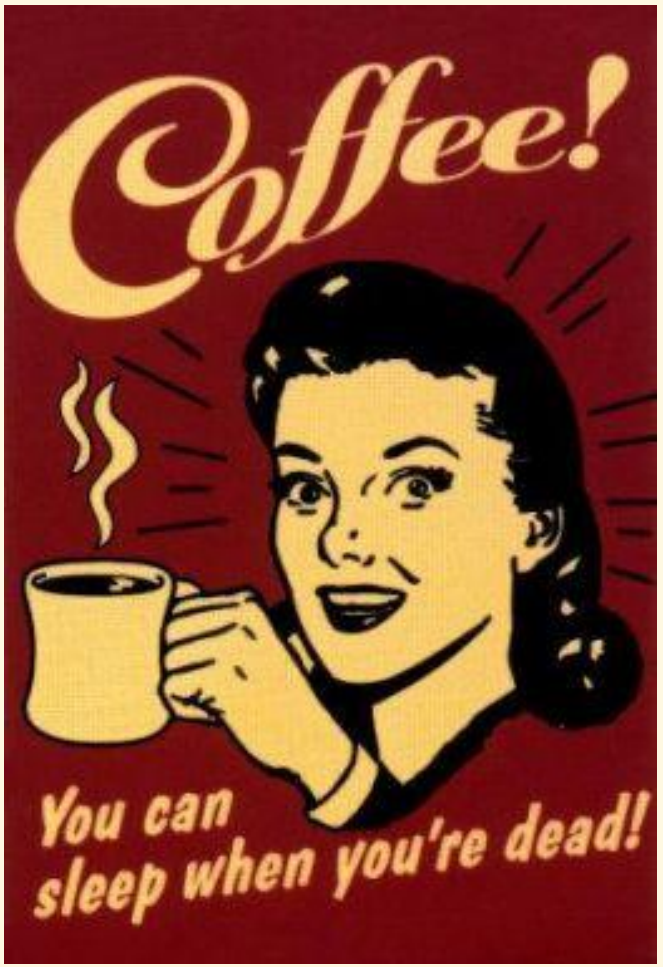
If you are not making and engine – and you are not!

Make it playable from day 1 (or at least week 1)

Get a head start – work like crazy!

Don’t expect help from anyone.
Roll up your sleeves and get dirty.

Don't Stray Too Far From Your Vision!



Iterate

```
while (game->quality() != AWESOME)
{
	void (*task)(CGame*);
	task = find_possible_improvements(game):
	(*task)(game);
}
Rejoice();
exit(0);
```

Iterate, iterate, play, play, iterate, play… however I have a nagging suspicion that this must is the case:

```
CGame::quality() { return NOT_REALLY_THERE_YET; }
```

fatshark

Maybe closer to the truth…

```
while (game->quality() != AWESOME)
{
	void (*task)(CGame*);
	task = find_possible_improvements(game):
	(*task)(game);
	if(out_of_time() || out_of_money())
	{
		int result = Pray();
		exit(result)
	}
}
Rejoice();
exit(0);
```

fatshark

Find a balance between general/reusable and quick/dirty.

If you make everything according to the schoolbook or everything prepared for the next iteration of the game:

- you will never ship…
- actually - you will never be close to shipping.

Find a balance between general/reusable and quick/dirty.

If you make everything quick and dirty – you might ship…

…but the release will probably kill you!

This goes for design as well as code.

And remember: Don't Stray Too Far From Your Vision!

Use external testers/judges

Since you need to work like hell you will lose perspective.

Bring in external people for feedback and judgment.

But at the same time – Don't Stray Too Far From Your Vision!

A bit more formal approach on design…

Take a look at *Joakim Setterbergs* blog.

Joakim is a lead level designer at Fatshark who has some interesting ideas about design.

http://ldcompanion.wordpress.com/

Limitations

As a programmer you will often be the one highlighting limitations:

- *You can only have X AI agents active at any time.*
- *The level can only be this big cause of memory constraints.*

Limitations

The most common technology related limitations of today:

- *Rendering quality, details, realistic lightning, shadows, refraction, reflection, transparency.*
- *Simulation of complex systems – physics, liquids, fire, hair, destruction of objects and environment.*
- *Number of agents simulated, complexity of agents.*
- *Network traffic, network latency, server processing power/c.*
- *Input quality and latency.*
- *Number of objects in the simulated world.*

fatshark

Use a “Gameplay Grid” to increase without adding new systems

	Camera	*Place*	*Time*	*Controls*	*GUI*
Camera					
Place					
Time					
Controls					
GUI					

Place challenge examples may include platforms that are traversable only during certain time intervals (*time* and *place*), a location in darkness that limits the view (*camera* and *place*), changes in floor traction or a difficult jump (*movement* and*place*), or a place that imposes restrictions on save and load functions (*gui* and *place*).

Platform diversity is a challenge

Mobile (pads and phones)

Virtual Reality Headsets (Oculus Rift)

Next-gen consoles (PS4 and XboxOne)

Set-top boxes, micro consoles, steambox…

Web based gaming

PC still strong (Mac not so…)

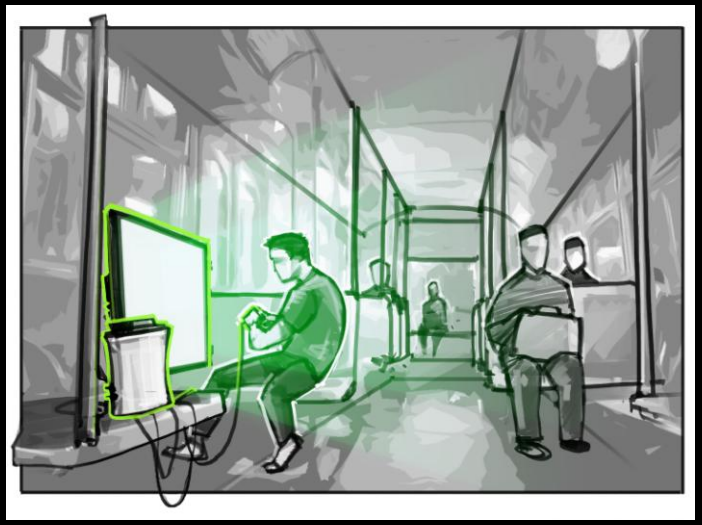
More to come…

Platform diversity

The number of platforms is a technology challenge.

But even more – it is a design challenge!

What is the potential of cross-platform gaming?



High-end gaming on the go?


Sharing the experience of
a mobile game?

Cross platform gaming

Clients needs to be optimized and adapted for each platform →

- Having a cross platform engine is essential

Data that clients works on needs to be centralized →

- An infrastructure for handling data is needed

Looking in the crystal ball…

If we believe in the assumption that cross-platform games are the future – data hosting becomes even more important. What can we expect from these (not so small) companies?

Akamai

rackspace HOSTING

facebook

Google

amazon

YAHOO!

End of this presentation

I really hope to get some questions…