



# **SAFE AND SOUND**

**Risk free Audio Development**

**GDC Audio Boot Camp 2009**

Tomas Neumann – Senior Audio Programmer

Frankfurt, March 2009

# Introduction



## Games

FarCry (2004)

Crysis (2007)

Crysis Warhead (2008)

Crysis Wars (2008)

## Studios

Frankfurt, Germany

Budapest, Kiev, Sofia, Seoul

Crytek UK

## Tomas Neumann

Senior Audio Programmer in  
R&D since 2005

[Tomas\(at\)crytek.com](mailto:Tomas(at)crytek.com)



# Audio In Game Development : Everywhere



- **Pre-Production**
- **Implementation**
- **Music / Dialogue**
- **Milestones**
- **Reviews**
- **Technical settings**
- **Alpha / Beta / Gold**
- **Demo**
- **Patches**

## People outside of Audio

- Might have wrong impression
- Fear what they don't know

# Risk free Audio Development ?



- **Stability**
  - Fewer crashes
  - Fewer show-stoppers
- **Efficiency**
  - Fast turn around
  - Fast bug fixes
  - Safe bug fixes
- **Earns trust**
  - Take their fear
  - Prove Audio is low risk
- **Quality**
  - Better Audio in final product

# Risk free Audio Development !



- **How to achieve?**

- Not done in a day
- Not a single task
- Baby steps

- **“Never-ending story”**

- Ongoing development
- Ongoing effort
- Ongoing improvements
- You will never succeed!

## **This talk:**

- **I will not tell you how to improve 1000 details!**
- **HOPE**



# General Strategies To Follow

1. Limit access
2. Inform others
3. Divide and Conquer
4. Do not put things where they don't belong
5. Prepare for content-locks
6. Find your own bugs
7. A bug is your friend
8. Keep track of changes
9. Exploit automation
10. Data-mine for gold

# I) “Limit Access”

- **Use abstract, data-driven sounds**
- **Use high level parameters**
- **Minimize external interface and behaviour**
  - (Code & Audio)

## **Cost:**

- More work
- More responsibility
- More bugs

## **Benefit:**

- More creativity (internal)
- Faster implementation
- Fewer coders help needed

## 2) “Inform Others”

**Write what others dont know.**

- **Internal communication**
  - Forum / Mail
  - Wiki
- **Show work early and often**
  - No embarressment
  - See Jason’s „Fail early“
- **Code:**
  - Optional debug messages

### **Cost:**

- More communication
- Extra code

### **Benefit:**

- Better transparency
- Help others
- People know:
  - you actually work(!)
  - what is WIP and what is new



## A close-up shot of a character's helmet, featuring a glowing red visor and a red 'D' logo on the side. The helmet is metallic and has various mechanical details. The character is wearing a white and grey suit.



### 3) “Divide And Conquer”

**Avoid mixing responsibilities or overlapping tasks.**

- **Sound Designer**
  - Source (raw) asset creation
  - Asset specification
- **(Automatic) build process**
  - Build (cooked) asset
- **Other areas:**
  - Audio implementation / Code
  - Localization / post-processing

#### **Cost:**

- Extra working steps

#### **Benefit:**

- Frees SD's time
- Easier to resolve issues
- Modular and structured work habits

## 4) “Do Not Put Things Where They Dont Belong”

- **Code**

- Avoid hard-coding audio specification and references
- Serialize from external database (xml, sql)

- **Audio**

- Clean up audio asset

- **Cost:**

- Coding effort
- Well organized SDs

- **Benefit:**

- Independence from code
- Quicker iteration (immune to code-lock!)
- More control (stripping)



## 5) “Prepare for code/content-locks“

**They are coming, whether you want it or not on milestones, demos, or randomly.**

**They will block your progress!**

- **Be prepared for downtime**
  - Reserve work to do in between.
  - Inform if no audio bug was blocking

### **Cost:**

- Administrative overhead
- More communication

### **Benefit:**

- Balanced work load
- Increase trust if audio was non-blocking
- Steady progress despite milestones

## 6) “Find Your Own Bugs“

### Dont expect:

- **QA** to know what is broken
- **Producers** to know how important or risky a fix is
- **Train QA**
- **Add bugs by yourself**
  - Add to bug tracker, even if you are about to fix
- **Increase bug priority**
  - Avoid low audio bug priority (known shippable in Beta)



### **Cost:**

- Extensive play tests
- Scanning through game logs



### **Benefit:**

- You know the game better
- More fixed bugs = better shipped audio
- Increase bug fix rate
- Decrease fix time
- Helpful statistics
- Happy producer = more trust



## 7) “A Bug Is Your Friend”

- **Document problems**

- Write about it : What? Why?
- Add to bug tracker, even if you are about to fix it

- **Code**

- Use `assert()`
- Log: Output warning
- Catch function call results
- Create test cases
  - In-house & middleware



### **Cost:**

- Admit your mistakes
- Overcome one's weaker self



### **Benefit:**

- Increased awareness (team, QA)
- Faster fix-rate
  - Done it before
  - Searchable mail
  - Producer's trust

## 8) “Keep Track Of Changes“

**Why was a decision made 2 years ago?**

**What was agreed on in that meeting?**

**Dialogue integration specialist:  
„btw, today is my last day!“**

- **Preserve knowledge**
- **Offer it to new team members**
- **Write what you know**

### **Cost:**

- Documentation effort
- Write down the obvious

### **Benefit:**

- Surprise-proof
- Database of experience
- Evaluate and rate old decisions

## 9) “Exploit Automation“

- **Automatic Build on several platforms**
  - Code (compilation errors)
  - Audio (integrity errors)
- **Automatic test run**
  - Stability tests
  - Mail filtered logs of warnings
- **Statistics**
  - Performances graph
    - Show impact of changes
  - Finalize compression settings



### **Cost:**

- Extra hardware
- Build / Audio engineer



### **Benefit:**

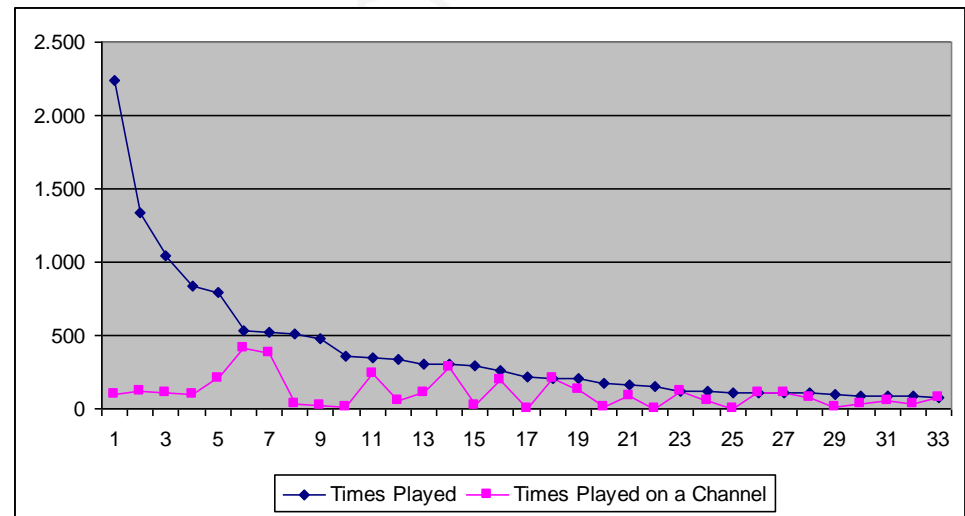
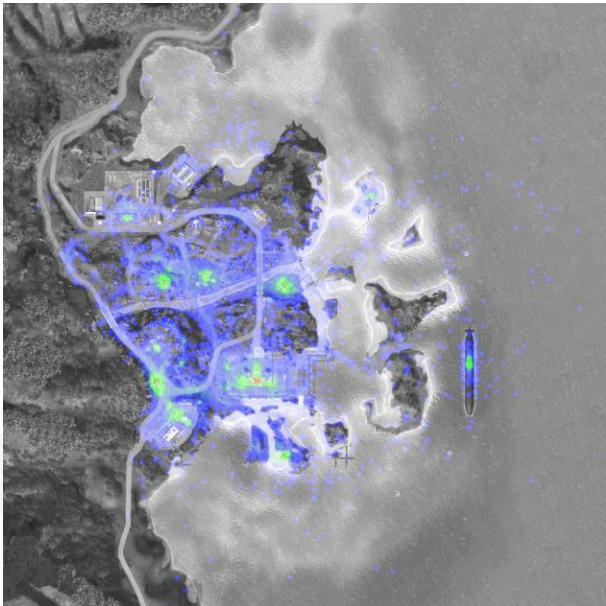
- Frees worker's PC
- Honest verification
- Stability
- Performance
- Short down time

## I 0) „Data-mine for Gold“



**Statistics help you to find the bang for the buck**

- **Graphs**
- **Lists**
- **Pictures**



# Conclusion



## Audio Development

- safer
- more stable
- more efficient
- more trustworthy
- increase quality

**You have to work on it –  
every single day!**



**KKTHXBYE**



**Thank you  
Please rate!  
Ask questions!**

