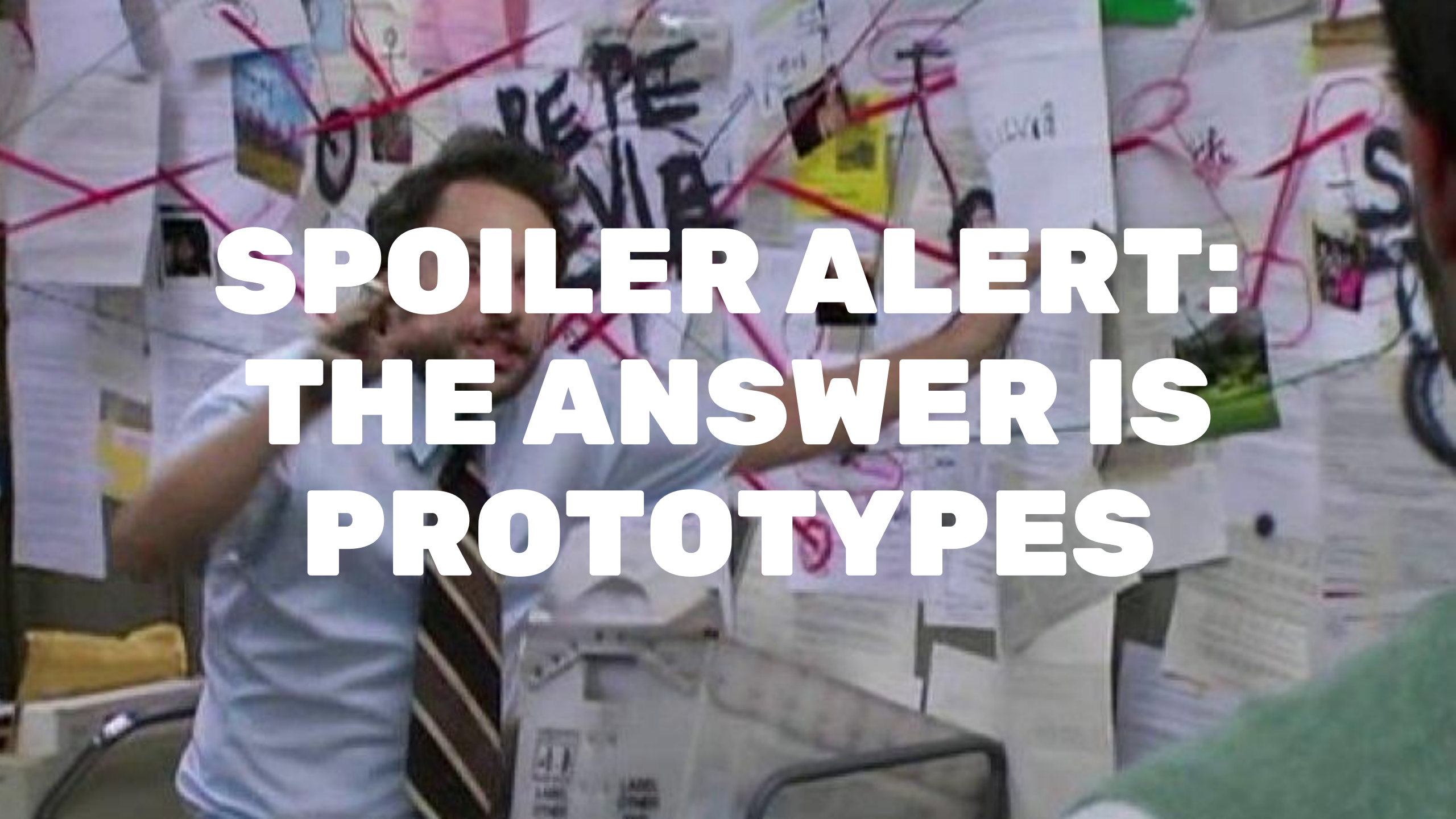
VAGUE BUT EXCITING

Dave Rupert @davatron5000

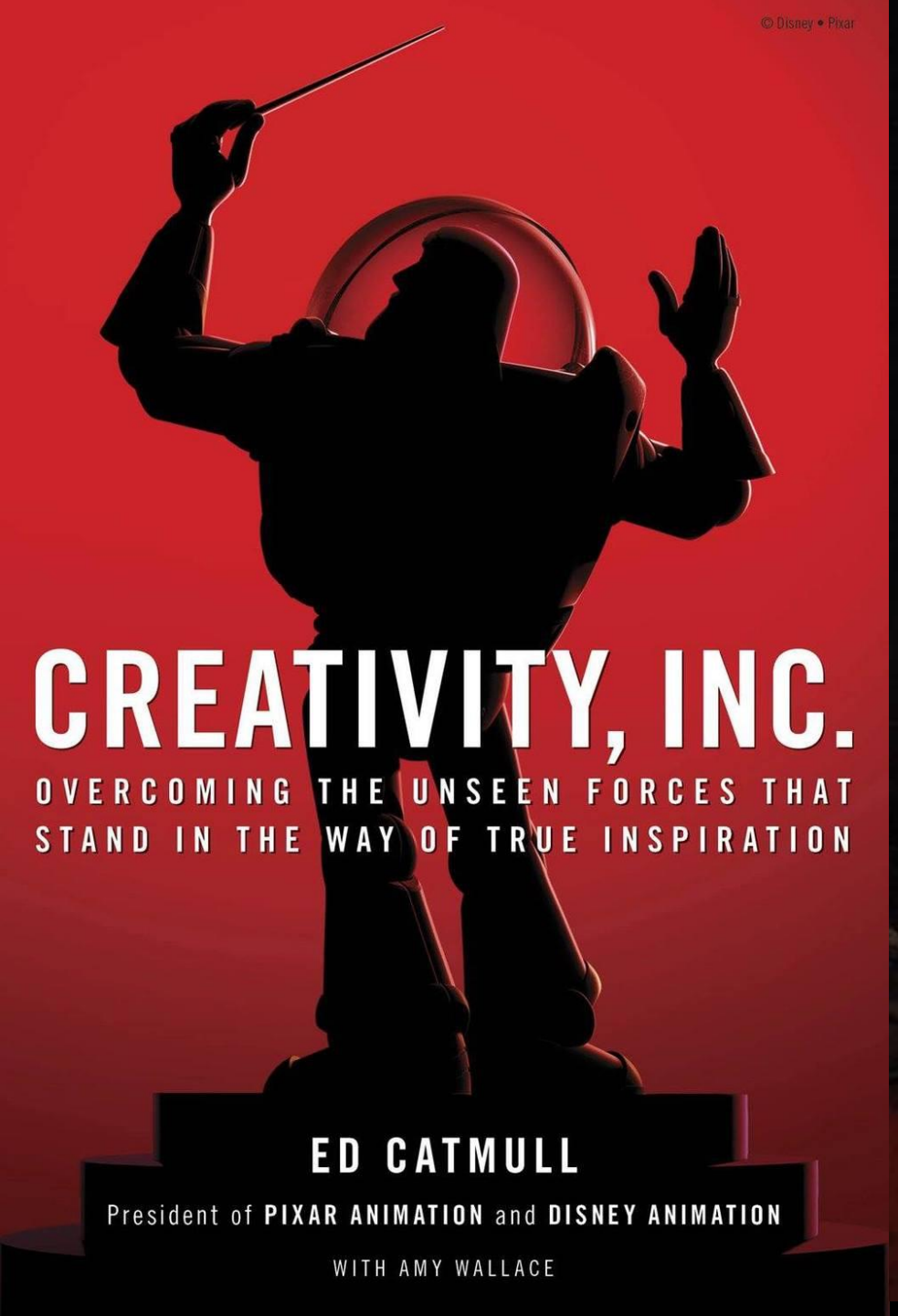

HOW THE F#$% DID
THEY MAKE THIS?


HOW DO LARGE
CREATIVE PROJECTS
SUCCEED?

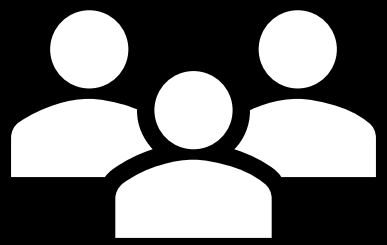

WHY DO LARGE
CREATIVE PROJECTS
FAIL?


SPOILER ALERT:
THE ANSWER IS
PROTOTYPES

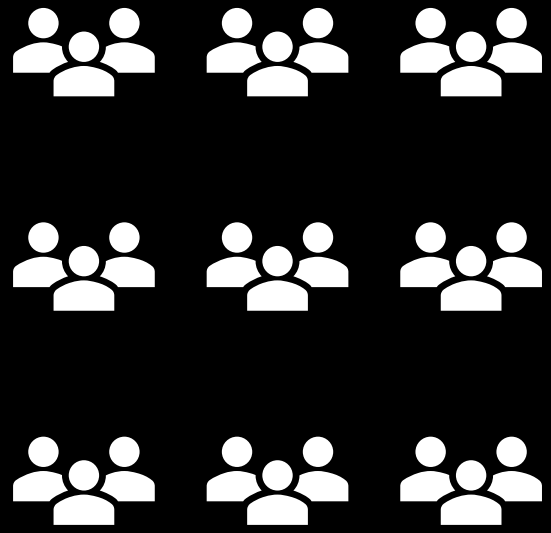

PIXAR


© Disney • Pixar
CREATIVITY, INC.
OVERCOMING THE UNSEEN FORCES THAT STAND IN THE WAY OF TRUE INSPIRATION
ED CATMULL
President of PIXAR ANIMATION and DISNEY ANIMATION
WITH AMY WALLACE

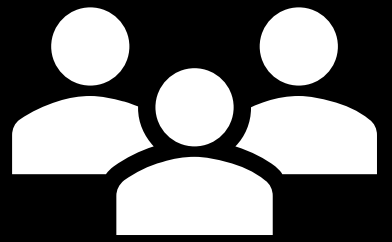
Making a Movie



Plan

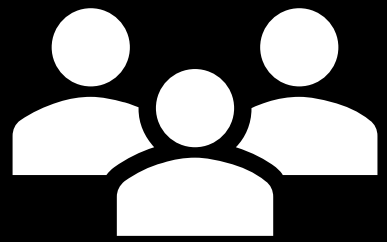


Capture

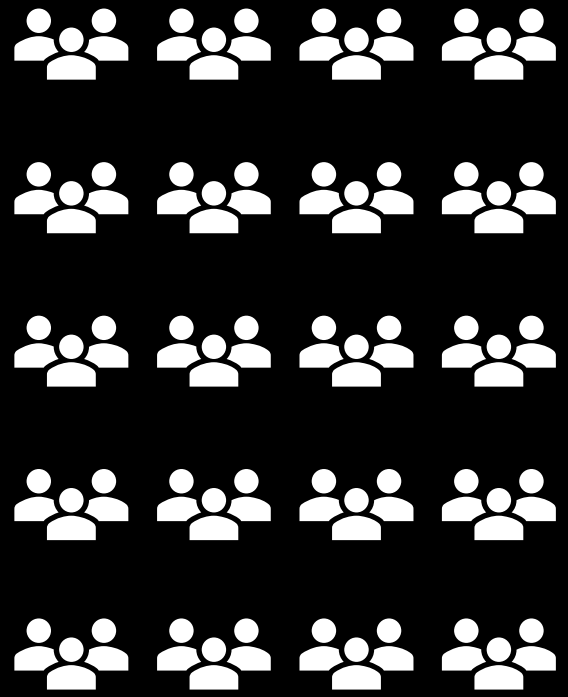


Edit

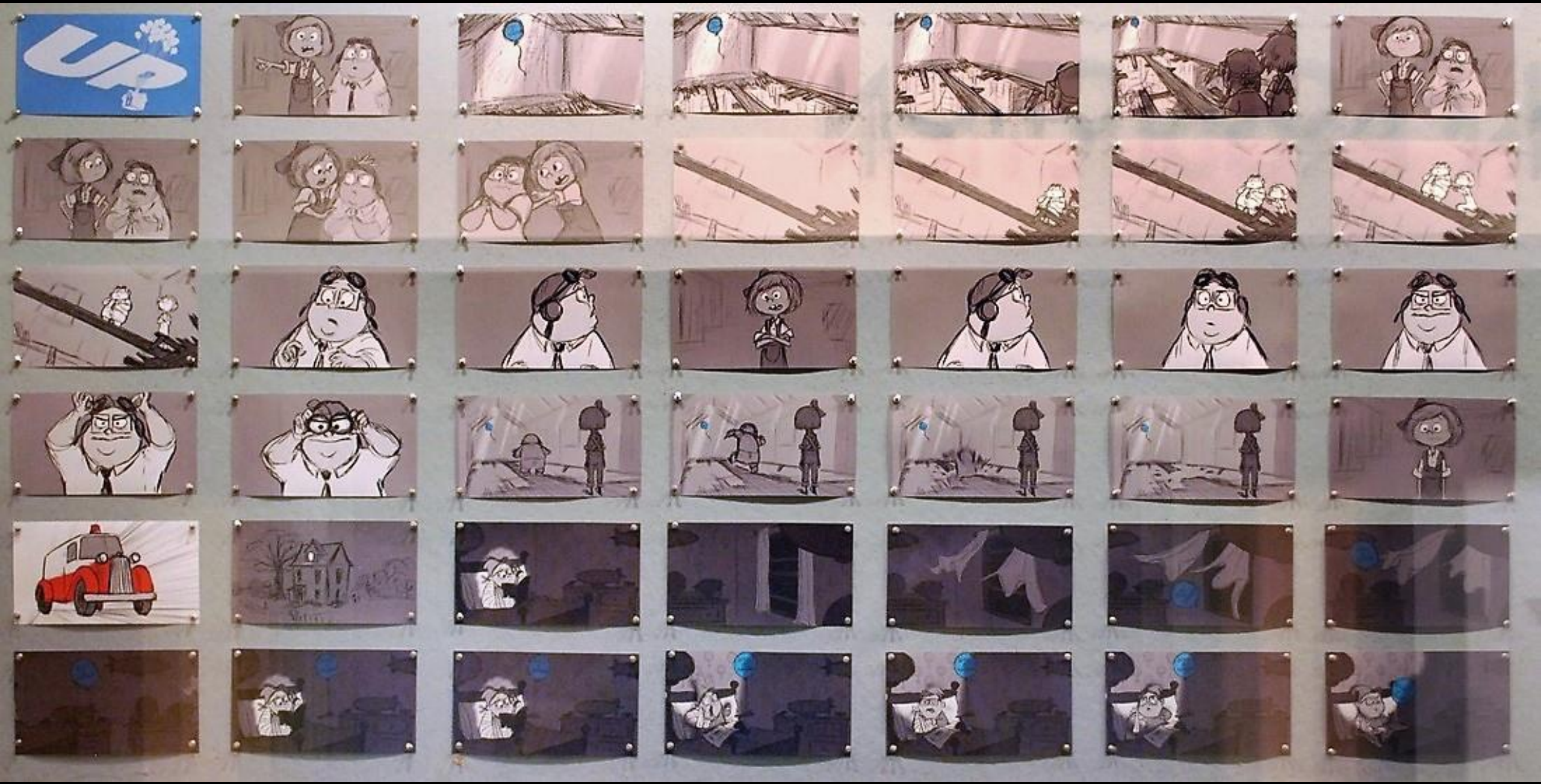
Making an Animated Film

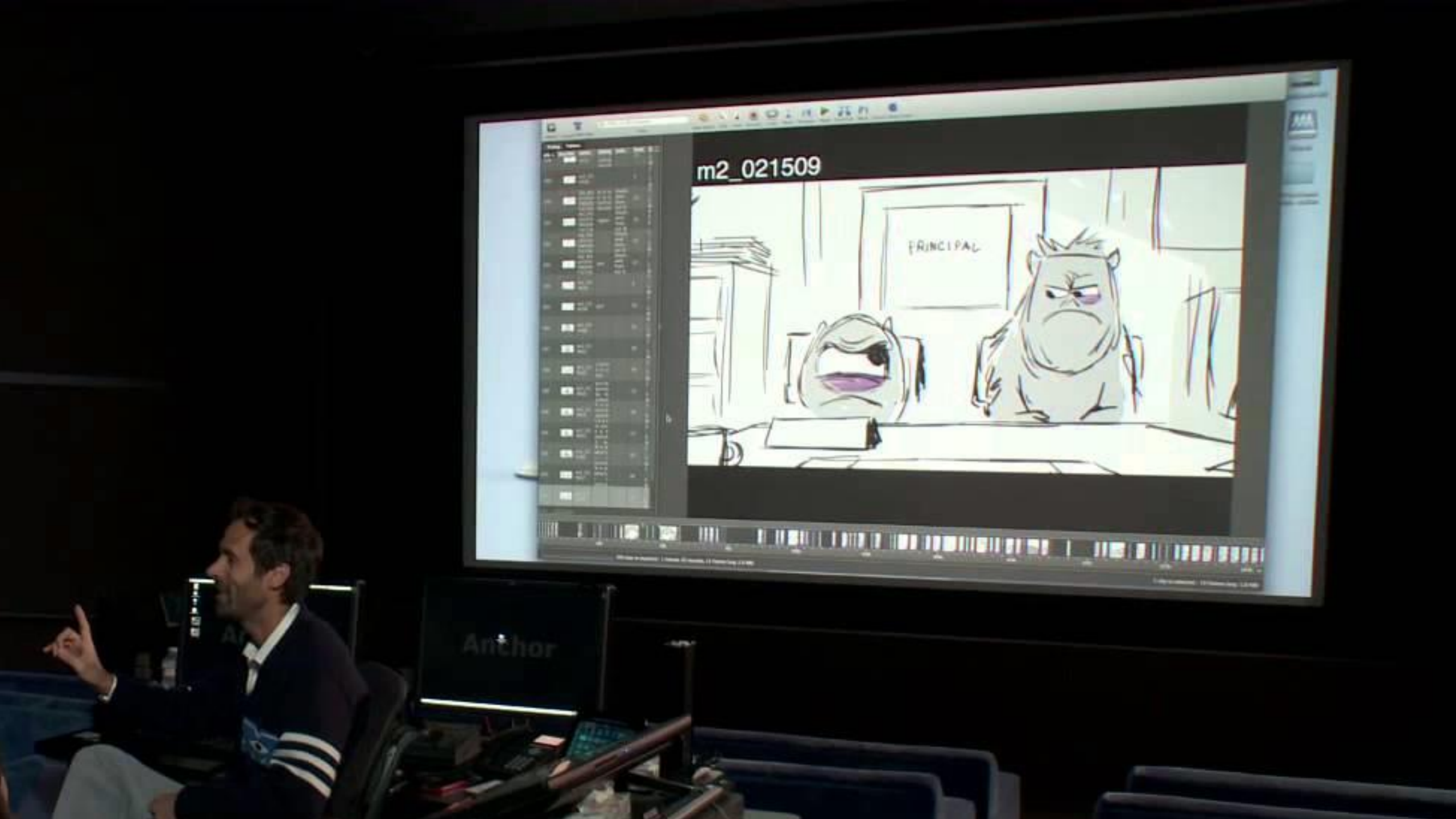


Plan



Animate


UP


m2_021509
PRINCIPAL

Introducing a revolutionary new Design Thinking Thought Technology…

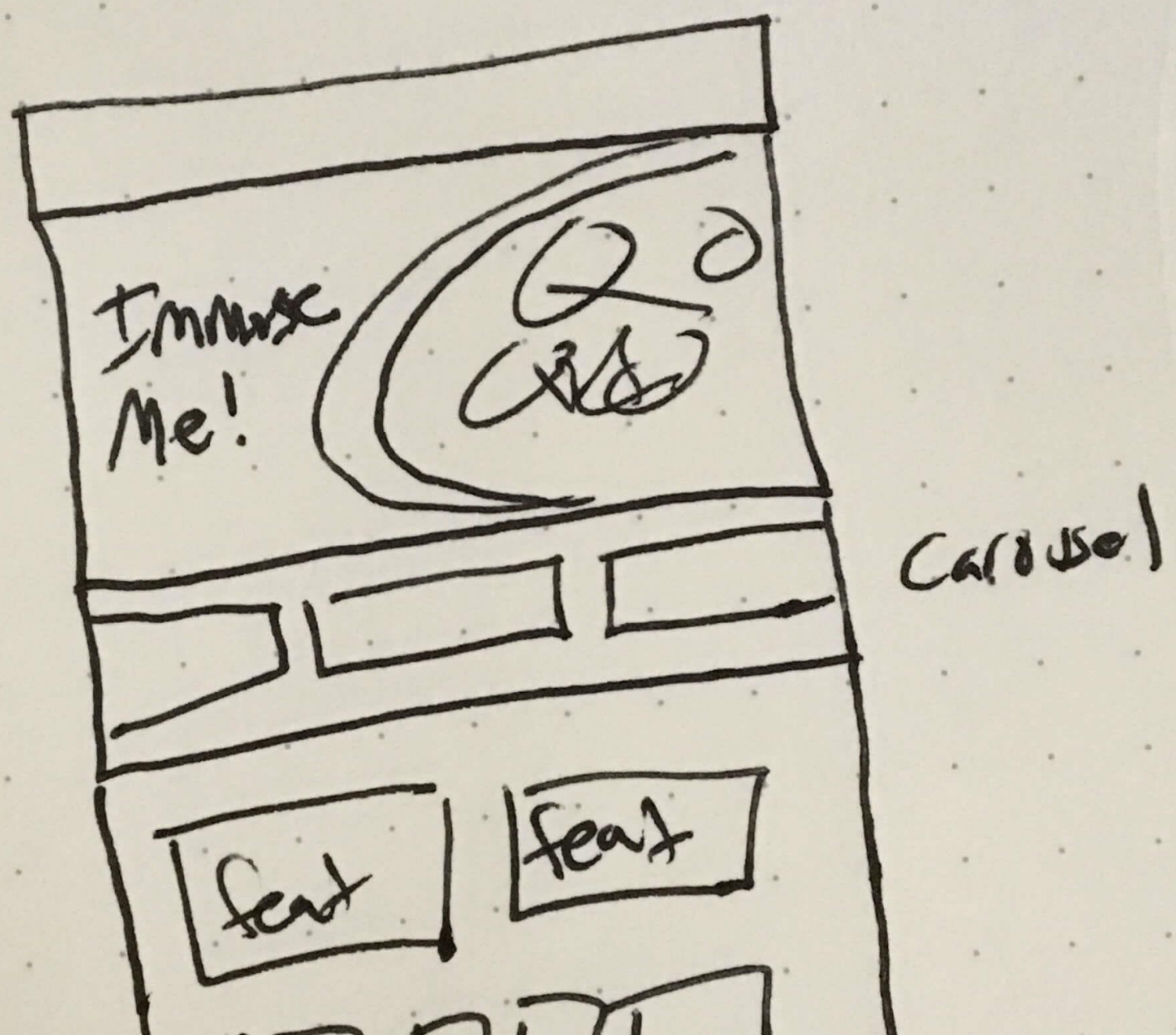
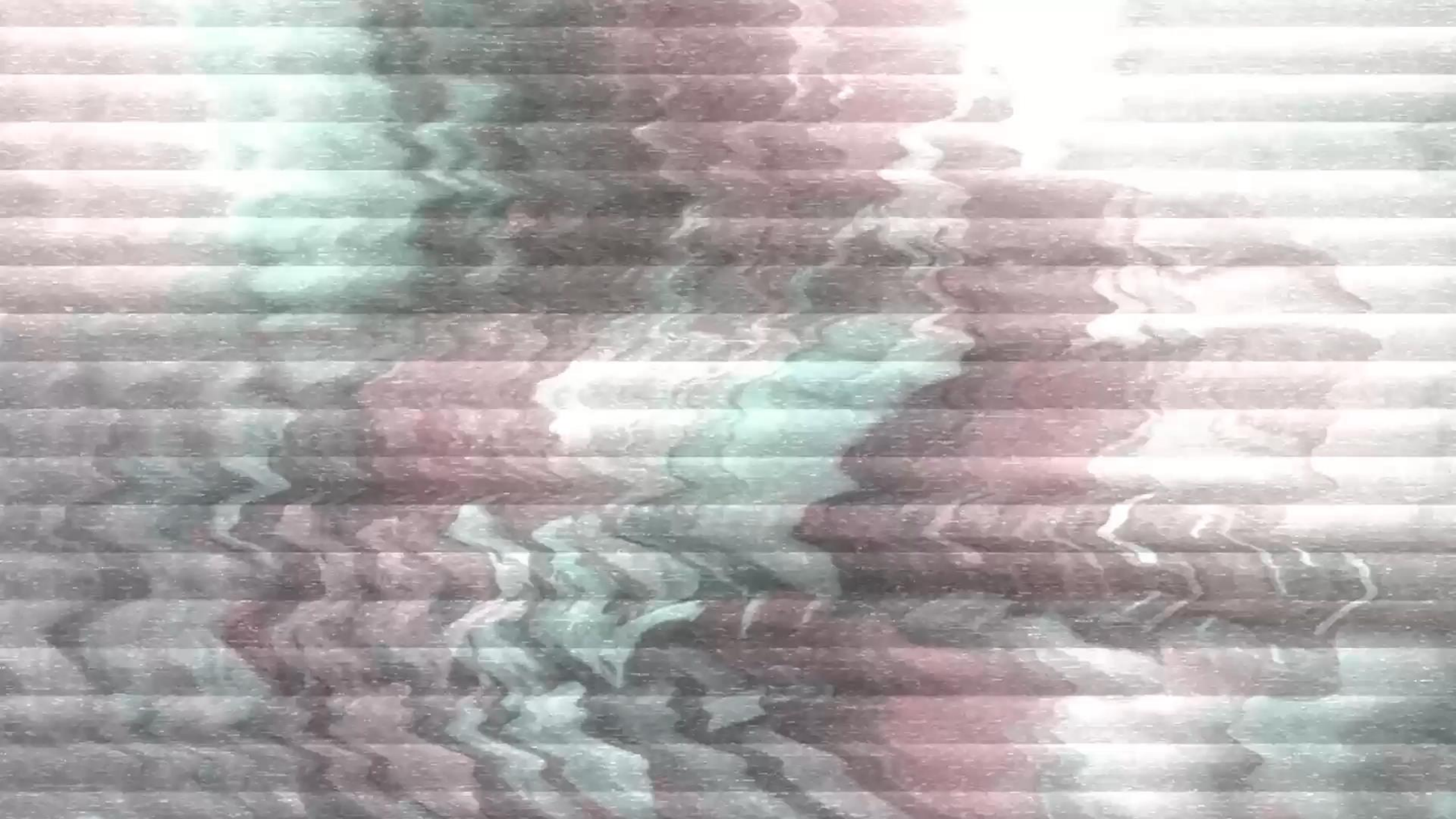


Image
Me!
feat
feat
Carousel

Think through problems before getting to the expensive part.

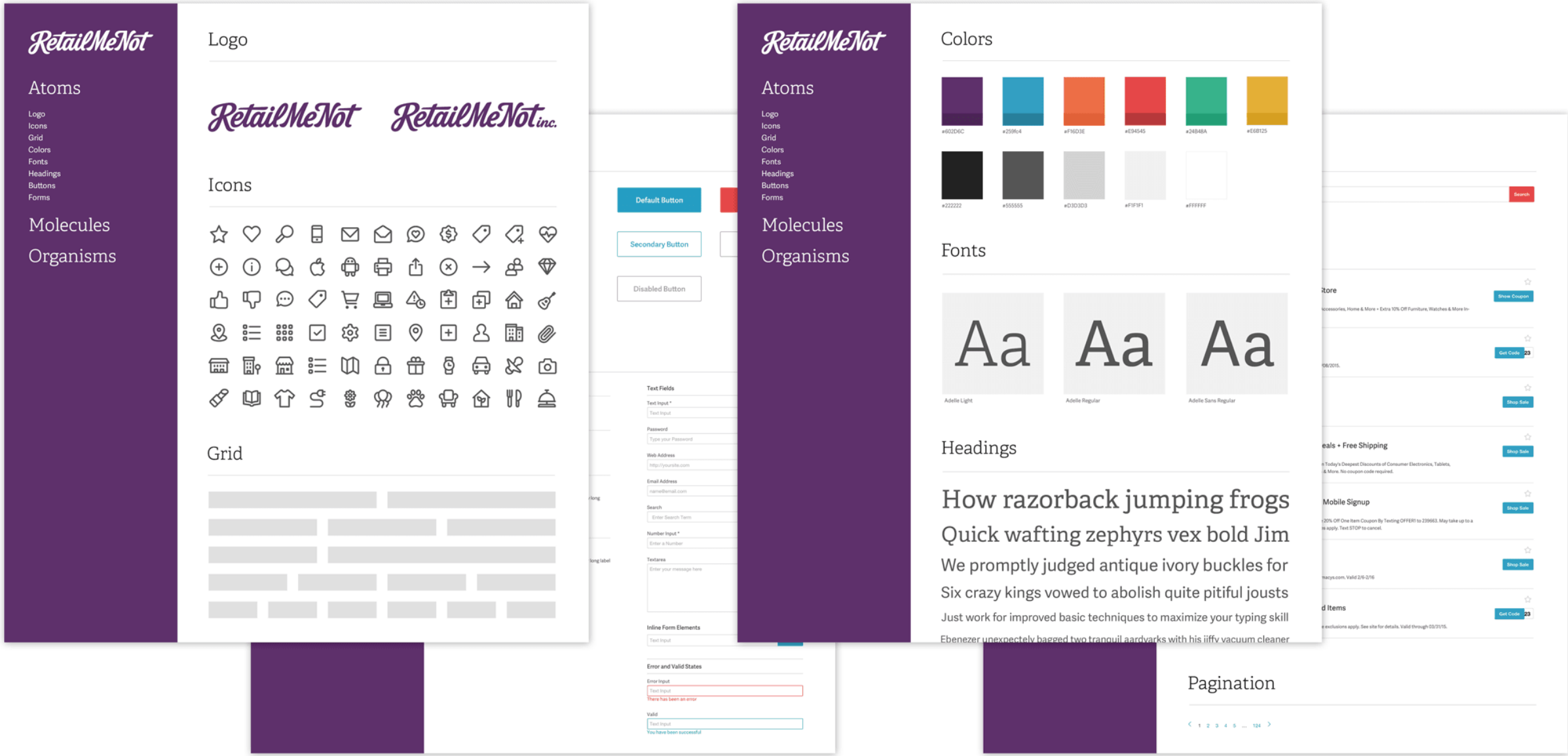
Find (or make) technology that allows you to prototype quickly.


OVERWATCH



A combination of cobbled together parts. But it was instantly fun. And that's how we knew...

– Jeff Kaplan, Game Director of Overwatch


RetailMeNot
Atoms
Logo
Icons
Grid
Colors
Fonts
Headings
Buttons
Forms
Molecules
Organisms
Logo
RetailMeNot
RetailMeNot inc.
Icons
Grid
Default Button
Secondary Button
Disabled Button
Text Fields
Text Input *
Password
Web Address
Email Address
Search
Number Input *
Textarea
Inline Form Elements
Error and Valid States
Error Input
There has been an error
Valid
You have been successful
RetailMeNot
Atoms
Logo
Icons
Grid
Colors
Fonts
Headings
Buttons
Forms
Molecules
Organisms
Colors
#602D6C
#259fc4
#F16D3E
#E94545
#24B48A
#E6B125
#222222
#555555
#D3D3D3
#F1F1F1
#FFFFFF
Fonts
Aa
Aa
Aa
Adelle Light
Adelle Regular
Adelle Sans Regular
Headings
How razorback jumping frogs
Quick wafting zephyrs vex bold Jim
We promptly judged antique ivory buckles for
Six crazy kings vowed to abolish quite pitiful jousts
Just work for improved basic techniques to maximize your typing skill
Ebenezer unexpectely bagged two tranquil aardvarks with his iiffy vacuum cleaner
Search
Show Coupon
Get Code
Shop Sale
Mobile Signup
Get Code
Pagination

Quickly create Minimum Viable Products.

Quickly create

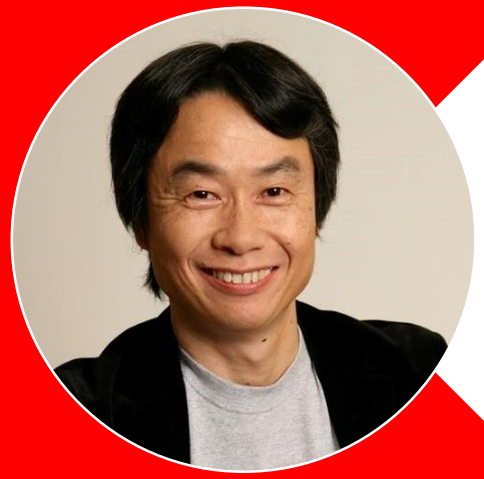
~~Minimum Viable Products.~~

“Find out if it’s a dumb idea as soon as possible.”

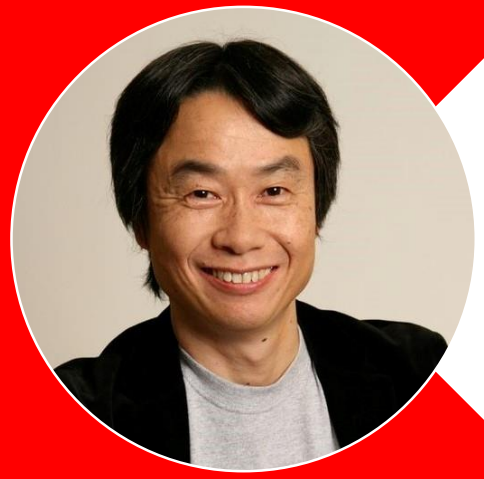
- Dave Rupert, Startup Guru


Nintendo®


SUPER
MARIO
64



There was a room made of simple Lego-like blocks, and Mario and Luigi could run around in there, climb slopes, jump around, etc. We were trying to get the controls right with an analogue 3D stick, and once that felt smooth, we knew we were halfway there.



As for the courses and enemies, those actually came at the very end. They were done in a single burst of energy, just thrown together, almost.

We get the fundamentals solid first, then do as much with that core concept as our time and ambition will allow.

– Shigeru Miyamoto

「手応え」

Hand Response or “Game Feel”


Splatoon
™


Player1
00:00
ゲームセット
計測中。。。
黒：38.63% vs 白：39.58%

The way that Miyamoto-san [makes] games, the idea comes not from the design, but from the function. The design comes after.

– Satoru Iwata

Prove ideas before committing to them.

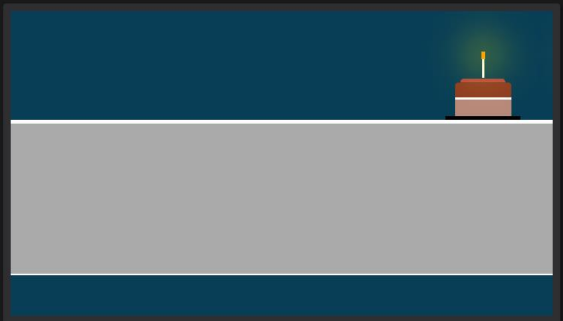
CodePen

https://codepen.io/#

CODEPEN Spark Jobs Blog Create

Picked Pens

VIEW MORE PENS →

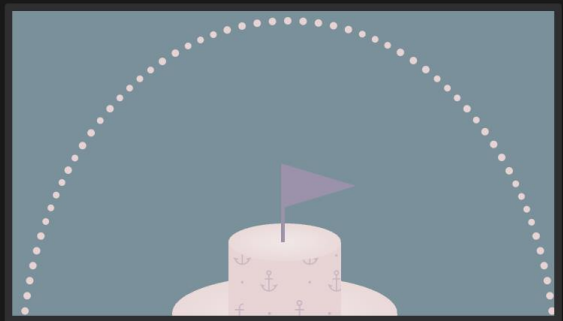


Codevember #2 (lots of) Cake - HTML, SCSS

Kye Buffery 48 0

#Codevember Day 2: Cake

Katherine Kato 129 0 5

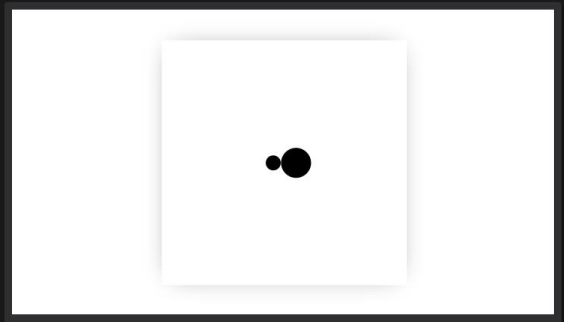


Cake - Codevember Day 2

Kristopher Van Sant PRO 156 0 3

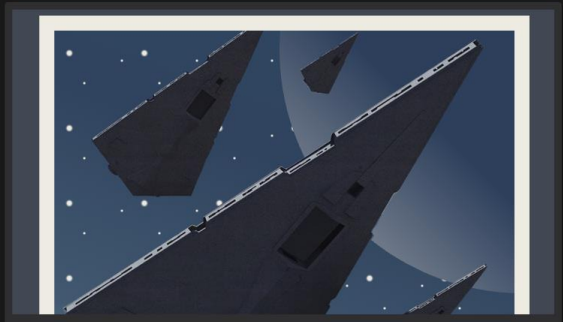
Codevember 2: Cake

Nathan Magyar 198 0 7



Conservation of 1D momentum

Derek Morash PRO 322 0 6



Galactic Empire Propaganda - Codevember Day 1

Kristopher Van Sant PRO 461 0 9

Next ›

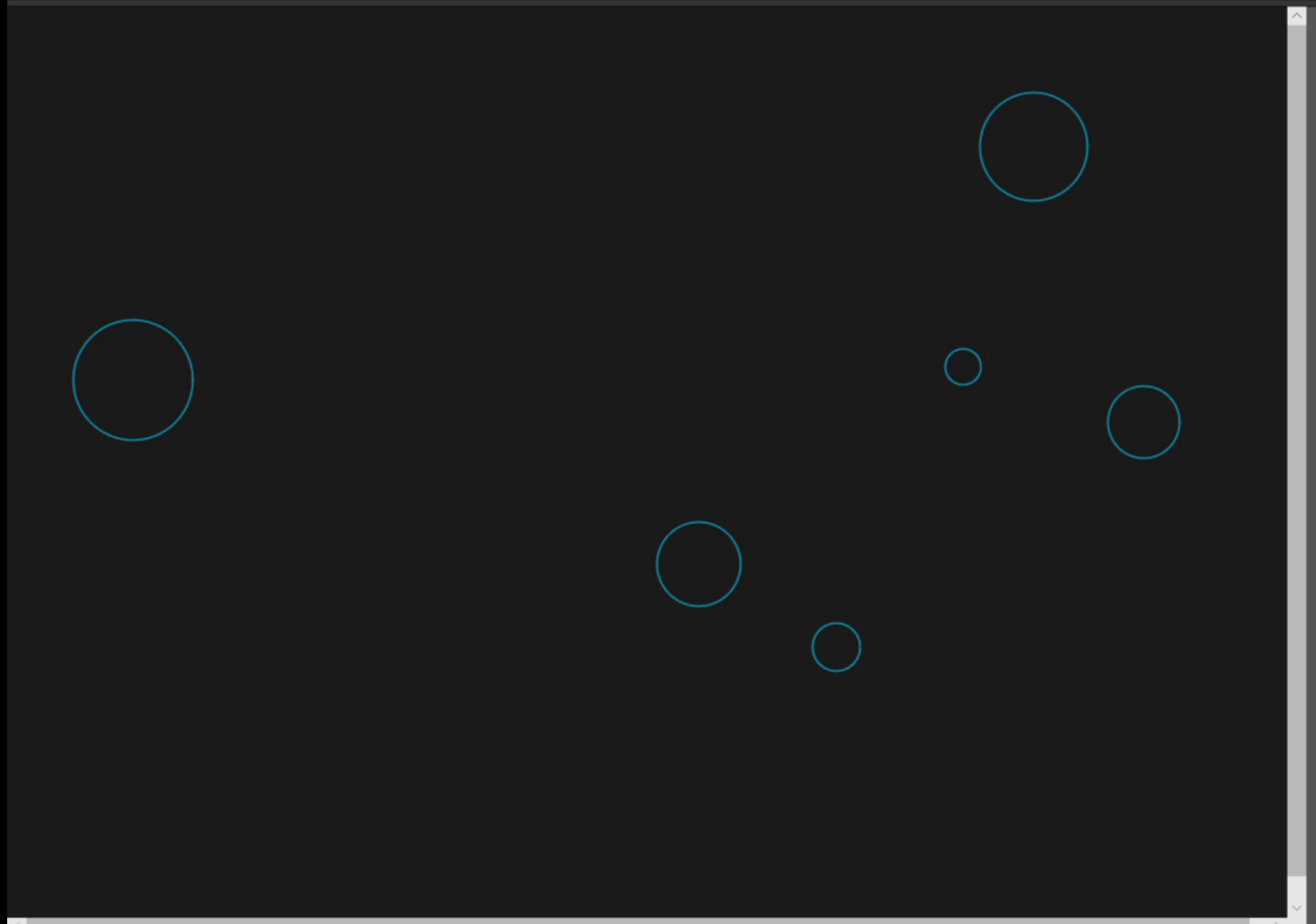
Global Defense v1: Mak

https://codepen.io/davatron5000/pen/9a857c7e89dfb37cfb508a865a73f6eb/

Global Defense v1: Make asteroids fall

A PEN BY Dave Rupert PRO

Save | Fork | Settings | Change View



HTML

```
<canvas id="canvas"></canvas>
```

CSS (SCSS)

```
html,
body {
  min-height: 100%;
}
body {
  padding: 0;
  margin: 0;
}

button {
  position: fixed;
```

JS

```
var canvas = document.getElementById('canvas');

var ctx = canvas.getContext('2d');

var bullets = [];
var asteroids = [];
var start = null;
var asteroidSizeModifier = 5;

var step = function( timestamp ) {
  if (!start) start = timestamp;
```

Console | Assets | Comments | Shortcuts

Last saved over 1 year ago | Delete | Share | Export | Embed | Collections

Global Defense v2: Make bullets shoot

A PEN BY Dave Rupert PRO

Save | Fork | Settings | Change View

HTML

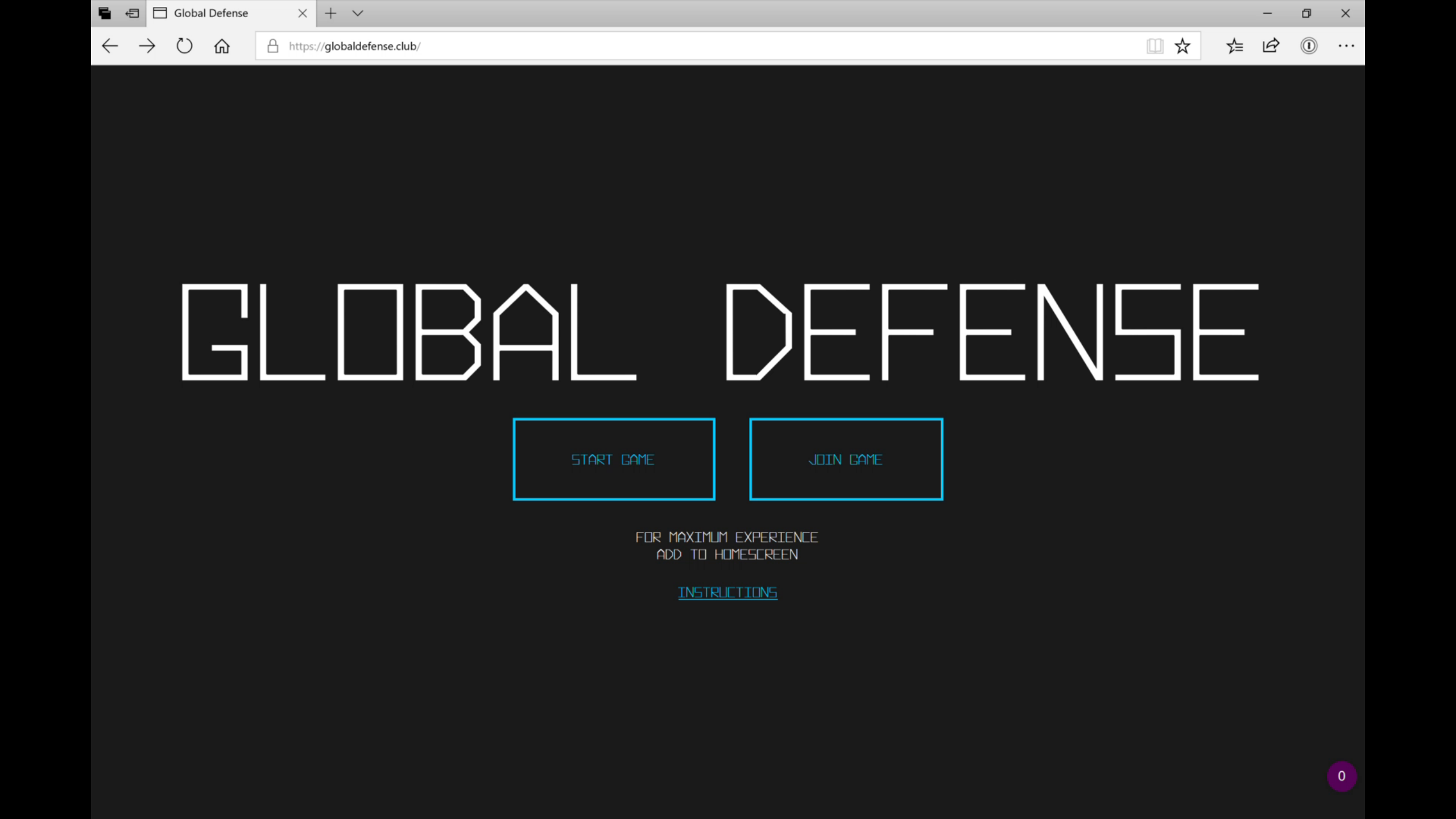
```
<canvas id="canvas"></canvas>
<button>Fire</button>
```

CSS (SCSS)

```
html,
body {
  min-height: 100%;
}
body {
  padding: 0;
  margin: 0;
}

button {
  position: fixed;
```

JS

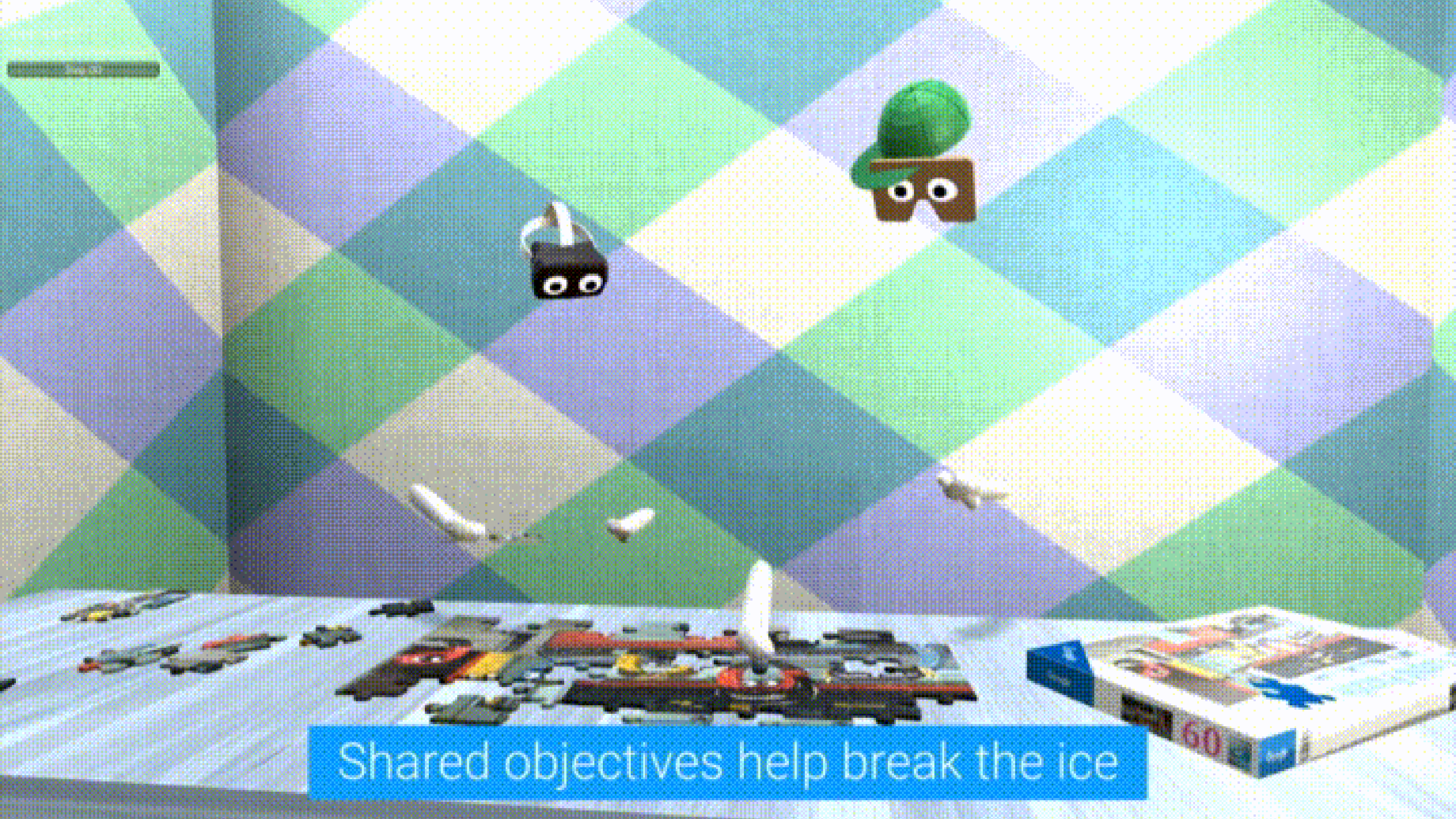
```
var canvas = document.getElementById('canvas');
var button = document.querySelector('button');

var ctx = canvas.getContext('2d');

var bullets = [];
var asteroids = [];
var start = null;
var deltaTime = null;
var asteroidSizeModifier =
```


Global Defense
https://globaldefense.club/
GLOBAL DEFENSE
START GAME
JOIN GAME
FOR MAXIMUM EXPERIENCE
ADD TO HOMESCREEN
INSTRUCTIONS
0


Daydream

60 PROTOTYPES IN 30 WEEKS

MON	TUE	WED	THU	FRI
				


Shared objectives help break the ice


Full body avatars are hard to get right

"80% IS BETTER THAN 90%"

You build the tip of the iceberg and people will come to you and describe the rest.

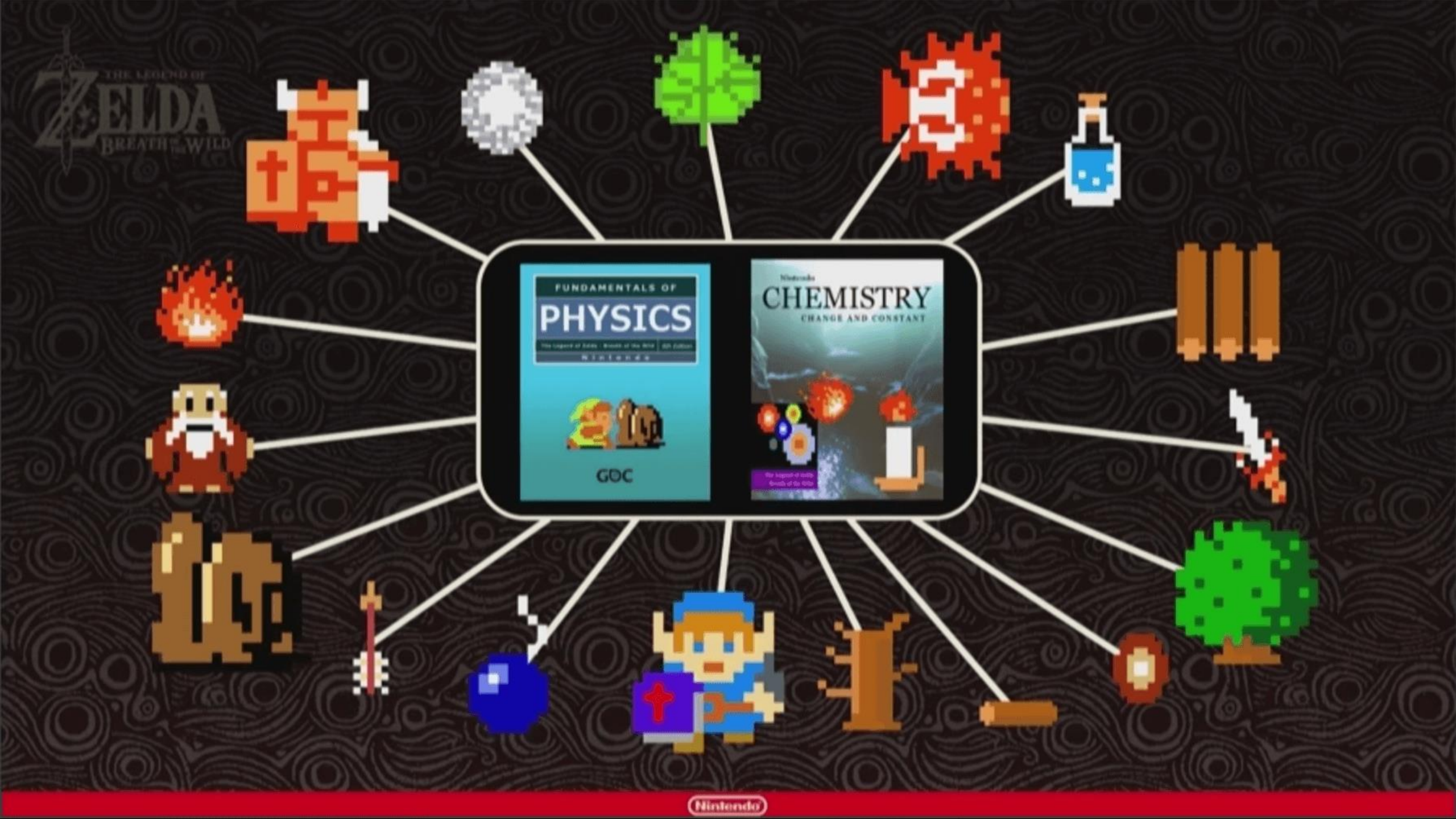
– Manuel Clément, Google

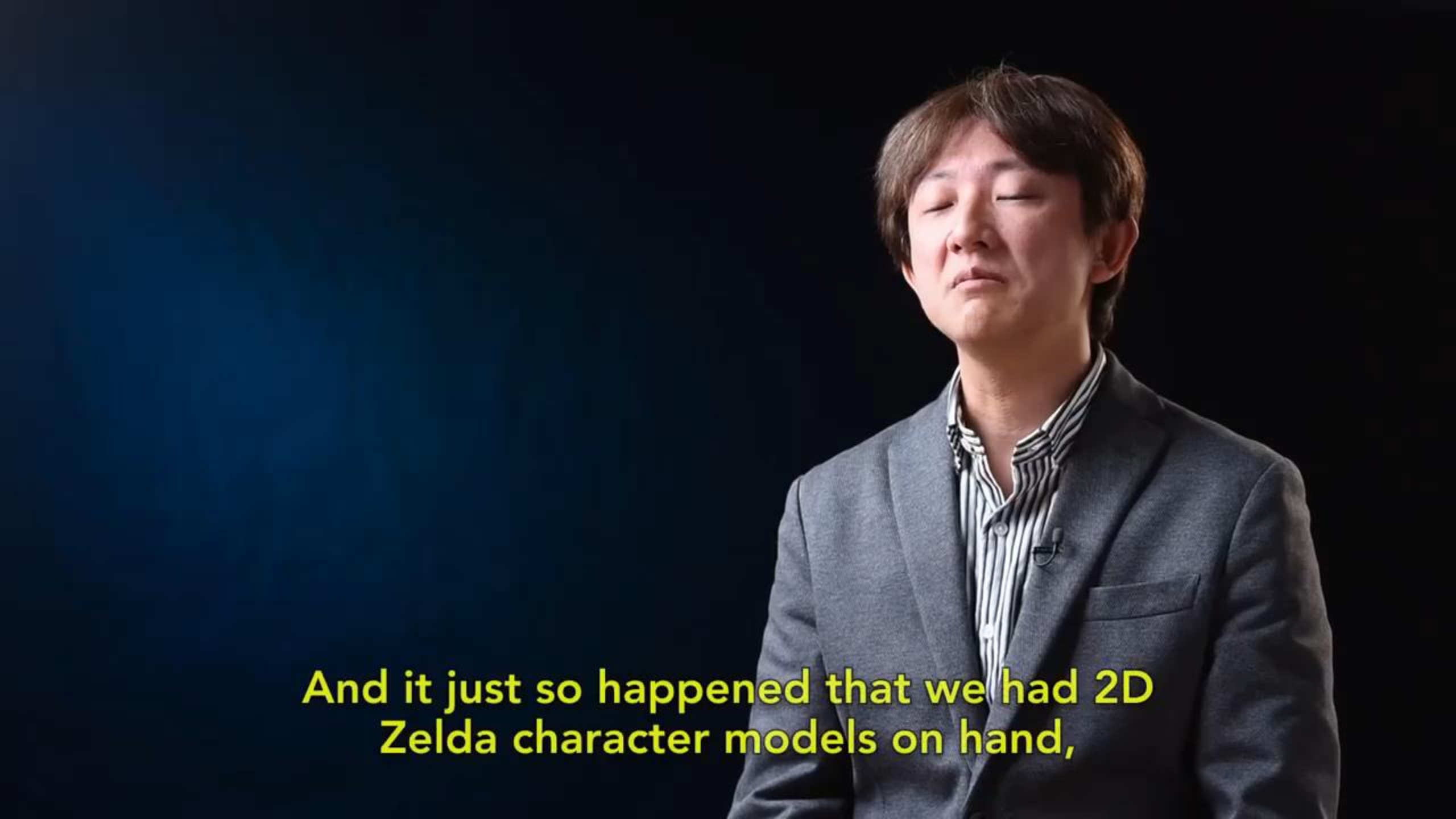
Bring people in to test.

A short timeframe forces quick thinking.

Less polished gets better feedback.


Prototype


THE LEGEND OF ZELDA
BREATH OF THE WILD
FUNDAMENTALS OF
PHYSICS
Nintendo
GDC
Nintendo
CHEMISTRY
CHANGE AND CONSTANT
Nintendo

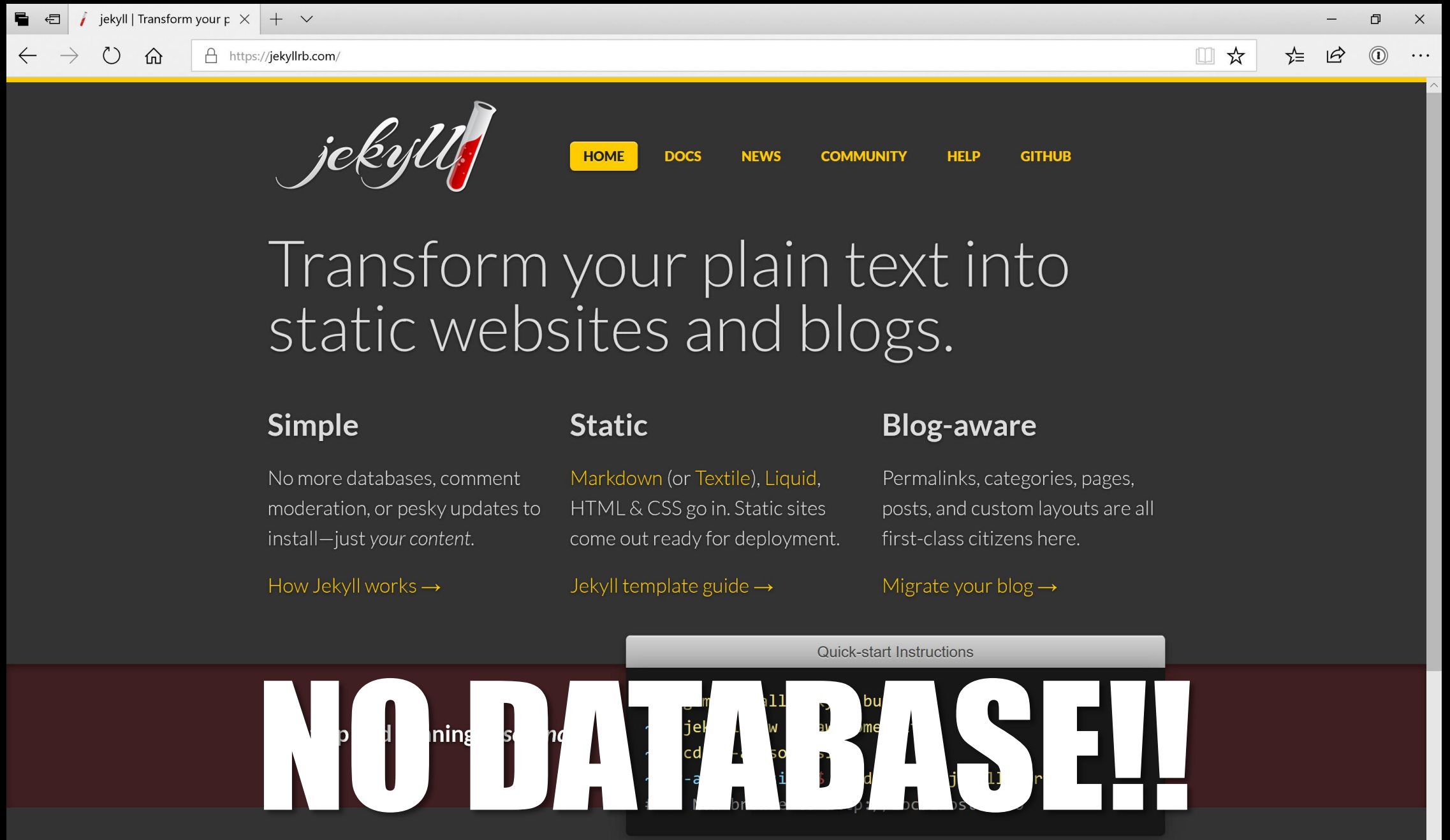

And it just so happened that we had 2D
Zelda character models on hand,

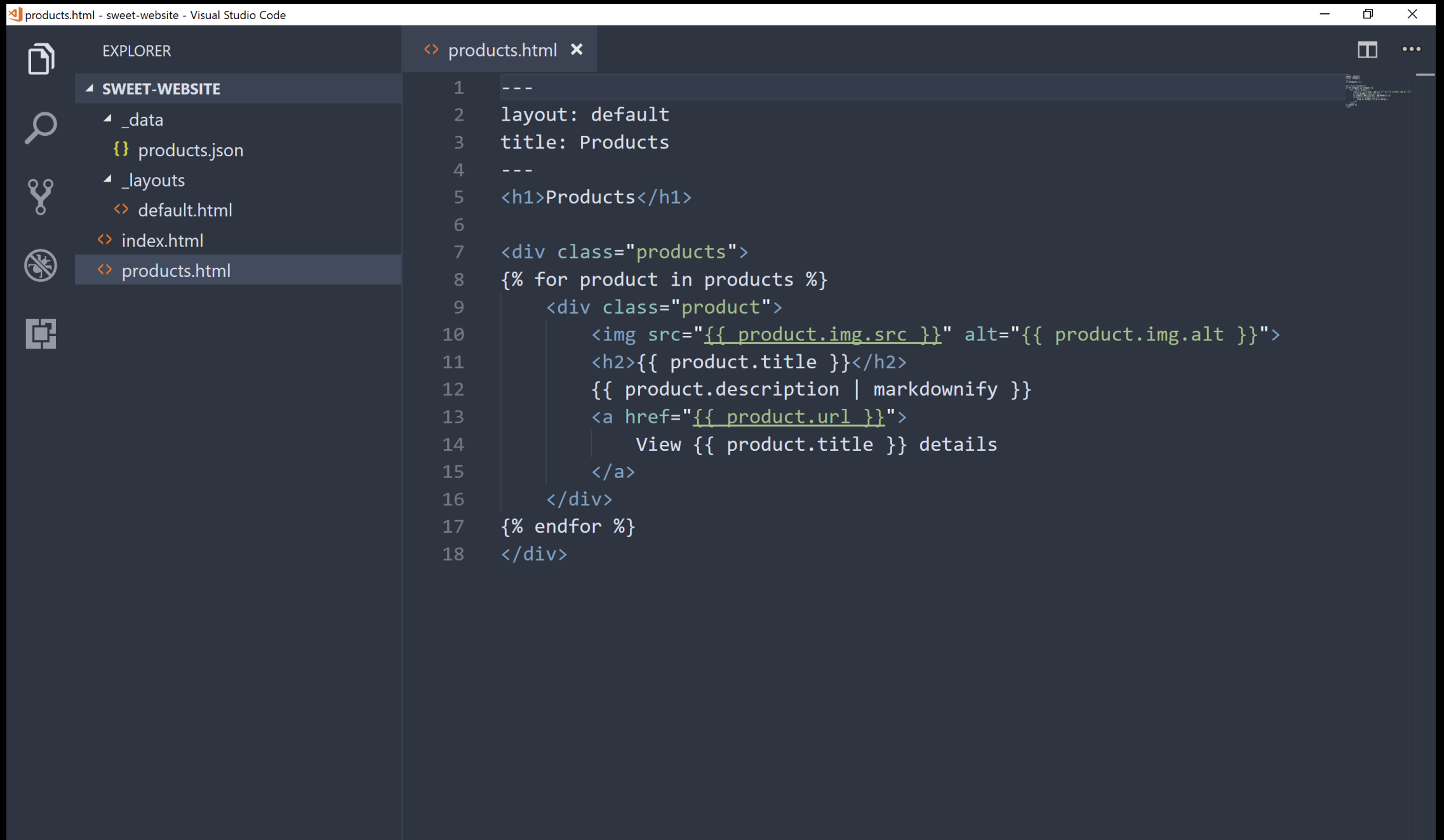
That experiment showed us how great the title could be, but also the amount of work required to make it.

– Hidemaro Fujibayashi

Build a prototype to validate ideas.

Experiment to get better estimates of the work required.


jekyll | Transform your p
https://jekyllrb.com/
jekyll
HOME
DOCS
NEWS
COMMUNITY
HELP
GITHUB
Transform your plain text into static websites and blogs.
Simple
No more databases, comment moderation, or pesky updates to install—just your content.
How Jekyll works →
Static
Markdown (or Textile), Liquid, HTML & CSS go in. Static sites come out ready for deployment.
Jekyll template guide →
Blog-aware
Permalinks, categories, pages, posts, and custom layouts are all first-class citizens here.
Migrate your blog →
Quick-start Instructions
NO DATABASE!!


products.html - sweet-website - Visual Studio Code
EXPLORER
SWEET-WEBSITE
_data
products.json
_layouts
default.html
index.html
products.html
products.html
---
layout: default
title: Products

<h1>Products</h1>

{% for product in products %}

<img src="{{ product.img.src }}" alt="{{ product.img.alt }}">
<h2>{{ product.title }}</h2>
{{ product.description | markdownify }}

View {{ product.title }} details

{% endfor %}


Microsoft

"What Went Right and What Went Wrong": An Analysis of 155 Postmortems from Game Development

Michael Washburn Jr.[1], Pavithra Sathiyanarayanan[1], Meiyappan Nagappan[1],
Thomas Zimmermann[2], Christian Bird[2]
[1]Rochester Institute of Technology, Rochester, NY, USA
[2]Microsoft Research, Redmond, WA, USA
{mdw7326, ps2908}@rit.edu, mei@se.rit.edu, {tzimmer, cbird}@microsoft.com

ABSTRACT

In game development, software teams often conduct postmortems to reflect on what went well and what went wrong in a project. The postmortems are shared publicly on gaming sites or at developer conferences. In this paper, we present an analysis of 155 postmortems published on the gaming site Gamasutra.com. We identify characteristics of game development, link the characteristics to positive and negative experiences in the postmortems and distill a set of best practices and pitfalls for game development.

Keywords

Games, Postmortems, Qualitative analysis.

1. INTRODUCTION

Over the past thirty years, the importance and market-share of video games in the world of software has grown by leaps and bounds. In lockstep with this growth, the scale of work required to develop games, whether in terms of budget, size of codebase, or team makeup, has ballooned and is on par with or exceeds any other software endeavors [13]. Games are arguably the most sophisticated and complex forms of software [18].

Indeed, games have been the driving factors behind many technological advances including high performance graphics cards, virtual reality, and distributed computing [16, 13]. Games also represent a substantial portion of software revenue; in 2013, video game revenue totaled over 93 billion dollars [21]! As such, the money, manpower, and effort put into video game development is likely to continue to increase in the coming years.

From a development perspective, games differ from more traditional software projects in a number of ways. Requirements are more subjective (e.g. "must be fun"), maintainability is often sacrificed for performance, testing and quality assurance are approached completely differently (e.g. live testers and few automated tests), most games require tools created from scratch, and deadlines are incredibly tight [12].

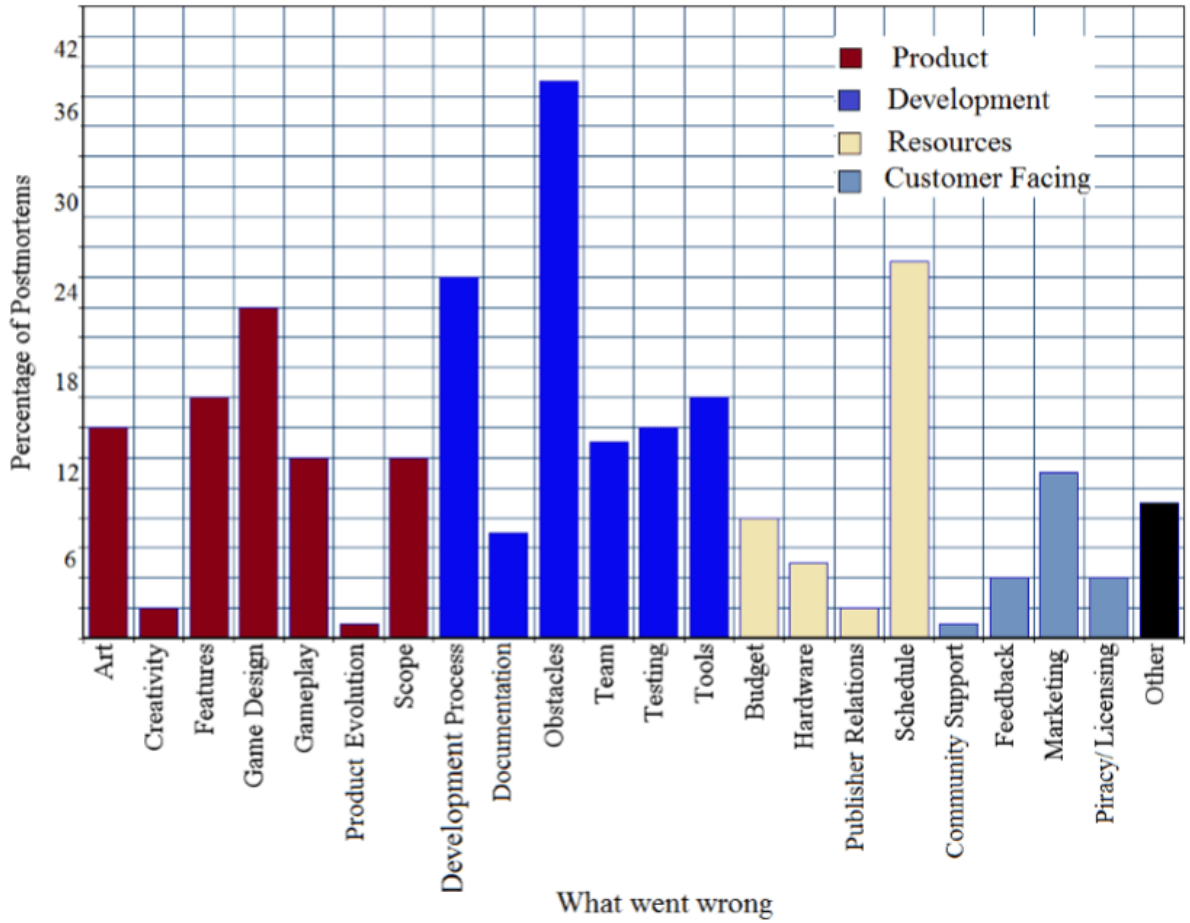
Therefore it is important to understand both the challenges that game development efforts face as well as the best practices that teams use to build games more effectively. The challenges are real problems faced by complex software efforts and represent avenues for research for our community. Successes and best practices embody knowledge that can aid future game development efforts and in some cases may generalize to or can be adapted for software development in non-game contexts. Because game development makes up a large slice of commercial software, a non-trivial proportion of students in computer science and software engineering programs will work on games during their careers. An understanding of game development can help educate and prepare such students.

Interestingly, game development has received very little attention in the academic community, as only three of the 116 open and closed source projects studied in the major software engineering conferences in two years were games [15]. Thus, one might reasonably expect that getting an inside view of game development is limited to a select few. Fortunately, the game development community has a unique practice that belies this assumption. Development teams often conduct postmortem retrospectives and share them publicly on gaming sites such as `Gamasutra.com` and at gaming conferences such as the Game Developers Conference (GDC). These postmortems offer an open and honest window into the development of games, often sharing the mistakes, setbacks, and wasted effort just as much as the successes and heroics that go into game building.

To address the limited study of this domain and shed light on the practice of game development we qualitatively and quantitatively analyze 155 retrospective postmortems published on `Gamasutra.com` over 16 years. These postmortems cover games for PCs, mobile devices, and consoles and range from small independent efforts to large AAA game franchises. As such, this represents the largest and most diverse study of game development to date and this data allows us to draw conclusions from a broad spectrum of game development. We make the following contributions in this paper.

- We present an empirically derived taxonomy of characteristics or dimensions of game development.
- We synthesize the best practices and commonly encountered challenges in game development and identify those areas that impact project outcomes the most.
- We provide recommendations for future game development based on the experiences shared in over one hundred postmortems.

ICSE '16 Companion, May 14-22, 2016, Austin, TX, USA
 ISBN 978-1-4503-4205-6/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2889160.2889253



"What Went Right and What Went Wrong": An Analysis of 155 Postmortems from Game Development

Michael Washburn Jr.[1], Pavithra Sathiyanarayanan[1], Meiyappan Nagappan[1],
Thomas Zimmermann[2], Christian Bird[2]
[1]Rochester Institute of Technology, Rochester, NY, USA
[2]Microsoft Research, Redmond, WA, USA
{mdw7326, ps2908}@rit.edu, mei@se.rit.edu, {tzimmer, cbird}@microsoft.com

ABSTRACT

In game development, software teams often conduct postmortems to reflect on what went well and what went wrong in a project. The postmortems are shared publicly on gaming sites or at developer conferences. In this paper, we present an analysis of 155 postmortems published on the gaming site Gamasutra.com. We identify characteristics of game development, link the characteristics to positive and negative experiences in the postmortems and distill a set of best practices and pitfalls for game development.

Keywords

Games, Postmortems, Qualitative analysis.

1. INTRODUCTION

Over the past thirty years, the importance and market-share of video games in the world of software has grown by leaps and bounds. In lockstep with this growth, the scale of work required to develop games, whether in terms of budget, size of codebase, or team makeup, has ballooned and is on par with or exceeds any other software endeavors [13]. Games are arguably the most sophisticated and complex forms of software [18].

Indeed, games have been the driving factors behind many technological advances including high performance graphics cards, virtual reality, and distributed computing [16, 13]. Games also represent a substantial portion of software revenue; in 2013, video game revenue totaled over 93 billion dollars [21]! As such, the money, manpower, and effort put into video game development is likely to continue to increase in the coming years.

From a development perspective, games differ from more traditional software projects in a number of ways. Requirements are more subjective (e.g. "must be fun"), maintainability is often sacrificed for performance, testing and quality assurance are approached completely differently (e.g. live testers and few automated tests), most games require tools created from scratch, and deadlines are incredibly tight [12].

Therefore it is important to understand both the challenges that game development efforts face as well as the best practices that teams use to build games more effectively. The challenges are real problems faced by complex software efforts and represent avenues for research for our community. Successes and best practices embody knowledge that can aid future game development efforts and in some cases may generalize to or can be adapted for software development in non-game contexts. Because game development makes up a large slice of commercial software, a non-trivial proportion of students in computer science and software engineering programs will work on games during their careers. An understanding of game development can help educate and prepare such students.

Interestingly, game development has received very little attention in the academic community, as only three of the 116 open and closed source projects studied in the major software engineering conferences in two years were games [15]. Thus, one might reasonably expect that getting an inside view of game development is limited to a select few. Fortunately, the game development community has a unique practice that belies this assumption. Development teams often conduct postmortem retrospectives and share them publicly on gaming sites such as `Gamasutra.com` and at gaming conferences such as the Game Developers Conference (GDC). These postmortems offer an open and honest window into the development of games, often sharing the mistakes, setbacks, and wasted effort just as much as the successes and heroics that go into game building.

To address the limited study of this domain and shed light on the practice of game development we qualitatively and quantitatively analyze 155 retrospective postmortems published on `Gamasutra.com` over 16 years. These postmortems cover games for PCs, mobile devices, and consoles and range from small independent efforts to large AAA game franchises. As such, this represents the largest and most diverse study of game development to date and this data allows us to draw conclusions from a broad spectrum of game development. We make the following contributions in this paper.

- We present an empirically derived taxonomy of characteristics or dimensions of game development.
- We synthesize the best practices and commonly encountered challenges in game development and identify those areas that impact project outcomes the most.
- We provide recommendations for future game development based on the experiences shared in over one hundred postmortems.

ICSE '16 Companion, May 14-22, 2016, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/2889160.2889253

9. CONCLUSIONS

We find that we were able to identify both best practices and pitfalls in game development using the information present in the postmortems. Such information on the development of all kinds of software would be highly useful too. Therefore we urge the research community to provide a forum where postmortems on general software development can be presented, and practitioners to report their retrospective thoughts in a postmortem.

Finally, based on our analysis of the data we collected, we make a few recommendations to game developers. First, be sure to practice good risk management techniques. This will help avoid some of the adverse effects of obstacles that you may encounter during development. Second, prescribe to an iterative development process, and utilize prototypes as a method of proving features and concepts before committing them to your design. Third, don't be overly ambitious in your design. Be reasonable, and take into account your schedule and budget before adding something to your design. Building off of that, don't be overly optimistic with your scheduling. If you make an estimate that initially feels optimistic to you, don't give that estimate to your stakeholders. Revisit and reassess your design to form a better estimation.

“Prescribe to an iterative development process, and use prototypes as a method of proving features and concepts before committing them to your design.”

VALUE PROTOTYPES

PROTOTYPES ARE FOR EVERYONE

THANKS!

Dave Rupert @davatron5000

Illustration by Kyle Ferrin @d20plusmodifier