What makes an interface feel fluid?
What makes an interface feel fluid?
What makes an interface feel fluid?
You know it when you feel it
A tool that feels like an extension of your mind
Paleolithic Handaxe
150,000 BC

Apple iPhone
2017 AD
An extension of the natural world
An extension of yourself
Interfaces that extend our minds
Interfaces that extend our minds
Designing with dynamic motion
Interfaces that extend our minds
Designing with dynamic motion
Responding to interaction
Interfaces that extend our minds
Designing with dynamic motion
Responding to interaction
Fluidity as a medium
Interfaces that extend our minds

Designing with dynamic motion

Responding to interaction

Fluidity as a medium
Aligning interfaces to the way we think and the way we move
Our interfaces need to be fluid because we are fluid
Respond to input instantly
10ms lag 50ms 100ms 200ms
Look for delays everywhere
Allow for constant redirection and interruption
What if it wasn’t redirectable?
Linear interfaces
Gesture happens \textit{after} thought
Gesture happens in *parallel* with thought.
Gestures in Parallel With Thought

Faster than thinking before doing

One gesture solves multiple problems

Helps with discovery

Allows you to layer gestures at the speed of thought
Monday, May 21, 2018

9 AM

10 AM
Weekly Team Meeting
3rd floor conference room

11 AM

12 PM
Noon

1 PM
Budget Review
3rd floor conference room

2 PM

3 PM
Swipe homescreen pages *while* going home.
Go to multitasking while app is launching
Close app while launching app
Interact with app as it launches
Respond to redirection as fast as possible
How can we detect this pause in motion?
How can we detect this pause in motion?

Timer too slow
Hey! Yesterday was so much fun. I really had an amazing time at my birthday party and the canoe trip was such an awesome surprise.

It was so great having all the family there. There were cousins I hadn’t seen in years, like before they could even walk. Can’t wait to go canoeing again. Promise we’ll do it sooner than later? 😊
Izzy Weberbauer
To: John Bishop

Birthday canoe trip
Today at 9:35 AM

Hey! Yesterday was so much fun. I really had an amazing time at my birthday party and the canoe trip was such an awesome surprise.

It was so great having all the family there. There were cousins I hadn’t seen in years, like before they could even walk. Can’t wait to go canoeing again. Promise we’ll do it sooner than later? 😊
Maintain spatial consistency throughout movement
Gray McCarty
Friday's show
You've got to check out his latest album first. I can't stop playing it!

Kevin Angel
Summer getaway
Road trip? I'm totally down for that. Anything to get me out of going to...

John Baily
Puppy training
Thanks for the advice yesterday. Everything worked out fine.

Lia Lange
@work life
I'm thinking about joining a sorority. Thoughts? They're having a rush next...
Izzy Weberbauer
Birthday canoe trip
Hey! Yesterday was so much fun. I really had an amazing time at my birthday party and the can... 

Brian Rekasis
The new digs
Do you think I should go with the designer we saw yesterday? Seems... 

Gray McCarty
Friday’s show
You’ve got to check out his latest album first. I can’t stop playing it! 

Kevin Angel
Summer getaway
Road trip? I’m totally down for that. Anything to get me out of going to... 

John Baily
Puppy training
Thanks for the advice yesterday. Everything worked out fine. 

Lia Longo
Greek life
I’m thinking about joining a sorority. Thoughts? They’re having a rush next...
Not spatially consistent
Hint in the direction of the gesture
Hinting

Initial

Final
Keep touch interactions lightweight
Keep touch interactions lightweight but amplify their motion.
Short interaction
infer trajectory with
position
velocity
speed
force
infer trajectory with position
velocity
speed
force
Amplified result of movement that still feels like an extension of you
Birthday cannes blog

Hey! yesterday was no much but I really had an amazing time at my birthday party and the came by was such an awesome surprise.

It was a great having all the family there. There were two days I haven't seen in years. We all talked and made jokes. Can't wait to go camping again tomorrow will visit another one.
Birthday canoe trip

Hey! Yesterday was so much fun! I really had an amazing time at my birthday party and the canoe trip was such an awesome surprise.

It was so great having all the family there. There were cousin's I hadn't seen in years. We all had so much fun and I can't wait to go canoeing again. Thanks for a fantastic day!
Manoj
Manoj
Manoj
April 14, 2019, 6:31 PM

Birthday canoe trip

Hey! Yesterday was so much fun — we had an
amazing time at your birthday party and
the canoe trip was such an awesome
surprise.

It was so good having all the family there.
There were some laughs too! Thanks for
inviting us to join you guys on your canoe
trip. Can't wait to go canoeing again.

 regard,

[Image of a canoe on water]
Birthday canoe trip

April 19, 2018

Hey! Yesterday was no much fun. We had a really amazing time at the birthday party and the singer was such an awesome surprise.

It was no great time at the lake there. There were dragonflies flying in my face. I think the people there should do something about the noise. Can't wait to go canoeing again! Thank you for the good time!
Softly indicate boundaries
Birthday canoe trip
April 19, 2018 at 6:38 PM

Hey! Yesterday was so much fun. I really had an amazing time at my birthday party and the canoe trip was such an awesome surprise.

It was so great having all the family there. There were cousins I hadn’t seen in years, like before they could even walk. Can’t wait to go canoeing again. Promise we’ll do it sooner than later? 😊
Soft transitions
Soft transition from dock to app
Design smooth frames of motion
Too much visual change
30fps looks smooth

30fps visually strobes
30fps looks smooth
30fps visually strobos
30fps

60fps

less strobing at same speed
Normal  Motion Blur  Motion Stretch
Normal  Motion Blur  Motion Stretch
Normal

Motion Blur

Motion Stretch
Work with behavior rather than animation
Fluid Interfaces Think Like Us
Fluid Interfaces Think Like Us

Instant response and constant redirection
Fluid Interfaces Think Like Us

Instant response and constant redirection

Maintain spatial consistency
Fluid Interfaces Think Like Us

Instant response and constant redirection

Maintain spatial consistency

Hint in the direction of the gesture
Fluid Interfaces Think Like Us

Instant response and constant redirection

Maintain spatial consistency

Hint in the direction of the gesture

Lightweight interactions, amplified output
Fluid Interfaces Think Like Us

Instant response and constant redirection

Maintain spatial consistency

Hint in the direction of the gesture

Lightweight interactions, amplified output

Soft boundaries and transitions
Fluid Interfaces Think Like Us

Instant response and constant redirection
Maintain spatial consistency
Hint in the direction of the gesture
Lightweight interactions, amplified output
Soft boundaries and transitions
Design smooth, dynamic behavior
Interfaces that extend our minds
Designing with dynamic motion
Responding to interaction
Fluidity as a medium
Interfaces that extend our minds

Designing with dynamic motion

Responding to interaction

Fluidity as a medium
Timed animations
Interactions result in *discrete animations*
Dynamic motion
Interactions influence continuous *dynamic behaviors*
Designing with Dynamic Motion
Designing with Dynamic Motion

Seamless motion
Designing with Dynamic Motion

Seamless motion

Crafting a character
Designing with Dynamic Motion

Seamless motion
Crafting a character
Understanding intent
Characteristics of the physical world make great behaviors.
Seamless Scrolling
Seamless Scrolling

Familiar, natural motion
Seamless Scrolling

Familiar, natural motion

Maintains throw momentum
Seamless Scrolling

Familiar, natural motion
Maintains throw momentum
Friction gracefully reduces speed
Seamless Scrolling

Familiar, natural motion
Maintains throw momentum
Friction gracefully reduces speed
Imperceptibly comes to rest
REST POSITION
Let's celebrate my 35th!
Today at 10:30AM

Hi everyone,

We’re well overdue for a camping trip to Big Sur, and I’d love to head there with you all for my birthday on the 24th of September!

I’m going to book a group campsite, so let me know if you’re free 😊

Love,
Ros
Let’s celebrate my 35th!
Today at 10:30AM

Hi everyone,

We’re well overdue for a camping trip to Big Sur, and I’d love to head there with you all for my birthday on the 24th of September!

I’m going to book a group campsite, so let me know if you’re free 😊

Love,
Ros
Bounciness can hint
Bounciness can hint
Think beyond motion
<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tue Feb 06</td>
<td></td>
<td>63</td>
</tr>
<tr>
<td>Wed Feb 07</td>
<td>9</td>
<td>57</td>
</tr>
<tr>
<td>Thu Feb 08</td>
<td>10</td>
<td>58</td>
</tr>
<tr>
<td>Fri Feb 09</td>
<td>11</td>
<td>59</td>
</tr>
<tr>
<td>Sat Feb 10</td>
<td>12</td>
<td>00 AM</td>
</tr>
<tr>
<td>Sun Feb 11</td>
<td>1</td>
<td>01 PM</td>
</tr>
<tr>
<td>Mon Feb 12</td>
<td>2</td>
<td>02</td>
</tr>
<tr>
<td>Tue Feb 13</td>
<td>3</td>
<td>03</td>
</tr>
<tr>
<td>Wed Feb 14</td>
<td>4</td>
<td>04</td>
</tr>
<tr>
<td>Date</td>
<td>Day</td>
<td>Time</td>
</tr>
<tr>
<td>------------</td>
<td>-----</td>
<td>--------</td>
</tr>
<tr>
<td>Tue Feb 06</td>
<td></td>
<td>08:50</td>
</tr>
<tr>
<td>Wed Feb 07</td>
<td></td>
<td>09:57</td>
</tr>
<tr>
<td>Thu Feb 08</td>
<td></td>
<td>10:58</td>
</tr>
<tr>
<td>Fri Feb 09</td>
<td></td>
<td>11:59</td>
</tr>
<tr>
<td>Sat Feb 10</td>
<td></td>
<td>12:00</td>
</tr>
<tr>
<td>Sun Feb 11</td>
<td></td>
<td>01:01</td>
</tr>
<tr>
<td>Mon Feb 12</td>
<td></td>
<td>02:02</td>
</tr>
<tr>
<td>Tue Feb 13</td>
<td></td>
<td>03:03</td>
</tr>
<tr>
<td>Wed Feb 14</td>
<td></td>
<td>04:04</td>
</tr>
</tbody>
</table>
Stay in character
Organic Molecular Structure

Countless Possibilities of Carbon Atoms
Organic Molecular Structure
Countless Possibilities of Carbon Atoms
Intent is expressed through motion
// Distance travelled after decelerating to zero velocity at a constant rate.
func project(initialVelocity: Float, decelerationRate: Float) -> Float {
    return (initialVelocity / 1000.0) * decelerationRate / (1.0 - decelerationRate)
}
// After the PiP is thrown, determine the best corner and re-target it there.

let decelerationRate = UIScrollView.DecelerationRate.normal

let projectedPosition = (  
  x: x.value + project(initialVelocity: x.velocity, decelerationRate: decelerationRate)  
  y: y.value + project(initialVelocity: y.velocity, decelerationRate: decelerationRate),
)

let nearestCornerPosition = nearestCornerTo(projectedPosition)

x.target = nearestCornerPosition.x
y.target = nearestCornerPosition.y
// After the PiP is thrown, determine the best corner and re-target it there.

let decelerationRate = UIScrollView.DecelerationRate.normal

let projectedPosition = (x: x.value + project(initialVelocity: x.velocity, decelerationRate: decelerationRate), y: y.value + project(initialVelocity: y.velocity, decelerationRate: decelerationRate),)

let nearestCornerPosition = nearestCornerTo(projectedPosition)

x.target = nearestCornerPosition.x
y.target = nearestCornerPosition.y
// After the PiP is thrown, determine the best corner and re-target it there.

let decelerationRate = UIScrollView.DecelerationRate.normal

let projectedPosition = (
    x: x.value + project(initialVelocity: x.velocity, decelerationRate: decelerationRate),
    y: y.value + project(initialVelocity: y.velocity, decelerationRate: decelerationRate),
)

let nearestCornerPosition = nearestCornerTo(projectedPosition)

x.target = nearestCornerPosition.x
y.target = nearestCornerPosition.y
// After the PiP is thrown, determine the best corner and re-target it there.

let decelerationRate = UIScrollView.DecelerationRate.normal

let projectedPosition = (
    x: x.value + project(initialVelocity: x.velocity, decelerationRate: decelerationRate),
    y: y.value + project(initialVelocity: y.velocity, decelerationRate: decelerationRate),
)

let nearestCornerPosition = nearestCornerTo(projectedPosition)

x.target = nearestCornerPosition.x
y.target = nearestCornerPosition.y
// After the PiP is thrown, determine the best corner and re-target it there.

let decelerationRate = UIScrollView.DecelerationRate.normal

let projectedPosition = {
    x: x.value + project(initialVelocity: x.velocity, decelerationRate: decelerationRate),
    y: y.value + project(initialVelocity: y.velocity, decelerationRate: decelerationRate),
}

let nearestCornerPosition = nearestCornerTo(projectedPosition)

x.target = nearestCornerPosition.x
y.target = nearestCornerPosition.y
Designing with Dynamic Motion
Designing with Dynamic Motion

Continuous behaviors, not timed animations
Designing with Dynamic Motion

Continuous behaviors, not timed animations

Draw inspiration from the physical world
Designing with Dynamic Motion

Continuous behaviors, not timed animations

Draw inspiration from the physical world

Springs don’t need to be springy!
Designing with Dynamic Motion

Continuous behaviors, not timed animations

Draw inspiration from the physical world

Springs don’t need to be springy!

Align motion with intent
Interfaces that extend our minds
Designing with dynamic motion
Responding to interaction
Fluidity as a medium
Interfaces that extend our minds
Designing with dynamic motion
Responding to interaction
Fluidity as a medium
Responding to Interaction
Responding to Interaction

Design of taps and swipes
Responding to Interaction

Design of taps and swipes

Principles of fluid interactions
Responding to Interaction

Design of taps and swipes
Principles of fluid interactions
Dealing with multiple gestures
Designing a tap
Designing a swipe
Designing a swipe
Touch and content move together
Organic Molecular Structure
Countless Possibilities of Carbon Atoms
BeautifulChemistry.net

Organic Molecular Structure

Countless Possibilities of Carbon Atoms
Provide continuous feedback
UISwipeGestureRecognizer
UISwipeGestureRecognizer
UITouches

UIPanGestureRecognizer

UISwipeGestureRecognizer
Combining gestures
What is the gesture?  

Starts  

Visual feedback  

Recognized
What is the gesture? | Visual feedback
---|---
Starts | Recognized
Taps are delayed because of double-tap
Responding to Interaction
Responding to Interaction

Design of taps and swipes
Responding to Interaction

Design of taps and swipes

Content and touch move one-to-one
Responding to Interaction

Design of taps and swipes

Content and touch move one-to-one

Provide continuous feedback
Responding to Interaction

Design of taps and swipes

Content and touch move one-to-one

Provide continuous feedback

Detect multiple gestures in parallel
Interfaces that extend our minds
Designing with dynamic motion
Responding to interaction
Fluidity as a medium
Interfaces that extend our minds
Designing with dynamic motion
Responding to interaction
Fluidity as a medium
Teaching
Visual Cues
Elevate interactive elements to a separate plane
Use behavior to teach gesture
Explanations
Fun
Playfulness
Playfulness

A natural consequence of a fluid interface
Playfulness

A natural consequence of a fluid interface
Happens when the interface feels in sync with you
Playfulness

A natural consequence of a fluid interface
Happens when the interface feels in sync with you
You feel comfortable exploring new areas
Playfulness

A natural consequence of a fluid interface
Happens when the interface feels in sync with you
You feel comfortable exploring new areas
Allow people to **discover** your interface through play
Play into our natural fiddle factor
Fluidity as a Medium
Fluidity as a Medium
Fluidity as a Medium

Design the interactions to be *inseparable* from the visuals
Fluidity as a Medium

Design the interactions to be *inseparable* from the visuals

A interactive demo is worth a million static designs
Fluidity as a Medium

Design the interactions to be *inseparable* from the visuals

A interactive demo is worth a million static designs

Prototyping helps you *think* by exploring
Fluidity as a Medium

Design the interactions to be *inseparable* from the visuals

A interactive demo is worth a million static designs

Prototyping helps you *think* by exploring

Sets a goal for the implementation
Fluidity as a Medium

Design the interactions to be *inseparable* from the visuals
A interactive demo is worth a million static designs
Prototyping helps you *think* by exploring
Sets a goal for the implementation
Hard to copy and gives your app a unique character