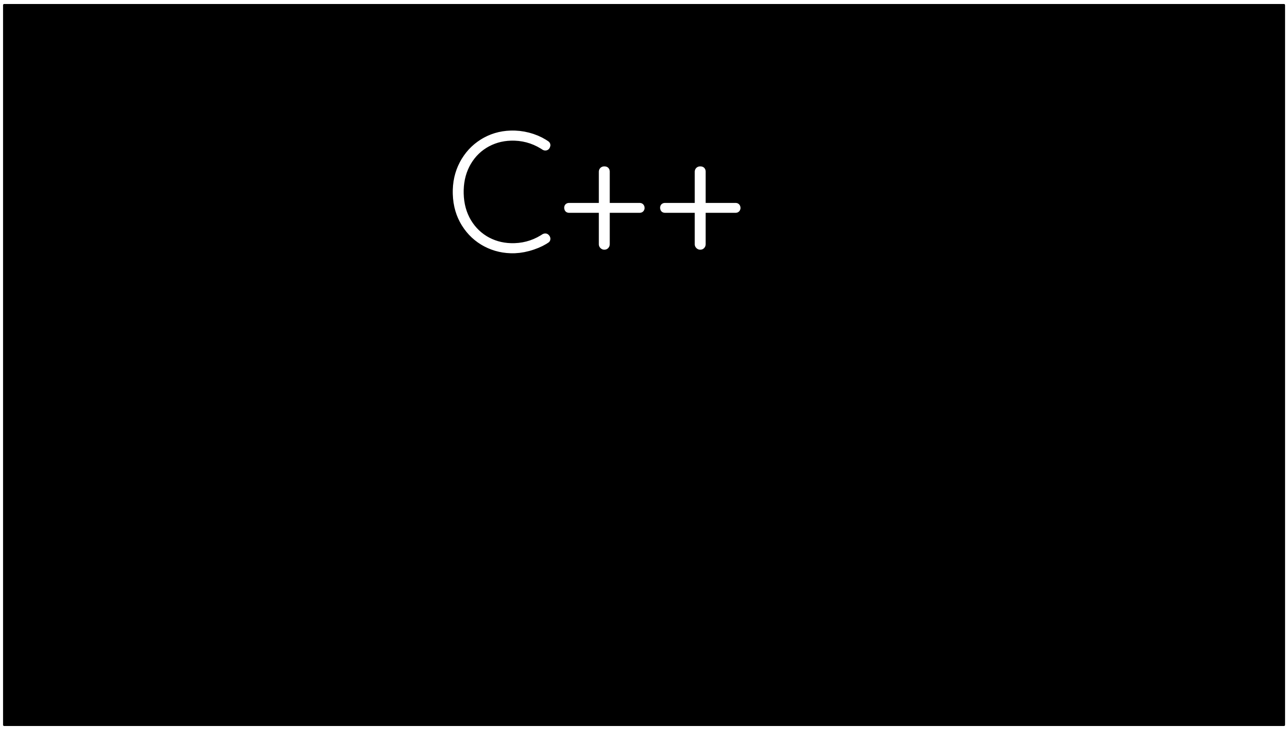
Being a Programmer in the Games Industry

How to arrive, survive & thrive!

Keith O’Conor

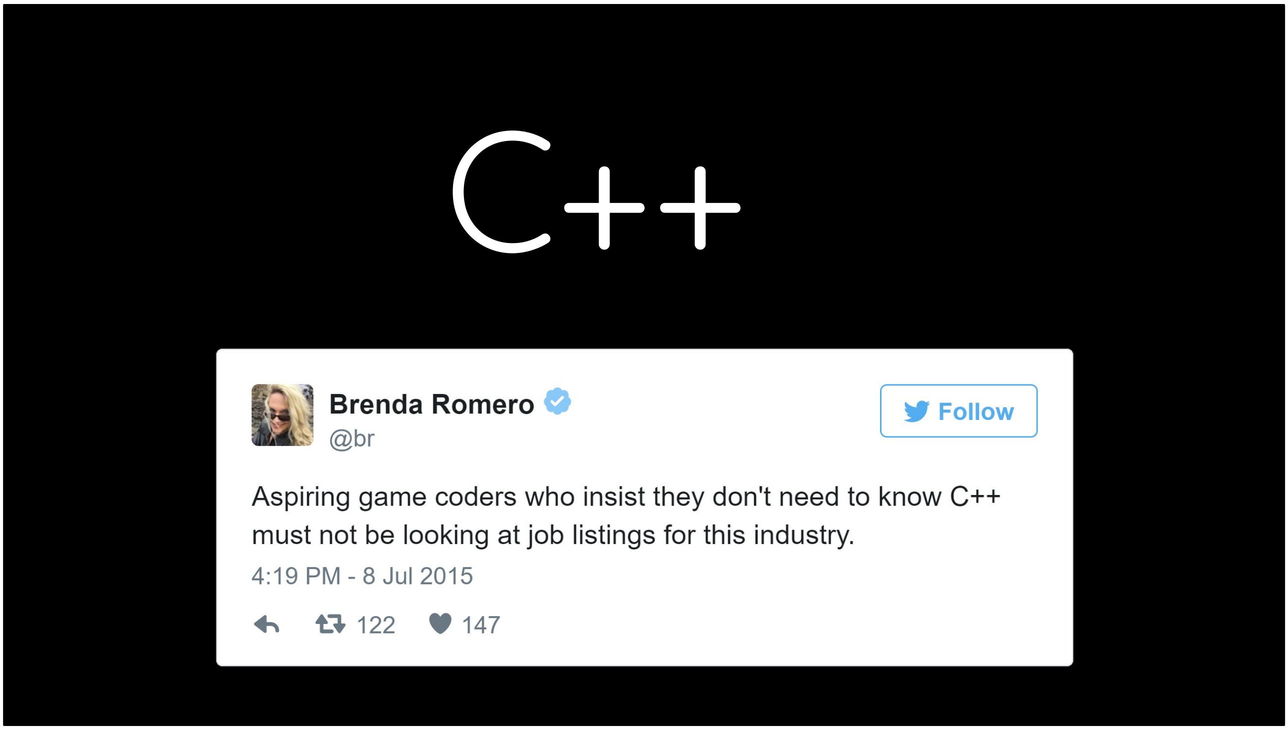
- **Who** I am
- Everything **except** programming
- **Caveat**: mainstream AAA industry
- **Interrupt** me! Or questions at end...

First thing needed: programming language...



C++

- **Control**: performance, memory, code generation
- Got legs: huge **legacy** battle-tested codebases
- Some legitimate **problems**: memory corruption, crashes



C++

Brenda Romero
@br

Aspiring game coders who insist they don't need to know C++ must not be looking at job listings for this industry.

4:19 PM - 8 Jul 2015

- Really the **only option** for making games

- But **Unity**/C#!
- Only smaller studios, not AAA (**prototyping**)
- Unity & Unreal - **engines written in C++**

- **Tools & scripting**: Lua/Python/C#

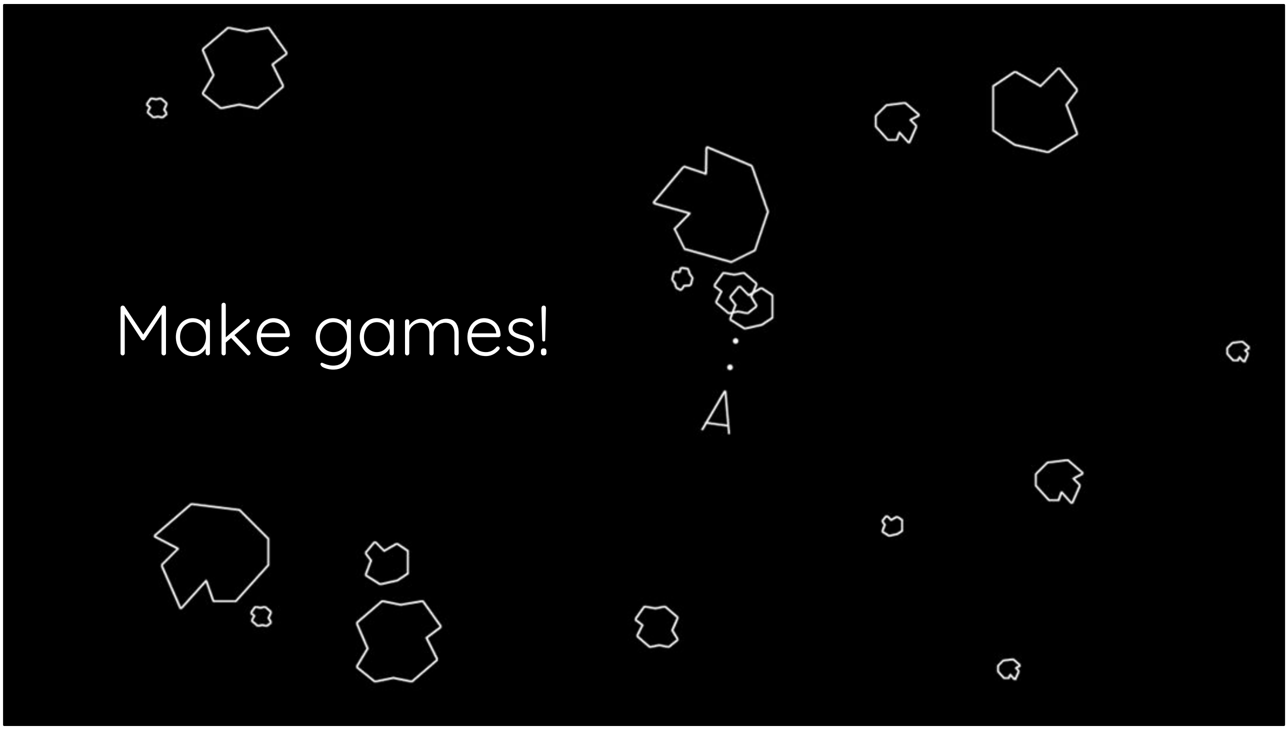
- Good coders are **multilingual**

Other obvious skill needed...

Make games!

- **Spare time** = programming
- ** HANDS UP! **
  - Aspiring game programmer
  - Full game - menu, audio
  - In C++
  - Without using STL or Boost
- Realistic **scope**
- **Finish** the game (UI, menu, high scores) - portfolio
- Don’t use **libraries**

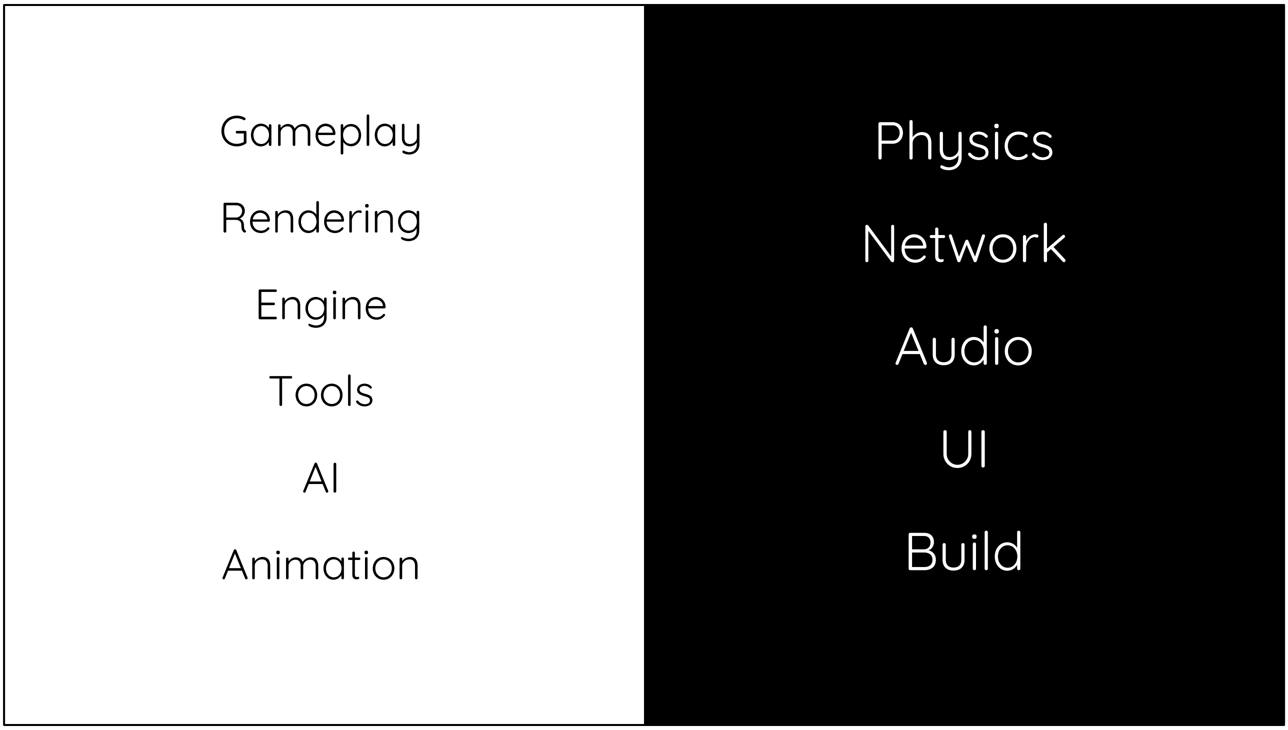
Never too late - may seem trivial, but...



Make games!

- **Asteroids**! Physics, vector math, collision detection, rendering, AI
- **Improve** - refactor, design

After a few games, you’ll know what you like...



Gameplay

Rendering

Engine

Tools

AI

Animation

Physics

Network

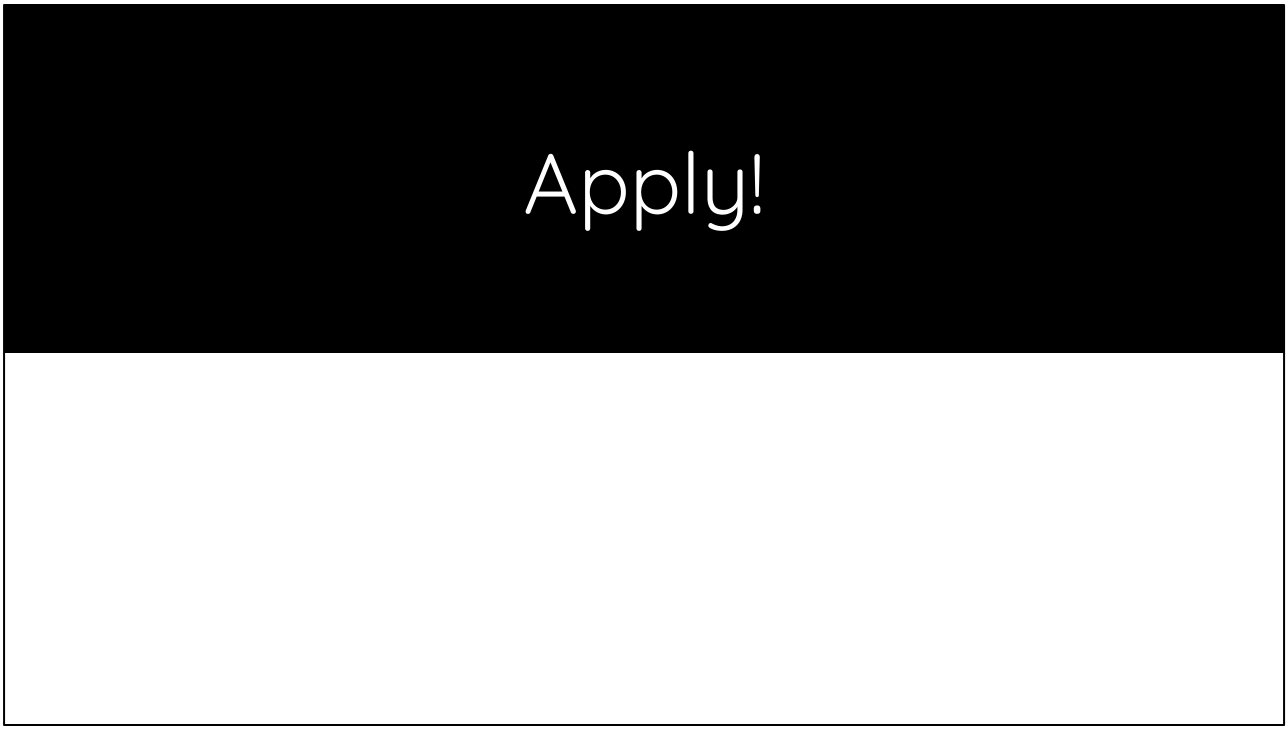
Audio

UI

Build

- Generalists good, bigger studio = specialization
- ***Describe*** *Roles*
- Radical - **camera** programmer

When ready, the only way to get a job is...

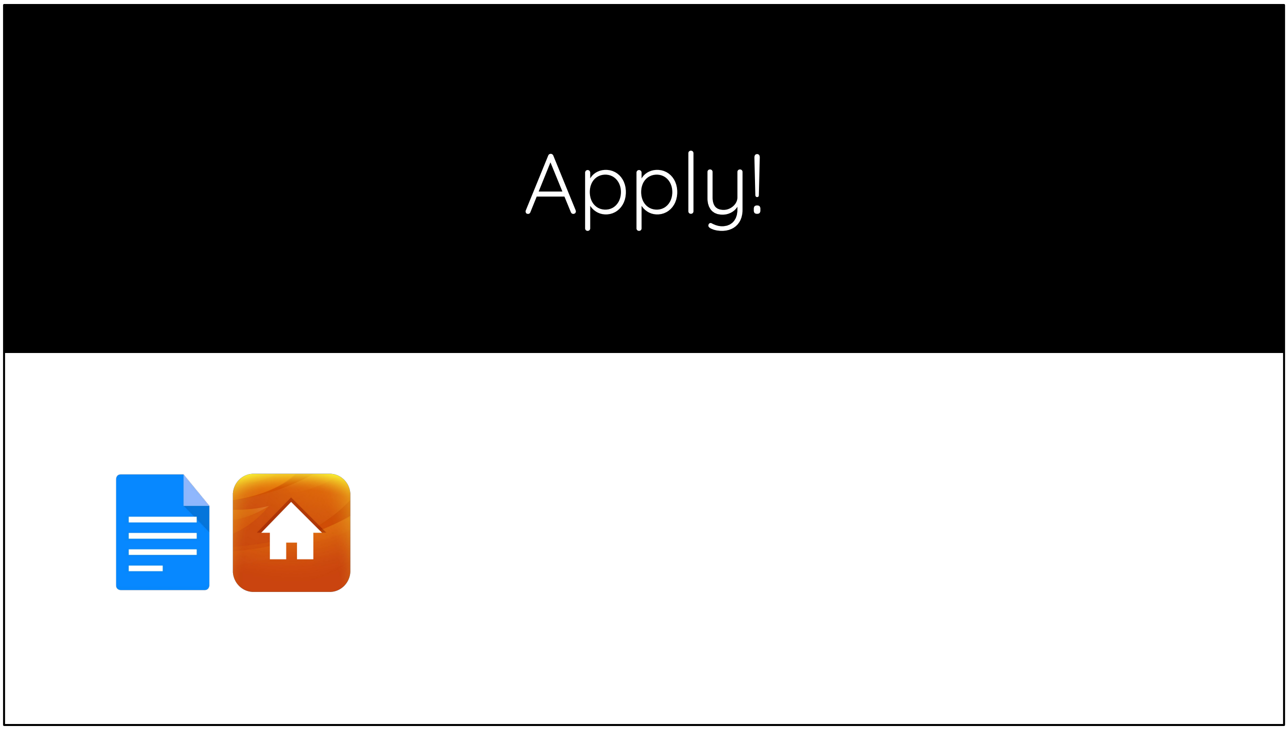


Apply!

- Make it **easy to decide** to interview you
- Expect to be largely **ignored**

Apply!

- Usual **CV** tips (2 pages, spell check!)
- **2** pages
- **Relevant** experience only - no Spar checkout roles
- SPELL CHECK!



Apply!

- **Website -** even GitHub pages
- **Simple**, clean, easy
- Contact details, online CV, blog, **Portfolio**
- **Games** or **tech demos**
- To the point **quickly** (< 3 minutes)

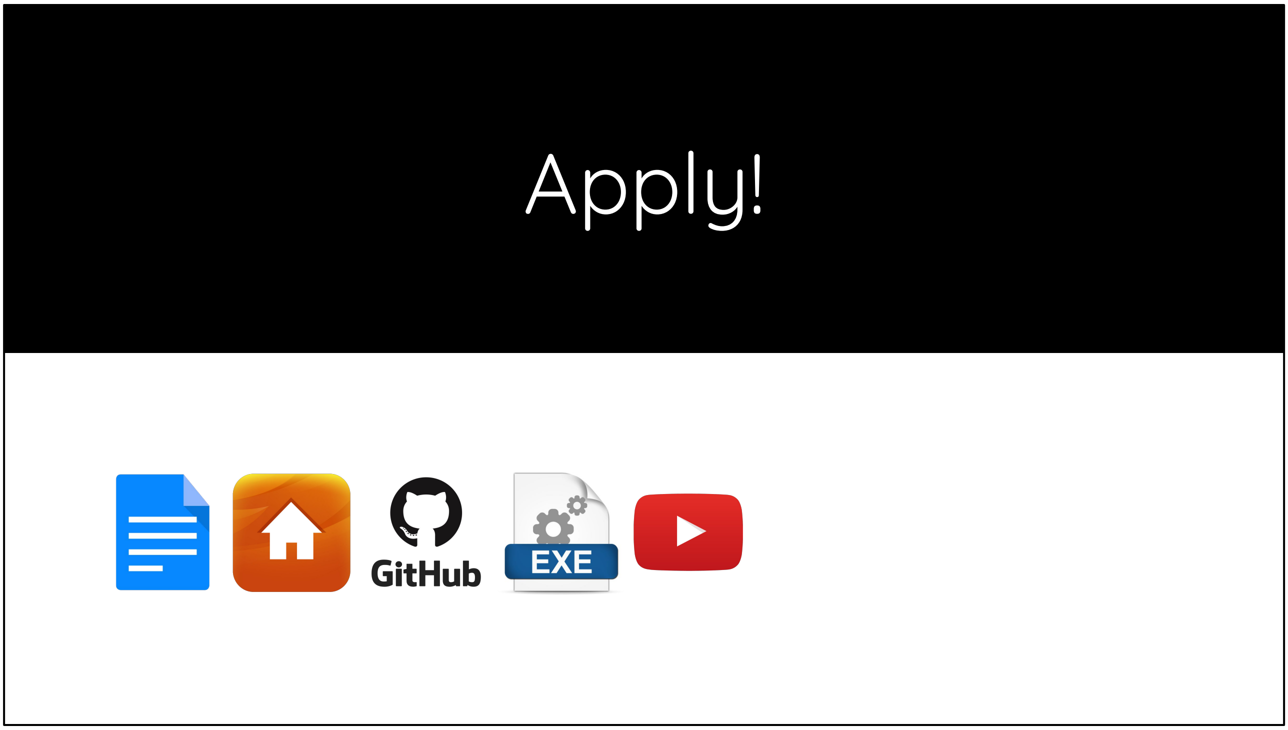
Apply!

GitHub

- Make **source** available
- Good idea: use **source control**
- **GOOD CODE!**
- **Clear**, concise, easy to **read**, well-**commented,** well-**structured**
- **Compiles!**
- Will be judged

Apply!

- Provide **executables**
- No missing **dependencies** (DLLs)



Apply!

- **Videos embedded** in YouTube or Vimeo

Apply!

- You will be **Googled** - behave!

- **LinkedIn** heavy with recruiters

- Apply **directly**, or ideally through **contact**

With any luck, you’ll be called...

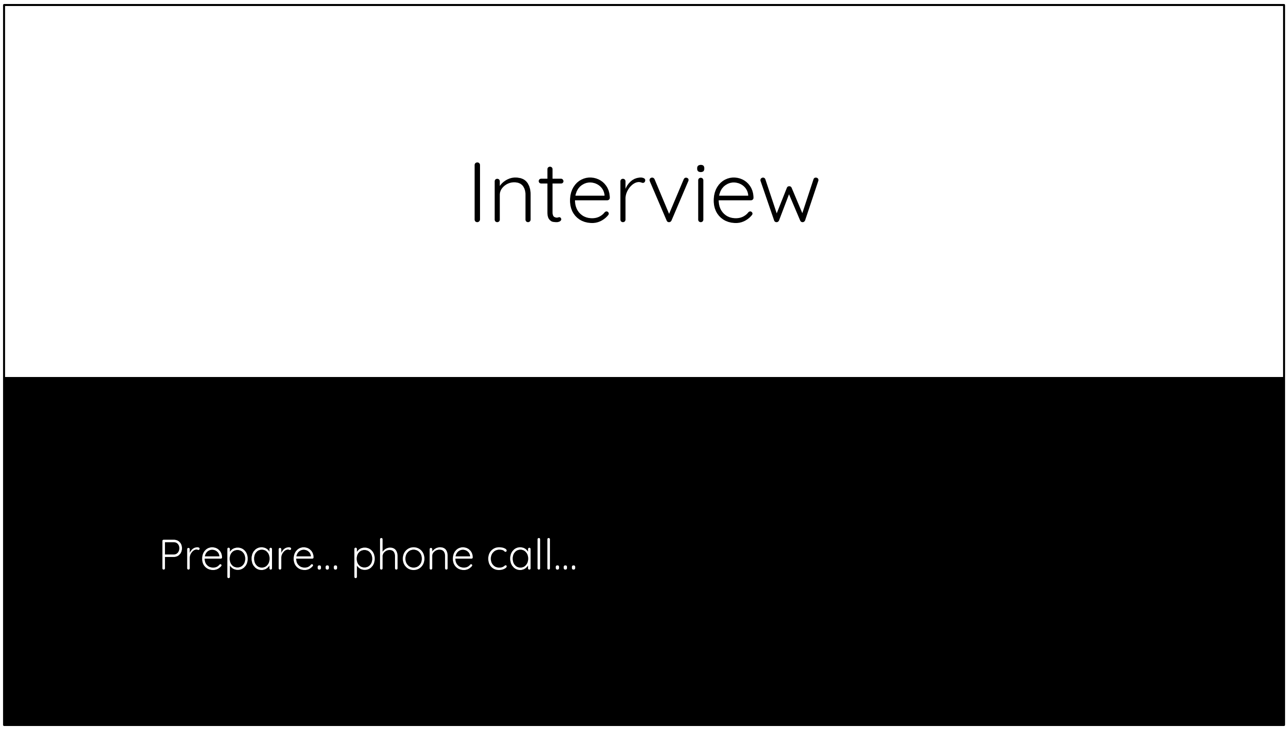
Interview

- **Different** everywhere
- **Typical** interview progression….

Interview

Prepare…

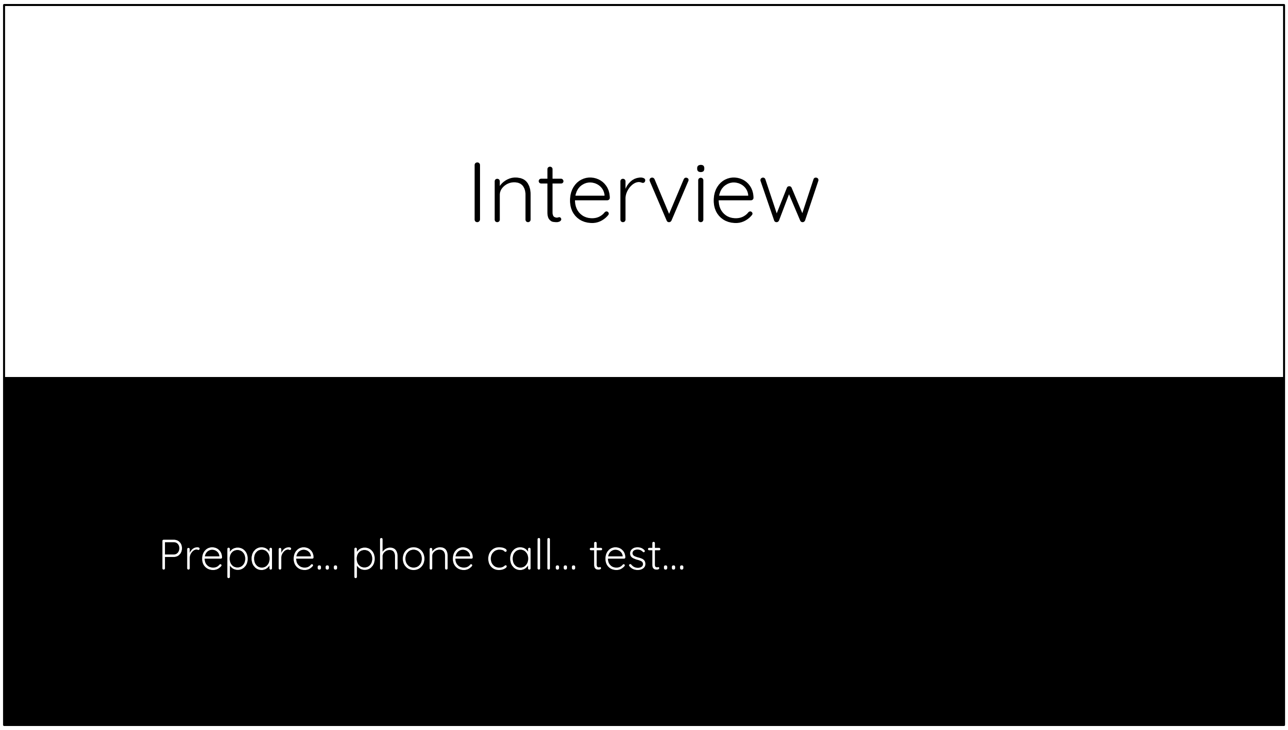
- Do your **homework**
- Play their **games** (be diplomatic if asked!)
- Have **questions** ready



Interview

Prepare… phone call…

- Initial call with **HR**
- Touch base, **briefing** on role & hiring procedure
- Find out details of **hiring process**
- Find out **names** of interviewers



Interview

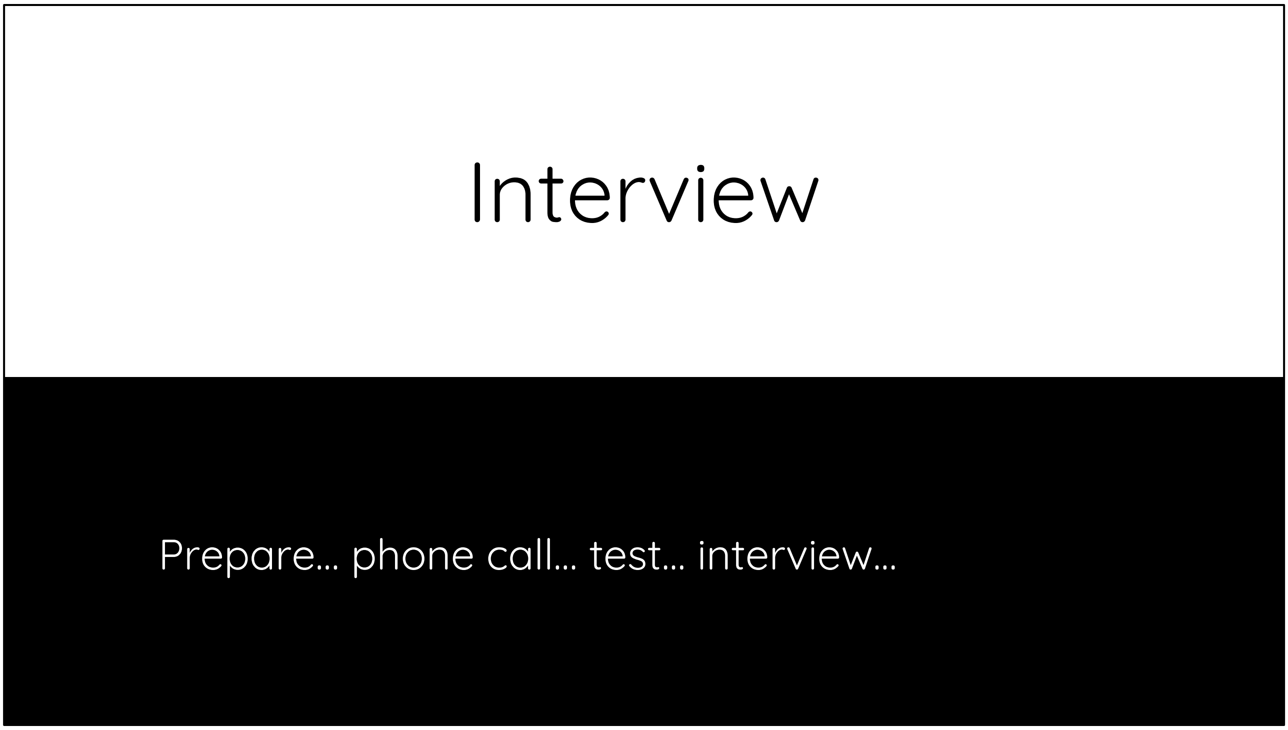
Prepare… phone call… test…

- Online timed **programming test**

- Be **prepared**

- Code **editor** open
- **Notepad**
- No **distractions**

- Good **code**!



Interview

Prepare… phone call… test… interview…

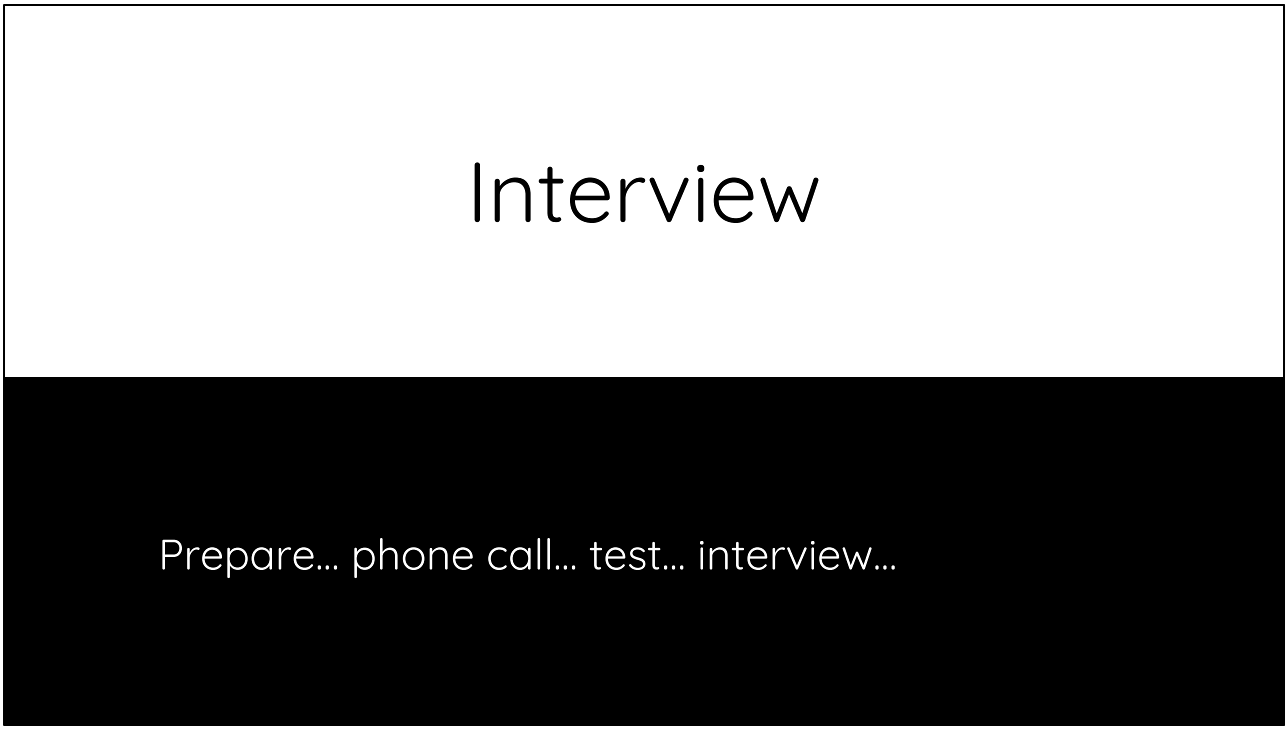
- Phone and/or on-site **tech interview**

- To find out **how** you think
- To find out **what** you know

- Always checking for **team fit**

- **Two-way**, ask questions & clarification

...continued…



Interview

Prepare… phone call… test… interview…

- Chit-chat, talk about **games & tech demos**
- **Steer** towards best work

- Code questions - **practice** the common ones
- C++ keywords, data structures & algorithms
- Virtual destructors, vtables, memory allocation

- **Domain-specific** - dot products, A*

- **Whiteboarding**

- **Don’t know** = don’t panic
- Break it down
- Identify assumptions

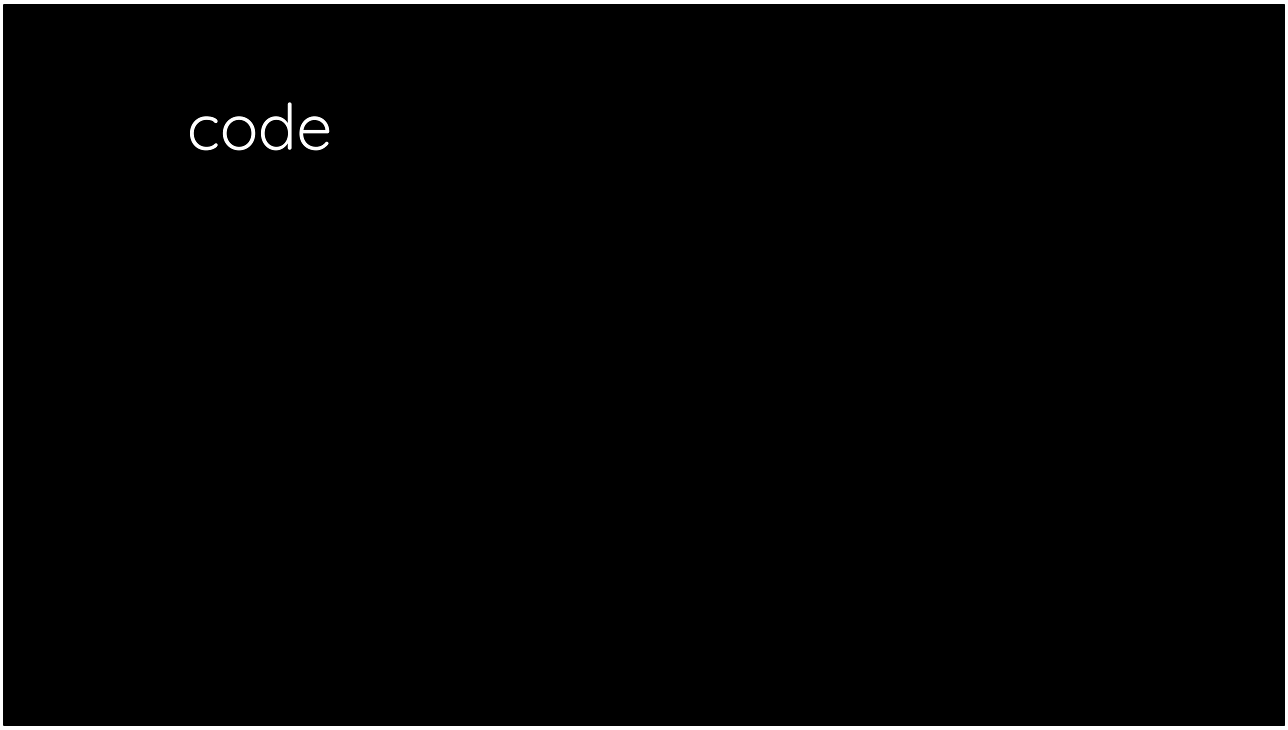
Interview

Prepare… phone call… test… interview… job?

- No luck - **learning experience**

- Follow up
- **Thanks**
- Ask for **feedback**
- **Take it** on board!

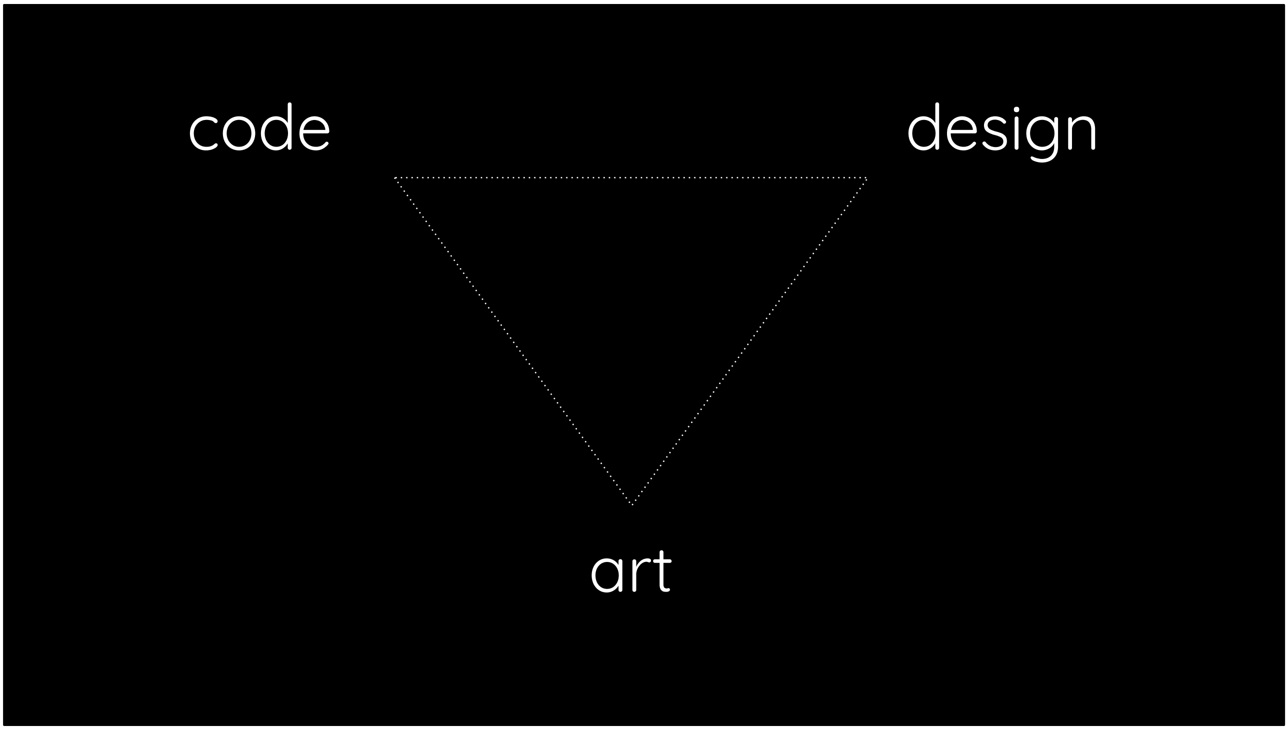
However, if you are successful, the real fun begins...



code

- **Knee-deep** in a (mostly undocumented) foreign code base
- **Overwhelming**… don’t panic!
- **Ask** questions, but don’t **pester**
- **Explore** the code and **learn**

But you might find the most interesting part of the job is...



code

design

art

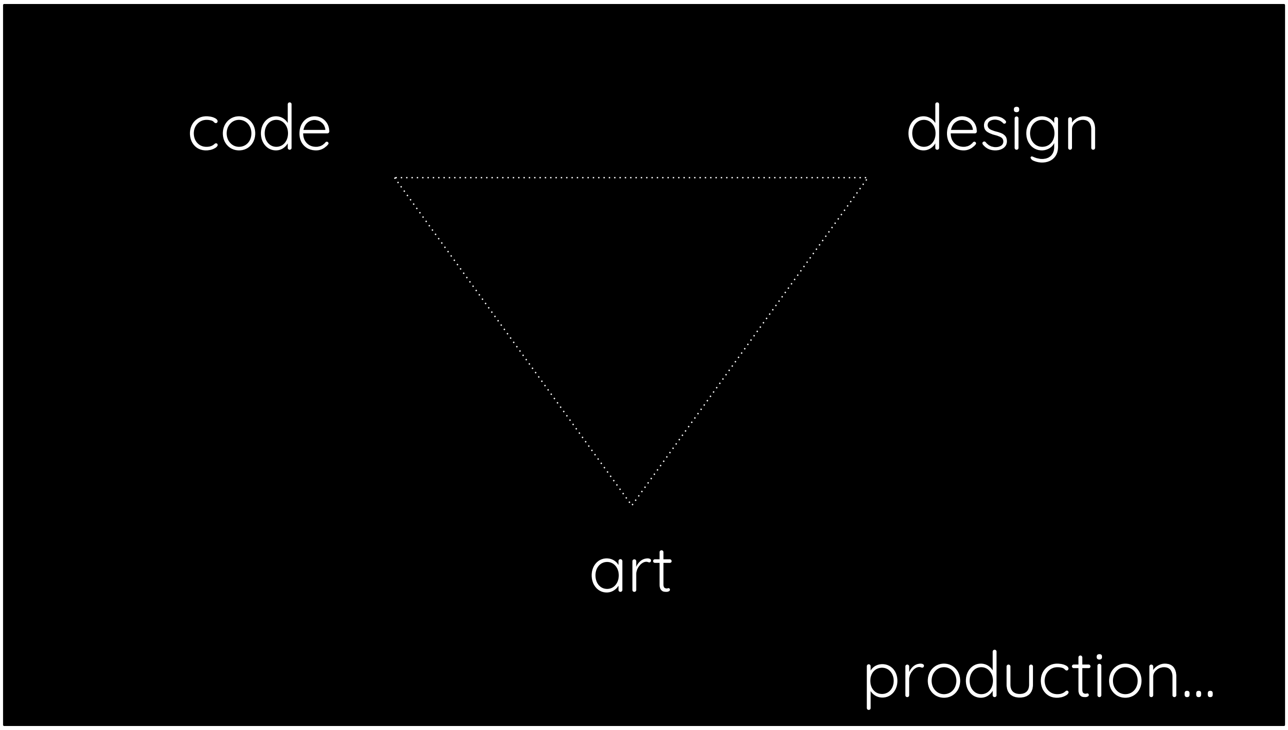
- Learn to work with **other disciplines**
- Deal with **big personalities**

- Good **communication** is paramount
- Why **team fit** is so important

- Everyone somewhat **selfish**
- Learn to **compromise**
- “What’s best for the game?”

- Good coders **don’t just say no**
- **Explain,** provide **alternatives**
- Or eventually they’ll just **stop asking**

There’s a sometimes-forgotten 4th side of the triangle, can be most important...



code

design

art

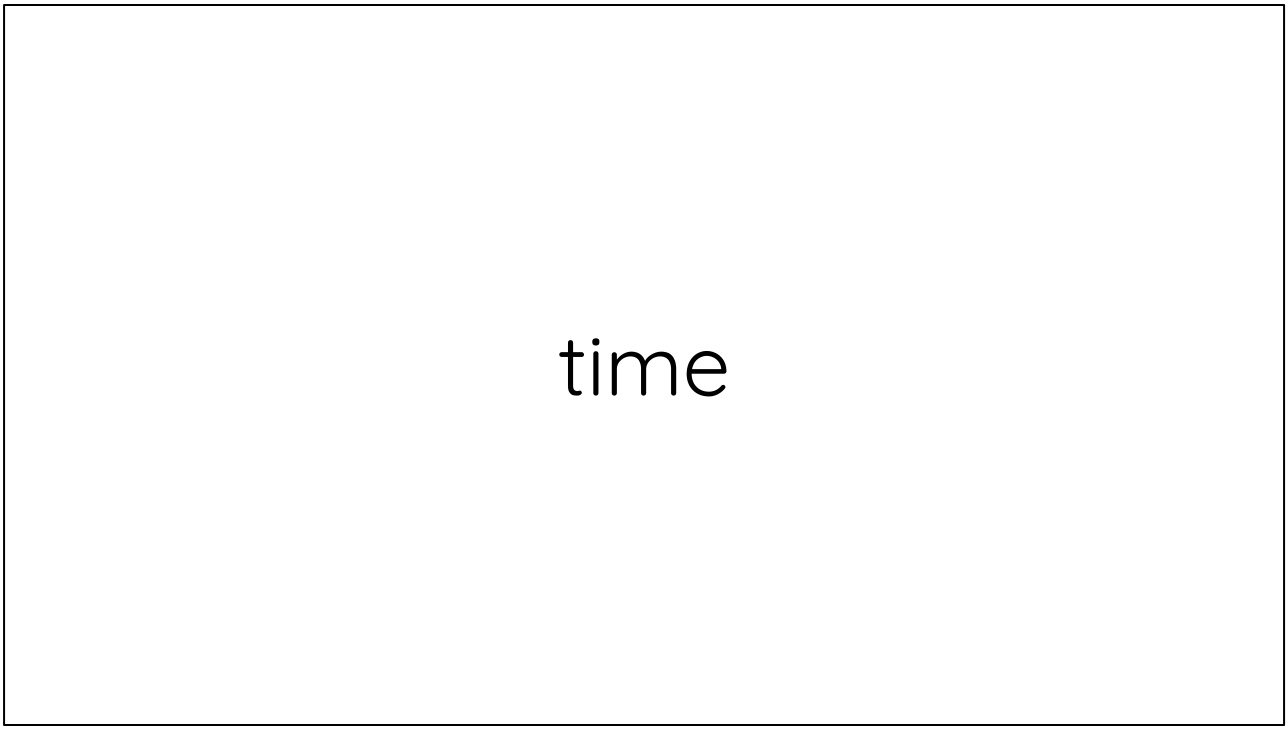
production...

- **Production** team

- Producers & project managers

- Two roles: keep everyone **working together,** and

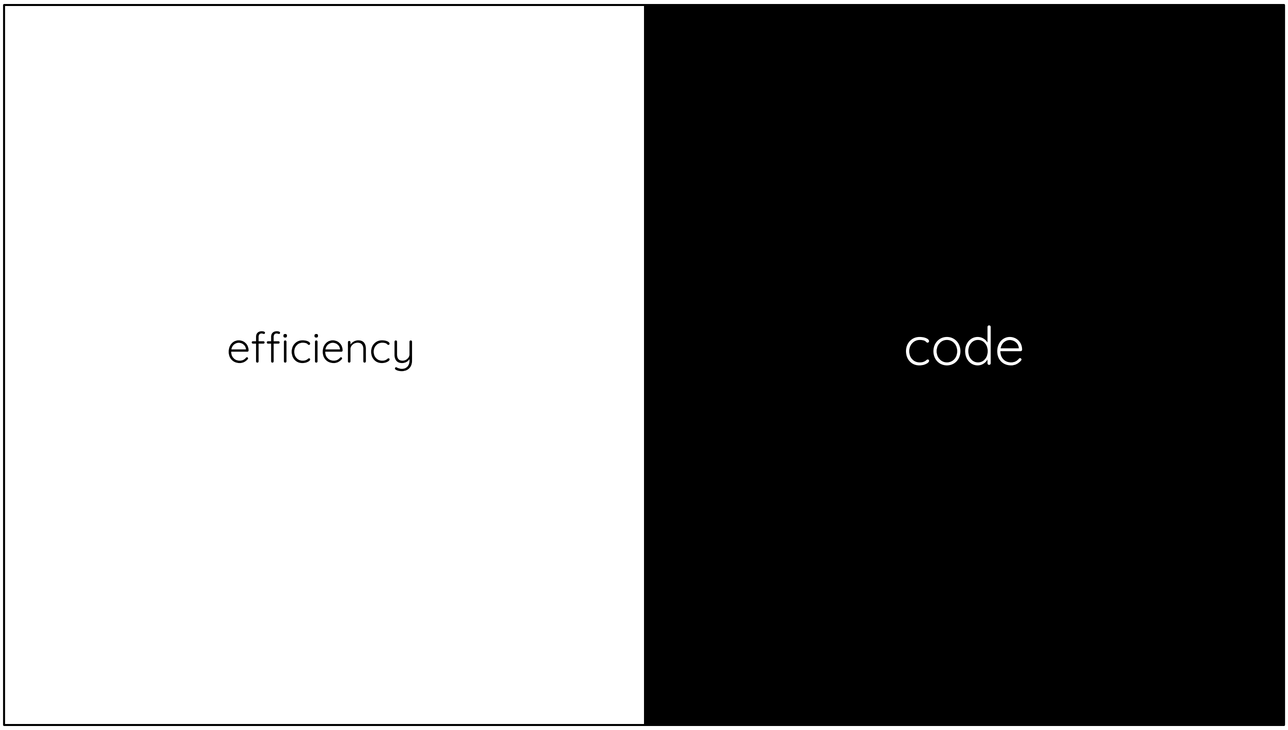
Keepers of the production’s most precious resource...



time

- Time!
- Can make **any game** given **enough time**
- Game **design changes** over life of production
- Constant **scoping** to make the best game in time available
- Good work **estimates** essential
- Poor time management = **crunch**

Time efficiency needed everywhere...



efficiency

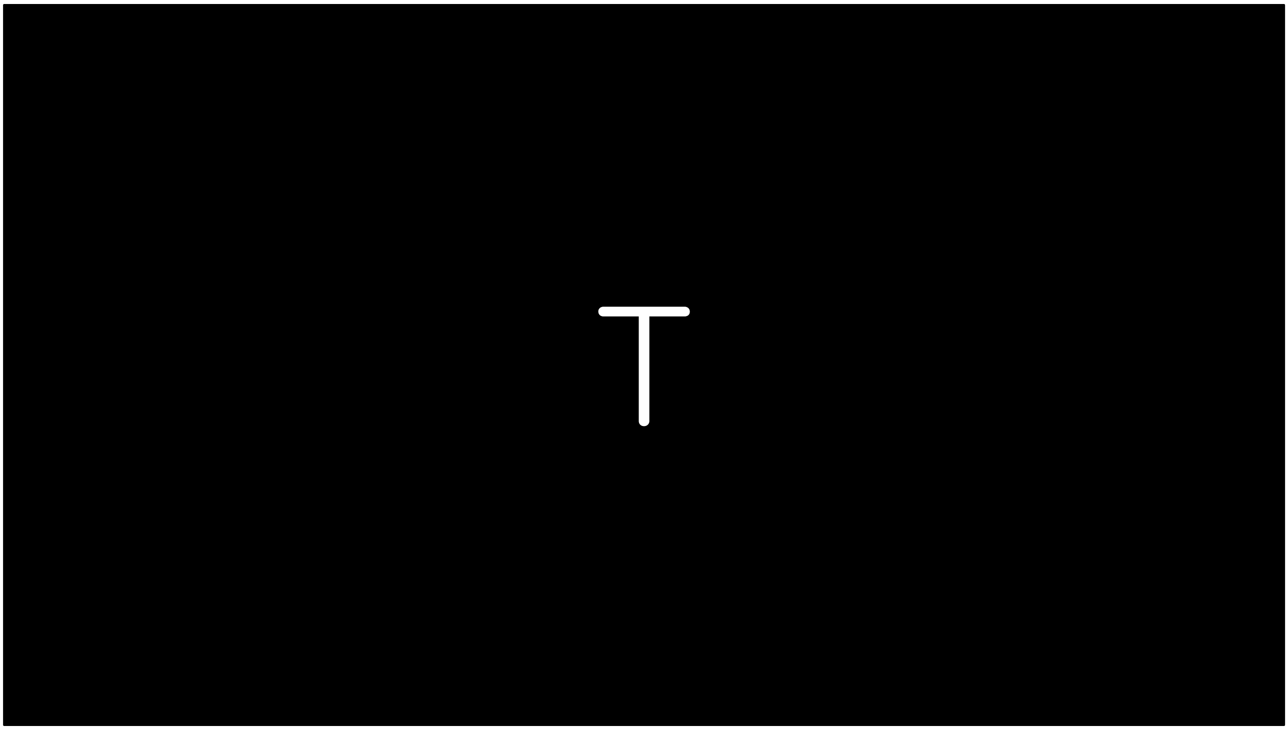
code

- Always keep time **efficiency in mind** when coding
- Code **maintenance**
- Future **debuggability**
- Build **stability**

efficiency

workflow

- Artist & designer **workflow**
- **Pipeline - huge** topic in itself
- **Convert source assets** into loadable data
- **Don’t do online** what you can do offline



T

- A good senior coder is **T-shaped**
- **Deep** knowledge of a particular discipline
- **Broad understanding** of everything else
- Make **decisions taking everything above** into account

@keithoconor

keith.oconor@gmail.com

www.fragmentbuffer.com