SPERASOFT
A KEYWORDS STUDIO

GAMEDEV CITYFEST

DESIGN AND PROGRAMMING
THE BEST OF FRENEMIES

Steve Thornton
Lead Game Designer

Ben Sharples
Technical Design Director

WHO IS WHOM

Steve Thornton
Lead Game Designer

- 8+ years experience
- 10+ shipped games
- Handheld/Console/PC
- Previous roles
  - *Game Director*
  - *Lead Level Designer*
  - *Assistant Lead Designer*
  - *Senior Designer*
  - *Games Designer*
- Cool and fun, nice hair 😘

Ben Sharples
Lead Technical Designer

- 10+ years experience
- 10+ shipped games
- Mobile/Handheld/Console/PC
- Previous roles
  - *Gameplay programmer*
  - *Script Team*
  - *Level Designer*
  - *Technical Designer*
  - *QA Technician*
- Old and cranky, nearly bald 🙄

OVERVIEW

- Design are the high level vision holder and content driver of the game
- Programming make everything in the game actually work
- A game starts with design and ends with programming

What is this talk about?

- The friction between conceptual vision holders and technical implementers
- Resolving common conflicts and dispelling myths
- Finding ways to improve and encourage trust between the disciplines

OVERVIEW

SPERASOFT A KEYWORDS STUDIO @ GAMEDEV CITYFEST

Clarifications

1. Observations are based on AAA game development
2. Observations are based on a “vision-led” hierarchy
3. We will eventually address hybrid roles
4. Problems identified are based on common trends

THE SINS

Design Sins
(according to programming)

- Don’t understand how things work
- Don’t consider the cost of requests
- Don’t provide clear instructions
- Aren’t reliable, change their mind
- No hard skills

Programming Sins
(according to design)

- Exaggerating the cost of requests
- Letting personal agendas affect technical evaluations
- Creatively unambitious
- Don’t care about the game

DEPARTMENT FOCUS

SPERASOFT A KEYWORDS STUDIO @ GAMEDEV CITYFEST

Design

FLOW

Programming

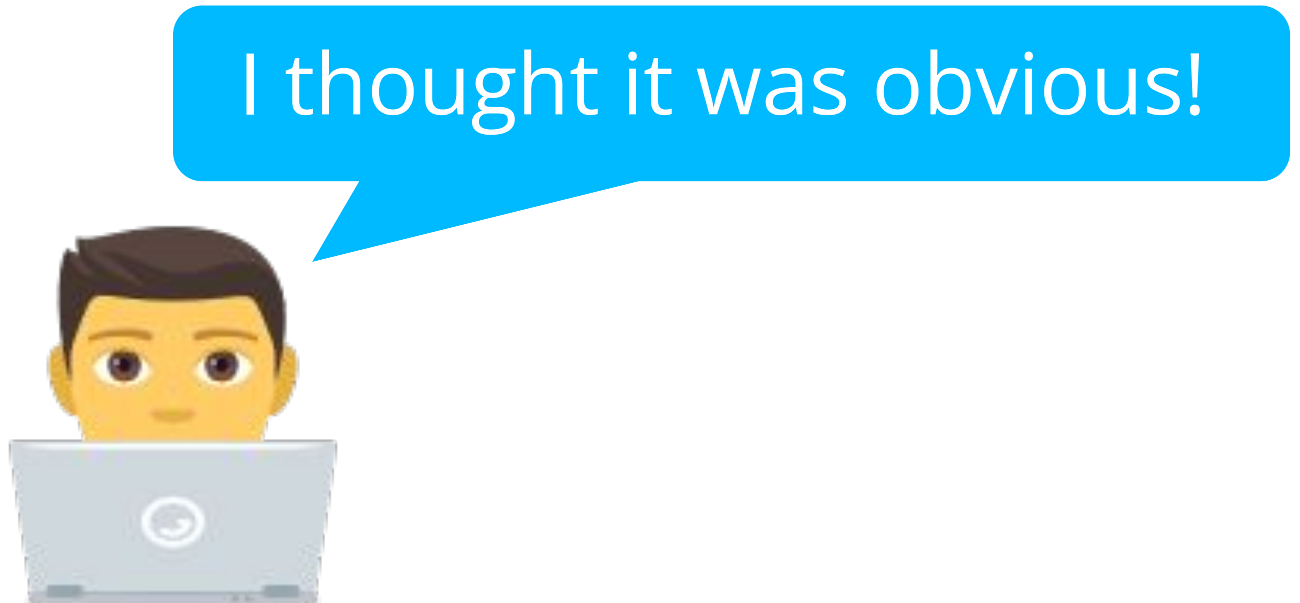
MECHANISM

BAD SPEC

Design must properly communicate their vision to the team, but…

What can go wrong?

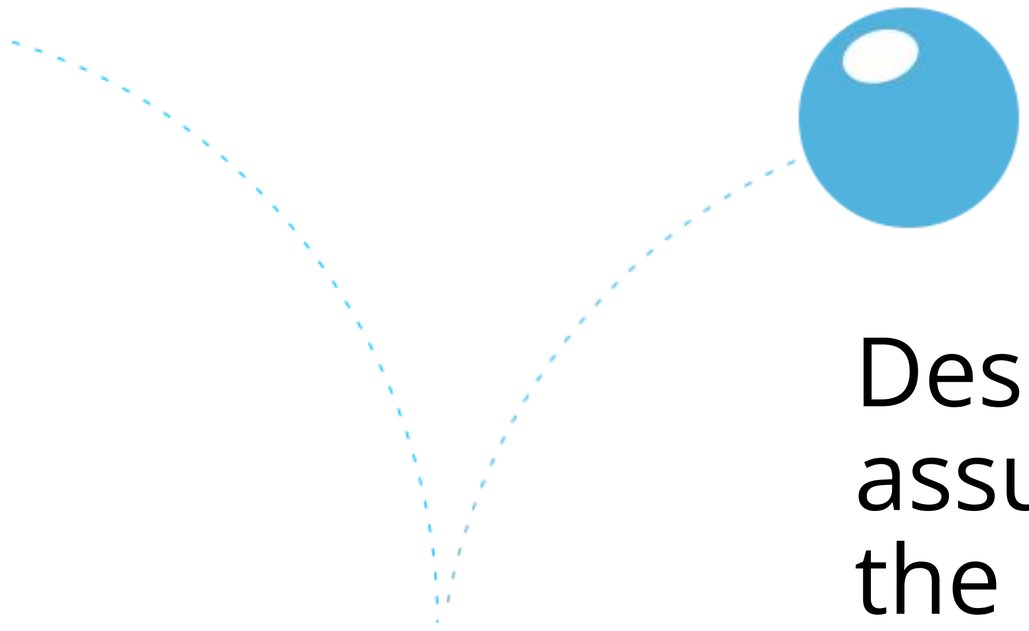
- Not identifying all required assets
- Not covering full feature flow
- Not considering variables outside the intended flow



COMMON SENSE

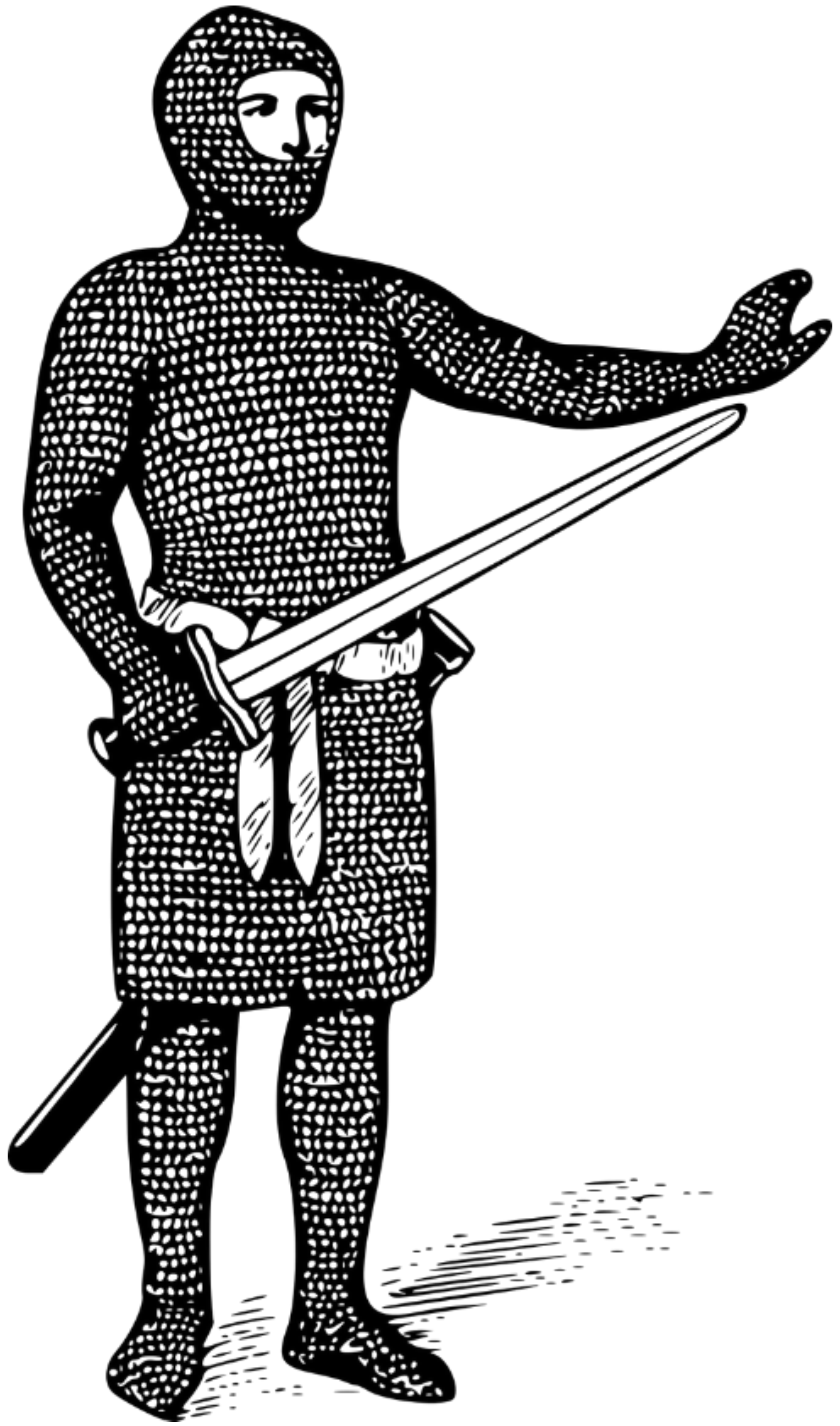
Why wouldn’t a non-designer use “common sense” to solve a problem themselves?

- Assumption is a risk
- Everyone has their own frame of reference
- Any detail not specified can be easily misinterpreted



Designers often make the mistake of assuming the game world will behave like the real world for “free”.

PLAYING DEFENCE



Designers can become defensive under questioning

dership

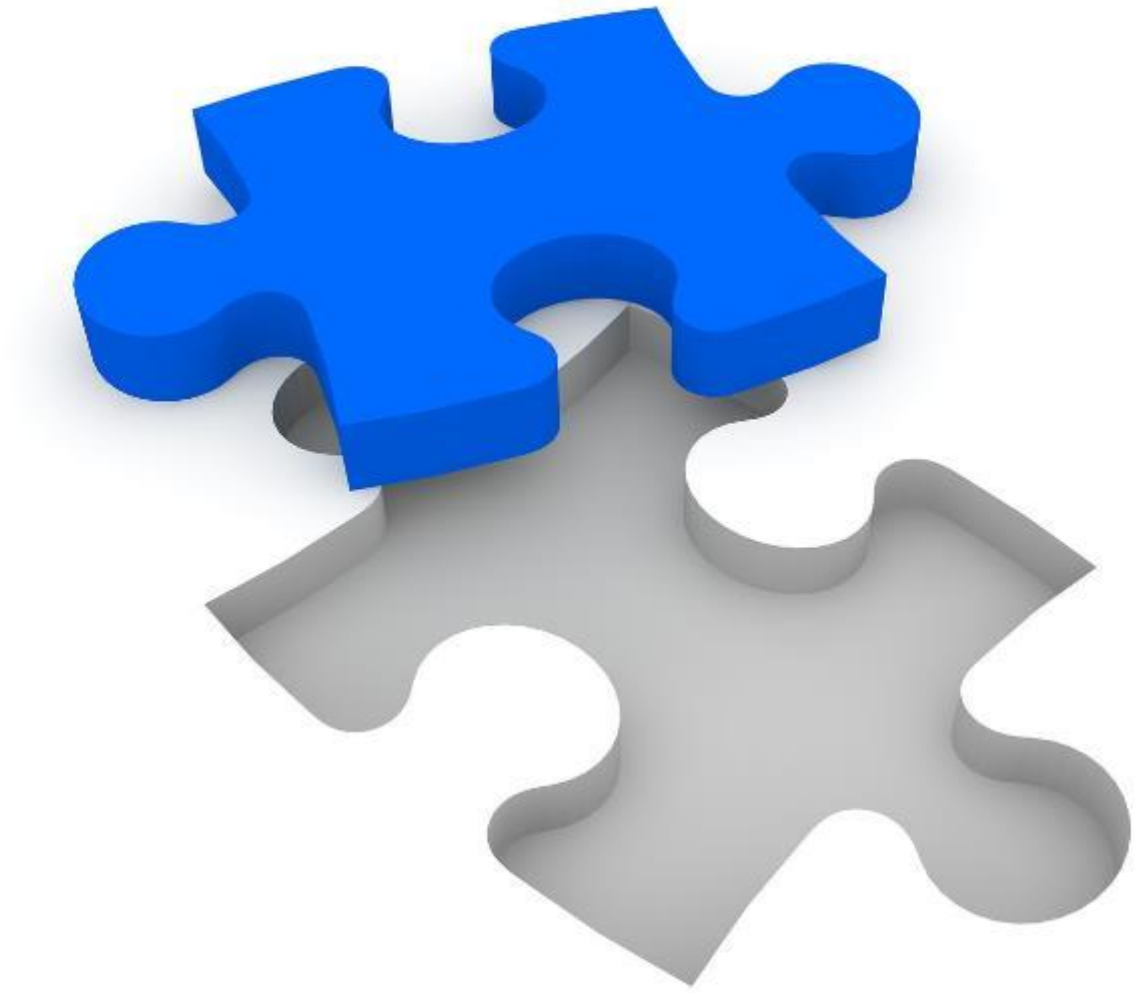
pt

DETAIL ORIENTATED

Consider the programming perspective…

- Programmers are detail orientated by necessity – in their world a single detail that is missing or wrong can break everything
- Programming tasks are unpredictable
- Programmers worry something cannot be done
- When asked to estimate an intimidating task, programmers will play it safe to avoid looking bad

Programmers can help the discussion process by polishing their own soft skills.



“THAT’S IMPOSSIBLE”

The truth? Anything is possible

W ”?

- It exceeds the current limitations of the chosen tech

THE IMPOSSIBLE DREAM

What should a designer do in the face of the impossible?

- Don’t try and pitch the “how” to experts
- Ask to actually hear the estimated cost
- Make clear the core goals of your request
- Find the key places where you can compromise
- Give the programming team time to research
- Make a prototype

Assure your experts that you won’t make a decision without them!

DECISION TIME

Help programmers inform your decisions

- Sell the vision, the "why" with the "what"
- Avoid pressure to come to a decision "in the room"
- Give time to research and prototype

Why do engineers fear "bad" decisions?

- They don't want their time wasted
- They don't want to spend their skills on something they don't agree with
- They fear it will lead to stressful schedules or over-time

What can designers do to reduce this fear?

- Be categorically opposed to over-time and crunch
- You are a team leader, do not allow people to work too hard for your goals

FEAR OF CHANGE

Why do designers change their mind?

- All creative tasks require iteration
- Feedback from higher in the hierarchy
- Requests from other departments

How can designers limit the impact of changes

- Be sensitive to morale when passing on feedback
- Don't judge the idea by the prototype assets
- Future Proof your ideas against the most likely changes

HYBRID THEORY

- The most common “hybrid” role is designers working in the editor
- Generally designers only work on high level editor tasks, using tools exposed / created for them by programming

Advantages

- Designers can iterate manually
- Design get to see the impact of their requests first-hand
- Improvements to relationship due to hard skill demonstration
- Technical can inspire the creative

Disadvantages

- Sometimes a little information can be worse than no information
- Design do not always follow “best practice” when working in tech – No training
- Creates tool requests from design in addition to feature / asset requests
- New type of design support is needed – usually creates a tools team

HYBRID THEORY

- It’s rare in AAA for programming to be directly involved in creative decisions, outside of suggesting alternatives and compromises to requests
- Some studios have a technical expert role between design and programming

Advantages

- Acts as an ambassador between design and programming
- Can assist in future proofing, such as suggesting key balancing variables

Disadvantages

- Being too close to the implementation cost to make objective decisions
- Technical tasks are time consuming, and require too much personal focus
- Risk of “bottle-necking” production

FIN

“Making a game combines everything that’s hard about building a bridge with everything that’s hard about composing an opera. Games are basically operas made out of bridges.”

- Frank Lantz

Director of the N.Y.U. Game Center and designer of the iPhone puzzle game, Drop7

SPERASOFT
A **KEYWORDS** STUDIO