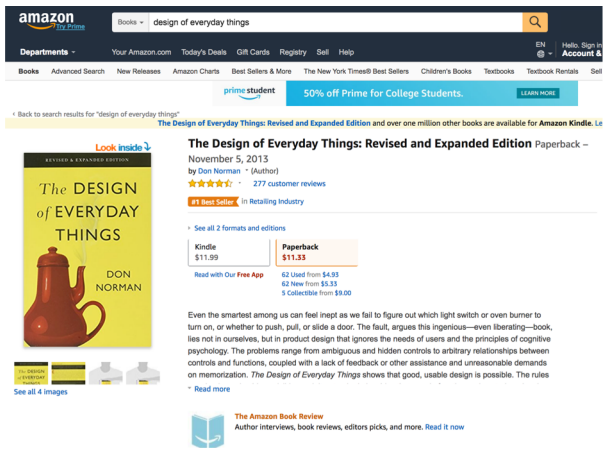
Events and Feedback

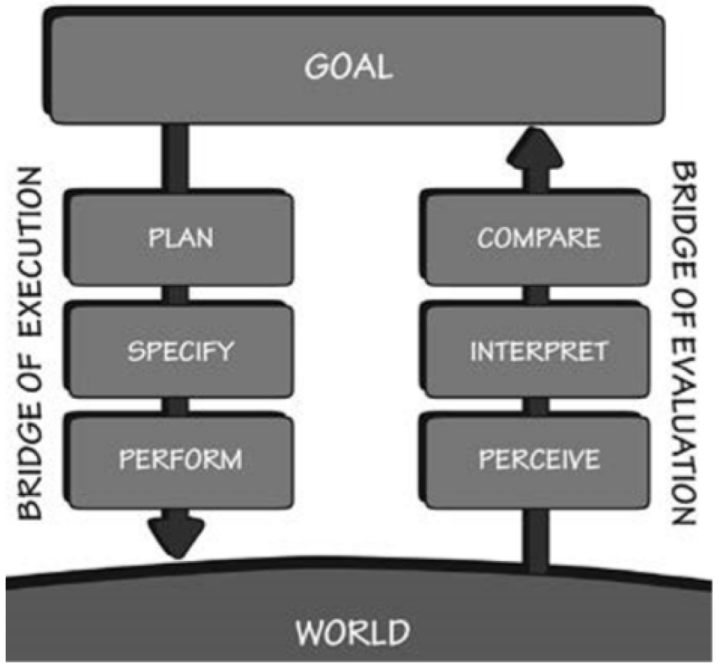
No screens

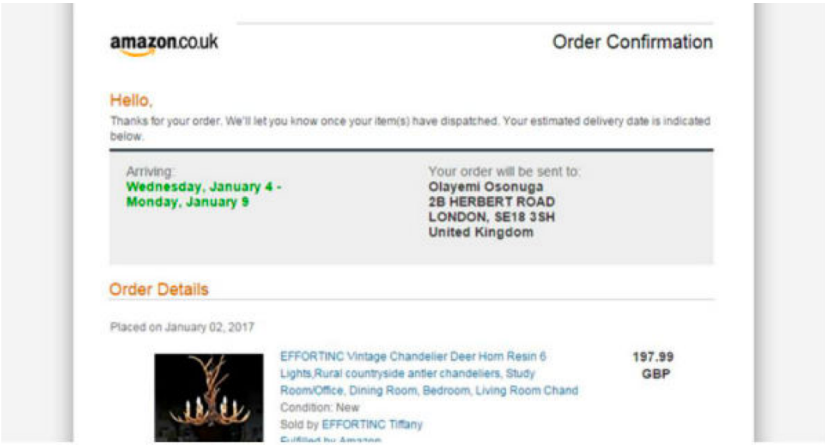
Prof. Lydia Chilton
COMS 4170
10 February 2020

Say your name

To accomplish a goal, users must
execute an operation and **evaluate** the result







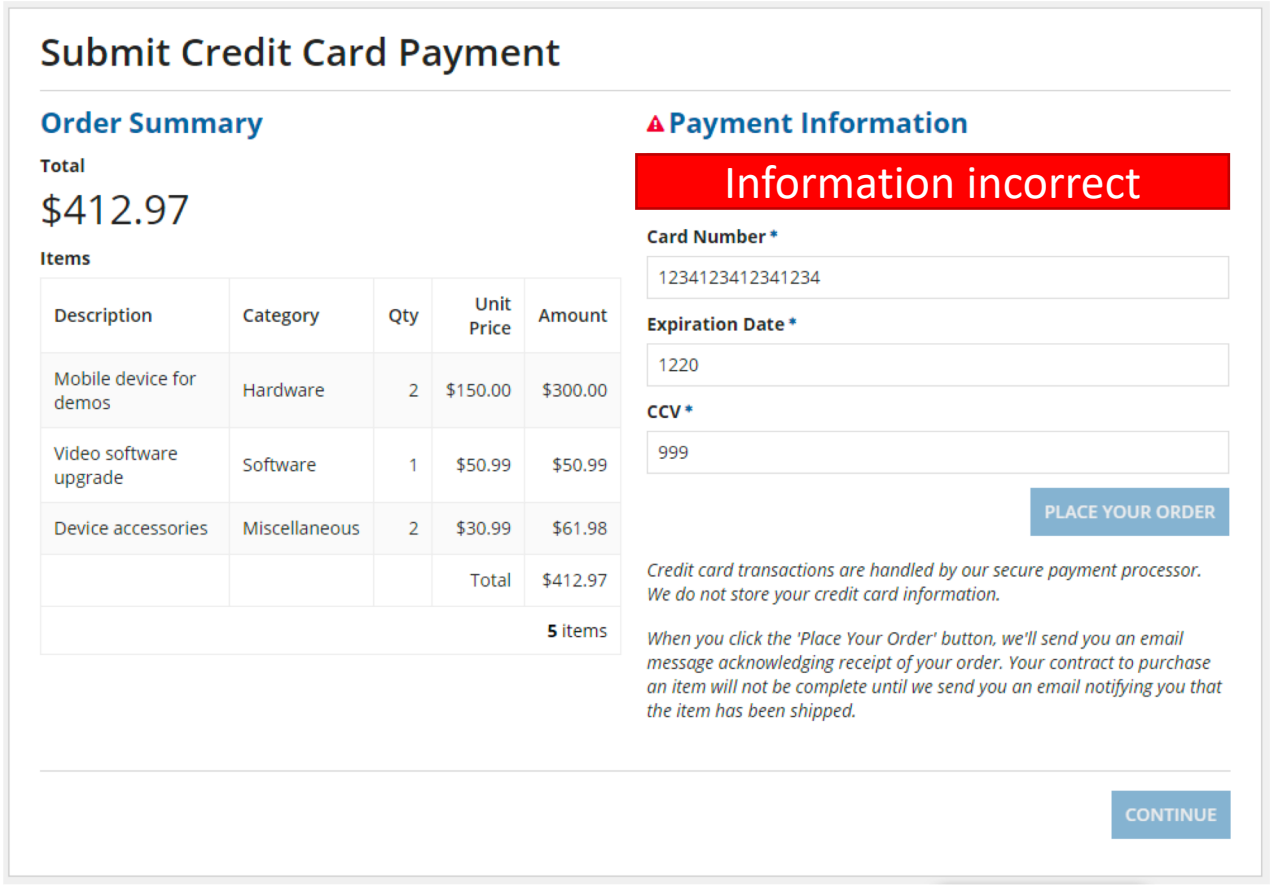
To help users **evaluate the result**,
designers must provide **feedback**.

What goes wrong when you provide no feedback?

Feeling: **Confusion.** Users wonder whether their goal has been achieved.

Action: They continue to **expend extra energy** to accomplish the goal.

What goes wrong when you provide too little feedback?



Feeling: **Frustration**. Users **don't know what** went wrong.

Action: They continue to **expend energy** to **figure out what to do**.

else

What goes wrong when you provide too little feedback?

BEEP

Feeling: **Anxiety**. Users **don't know if the feedback if it's important**.

Action: They continue to **expend energy** to **figure out if it's important**.

What goes wrong when you provide too much feedback?

I am now booking your flight

I am now using Google flight search

I am now typing JFK into the departure location

I am now typing LAX into the arrival location

I am now selecting February 26, 2018 from the departure date box

I am now confirming the date I just selected from the Departure date box

Feeling: **Annoyed** at being interrupted by irrelevant information.

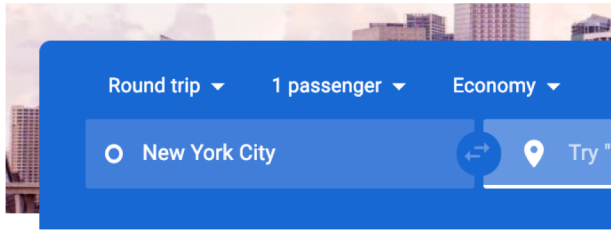
Action: They **ignore all the feedback**.

What goes wrong when feedback too late?

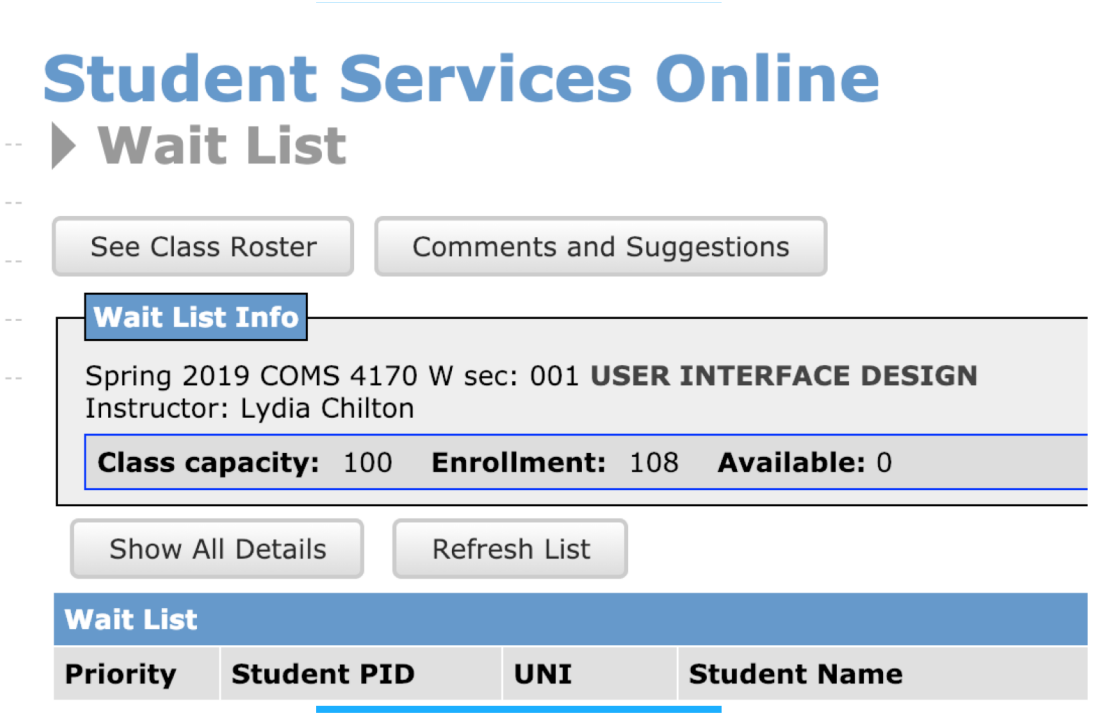
By the way, I booked that flight you asked for yesterday!

Feeling: **Uncertainty**. They're not sure anything happened.

Action: They **find another way** to reach their goal



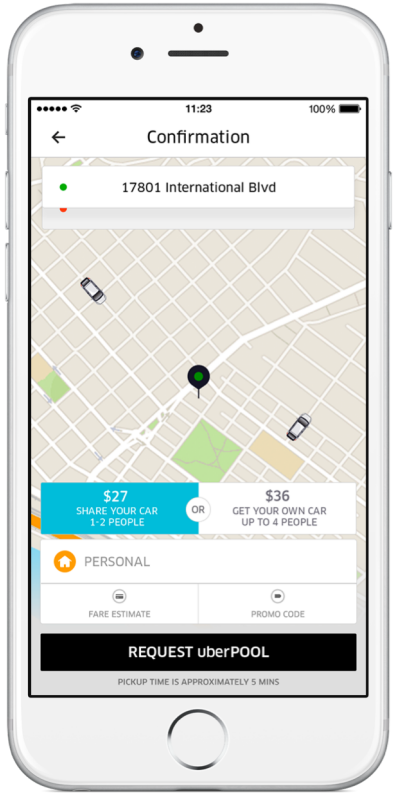
What goes wrong when feedback is not continuous?

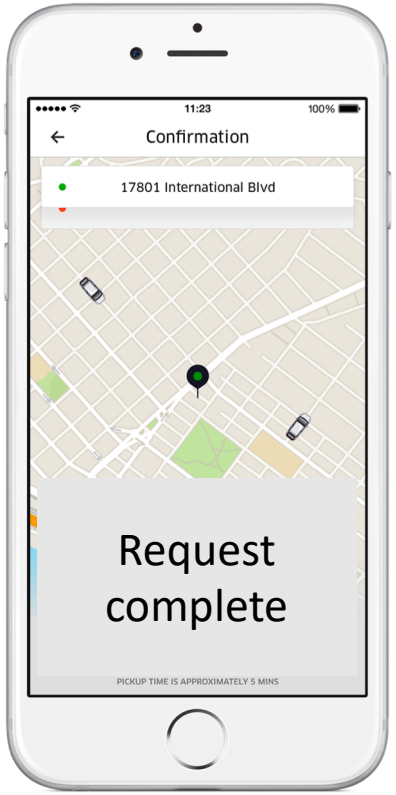


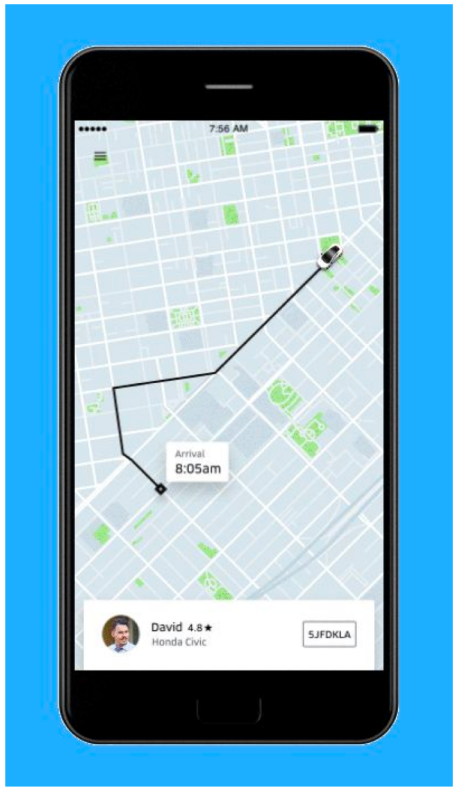
Feeling: Unsure. Users are **unsure** whether the system is doing it or not.

Action: Users have to **ask the system** for feedback frequently.

What goes wrong when feedback acknowledges the action but does not communicate the new state?







Feeling: **No closure**. Users feel like there's still more to do in the old state.

Action: Users will continue to perform actions from the previous state

Design goals for feedback:

Communicate
full and continuous information about
the results of an action and
the current state of the system

to help people achieve their goal

Ways of perceiving feedback

How do we perceive this feedback?

BEEP

I’m sorry, Dave.
I’m afraid I can’t do that.

Sound

Examples of sound feedback?

How do we perceive this feedback?

Submit Credit Card Payment

Order Summary

Total

$412.97

Items

Description	Category	Qty	Unit Price	Amount
Mobile device for demos	Hardware	2	$150.00	$300.00
Video software upgrade	Software	1	$50.99	$50.99
Device accessories	Miscellaneous	2	$30.99	$61.98
			Total	$412.97

5 items

Payment Information

Information incorrect

Card Number *

1234123412341234

Expiration Date *

1220

CCV *

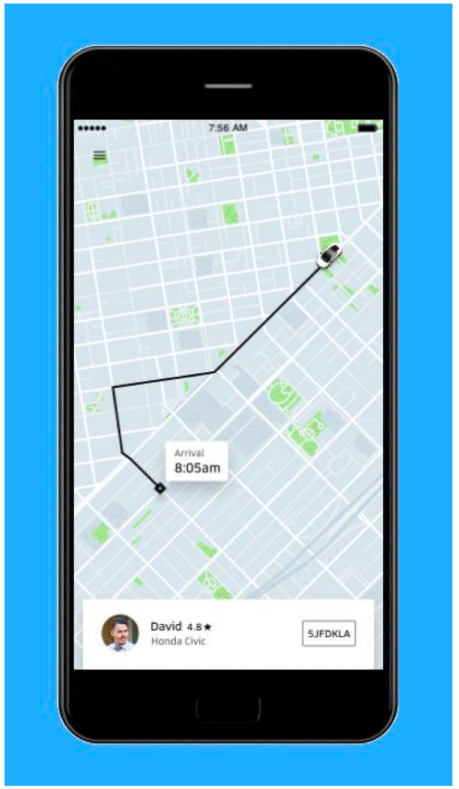
999

PLACE YOUR ORDER

Credit card transactions are handled by our secure payment processor. We do not store your credit card information.

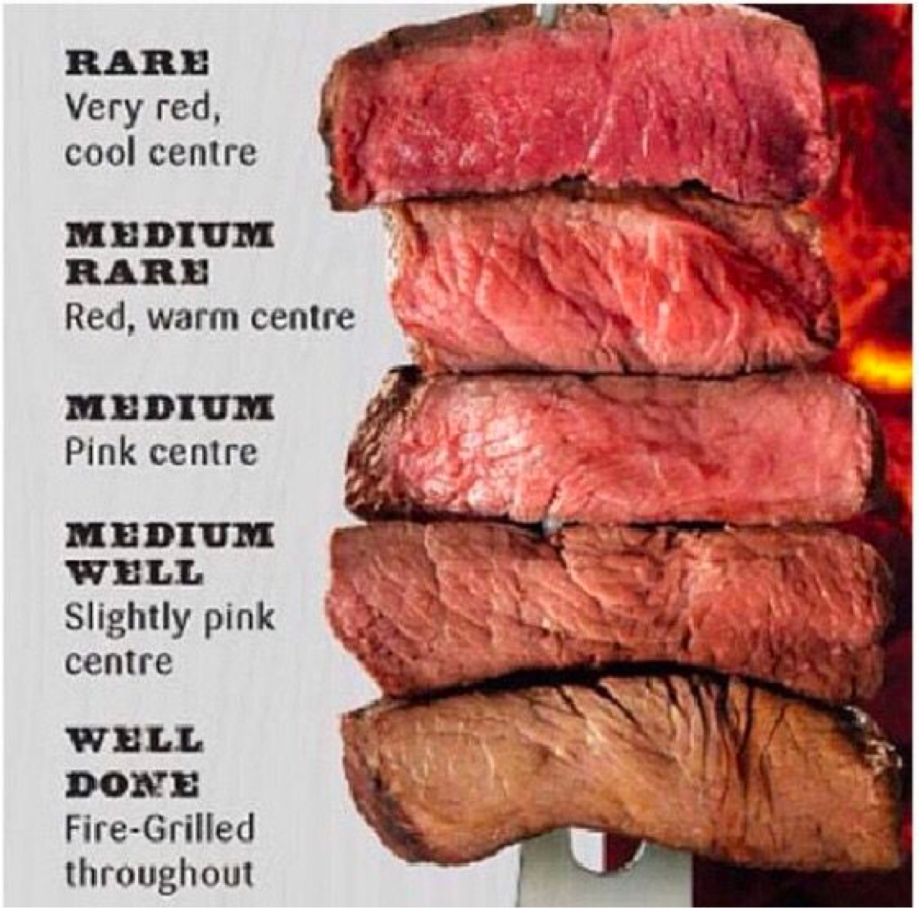
When you click the 'Place Your Order' button, we'll send you an email message acknowledging receipt of your order. Your contract to purchase an item will not be complete until we send you an email notifying you that the item has been shipped.

CONTINUE



Sight

Examples of visual feedback (non-digital)?



How do we perceive this feedback?

Smell

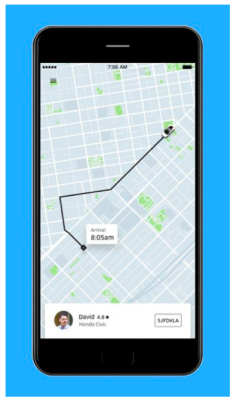
Examples of smell feedback?

How do we perceive this feedback?

Touch

Examples of haptic (touch) feedback?

The human nervous system is designed to perceive feedback in many forms.



Sight

Sound

Smell

Touch

I just gave a recap of Sesame Street

Sight

Sound

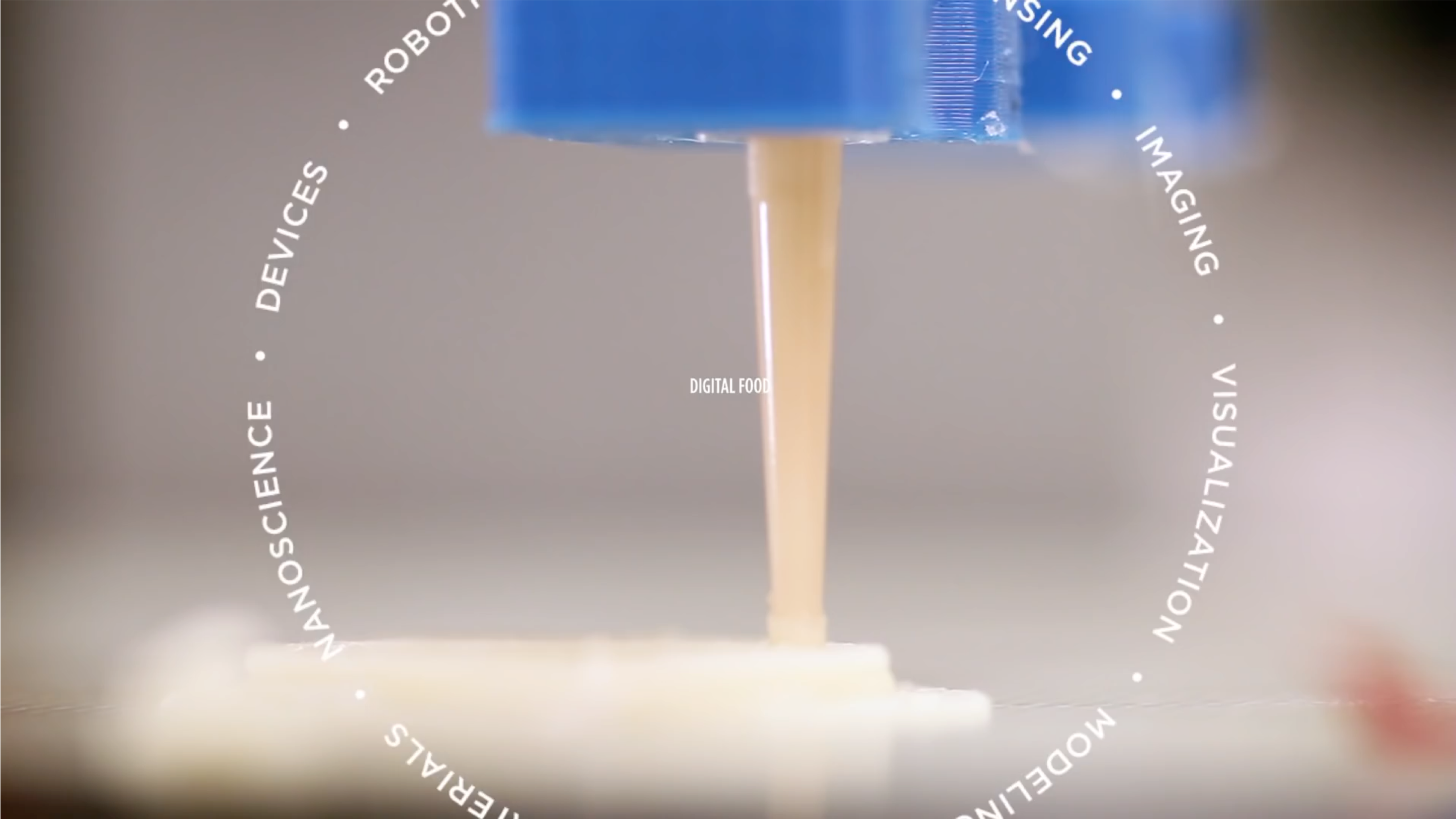
Smell

Touch

Very often, design requires us to go **back to basics**.
Basic feelings like fear and confusion.
Basic senses like sight, sound, and touch.

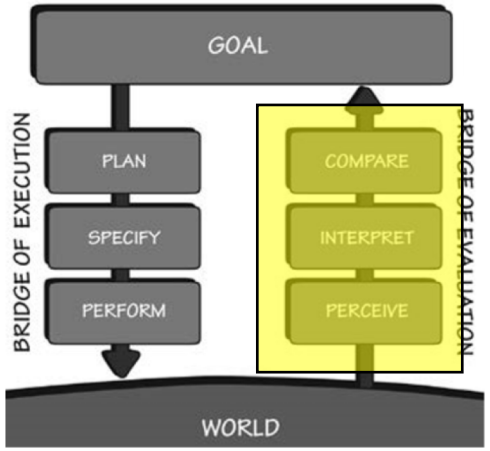
Taste

3D Printed Food?


DIGITAL FOOD
IMAGING
VISUALIZATION
NANOSCIENCE
DEVICES

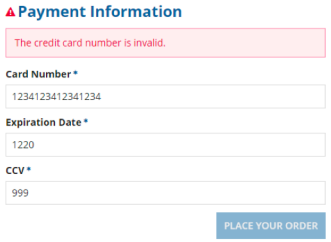
Physical Input Events and Feedback

Every time the user executes an **action**, the interface should provide **feedback**

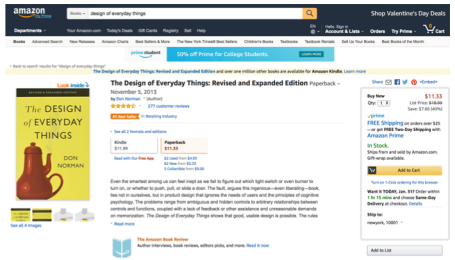


Low-level physical actions, like pressing a key

Low-level virtual actions, like clicking a button

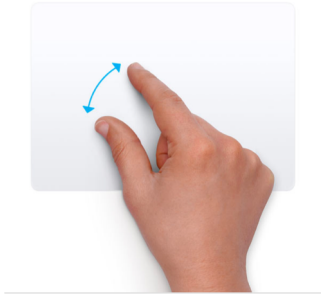


Mid-level actions, like filling out a form



High-level actions, like buying a book

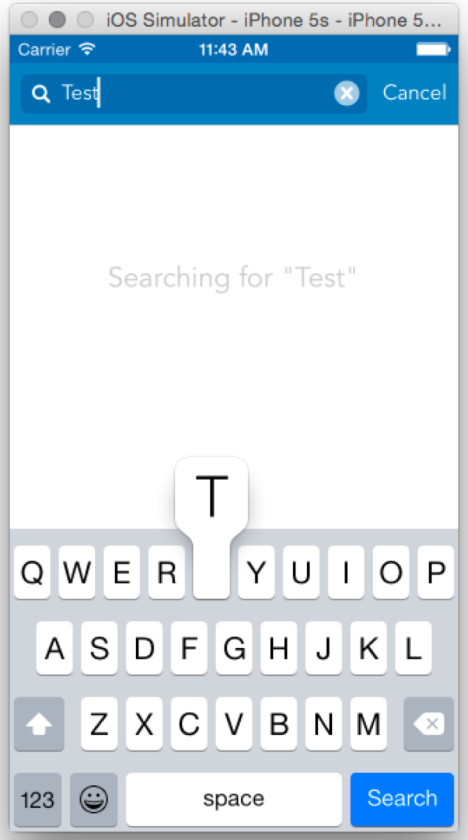
Low-level user actions are represented in the system as **events**.

Action	Event
	Keypress event
	Mousemove event Mousepress event
	Pinch gesture event

Types of hard keyboard **keypress** feedback?

haptic (can feel the key), **sound**, and **visual** (screen action)

Types of soft keyboard **keypress** feedback?



~~haptic (can feel the key)~~, **sound (simulated)**, and **visual** (screen action)

Types of hard keyboard **key**down feedback?

haptic (can feel the spring pushing back),
sound (click), and
visual (character appears or NOTHING (shift key))

Types of hard keyboard **keyup** feedback?

haptic (can feel the spring pushing up),
sound (click), and
visual (characters stop appearing or NOTHING)

Types of trackpad **Mousepress** feedback?

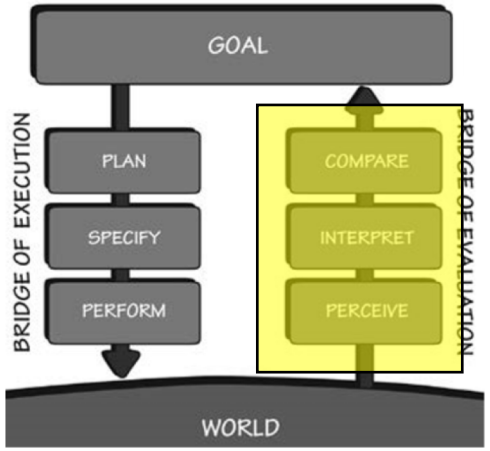
haptic (spring resistance),
sound (simulated click), and
visual (NOTHING)

Types of trackpad **Mousemove** feedback?

haptic (can feel the friction),
~~sound (no), and~~
visual (cursor moves)

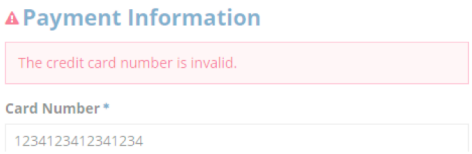
Low-level **Virtual** Feedback

Every time the user executes an action, the interface should provide feedback

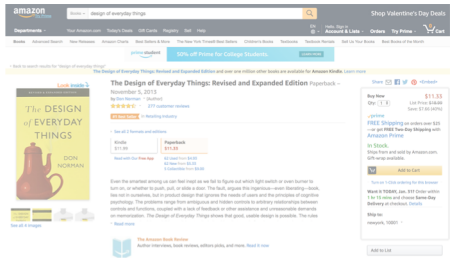


Low-level physical actions, like pressing a key

Low-level **virtual actions**, like clicking a button



Mid-level actions, like filling out a form

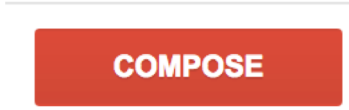


High-level actions, like buying a book

Subgoal: Press the compose button

What is the **first event** we respond to?

Normal state

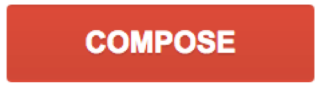



```
.T-I-KE {
  background-color: #d14836;
}
```

Subgoal: Press the compose button

What is the second **event** we respond to?

Normal state




```
.T-I-KE {
  background-color: #d14836;
}
```

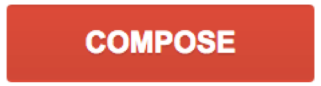
Mouseover feedback


```
.T-I-KE.T-I-JW {
  background-image: linear-gradient(to
    bottom, #dd4b39, #dd4b39);
}
```

Subgoal: Press the compose button

What is the third **event** we respond to?

Normal state




```
.T-I-KE {
  background-color: #d14836;
}
```

Mouseover feedback

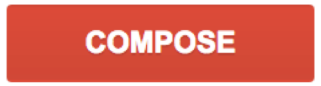

```
.T-I-KE.T-I-JW {
  background-image: linear-gradient(to
    bottom, #dd4b39, #dd4b39);
}
```

Mousedown feedback


```
background-image: linear-gradient(to
  bottom, #dd4b39, #400000);
```

What is the final **event** we respond to?

Normal state




```
.T-I-KE {
  background-color: #d14836;
}
```

Mouseover feedback


```
.T-I-KE.T-I-JW {
  background-image: linear-gradient(to
    bottom, #dd4b39, #dd4b39);
}
```

Mousedown feedback


```
background-image: linear-gradient(to
  bottom, #dd4b39, #400000);
```

Mouseup feedback


```
.T-I-KE {
  background-color: #d14836;
}
```

Implementing Low-level Feedback

How do you implement visual feedback?

Normal state


```
.T-I-KE {
   background-color: #d14836;
}
```

Mousedown


```
.T-I-KE.T-I-JW {
   background-image: linear-gradient(to
}     bottom,#dd4b39,#400000);
```

```
1  $(document).ready(function(){
2      $("#compose_button").mousedown(function(){
3
4          //?????????
5
6      })
7  })
```

1. Register an event handler on the object

2. Change the style

Can you change style like this?

Normal state

COMPOSE

```
.T-I-KE {
   background-color: #d14836;
}
```

Mousedown

COMPOSE

```
.T-I-KE.T-I-JW {
   background-image: linear-gradient(to
}     bottom,#dd4b39,#400000);
```

```
1  $(document).ready(function(){
2      $("#compose_button").mousedown(function(){
3
4          $(this).css("background-image", "linear-gradient(to bottom, #dd4b38, #400000")
5
6      })
7  })
8
```

It will work, but it's **ugly**.

This is the better way to change style. Why?

Normal state


```
.T-I-KE {
   background-color: #d14836;
}
```

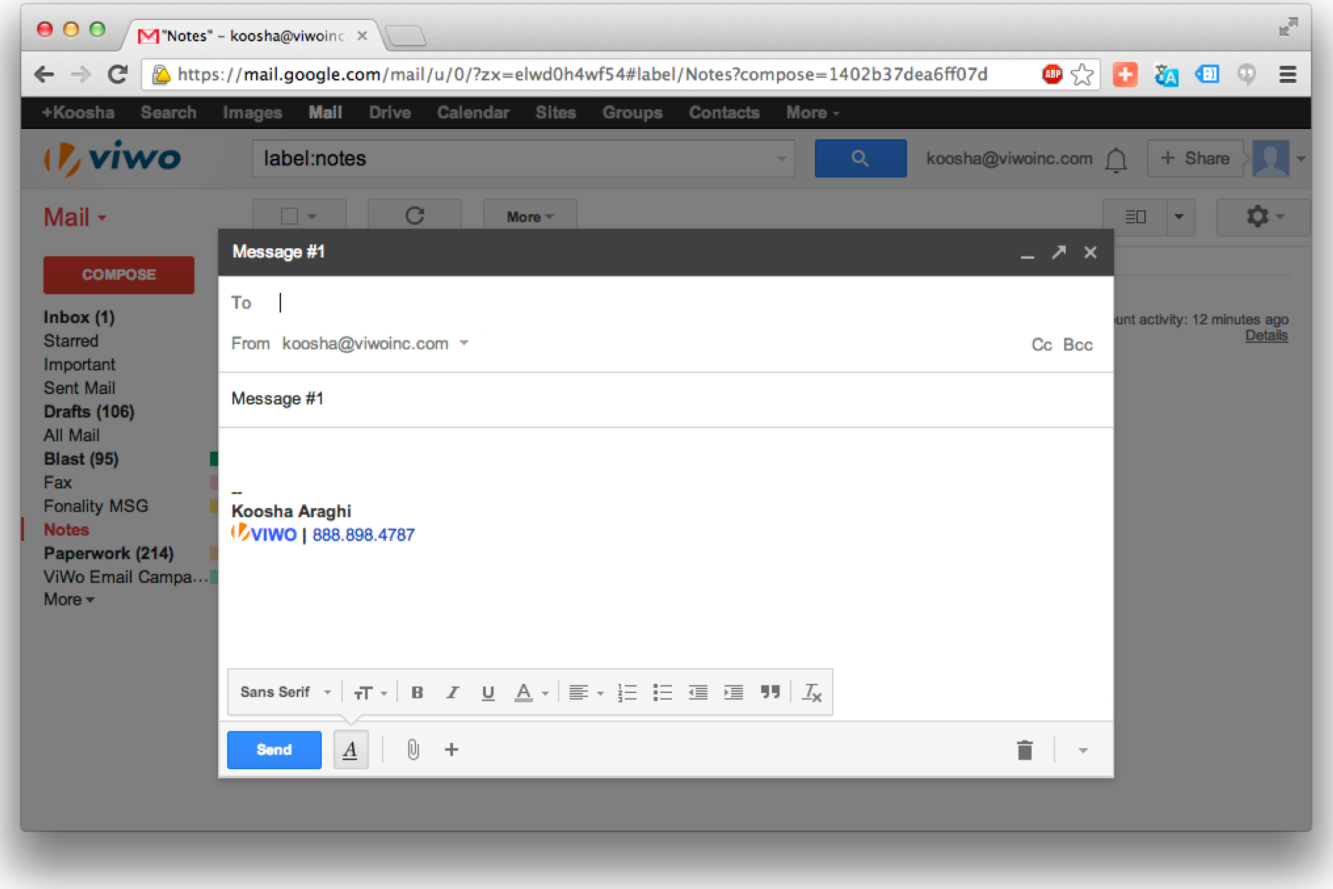
Mousedown


```
.T-I-KE.T-I-JW {
   background-image: linear-gradient(to
}     bottom, #dd4b39, #400000);
```

```
1  $(document).ready(function(){
2      $("#compose_button").mousedown(function(){
3
4          //$(this).css("background-image", "linear-gradient(to bottom, #dd4b38, #400000")
5          $(this).addClass(    "T-I-JW"     )
6
7      })
8  })
```

Classes abstract out designs to make them easier to add / remove.

After you press COMPOSE What happens?



You enter the compose email state

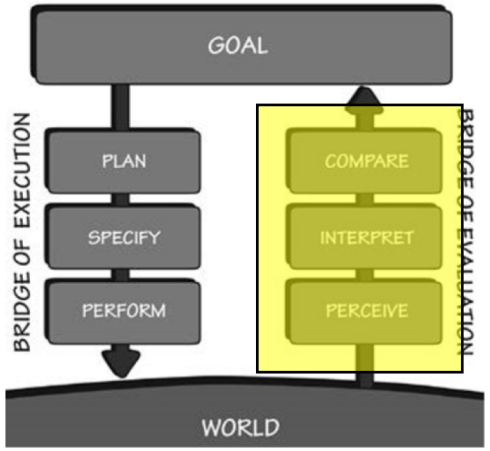
Feedback:

Communicate
full and continuous information about
the results of an action and
the current state of the system

to help people achieve their goal

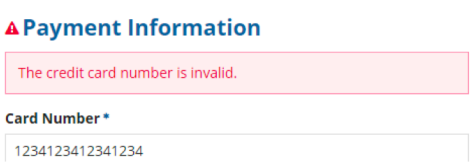
Mid-and High-level Action Feedback

Every time the user executes an action, the interface should provide feedback

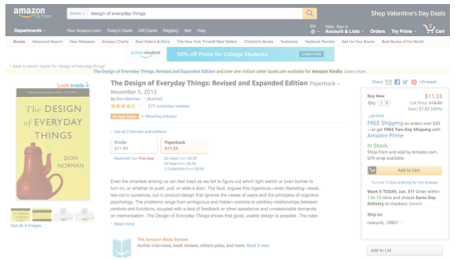


Low-level physical actions, like pressing a key

Low-level **virtual actions**, like clicking a button

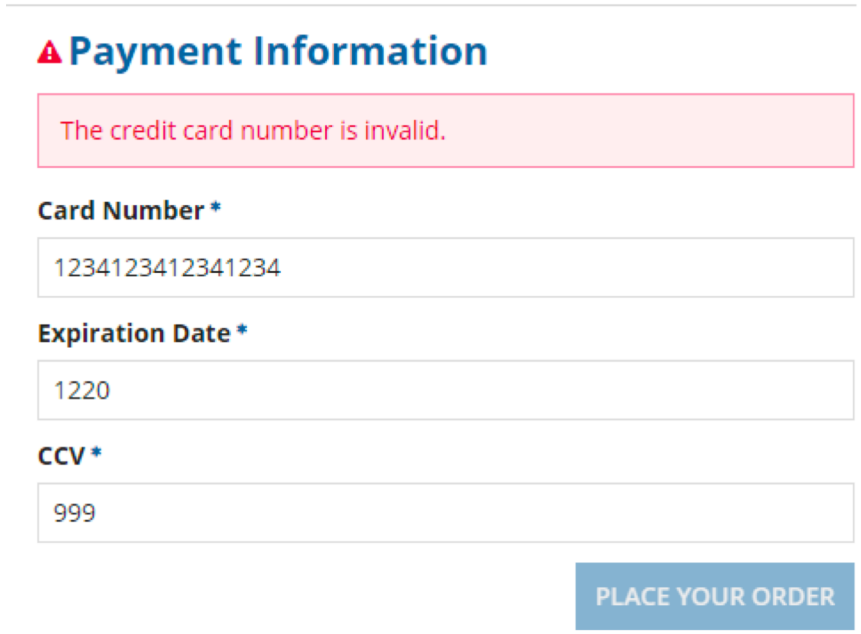


Mid-level actions, like filling out a form



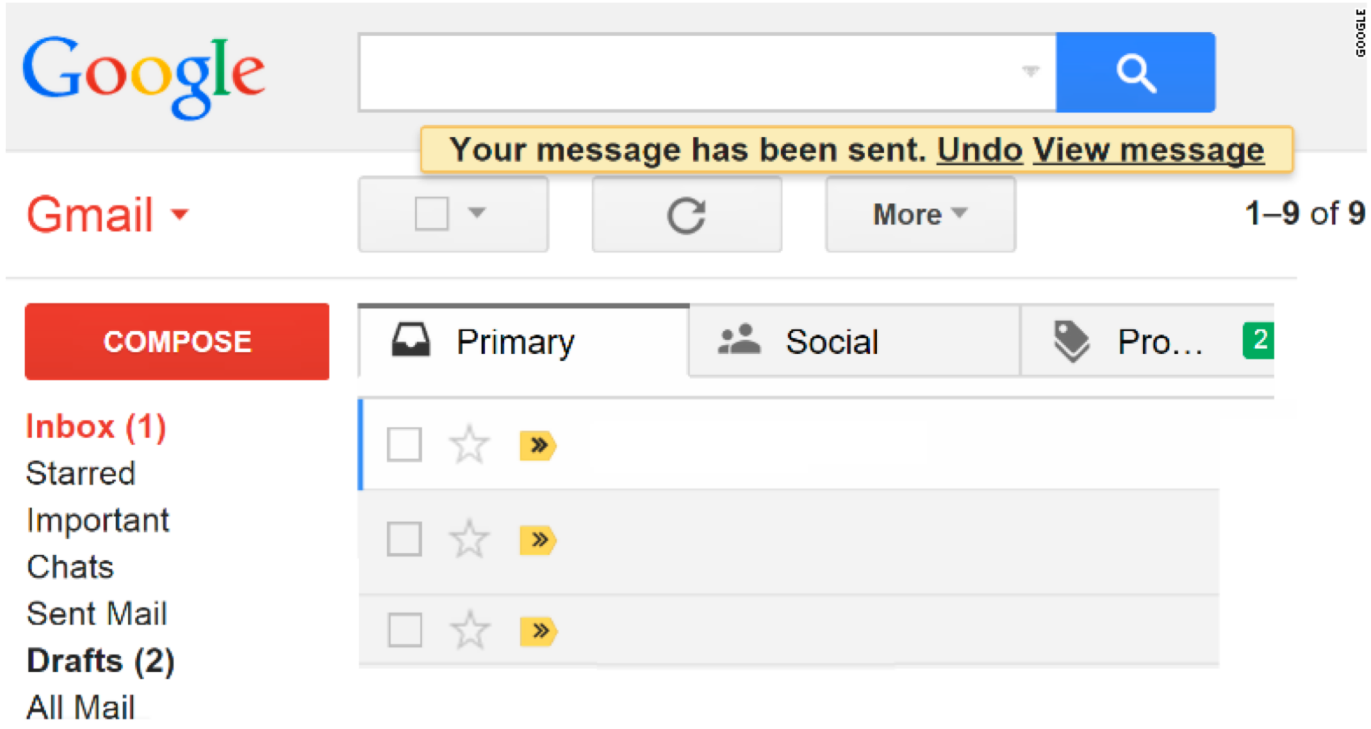
High-level actions, like buying a book

What is the new state?



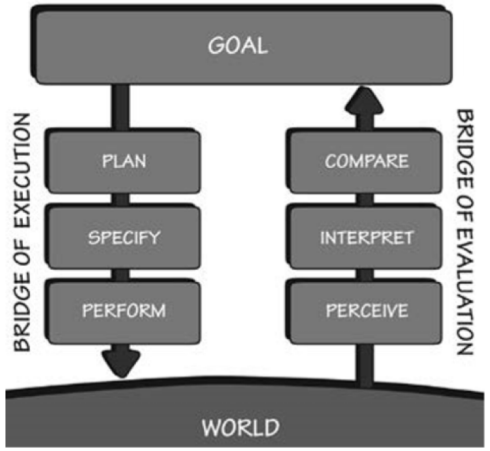
Error recovery state

What is the new state?

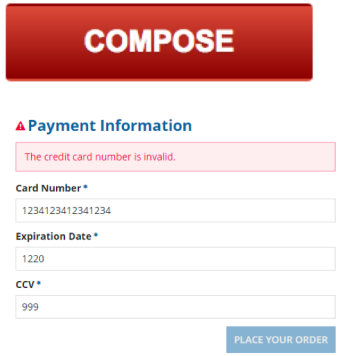


Normal inbox state

Every time the user executes an **action**, the interface should provide **feedback**

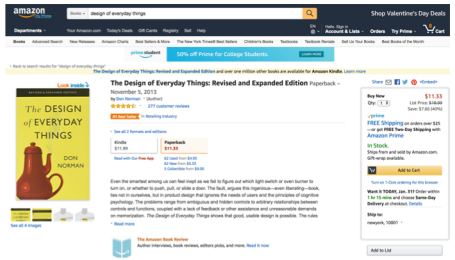


Low-level physical actions, like pressing a key



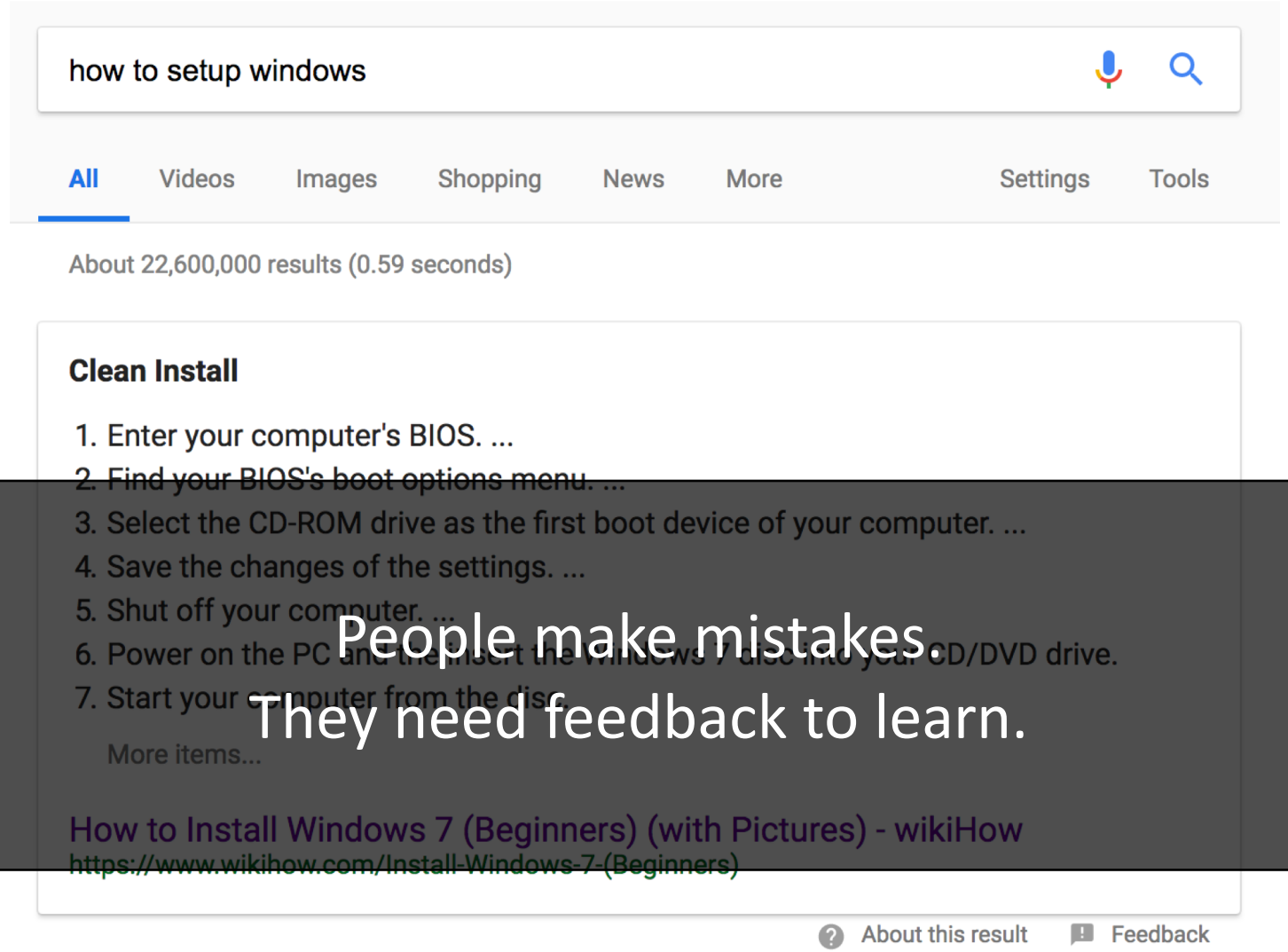
Low-level virtual actions, like clicking a button

Mid-level actions, like filling out a form



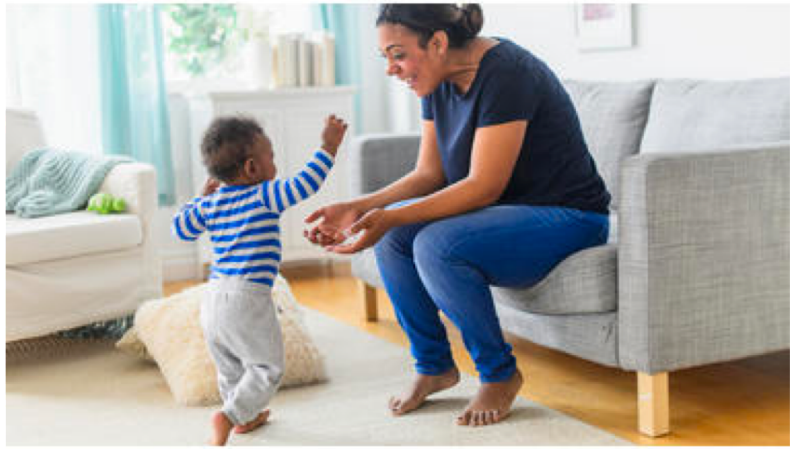
High-level actions, like buying a book

Following instructions suck. Why?



People make mistakes.
They need feedback to learn.

Feedback is essential to achieving goals



Learning to walk

Playing an instrument

```
void loop() {

float humidity = 452;
for( int counter = 0; counter < 1000; ++counter ) {
  digitalWrite(13, HIGH);   // set the LED on
  delay( 200 );             // wait for a second
  digitalWrite(13, LOW);    // set the LED offer
  Serial.pirnt( "Hello " );
  Serial.println( counter );
  delay(200);               // wait for a second
  humidity += 0.5;
}
```

100 %

Error List

2 Errors | 1 Warning | 0 Messages

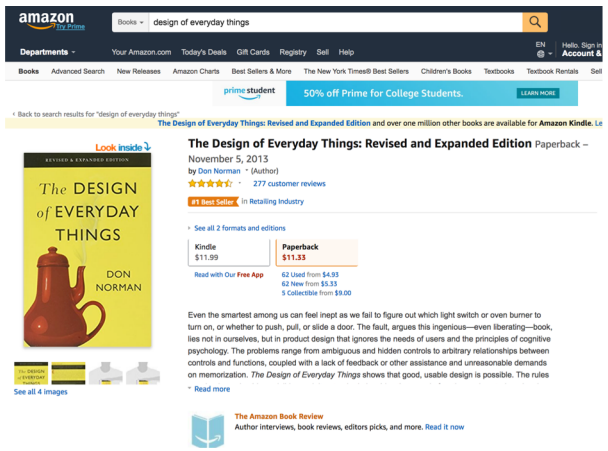
	Description
1	IntelliSense: class "Serial_" has no member "pirnt"
2	In function 'void loop()'
3	23:10: error: 'class Serial_' has no member named 'pirnt'

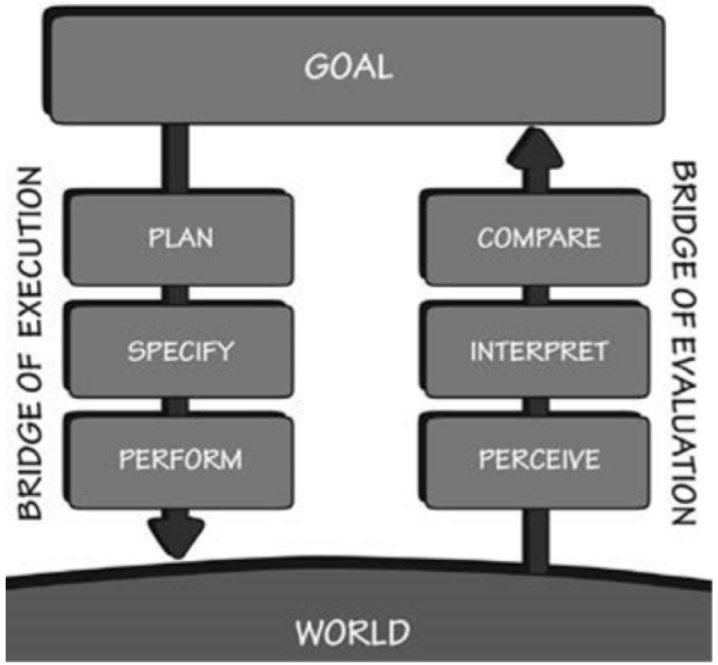
Code

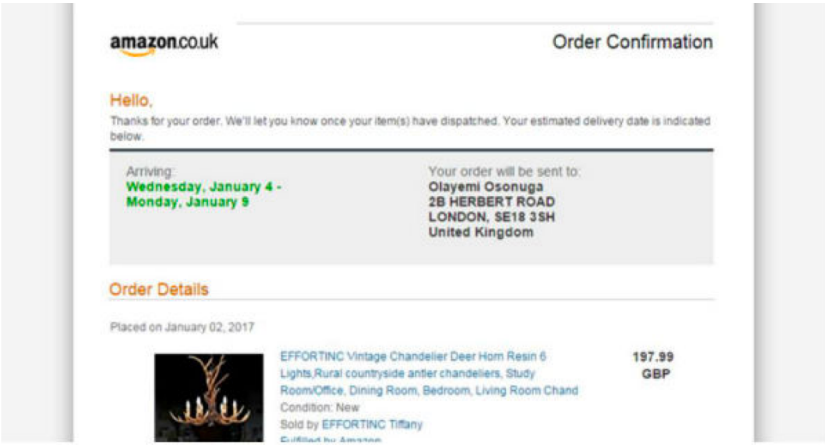
Feedback is how we learn

Summary

To accomplish a goal, users must
execute an operation and **evaluate** the result

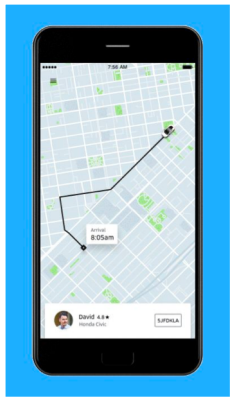






To help users **evaluate the result**,
designers must provide **feedback**.

The human nervous system is designed to perceive feedback in many forms.



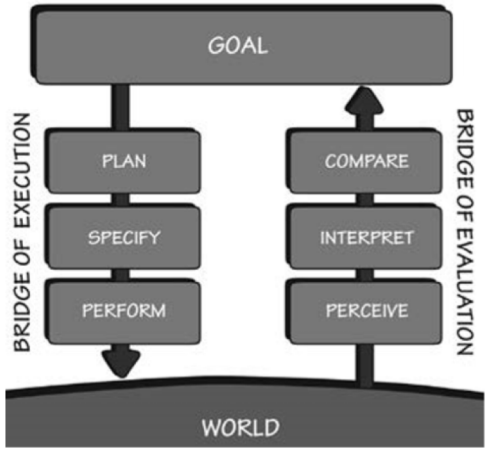
Sight

Sound

Smell

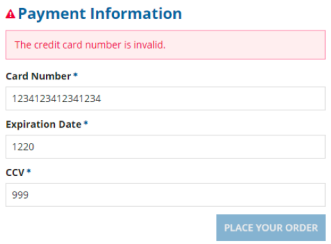
Touch

Every time the user executes an **action**, the interface should provide **feedback**

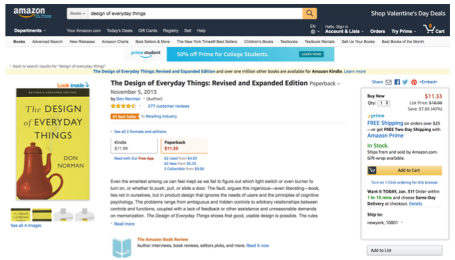


Low-level physical actions, like pressing a key

Low-level virtual actions, like clicking a button



Mid-level actions, like filling out a form



High-level actions, like buying a book

Adding/Removing CSS Classes are a modular way to implement visual feedback.

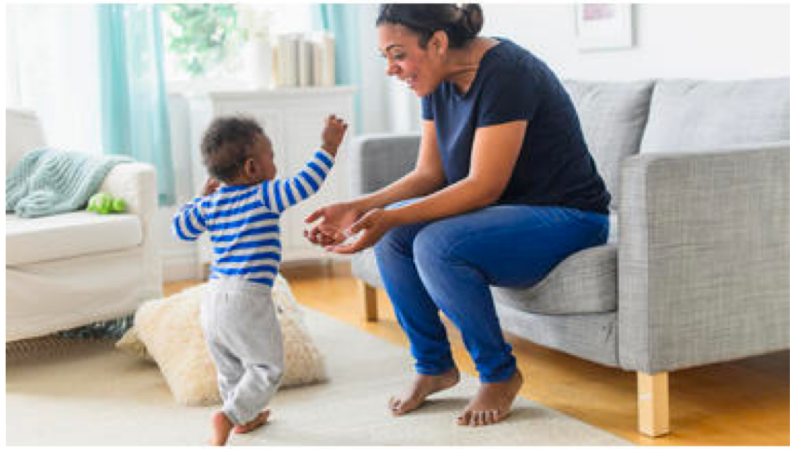

```
$(document).ready(function(){
    $("#compose_button").mousedown(function(){

        //$(this).css("background-image", "linear-gradient(to bottom, #dd4b38, #400000")
        $(this).addClass("compose_press_state")

    })
})
```

Normal state	Mouseover	Mousedown	Mouseup	Normal state
COMPOSE	COMPOSE	COMPOSE	COMPOSE	COMPOSE

Feedback is essential to achieving goals



Learning to walk

Playing an instrument

```
void loop() {

float humidity = 452;
for( int counter = 0; counter < 1000; ++counter ) {
  digitalWrite(13, HIGH);   // set the LED on
  delay( 200 );             // wait for a second
  digitalWrite(13, LOW);    // set the LED offer
  Serial.pirnt( "Hello " );
  Serial.println( counter );
  delay(200);               // wait for a second
  humidity += 0.5;
}
```

100 %

Error List

2 Errors | 1 Warning | 0 Messages

	Description
1	IntelliSense: class "Serial_" has no member "pirnt"
2	In function 'void loop()'
3	23:10: error: 'class Serial_' has no member named 'pirnt'

Code

Feedback is how we learn

Fill out participation now!
HW 3 is out.