

The banner features a dark background with colorful geometric shapes (polygons and cubes) in shades of blue, purple, yellow, and red. In the top left corner is the 'INDEPENDENT GAMES SUMMIT' logo, which includes a stylized 'i' inside a circle. The main title is 'From Indie to Fable and Back. 30 Years of 'Wisdom''. Below the title is the speaker's name 'Dene Carter' and his title 'Managing Director Fluttermind LLC @Fluttermind'. At the bottom left is the 'GDC' logo, followed by the text 'GAME DEVELOPERS CONFERENCE | FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 #GDC17'. A small 'U' logo is visible in the bottom right corner.

INDEPENDENT GAMES  
SUMMIT

**From Indie to Fable and Back.  
30 Years of 'Wisdom'**

**Dene Carter**  
Managing Director  
Fluttermind LLC  
@Fluttermind

**GDC** GAME DEVELOPERS CONFERENCE | FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 #GDC17

Before we start I need to do a couple of basic housekeeping things:

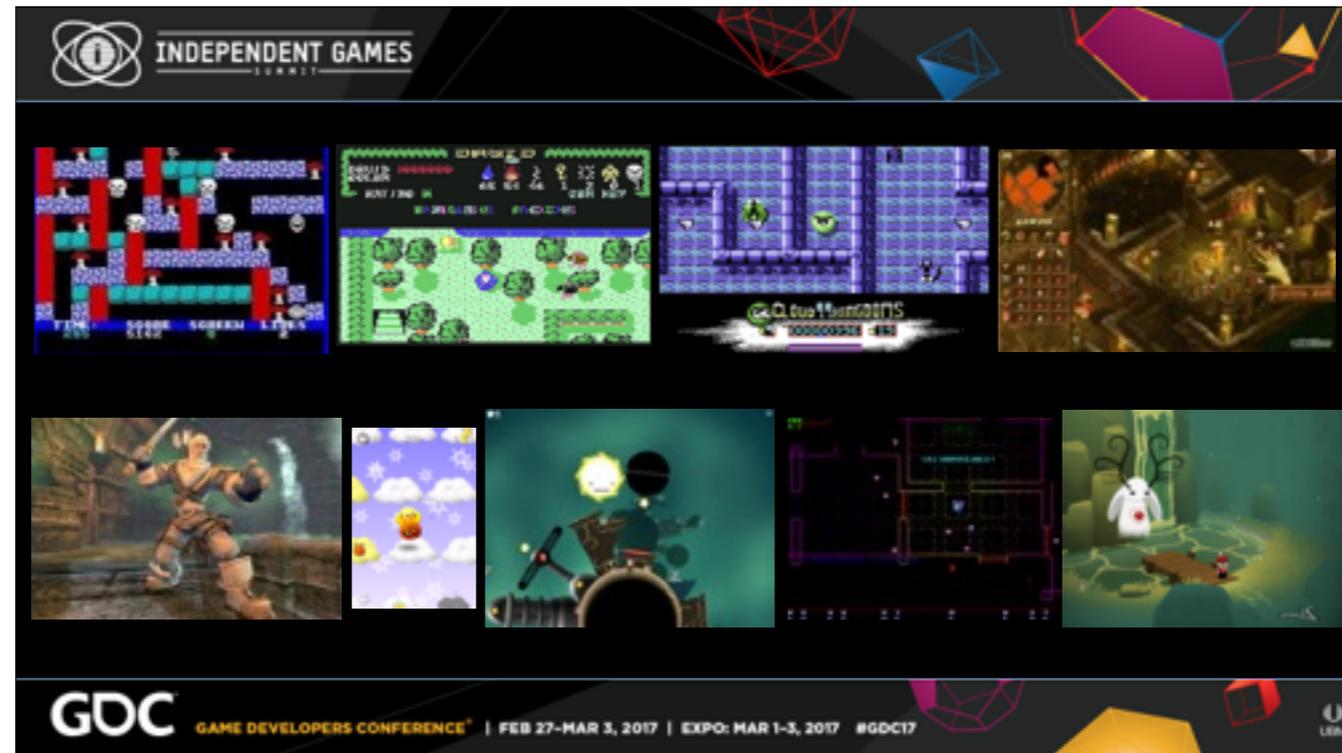
Ask you to turn off your mobile phone-ringers

Fill in the online review forms so the organizers know NOT to book me next year.

So... Hello! I'm Dene Carter, and I run a **one-man** indie studio called **Fluttermind**.

I've had the **privilege** of writing games professionally since I was 14.

That was back in 1984, so that's 33 years.



Over that time:

- I created *Druid* on the C64 (see top and middle-left-ish)
- I was part of the core team on the original *Dungeon Keeper* (top-right)
- I started the *Fable* franchise with my brother, Simon and artist Ian Lovett (bottom left)
- I wrote *Incoboto*, which was nominated for a BAFTA (bottom-middle)
- my current project is **Wardenclyffe** (bottom right)

So, I have worked in companies of all scales:

From tiny solo '80s bedroom indie coder, all the way through to AAA and back again. Every scale.

So that's me. I'd like to ask a few questions so I know the audience a little better.

How many people here have released a game? - well done, you

How many of you are artists? - the ones who look stoned but stylish

How many of you are coders? - the pale, precise ones

How many of you are solo devs? - the lonely, haunted ones

Okay. Thanks for that.

This talk isn't like a lot of other talks.

It isn't a How To manual for making money,

Or 'Survivor Bias - the seminar'.

It's a collection of things I learned over the last 30 years that I wish I'd known earlier.

So with that, let's look at the first thing I learned.

## I Am An Idiot (first learned in '84, then '85, '86...)

- Rockman: made \$7500 for 2 weeks of work
- Wow! I could make 26 games a year!
- \$195000 a year! (\$438,077 in today's money)
- Takeaway: I was really stupid when I was 14
- The *best* '80s developers only got 12-14% royalties
- Everything is cyclic. Learn from previous cycles.

Rockman: made \$7500 for 2 weeks of work

Wow! I could make 26 games a year!

\$195000 a year! (\$438,077 in today's money)

Takeaway: I was really stupid when I was 14 because. There was no way that this model was sustainable because...

### **MAKING GAMES PROFITABLY WAS NEARLY IMPOSSIBLE**

**The best '80s developers only got 12% to 14% royalties for FINISHED games**

**This** is why many '80s developers left the industry!

**Everything is cyclic. A lot of developers are struggling again.**

It's nothing new. But this time, you've got a larger **audience**, more **distribution** channels, and decades of **development** to learn from.

You can avoid mistakes people like me made the first time around...

## You Are Not Your Players ('84)

- My first game & my first stupid, public mistake

...like this

This lesson started with my first game, Rockman and my first stupid (public) mistake.



Poor. Only had a Black and white TV. So, to me, the game looked like this.

[Black and White Rockman] Of course, a majority of players weren't me in my house with a B&W TV.



[Switch to colour] Comment on how hideous it is.  
Reviews: 'Hideously garish graphics'. I was 15. It hurt <sniff>

## You Are Not Your Players ('84)

- My first game & my first stupid, public mistake!
- Players see things completely differently to you
- Avoid difficulty/complexity-creep
- The real world has distractions and beer
- Players live inside their brains, not yours
- Those you ignore will ignore you in return

### **Your development environment is not representative of the final experience.**

The Black and White TV was not representative of my players' final hardware.

While nobody here is developing in black and white...

### **Players perceive things COMPLETELY DIFFERENTLY to you as a developer.**

Your players are not hunched over a copy of Unity with a grey border surrounding a tiny screen

Your players are not looking for bugs

Your players have not seen the same scenarios 50 times over that day alone

If you forget this, it can lead to...

### **...difficulty/complexity-creep**

Your players have not built muscle memory over a year of development

And so, players will find your game **harder** than YOU do

### **And there's the real world to consider, too**

Real players lounge on a sofa, or relax near their PC... with a beer.

Real players don't pay attention, or understand that crucial information **is** crucial.

Real players go on holiday and need to remember what they were doing when they return!

Real players take a break because they need the toilet after all that beer

And if you haven't tested your mobile game on the **toilet**, you're in trouble.

Real players don't care about your game as much as you do

### **Fundamentally, players live inside their own brains, not yours**

Your players aren't living **your** life, experiencing the world with **your** brain

Some of your players don't speak English as a first language.

Some aren't straight white guys.

But most of all, stuff that's **obvious** to you will **seem utterly bizarre** and alien to someone who is not you.

Design for **players'** minds, not yours

### **Every time you decide some group of players aren't worth caring about...**

...you guarantee that group won't care about your game.

## Generalizing is *Really* Useful ('84)

“Specialization is for insects”

-*Robert A. Heinlein (and '80s developers)*

“An expert who is too narrow has difficulty collaborating.”

- *Valve Employee Manual (referring to “T-Shaped” individuals)*

I know a lot of people are going to disagree with me on this one.

So I'm going to start off with some quotes to help me sound more authoritative.

“Specialization is for insects” - writer of *Stranger in a Strange Land*

“An expert who is too narrow has difficulty collaborating” - Valve - T-Shaped individuals



# Generalizing is *Really* Useful ('84)

- In the early days most developers generalized
- Developing a broad skill-set aids communication
- Knowledge breeds respect (never say 'Just')
- Ignorance is an energy. Use it!
- Be a Human. Learn a skill outside your area



GDC GAME DEVELOPERS CONFERENCE | FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 #GDC17

### **In the early days, everyone was a generalist!**

In the '80s I made **typefaces, art, coded engines, created sound effects** and **music**.

Even during my time at **Bullfrog** I was coding the user interface on **Dungeon Keeper** one day and shadow renderer the next.

It's not because I'm special, it's because that's just what companies had to do to keep costs down!

I'm sure this sounds familiar to the solo devs in the audience.

**Indie** teams are getting **bigger**, and production values **are** rising.

So you might be thinking that it's wiser to specialize.

### **Developing a broad skill set isn't just for you**

Larger teams require collaboration and clear communication.

Being a generalist doesn't just help you get more work done with fewer people...

...it helps you learn how different disciplines think, their vocabulary, their workflow, **their pain**.

### **Knowledge breeds respect for others, and that helps everyone you work with.**

It stops you making assumptions about how difficult a task is.

It stops you saying: 'Just'.

'Just make the reds in that image brighter'

'Just make the protagonist run 10% faster'

Saying 'Just' suggests that you know a task is easy. It might not be! Learning more about others' areas of work makes you less of an arse to work with.

### **Ignorance is an energy**

Those of you who put up your hands saying you were coders or artists or whatever.

Don't forget that most of all, you are **HUMAN. You are designed to adapt**, and you can do **pretty much anything** with a bit of effort.

If you feel you can't do something because you lack the skills, or feel too ignorant...

If you're sitting there saying "I can't do art/math/coding"... I once said that, too. And I was wrong. Because I'm an idiot.

At various points I was convinced I could never learn:

### **3D vector maths**

### **Music theory & production**

### **Colour theory (as you saw in Rockman)**

In each case, I was wrong.

### **Be a Human. Learn a skill outside your area of specialty.**

## Rapid Team Growth is Dangerous ('01)

“If your team can eat more than two pizzas,  
it’s too large”

- *Werner Vogels, Amazon VP/CTO*

Another quote for you...

“If a project team can eat more than two pizzas, it’s too large.”

- Werner Vogels, Amazon’s Vice President and CTO

I love the fact that Pizza is considered a metric. My people.

## Rapid Team Growth is Dangerous ('01)

- Early Fable (peace), later Fable (fear & rumours). Why?!
- Communication paths do not grow linearly with team size
- Fred Brooks:  $[n * (n - 1)] / 2$

### From Peace to Fear

Early on in Fable's development at Big Blue Box, there were just 5 of us.

We were living off savings and a bank-loan.

No publishing deal. Unstable and stressful. BUT, **team morale and cohesion** remained high.

Then got a Deal = bigger office & our company grew. All good up until ~12 people.

Rumour Mill: "Is it true? I heard that Fable is being canned!"

We didn't know why. More stable, so why more fear?

Apparently, this kind of thing is quite common, because...

### Communication Lines do not grow linearly with team size

Small: Simple communication - turn chair & shout

Large: Can't have everyone in every single meeting. Cliques & gaps in knowledge develop.

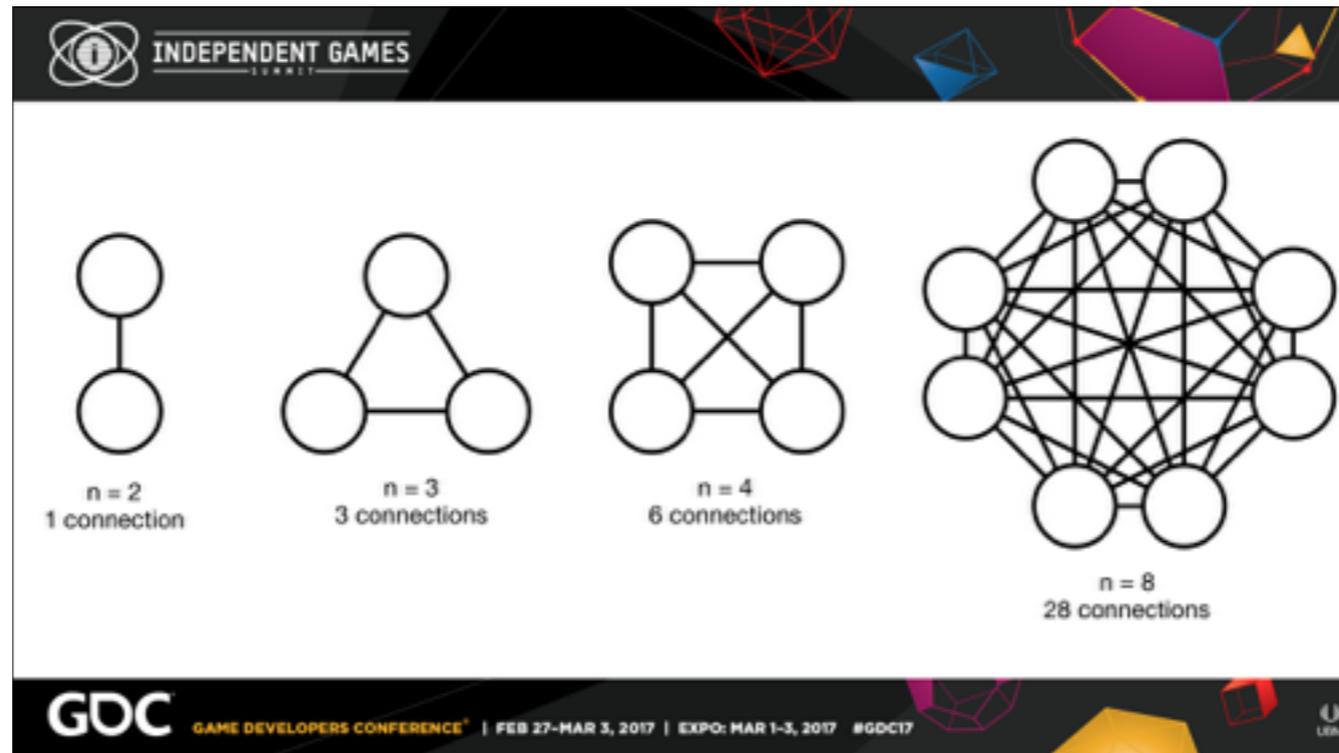
This causes fear & uncertainty. People will fill in those gaps with fear and rumour.

### $n*(n-1) / 2$ (Fred Brooks Mythical Man Month)

Here's a lovely equation for calculating number of communication paths in your team.

N is the number of people in your group.

What you end up with is this:



2 people, only one connection.

4 people, 6 connections.

It all stays fairly reasonable up until you hit 8 people where you have 28 connections.

28 flows of information where things can be misinterpreted or... simply made up!

**This is why producers exist!**

If you've ever wondered what producers are for, this is why. Managing these lines is a \*job\*.

It doesn't exist at the birth of a startup & seems like it isn't real work.

It is. Without it, you end up with a demoralized, confused team.



# Rapid Team Growth is Dangerous ('01)

- Early Fable (peace), later Fable (fear & rumours). Why?!
- Communication paths do not grow linearly with team size
- Fred Brooks:  $[n * (n - 1)] / 2$
- The Ringelmann Effect:
  - "Larger groups lead to less motivation & coordination & productivity"
- Fix? Help whole team understand every member's value
- Think before growing your team!



GAME DEVELOPERS CONFERENCE | FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 #GDC17

The Ringelmann effect

**1913, a French agricultural engineer, Maximilian Ringelmann discovered that...**

**"Larger groups lead to less motivation & coordination & productivity"**

Why decreased motivation? In a larger group, it's harder for each person to see their individual impact.

Why decreased coordination? In a larger group, the communication pathways have exploded, as I pointed out earlier.

Again - this is why we need producers!

**What are the suggested solutions?**

If you look this stuff up online, most of Max's 'fixes' are horrible and draconian.

Things like **"increasing surveillance** and **creating internal competition."**

But who wants to make their company a tiny version of 'The Hunger Games?'

But there is suggestion that has worked in my experience:

Increase identifiability.

In a larger team, it's important to ensure everyone knows the meaning of each person's contribution.

People WANT to feel useful and needed, and not like they are just another cog in a machine.

In **Bullfrog**, **Big Blue Box** and **Lionhead** this involved making sure we took time to talk to **everyone**

We'd also have Friday meetings which gave people an opportunity to show off what they were doing.

**Most of all think before growing your team!**

Think really hard about whether it's the right time, and whether you have the right support in place.

And most of all, be nice to producers.



# Blandness Has Its Uses ('02)

- The pursuit of excellence can be wasteful
- Blandness helps disguise repetition
- Communicate the context of your assets



GAME DEVELOPERS CONFERENCE | FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 #GDC17



### Excellence has a cost.

On Fable I once asked an animator for a 'digging animation'.

What I got back was a Pixar-quality 12 second animation of a tired farm laborer..

[Mime the farm labourer]

...resting on his shovel, wiping his brow, pushing the spade into the earth, levering up a chunk of turf, and suffering acute back-pain.

It was beautiful. It was characterful. It was art.

It was **unusable**, because this was for the player's dig animation.

It would be seen 100 times or more during a playthrough.

It was not only far too long, but **far too specific**.

I had to ask for the animation to be made worse. **More bland. Duller**. Just think how counter-intuitive that sounds, for a moment.

If you have good people, they take pride in their craft.

Pride in work leads to excellence... and memorable assets.

But this level of excellence is not always a good thing.

### Blandness helps disguise repetition

There usually isn't enough manpower to make every single asset in your game one-use.

And if you're reusing assets, specificity is bad.

People's brains are designed to spot patterns, in both visuals and audio.

Players notice NPCs talking about: "having taken an arrow to the knee." for the 50th time

Players notice the same boulder engraved with eldritch runes used over and over again.

Players notice these patterns, these specifics, and consider the game worse than a game with a blander asset!

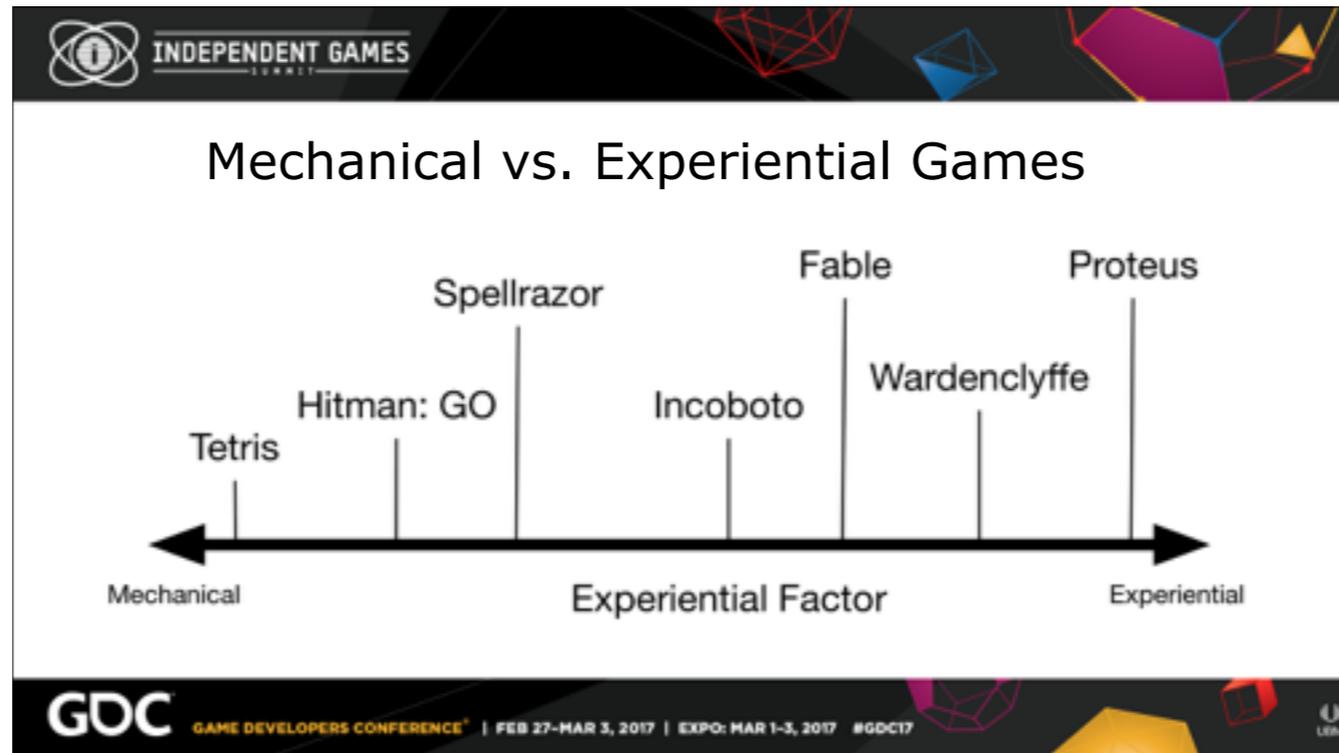
### So it's not enough to ask for content to be made.

Be really clear about how often it will be seen or heard.

Tell content creators how specific content needs to be.

In a lot of cases you'll find it needs to be bland, and anything else will make the game worse.

You might find that your content creators thank you for not wasting their time.



Before the next subject - here's a quick diagram showing the continuum of game types, from Mechanical (Tetris) to Experiential (Proteus).

From rules and systems at one end...  
...to mood, look and feel at the other.



# Prototypes Can Be Misleading ('03)

- Experiential game prototyping is difficult (and maybe polish *is* the game!)
- Fable was a total mess until the very end
- Bland prototypes are better than ugly ones
- Define criteria and reject for the right reasons!



FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 #GDC17

### Experiential game prototyping is difficult

With a mechanics-driven game: a janky prototype is okay. As long as it lets you test your risk/reward rules, you're alright.  
But experiential games are all about look and feel, which means a janky prototype isn't helpful!

In experiential games, sometimes 'polish' IS the game.

### Fable was a lurching, glitchy, joyless experience until the last couple of months.

Despite having lots of game mechanics in it, Fable was very much an **experiential** game.

The music, the mood, the charm...

These things defined Fable, not the raw mechanics of rules and resources.

Not the things you can prototype easily.

But here are the lessons I learned from that experience.

### Bland prototypes are better than ugly ones

There's a weird effect in music production where the louder a track is played, the better the listener thinks the sound quality is.

Quality judgements can be distorted by unrelated factors.

Ugly prototypes lead you to all sorts of weird cognitive biases that utterly undermine their value.

For experiential games, a bland prototype is often better than an ugly one.

If you don't have representative graphics, prototype **without** graphics. Use plain boxes and spheres.

If your game is experiential, **choose bland** over **ugly** every time, or you'll make the wrong decisions.

### Define Criteria and Be Sure You Reject For the Right Reasons

Before you bodge together a prototype, make sure you know:

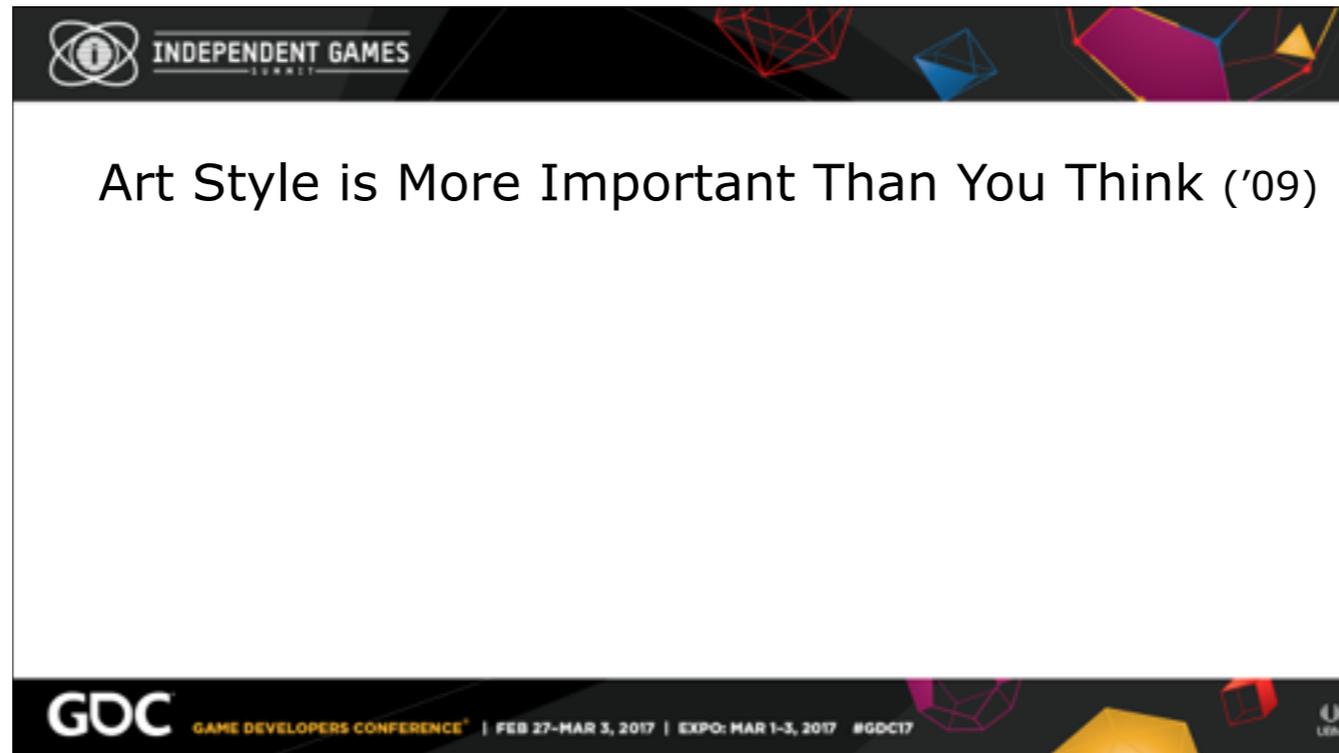
Are you prototyping something experiential?

If so, have you limited the scope of your prototype.

Have you defined the criteria for success?

Is 'success' even achievable at this point in the project?

If not, think again.

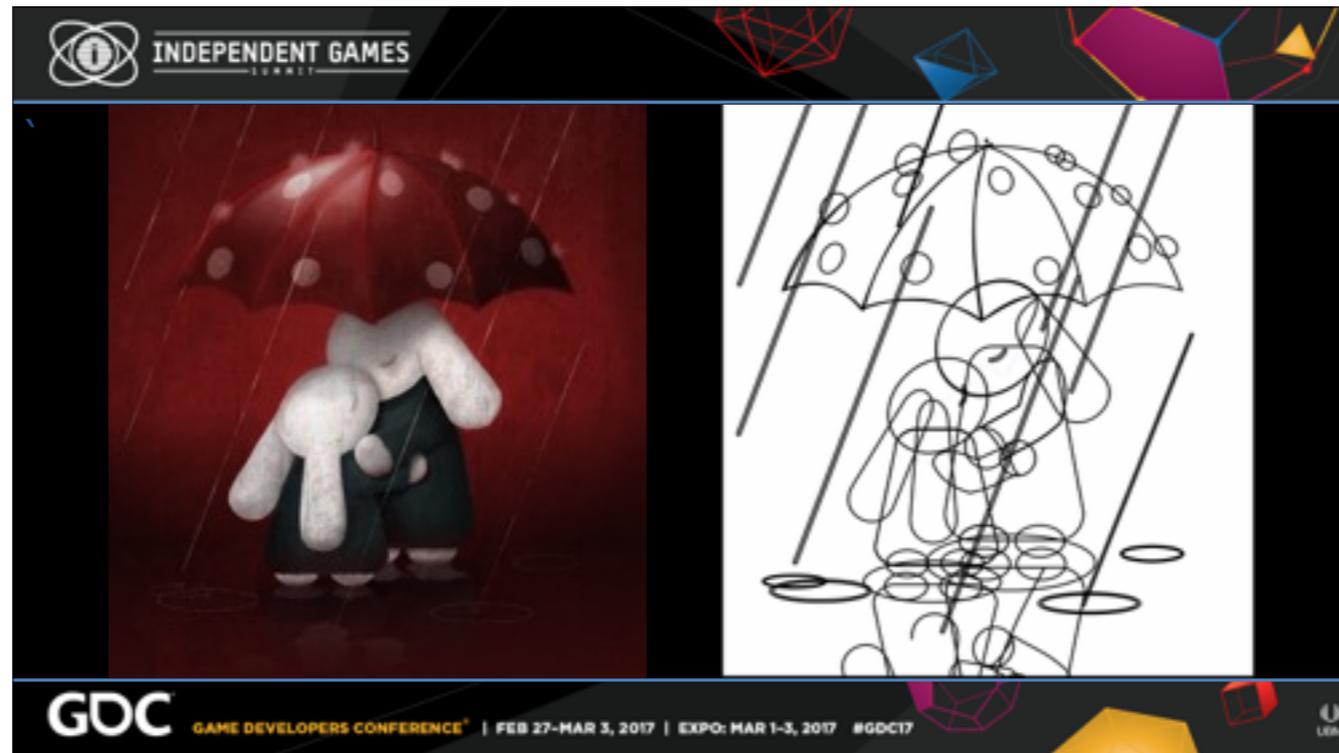


Proper artists here are probably saying things like:

"An art style is what you discover once you've learned to draw properly, you hack!"

True. But that's only if you're going to make art your speciality.

I'm not an artist. I cannot draw. You know how I feel about being a specialist.



But I made this for my wife a few years ago. I used a 2D vector package.

It's a really simple image.

[NEW slide]

Without colours and gradients, it looks like this. You can see how simple it is.

Because I'm not an artist, I needed a strong style to introduce **limitations**.

Limitations made it easier come up with variations **quickly**...

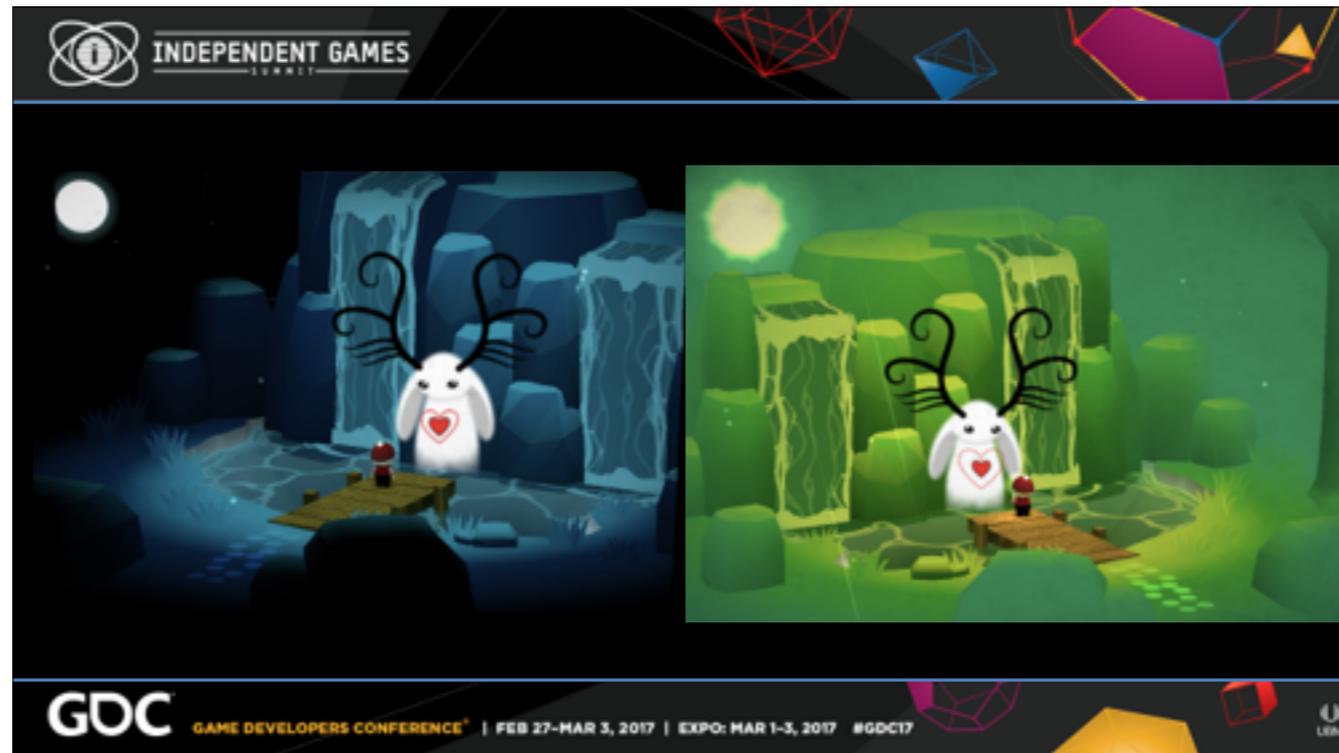
...allowing me to **curate** (easy for me) rather than **create** (hard for me).

Choosing the right style also does something magical...

## Art Style is More Important Than You Think ('09)

- Style can hide the fact that you can't draw!

Style hides the fact that you can't draw!  
This my current project, Wardencllyffe...

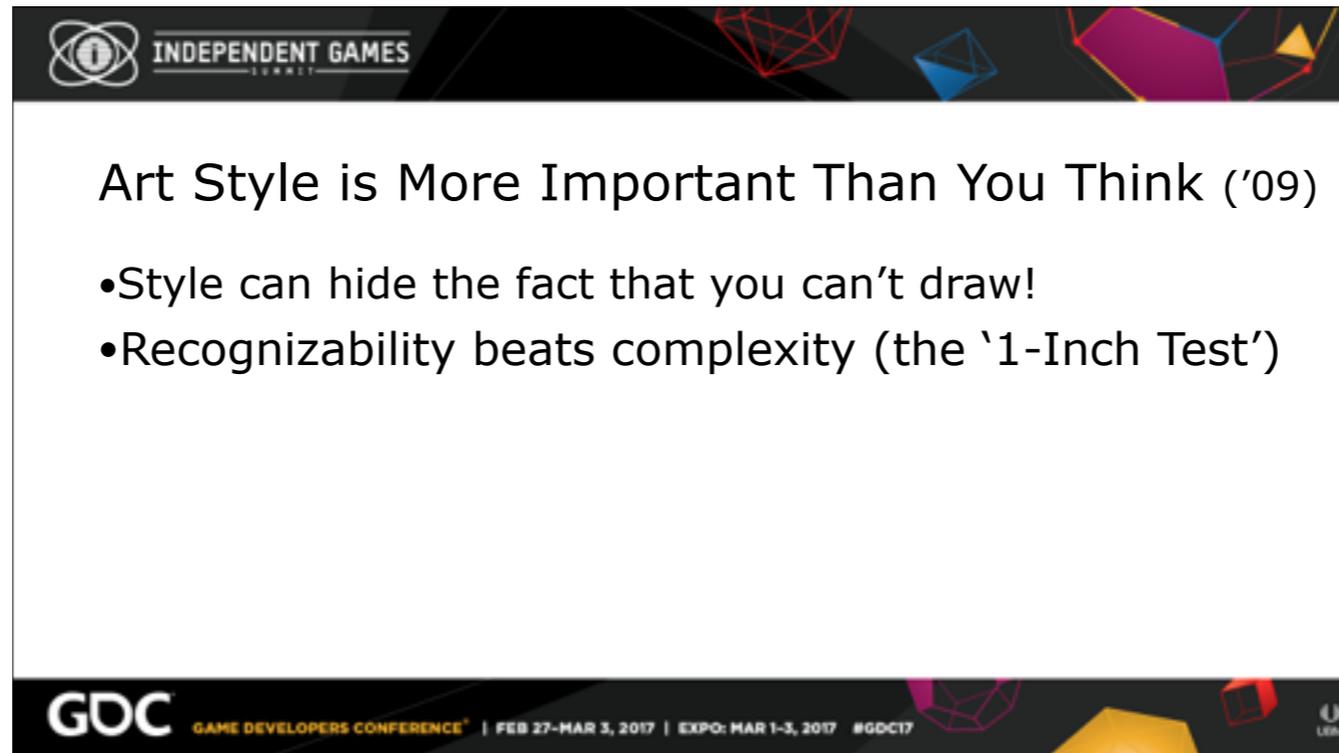


[Wardencllyffe pic]

I'm not an artist. I can't really model in 3D.

So I chose a style that restricted the complexity of my scenes.

Again, it allows me time to curate rather than create.

A presentation slide from the Game Developers Conference (GDC) Independent Games Summit 2017. The slide has a black header with the 'INDEPENDENT GAMES SUMMIT 2017' logo and a decorative border of colorful geometric shapes. The main content is on a white background with black text. The footer is black with the 'GDC GAME DEVELOPERS CONFERENCE' logo, dates 'FEB 27-MAR 3, 2017' and 'EXPO: MAR 1-3, 2017', the hashtag '#GDC17', and the Ustream logo.

INDEPENDENT GAMES  
SUMMIT 2017

## Art Style is More Important Than You Think ('09)

- Style can hide the fact that you can't draw!
- Recognizability beats complexity (the '1-Inch Test')

GDC GAME DEVELOPERS CONFERENCE | FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 | #GDC17 Ustream

**Successful games often have a strong style**

My brother, Simon, and I always used to do the 1-Inch Test when looking at successful games.

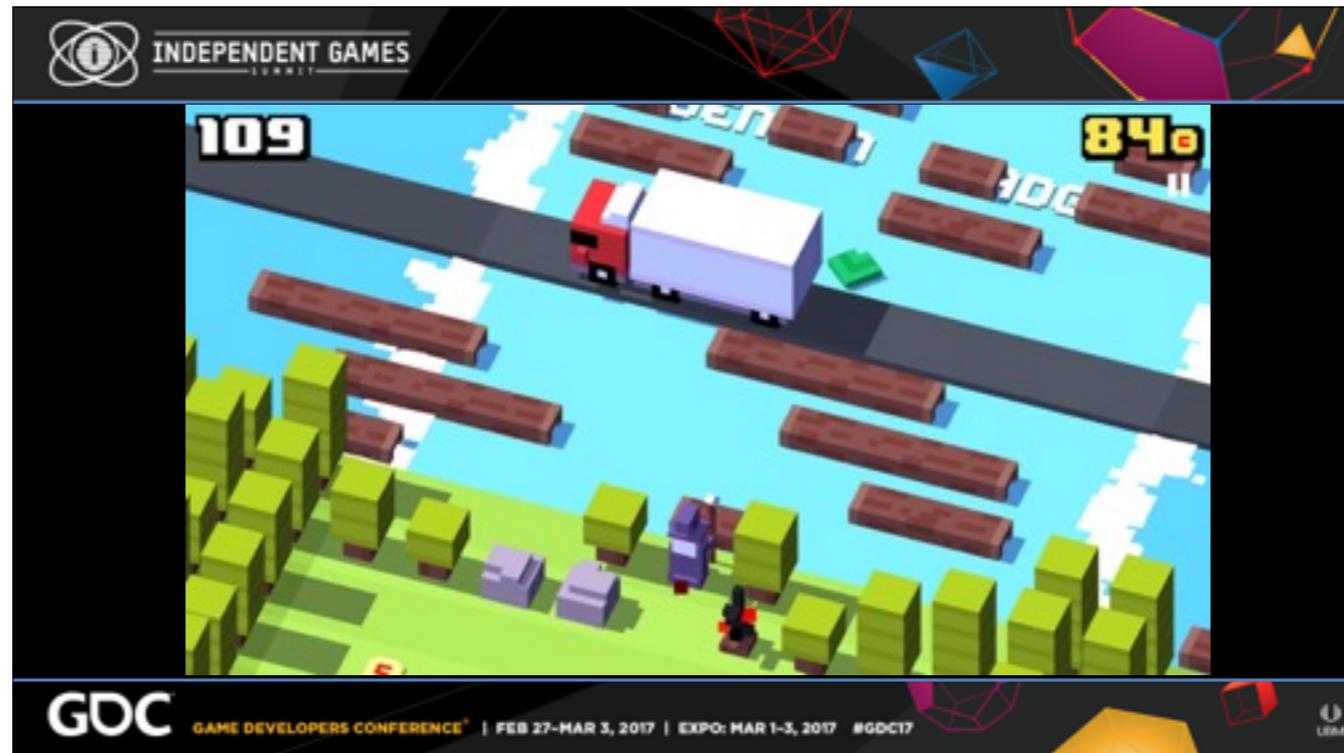
It's pretty simple, so we're going to do it now.

I've got a bunch of screenshots here.

I'm going to show you a tiny amount of each one.

The shots won't show main characters or other obvious clues to their identity.

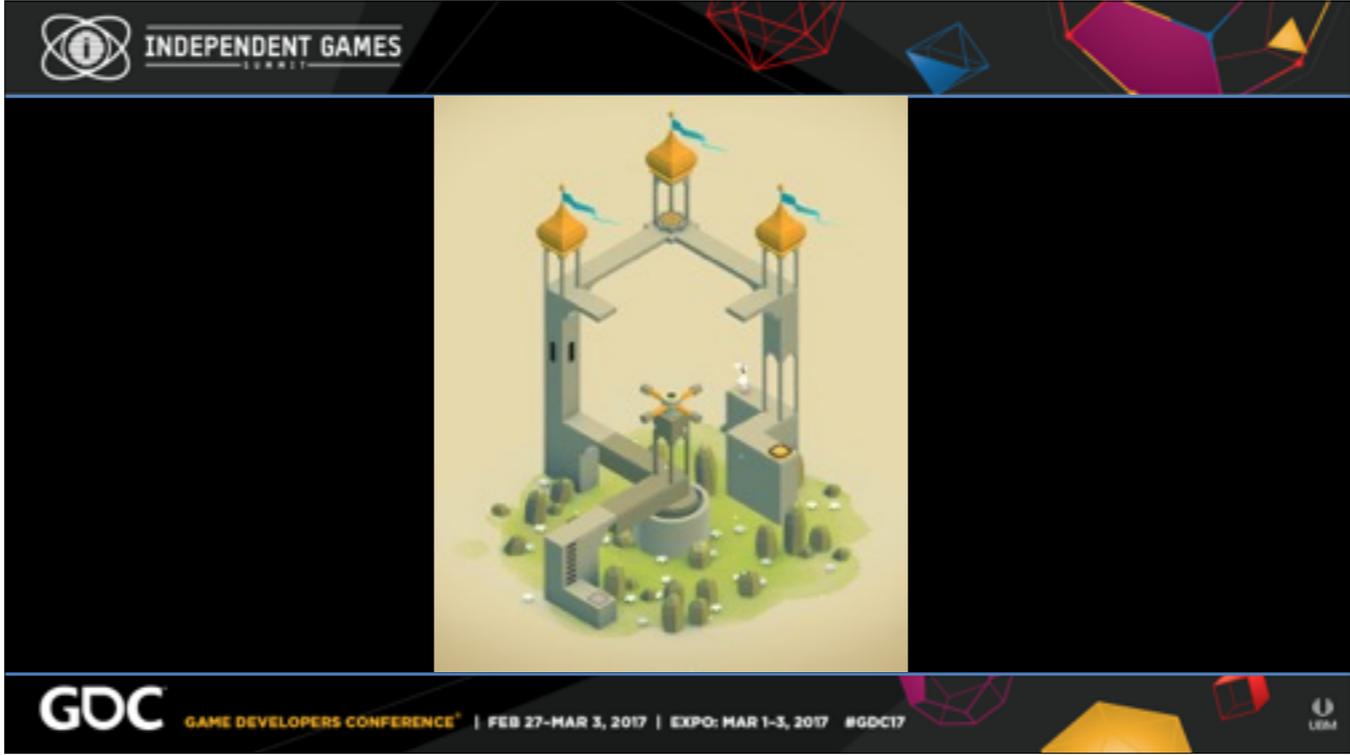
I want you to shout out the name of the game when you've guessed it.



[Wait]  
Crossy Road



[Wait]  
Limbo



[Wait]  
Monument Valley



INDEPENDENT GAMES  
SUMMIT

## Art Style is More Important Than You Think ('09)

- Style can hide the fact that you can't draw!
- Recognizability beats complexity (the '1-Inch Test')
- Style is marketing
- Buy the book 'Pictoplasma!' (not mine, honest!)

GDC GAME DEVELOPERS CONFERENCE | FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 #GDC17



Isn't that amazing? You recognized those games out of thousands, with only minimal information.  
The takeaway here should be that...

**Style is marketing!**

Ultimately, a unique style tells players there might be something unique about your game.  
And having players believe your game is unique is half the battle.

**Buy Pictoplasma!**

I learned so much by looking at street art and character design in this book.

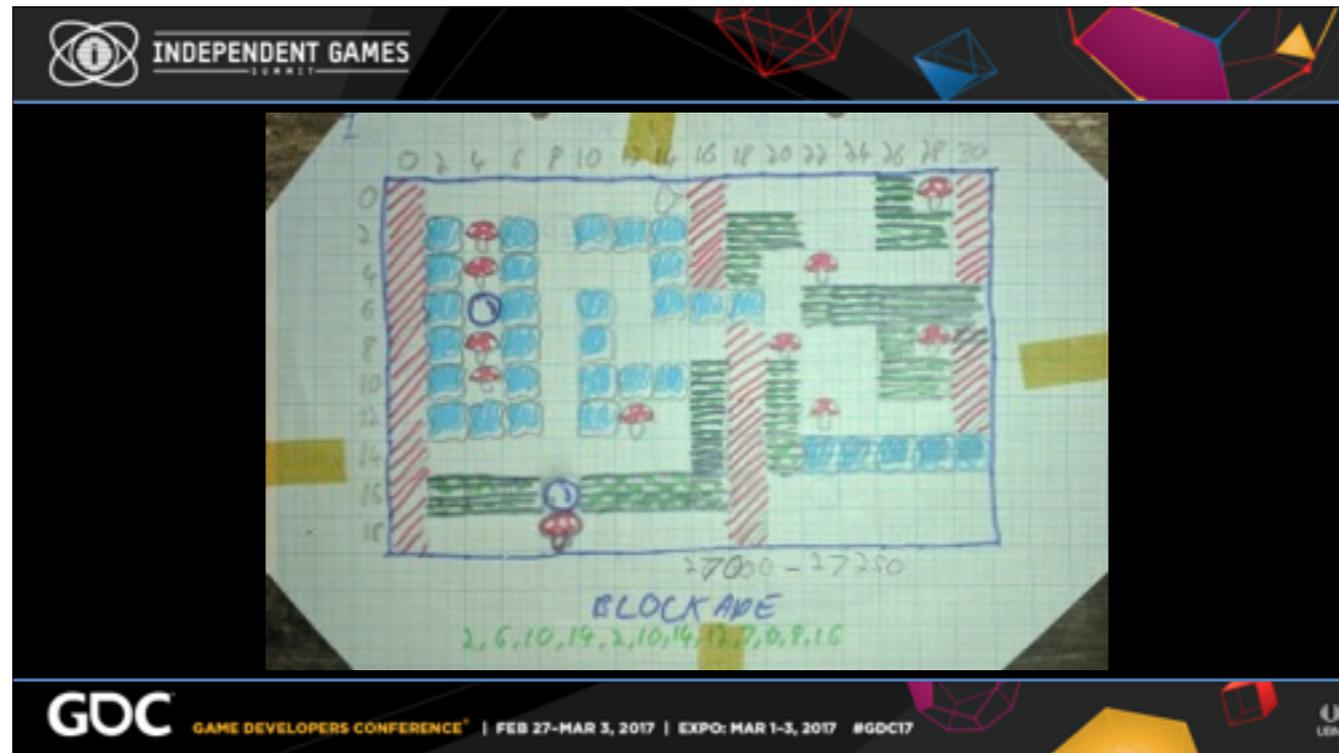
## Streamline Your Process ('84-present)

- What is your iteration cost? No, *really*.

Do you know how much time you waste on iteration gaps?

You know, that weird time period between making a change and evaluating it?

If you think you do... have you been honest?



**Slide 1:**

This is my first editor. It's a piece of graph paper I scribbled on during 'Sport' at school.

Yes, I was that nerd.

After drawing the levels, I hand-coded the various elements into data tables in BASIC. It took about 10 minutes. Oh wait...

**Drawing it on the paper** took 10 minutes. Typing it in took another 5.

And **iterating** took forever.

It was stupid.

Yet for some reason I resisted writing an editor for the entire game.



**While I never repeated that mistake, my brain still resists certain types of work.**

In my current game, Wardencllyffe...

...I found myself walking to a specific area for 10 seconds each time I wanted to test a ritual.

Iterating took too long, so the **sensible** thing to do was to write a small tool allowing me to teleport anywhere I wanted.

BUT getting around to doing it took... a ridiculous amount of time.

## Streamline Your Process ('84-present)

- What is your iteration cost? No, *really*.
- Making your life easier **is** 'real work'! Resist self-sabotage!

**This was because my brain decided that making life easier wasn't 'real' work.**

Because **it didn't feel like progress**. It's a form of self-sabotage.

So, look for similar pointless wastes-of-time in your own process and kill them **even when your brain resists**.



# Closing Thoughts ('84-present)

- Don't let technology dictate your game's core promise
- Static screenshots have to look good, too (no 1 pixel lines!)
- David Lynch: People only see 'doughnuts', not 'holes'...
- ...unless you told people about the holes (or acorns!)
- Don't panic if your current game sucks. Fix it.



GAME DEVELOPERS CONFERENCE | FEB 27-MAR 3, 2017 | EXPO: MAR 1-3, 2017 #GDC17



Okay, I'm going to end this session with a few shorter points.

First up:

**Don't let technology dictate your game's core promise**

My brother and I made this mistake this on Dungeon Keeper.

The navigation system took 1 second to update after any map change... so we altered the game to minimize the amount of map-building which utterly ruined the game.

Peter Molyneux saved us by demanding that navigation was optimized. Immediately.

And that allowed us to make the correct game, the one that lived up to the premise's potential.

So, yes, don't let tech limitations dictate your game's fundamental features.

**Static screenshots have to look good, too (Say no to 1 pixel lines!)**

I learned this one on Spellrazor, a game that features a weird, semi-vector aesthetic with lots of 1 pixel lines.

It turns out that they compress **really** badly.

In addition, while the game is actually really pretty... in motion....

...static shots for Spellrazor don't do it justice. That's a real problem when trying to get people interested in your work!

**Focus on the Doughnut not the hole**

This is something the Director David Lynch once said.

By this he meant that your audience never knows what you had to cut, so don't worry so much about things that end up on the cutting-room floor.

**People only see 'doughnuts', not 'holes'...**

**...unless you told people about the holes... or acorns.**

**Finally:**

**Don't panic if your current game sucks. Fix it.**

Making games is all about making **bad** games.

Being a successful designer means you need to be comfortable with your game sucking... and, of course, discovering the steps to improve it.

If your game is bad right now... that's okay. Don't beat yourself up. The chances are that you're only part way through your journey.

**And that's it. Thanks for coming today, and I'll be in the wrap-up room if anyone wants to chat.**