SmashingConf NYC 2018

Day 2, October 24th, 2018

http://smashed.by/nyc02

Hello friends! :)

Let's take notes together.

Please be cool and **friendly**.

Notes from Day 1: ***https://smashed.by/nyc01***

09:40 — And The Mystery Begins!

Who Am I?

John Maeda!!!!

What am I talking about?

-

Questions:

-

Notes & Key takeaways:

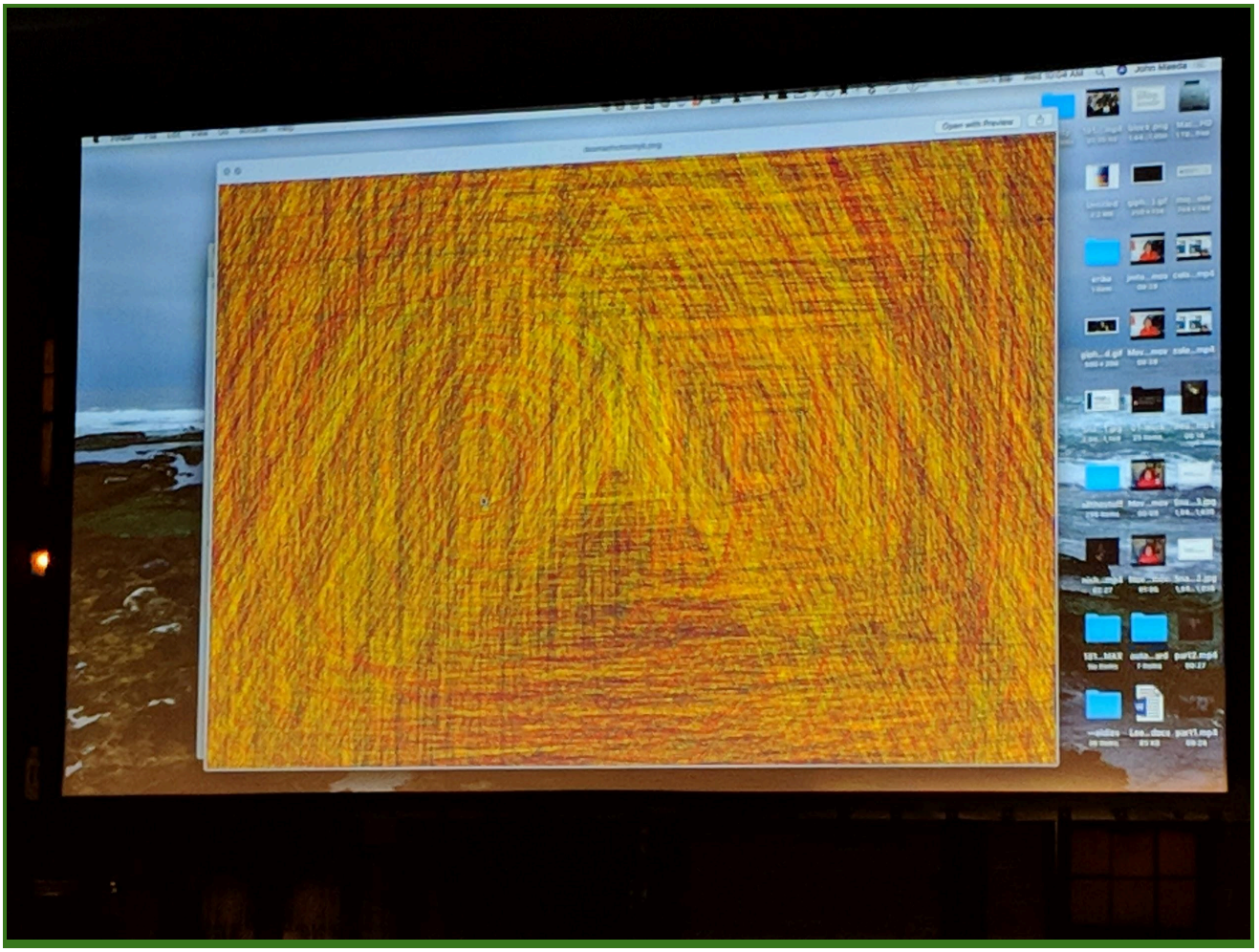
- John is the global head of Automattic
- “The great pleasure in life is doing what people say you cannot do.”
    - Defy the path that has been given to you.

    - Walk through the doubt to find your new self.
- Art school begs the question, “what should you make?”
- Maeda: As a Technologist, Good at making things, but don’t know what to make.
    - Developed respect for physical, non-computational materials
    - “Only goodness lies in understanding your material.”
    - “One person can change everything.”
    - “The classics will always be there when you get older”
- Illustrator: Oliver Jeffers
- Design is a “People challenge not a people problem”
- Remote collaboration vs. In-house
    - Slack, zoom, mural. Virtual tools: vehicles for transparency and collab., even for in-house teams
    - Importance of working onsite? Working remotely impacts peoples’ lives (more time for eldercare, childcare)
    - Time zone inclusivity
- Vitally: best case scenario in his experience: all in-house or all remote.
- Candid feedback / “Brutally honest feedback” shared with teams
    - Improves transparency
    - Radical candor (book)
- Maeda: Voice and face communicate a lot.
- Quote, that has stuck with Vitaly for some time. “What can I do to make your life easier right now?” - Dan Mall

- Asynchronous communication across time zones using recorded video / voice messages
- What are the unique capabilities of emerging technologies?
    - E.g. chatbots. Does this help me solve my problem? Is this making the process more efficient or overcomplicating it?
- What are the names and titles of the two other speakers?
    - Alison Rand, Head of Design Operations @alisonrand
    - Alexis Lloyd, Head of Design Innovation @alexislloyd
- “Use the material of the internet to make things.”
    - Create tools that are functional, but also enable virtuosity
- John: I would like to see people who are not like us be able to use this technology to make some money.

Links & Slides:

- **https://maedastudio.com/**
- **http://www.hotstudio.com/**



10:30 — Sarah Drasner on Serverless Functions and Vue.js

Sarah is an award-winning Speaker, Sr Cloud Developer Advocate at Microsoft, and Staff Writer at CSS-Tricks. Sarah is also the co-founder of Web Animation Workshops, with Val Head. She's the author of SVG Animations from O'Reilly and has given Frontend Masters workshops on Vue.js and Advanced SVG Animations. Sarah is formerly Manager of UX Design & Engineering at Trulia (Zillow). Sarah won CSS Dev Conf's "Best of the Best Award" as well as "Best Code Wrangler" from CSS Design Awards.

———————

Despite the name, Serverless does not mean you're not using a server, rather, the promise of Serverless is to no longer have to babysit a server. Scaling is done for you, you're billed only for what you use. In this session, we'll cover some key use cases for these functions within a Vue.js application: we'll accept payments with stripe, we'll gather geolocation data from Google Maps, and more! We'll make it all work with Vue and Nuxt seamlessly, simplifying how to leverage this paradigm to be a workhorse for your application.

Questions:

- Is it difficult to transition from React to Vue, and what should developers know about the differences to smoothen the transition?
- What are your strategies for refactoring and debugging code?
- You've been a big proponent of Visual Studio Code for years now. Can you show some of the useful extensions that save you time and headaches every single day? Perhaps some other obscure tools you are using all the time? Any particular plugins/extensions you'd recommend for Visual Studio Code?
- With so much happening in front-end these days, are there any things that you are particularly excited about these days?
- You've transitioned from teaching art in the middle of Greece to coding serverless functions with Vue.js: **how does that happen?**

Notes & key takeaways:

- Serverless Functions: actually interesting thing with a "Clickbait-y Title"
- FaaS (Functions as a Service)
    - Much clearer name than "Serverless"
    - Executing your function on someone else's computer

- Benefits
    - Only pay for what you use (usually a lot less)
    - Less time babysitting a server
    - Reusable functions – great for maintaining
    - Principle of least power
- With FaaS functions only run when a request comes in
- Not good for real-time apps
- Good for:
    - Take data and create visualizations
    - Cleaning data on a cron job
    - Running a computationally heavy process
- Use cases: Vue.js, Stripe and Serverless
    - Vuex || Redux (React)
        - Vue Devtools: can view the state
    - Get the testing key for the API

- `pay ()`
    - Uses axios to post our function to our function URL
    - Tracks whether we've submitted the form
- Google MAps API powered Data Vis
    - Three.js, gio.js library, vue.js
        - Three.js || the canvas element; called when mounted

Links & Slides:

- Sample Vue.js ecommerce site
- https://sarahdrasnerdesign.com/
- Sara Drasner - css tricks
- Gio.js
- three.js

Local Community

- **Hudson Valley Tech Meetup**
- **https://artofworkingremotely.com/**
- **https://londonwebstandards.org/**

11:50 — Michael Flarup on Designing for Augmented Reality

Michael is a Danish designer & entrepreneur. He runs entertainment studio Northplay, Pixelresort and design resource platform Apply Pixels. He's got a treasure-trove of experience creating his own companies, leading teams of developers and designers and working on countless of successful products, games and services.

———————

We thought we could port our existing game to an AR experience. We couldn't. Here's what happened and everything we learned.

In this talk I'll take the audience through the journey we went on creating our first AR-only experience. Our game Conduct AR! I'll talk about how many of the assumptions we had going in turned out to be wrong and how we worked to change them in a race to launch alongside ARkit.

You'll Learn

- What makes AR Special (hint: it's not what people think)
- What the biggest challenges are for AR apps and games
- What you need to consider when doing UI in AR

- What AR enables us to do right now, and what it doesn't (but might in the future)

It'll be an explosive, entertaining and visual journey into one of the most exciting frontiers in technology today.

Questions:

- Do you think your products suffer from not having a diverse team? Specifically women? ***Noted ✓ (Vitaly)***
- How would you best use AR in an educational setting to teach emotional intelligence to adults or children? ***Noted ✓ (Vitaly)***
- Is there a way to create responsive AR/VR experiences, or do we need to design dedicated experiences for Oculus Rift and all the other headsets?
- What are your thoughts on beacons and AR?
- What are some of the things you wish you'd done differently in your AR application?

- Any resources and tools and techniques you would recommend to designers and developers who want to start out with AR/VR?

- Do you see AR/VR becoming a mainstream for traditional applications, or do you see it remaining a tool of choice for very particular cases, e.g. education, immersive games etc.?

- Sometimes we forget the sense of reality when using the devices... but the form factor is a big issue. It's unlikely to see many people heading out with a headset like Oculus Rift. Are there any developments of adjusting the form factor? Google Glass seems to have failed in that department.

- You've been designing interfaces for quite some time. What techniques and strategies help you avoid common icon design mistakes and issues creeping into our interfaces?

- You mentioned one member of your team pivoted his career from front end development to game development. What was his transition like? What professional development was necessary and what challenges did he face? What skills was he able to transfer over? Any advice for someone looking to transition? ***Noted ✓ (Vitaly)***

- What are some great examples of AR done with accessibility in mind? Or how would you hope to push AR to be more inclusive? ***Noted ✓ (Vitaly)***
    - Worked on be my eyes, an app for vision impaired people
    - New frontier
- Does Conduct AR use markerless placement or does it use markers? How does that work? ***Noted ✓ (Vitaly)***
    - Comes from AR kit, it's like a black box, secret sauce
- How do you debug AR experiences? Any tools / strategy you'd recommend?

Notes & key takeaways:

- Known for work on Icons
- Designing icons can make you a better designer
- 2016, started NorthPlay
- **Conduct AR**, Conduct This!, Conduct Deluxe
    - Look at the beginning with Conduct This, before AR
- "AR is not an afterthought."
    - AR is a core interaction method.
    - Game was hard enough in Conduct This
- AR design challenges

    - Inability to control the camera
    - Had to build from scratch
- AR Lessons
    - Try not to design against the constraints of the medium
    - Use spatial awareness of gamer player
    - Controlling the camera and playing the game was challenging for players
    - Themes help you make decisions about products, creates a narrative
    - If you want user exploration, create a world worth exploring
    - Testing is difficult, laborious
    - UI in AR is hard – what's in world / out of world?
- Solution for holding device and interaction - Added a stop-go-switches on each side with center reticle
- Tools we use
    - Blender - 3d models and animation
    - Photoshop - 2d, ui, art direction
    - Unity - game engine
    - Xcode (build and distribute) ↠ TestFlight (game test)
- Experiment with AR
    - Gamify experience
    - Shopping, education
- Game design: allows you to reinvent world with each attempt
- Cost ~50,000 USD to make Conduct AR

Links & slides:

- https://northplay.co/
- https://developer.apple.com/arkit/
- IKEA Place, AR

12:40 — Monica Dinculescu on Bet you Didn't Think Your Browser Could do That

Monica works on Magenta, in Google Brain, where she makes generative music and art with Machine Learning. In the past she's worked on Polymer, web components, and Chrome, and has probably at least once broken the Internet for you. She is unreasonably excited about emoji, wary of web fonts, and will become your best friend if you bring her cheese. On second thought, she may be a mouse.

———————

Building a modern web app is pretty complicated! You need to make pretty things! That are fast! But also useful! Ideally offline! And can accommodate many users! One way you can simplify this mountain of requirements is to think of your app as a static site, and build it like one. With things like Service Workers, serverless architectures and Firebase authentication, a lot of the features that used to need a server can live entirely on the client side, leaving you with one less thing you need to worry about. Let's do some myth busting and see that static sites can do more than you expect!

Questions:

- Does the choice of a static site generator matter? How do you choose one for a particular situation?
- But what about performance, Monica, what about performance and dependencies? Do we really want to push everything to the client and all those services? Remember the AWS, Firebase outage? Shouldn't we keep server-side rendering as fallback? ***Noted ✓ (Vitaly)***
- Resilience. JS requests timeout (Buzzfeed talk). How do we keep our applications resilient to failure when DynDNS, AWS etc. go down?
- Any particular tools you can recommend to save time and trouble and headaches when working on a fancy new app?
- How can dynamic widgets co-exist with a static environment? Does it cause any issues? Re-deployment etc.?
- What's your maintenance strategy?
- How do you start refactoring or debugging?
- What are some of the important things you've learned in your dev career so far?

- I work for a company that requires all data to be on-premises for security purposes (hipaa, ferpa, etc.). I understand the pitfalls of self hosting. Are there solutions out there that meet halfway between self hosting and using the cloud? Perhaps some self hosted SaaS that abstracts away the micromanagement concerns? ***Noted ✓ (Vitaly)***

Notes & key takeaways:

- IBM Plex mono
- Inktober
- Magenta
- “More like me, the laziest developer in the universe”
- Static sites are dynamic these days
- “The more complex your setup is, the more likely it is that it will break.”

  The less complex, the more you can actually deliver!
- Be a lazy developer and use the stuff other people have already built!

- Buying hardware for a server is expensive. Maintaining a server is expensive.

    Need backups for your backups

- **DBs**:

    Sql : azure

    no-sql : firebase

    Wat-sql : local-storage

- Static site Servers uses:

    - Submit a form

        Cloud servers

    - OAuth, validate passwords

        Firebase! Provides a clean UI

    - Get around CORS (cross origin ritual sacrifice) – dynamic HTML, web scraping

    - Serverless functions: Azure, firebase, aws, netify

    - Actually write server code with: digital ocean, heroku,

- “Chrome is an operating system!”

- WebGL shaders

    - Runs in the browser
    - Enables badass graphics on the web
    - https://split82.itch.io/tisic-izieb
    - really good at matrix math
    - http://oatthegoat.co.nz/
    - Machine learning problems https://js.tensorflow.org/
    - Get data ↠ train model ↠ ask the model to do something
        - Use the model to do something for the user
        - https://magenta.tensorflow.org/assets/sketch_rnn_demo/index.html
        - http://incredible-spinners.glitch.me/

Links & slides:

- https://github.com/notwaldorf
- twitter
- https://inktober.com/
- google polymer
- IBM Plex mono

- Glitch **is the friendly community where everyone can discover & create the best stuff on the web**
- Google Firebase
- Zeit
- awesome serverless
- https://serverlesslibrary.net !!!
- https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7ddobc881cd5
-
-

3:00 — Paul Boag on Encouraging Clicks - How to Encourage Action Without Alienating People

Paul Boag has been working on the Web since 1993. He is a user experience strategist who helps companies make use of digital to better serve connected consumers. Paul also hosts the award-winning user experience podcast at boagworld.com. He is a regular speaker at conferences and author of four books including *Digital Adaptation*, as well as *User Experience Revolution*.

———————

Every website seeks to encourage visitors to do something, whether it is placing an order, make a donation or signup to a mailing list. But how do you do that without alienating users?

After all, we now live in a world where one disgruntled customer can destroy a brand. Paul Boag explains how we don't need to resort to dark patterns to

encourage users to take action.

Questions:

- We see many conversations circling around ethical, responsible and respect-driven UX practices at this point but we still see many companies using and misusing privacy, data, and dirty tricks to manipulate customers. How do we change it on big scale? And is being ethical, honest and transparent has become a new competitive advantage these days?
- How do you know when something is simple enough? Or how do you know if something is too simple? Testing?
- “We had an interesting conversation / redesign / personality / cats / you told me we did too much / went over the board…” (Vitaly) :)
- There are things you can track and things you can’t track. We can track conversion rates, but what about long-term loyalty.
- Humanize / connect with people: that’s the opposite of generic designs?
- Bruce is asking when does rewriting become too manipulative? How do you know when you are going overboard?
- Internal note to myself: Netlify, Nintendo Switch giveaway

Notes & key takeaways:

- Manipulative ecommerce techniques, “Dark patterns”
    - A dark pattern is "a user interface that has been carefully crafted to trick users into doing things, such as buying

insurance with their purchase or signing up for recurring bills."
    - “3 people also viewing this product, limited quantity”
- **Problem with dark patterns**
    - Consumers are very aware that companies are trying to manipulate them online ↠ undermines trustworthiness of the brand ↠ leads to a financial cost
    - Creates a generation of cynical consumers. We expect online retailers to screw us over ↠ damages conversion.
    - Increased sense of buyer’s remorse, damages your brand.
        - Average facebook user has 338 friends (of which one can tell about your brand)
        - Hasan Sayed bought a promoted tweet bashing British Airways that would send a tweet saying “don’t fly BA, their customer service is terrible” whenever someone tweeted about BA - single-handedly caused a PR trainwreck
        - Costs your brand financially: greater cost in customer service reviews.
- **How people make decisions**
    - System I: Instinctive response, automatic, intuitive, emotional mode of thinking. This is favored.
    - System II: slower, deliberate. Used to make logical decisions.

- **Reducing Cognitive Load**
    - To engage people, get them to take action; Keep cognitive load low, System I
    - High cognitive load feels like hard work, unfamiliar, uncomfortable
    - User tests are *paid* users, so they are concentrating more than *real* users
    - To reduce cognitive load
        - Remove clutter on webpage
        - Create content that's scannable
        - Content/form that's familiar
        - Avoid redesigns – or expect resistance from users
        - Seek consistency
        - Embrace convergence of design patterns
- **Set expectations through priming**
    - Usability testing - asked testers about Oxford University, "Do you think this university is for you?" - the images didn't look like the participants, and they said no.
        - If we don't see people who look like us...we assume it isn't for us.
    - Focus on benefits and features
    - Be personal! Not Corporate!
        - Avoid using the 3rd person
        - Using the 2nd person sounds more inclusive, community-oriented.

        - Use Honesty! Flickr: “Sometimes we suck”
    - **Help people overcome choice paralysis to improve conversion**
        - Make options distinct
            - Amazon’s buttons at the end of a product page: “Preorder item today” & “Preorder it now” -- seriously?
        - Simplify the question
        - Smaller steps, take the user step by step
        - Better recommendations
        - Reduce risk
        - Objection handling
            - Answer a user’s questions before they even ask them!
            - Take possible objections and address them one by one, e.g. “are there hidden charges?” “what if I have a problem?” “what will you do with my data?”
                - Maintains transparency and trustworthiness
                - Makes customers Happy!! - the holy grail.

Links & slides:

- Hasan Sayed British Airways story
- Monkey business illusion

    - Great exercise to explain why we can't add another element to the page.
- http://hemingwayapp.com/
    - Test how easier your copy is to read.
- https://boagworld.com/
- Boag masterclass

15:50 — Josh Clark on Design in the Era of the Algorithm

Josh Clark is founder of Big Medium, a design agency specializing in connected devices, mobile experiences, and responsive web design. His clients include Samsung, Time Inc, TechCrunch, Entertainment Weekly, eBay, O'Reilly Media, and many others. Josh has written several books, including "Designing for Touch" (A Book Apart, 2015) and "Tapworthy: Designing Great iPhone Apps" (O'Reilly, 2010). He speaks around the world about what's next for digital interfaces.

———————

The design and presentation of data is just as important as the underlying algorithm, and designers have an urgent role to play in crafting the emerging generation of AI interfaces.

Designer Josh Clark explores a rich trove of examples—at once entertaining and sobering—to unearth design principles for creating responsible machine-learning applications.

Learn to use AI as design material in your everyday work. Adapt to the machines' weird, unexpected, or flat-out-wrong conclusions. And above all, scrub data for bias to create the respectful and inclusive world we all want to live in.

———————

Questions:

- With how quickly Ai and machine learning is evolving, will we have jobs in 50 years (not even joking)
- Should we really give "the big guys" our data to use the APIs? Nothing is ever really free.
- What's better left to humans and what to machines?
- As designers, we aren't in the business of predicting the future, but looking back at the past 10-15 years of evolution of design — even moving from desktop-focused experience to designing for mobile to designing conversational interfaces now, can you see some patterns that keep occurring, over and over?

- As designers, we have to learn something new all the time — we've got pretty far when designing for screens, large and small, but now we are confronted with audio interfaces, VR, AR, XR, conversational interfaces and we have to figure out how to design for them. Which of these things will remain relevant in your opinion, and which are just trends that will fade away?

- Many designers fear that "systemizing" experiences and design essentially removes soul from the designs as we are creating and using a framework of pre-built blocks, hence restricting creativity. Where do art direction and creative direction fit in this context? How do we art-direct a chatbot that doesn't even have a screen sometimes?

- **Many people have real issues accepting the fact that machines can be creative.** Mostly because it's assumed that they are running on predictable step-by-step algorithms and hence can't create something new from scratch. However, when we think about the sheer amount decisions such machines can make in a blink of a second, isn't it more reliable to let machines build autonomous cars and perfect bridges and perfect architecture and our systems. Where does the "human factor" fits in this story?

- Does the future look bright? Is there any particular thing that you wish to come sooner than later?

Notes & key takeaways:

- IBM Watson
- Azure Machine Learning
- Design for failure and uncertainty
- Machines are way too confident in their answers
    - Google autofilling questions
- Machines should be designed to indicate when it isn't 100% confident in the answer its giving
    - A dinosaur (**maybe** on a surfboard)
- Google speech recognition is gender biased
    - Women speak more clearly and yet, machines have a harder time understanding them
    - Machines are designed by white men and work best for white men
- Amazon stopped using their machine learning hiring system because it was showing biased towards men
- We have to correct biases - that's one of our roles
- Devices have interests that aren't shared by their owners
    - If you change "I had a shitty day" to "I had a challenging day" on a FB post, FB keeps your original, unedited post, uses that data. Even if you don't post it.
- Design toward the user's needs
- We should be shaping the future, not letting technology shape it for us.

- Machines can be good companions to reveal flaws in our systems and data.

Links & slides:

- https://experiments.withgoogle.com/giorgio-cam
- https://azure.microsoft.com/en-us/services/machine-learning-studio/
- Popper Paradox
- Mindfultechnology.com
- Juvetagenda.org
- bigmedium.com

Notes from Day 1: ***https://smashed.by/nyco1***