Building Interactive Systems

Direct Manipulation

Professor Bilge Mutlu | Spring 2023

What will we cover today?

→ History and emergence of direct manipulation

→ Direct manipulation today

→ Touch interfaces

→ Implications for interactive system design

→ HACK 3, INTEGRATE Milestone 2 Q&A

History of Direct Manipulation

1

[1] YouTube

What did we watch?[3]

Ivan Sutherland's 1963 MIT PhD dissertation: *Sketchpad*.[2]

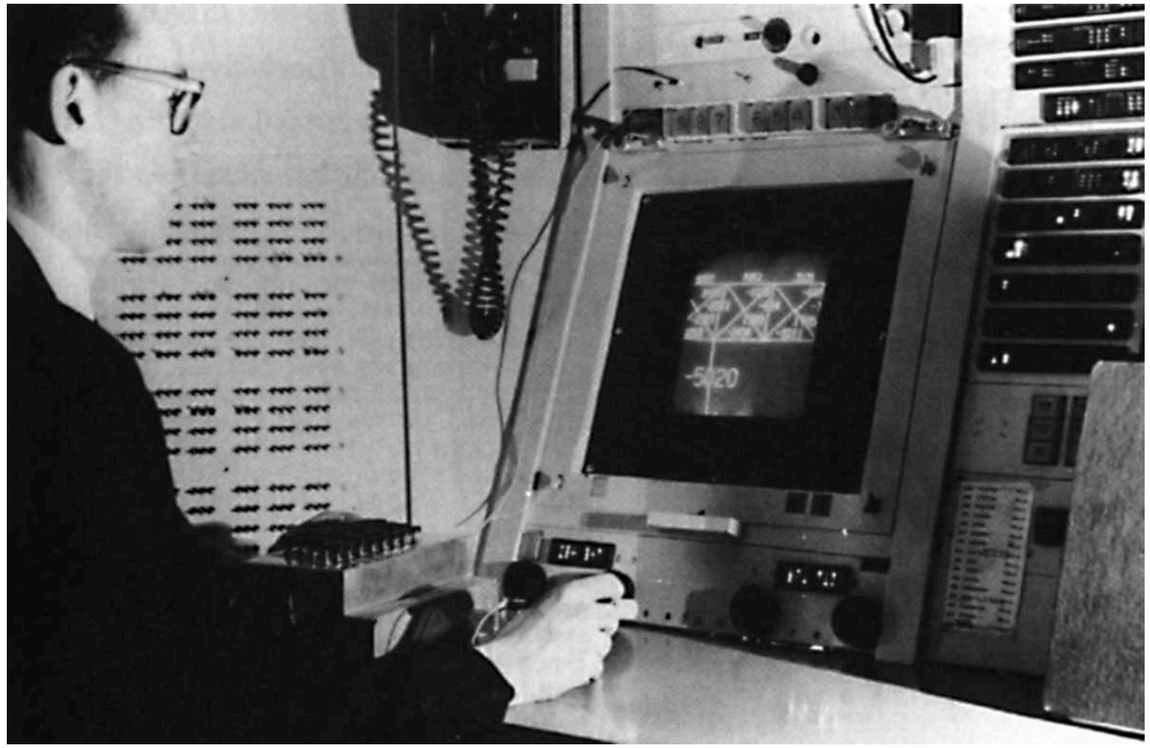
→ Direct manipulation of graphics

→ Widgets with controllers

→ Selection by pointing

→ Gesture recognition

→ Many more...



[3] Shneiderman (1982). The future of interactive systems and the emergence of direct manipulation. *Behaviour & Information Technology*.

[2] Sutherland (1964). Sketch pad a man-machine graphical communication system. In *Proceedings of the SHARE design automation workshop*.

Definitions

Direct Manipulation: … [a]n interaction style in which the objects of interest in the UI are visible and can be acted upon via physical, reversible, incremental actions that receive immediate feedback.[6]

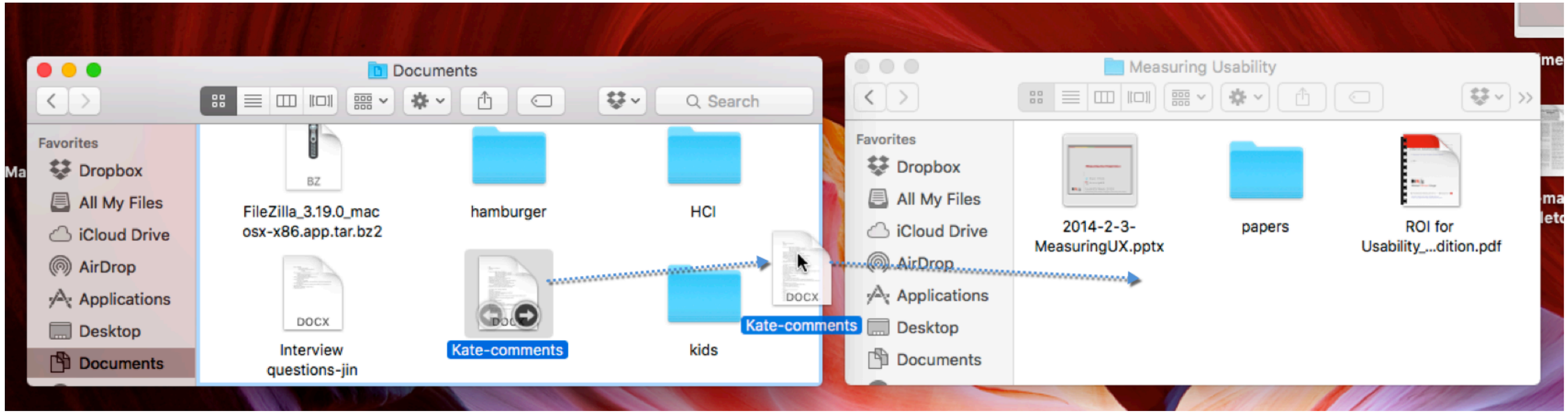
Style of interaction with the following properties:[3]

1. Continuous representation of the object of interest (instead of hidden affordances);
2. Physical actions or labelled button presses (instead of complex syntax);
3. Rapid incremental reversible operations immediately visible in objects of interest.

[6] Direct Manipulation: NN/g

[3] Shneiderman (1982). The future of interactive systems and the emergence of direct manipulation. *Behaviour & Information Technology*.

```
raluca — -bash — 94×6
Last login: Thu Aug 18 16:29:22 on ttys002
Ralucas-MacBook-Air:~ raluca$ mv Documents/Kate-comments Documents/Measuring\ Usability/
```



[7] Images

Timeline of Development[4]

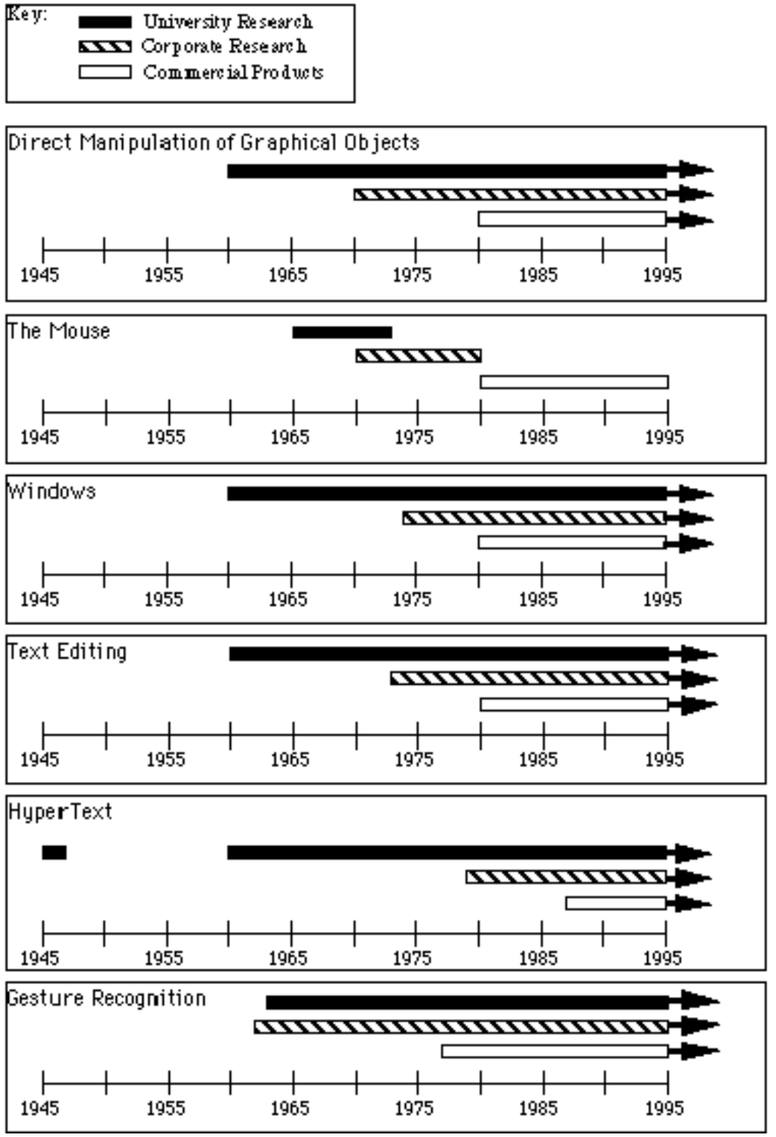
→ Sketchpad by Ivan Sutherland (MIT) in the early 60s

→ Pygmalion, which introduced *icons*, by David Canfield Smith (Stanford) in the early 70s

→ Development of Xerox Star, integrating direct manipulation, WYSIWYG, in the 70s

→ Dynabook by Alan Kay in 1977

→ Direct manipulation in Xerox Star (1981), Apple Lisa (1982), Macintosh (1984)



[4] Myers (1998). A brief history of human-computer interaction technology. *interactions*.

The Mouse

The invention of *the mouse* was essential for widespread use and adoption of direct manipulation interfaces.[4]

→ Envisioned as a cheap replacement for light pens

→ Developed at Stanford Research Laboratory (now SRI) in 1965 (project called NLS)

→ Demoed by Douglas Engelbart in the "mother of all demos" in 1968

[4] Myers (1998). A brief history of human-computer interaction technology. *interactions*.

"The Mother of All Demos"

[5]

[5] YouTube

Windows[4]

Interface elements become fully manipulable with the introduction of *multiple application windows*.

→ NLS demoed multiple tiled windows in 1968

→ Stanford Copilot, MIT Emacs used tiled windows in 1974

→ Overlapping windows in Alan Kay's University of Utah PhD thesis in 1969

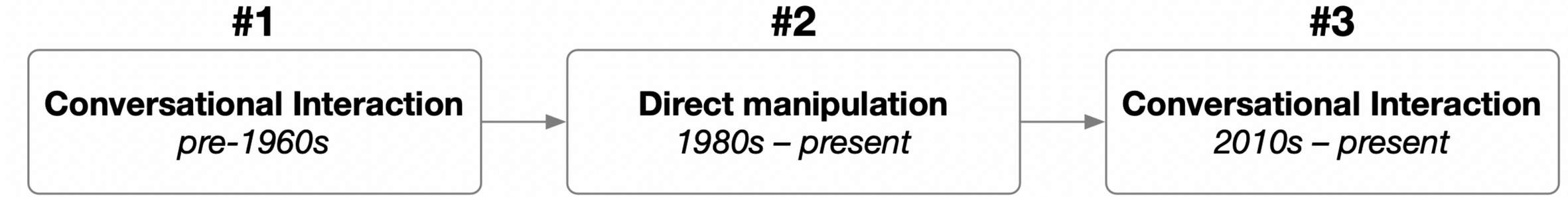
→ Appeared in the Smalltalk (1974), Xerox Star (1981) by Xerox PARC

→ Andrew window manager (1983) at CMU

→ X Window (current standard) was developed at MIT in 1984

[4] Myers (1998). A brief history of human-computer interaction technology. *interactions*.

An Observation



1. Command-based interfaces where the user had to speak the computer's language
2. World-model interfaces where the user manipulated graphical representations
3. Conversational interfaces where the computer speaks the user's language

Direct Manipulation Today

How is Direct Manipulation Implemented?

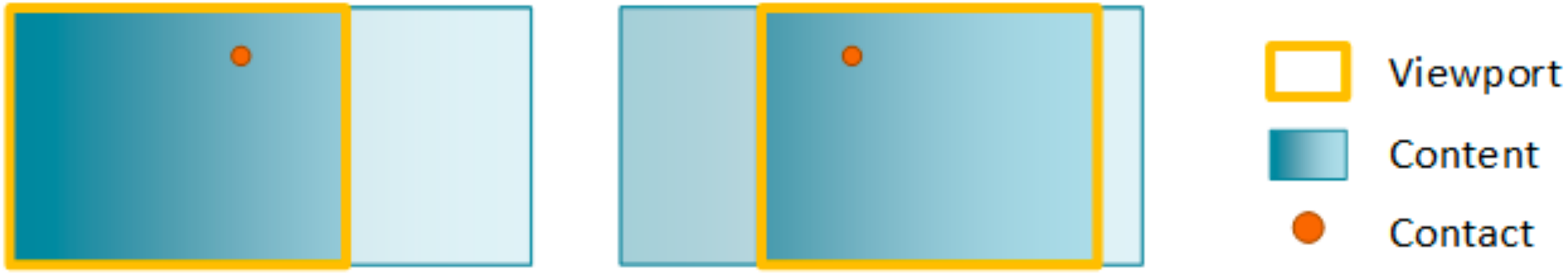
- → Governed by international standards
  - → **ISO 9241-16:1999(en)** Ergonomic requirements for office work with visual display terminals (VDTs) — Part 16: Direct manipulation dialogues[8]
- → Implemented differently across platforms

[8] ISO 9241-16:1999(en)

Example platform: *Microsoft Windows*[9]

→ Core components: *viewport*, *content*, *contact*

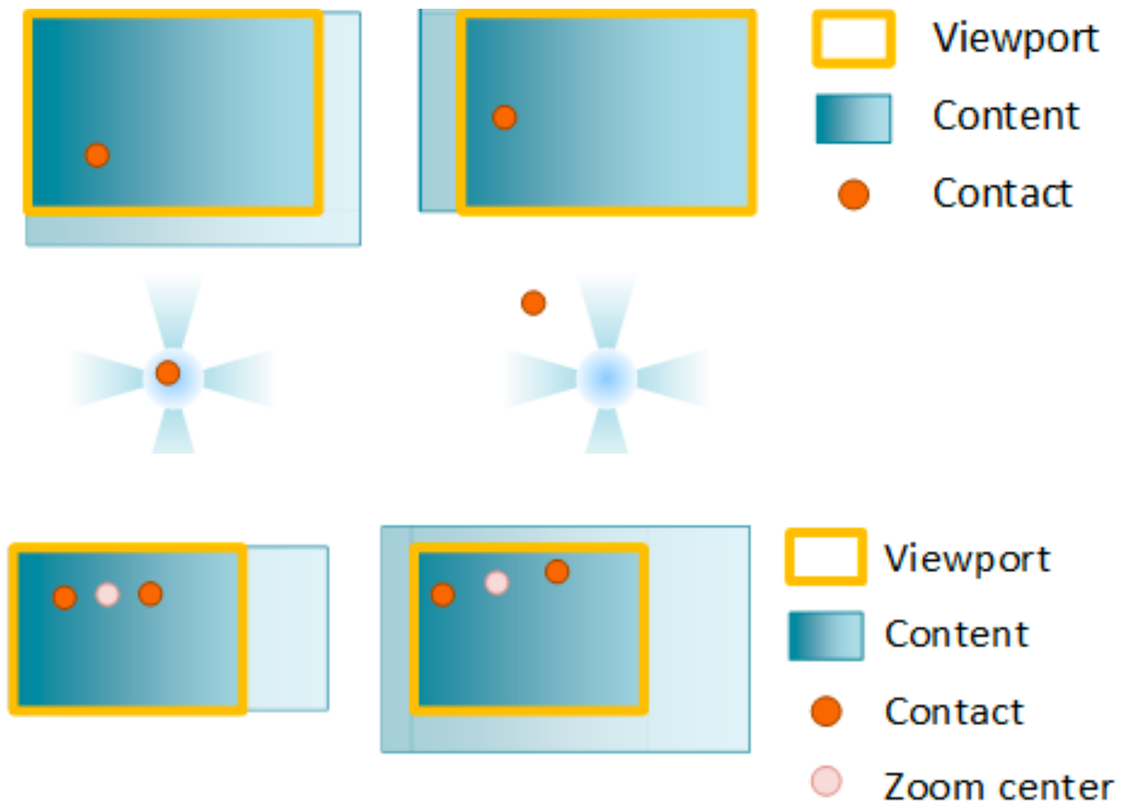
→ Key operations: *pan*, *zoom*



[9] Direct Manipulation: Microsoft Windows

Interaction

Interaction is facilitated by rails, zoom center, intertia, snap points, boundaries, behaviors, chaining, etc.[10]



[10] Using the HTML5 Drag and Drop API

Another Example: HTML5 Drag & Drop[10]

- → Core components: `draggable` objects, drag events, drag sequences, drop handling
- → Containers are enabled for direction manipulation with `draggable="true"`
- → Simple drag-and-drop application[11]

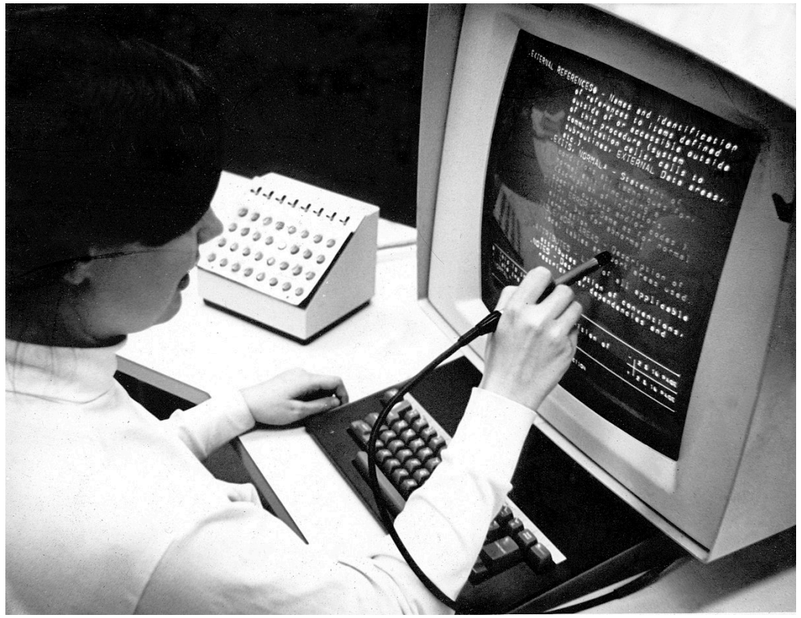

```
<div class="container">
  <div draggable="true" class="box">A</div>
  <div draggable="true" class="box">B</div>
  <div draggable="true" class="box">C</div>
</div>
```

[10] Using the HTML5 Drag and Drop API

[11] Simple drag and drop

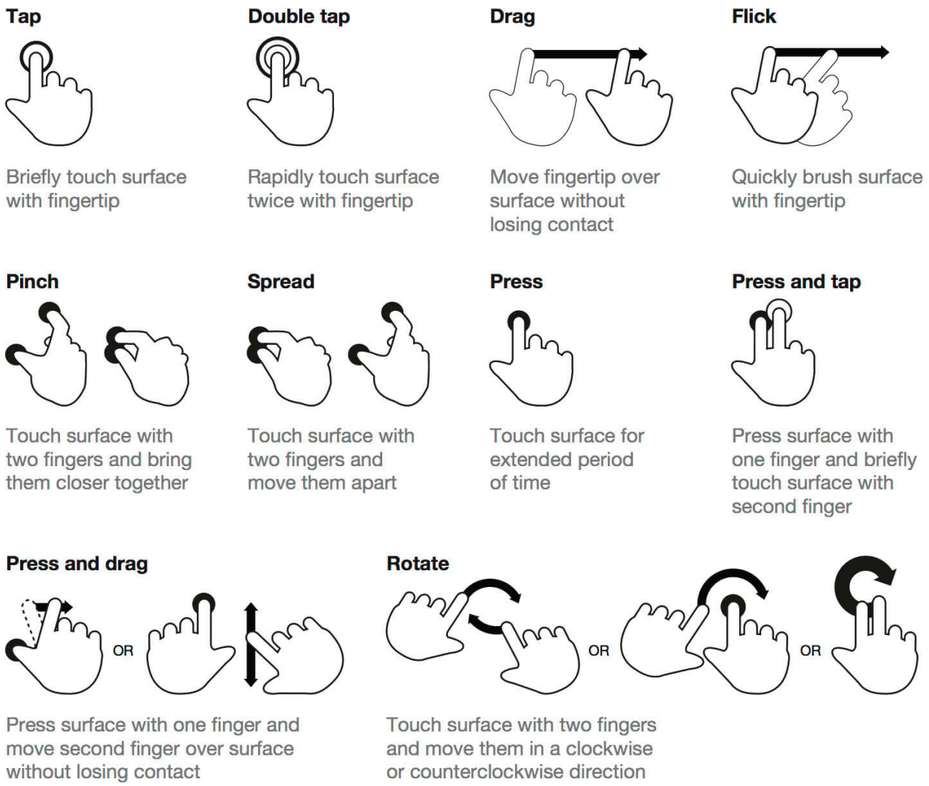
Touch Interfaces

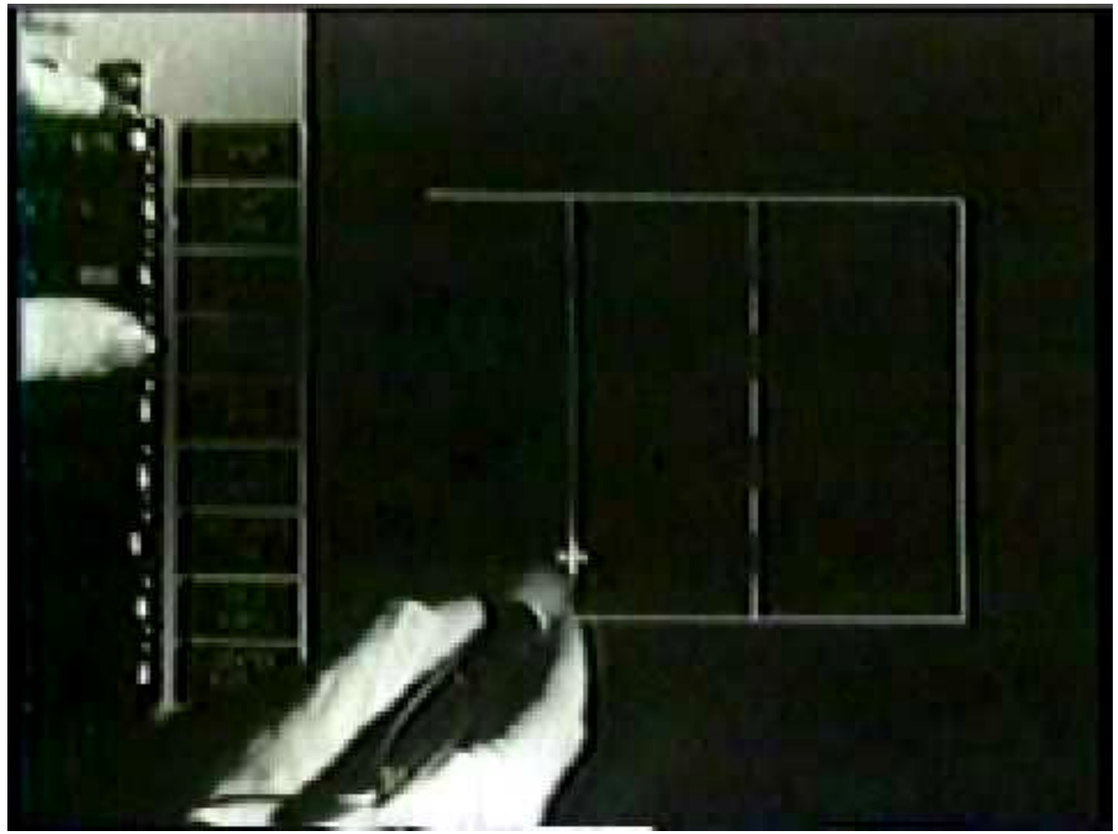
Methods of User Input[12]



[12] Images: left, center, right

Touch Gestures[13] [14]





[13] Touch Gesture Controls

[14] Sutherland's Sketchpad w/ comments by Alan Kay

How is Touch Implemented? Example Platform: React Native[15]

Components: `TouchableHighlight`, `TouchableOpacity`, `TouchableNativeFeedback` `TouchableWithoutFeedback`

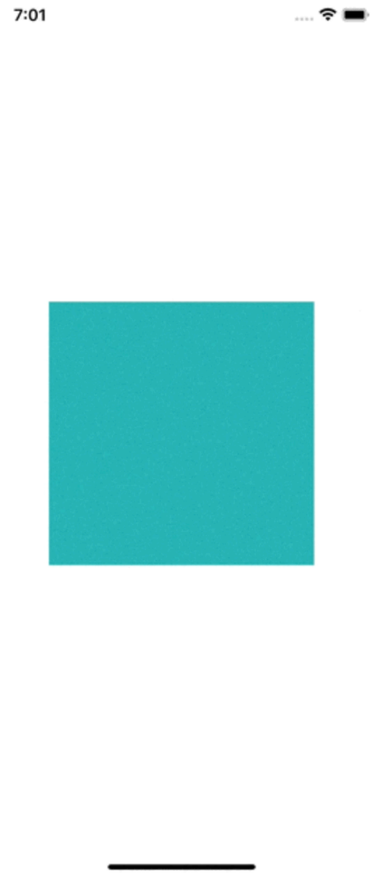
```
render() {
  return (
    <View style={styles.container}>
      <TouchableHighlight onPress={this._onPressButton} underlayColor="white">
        <View style={styles.button}>
          <Text style={styles.buttonText}>TouchableHighlight</Text>
        </View>
      </TouchableHighlight>
        ...
```

[15] Handling Touches, Expo Snack

Implementing Gestures[16]

Handlers: `PanGestureHandler`, `TapGestureHandler`, `LongPressGestureHandler`, `PinchGestureHandler`

```
render() {
    return (
      <PinchGestureHandler
        onGestureEvent
            ={this.onPinchGestureEvent}
        onHandlerStateChange
            ={this.onPinchHandlerStateChange}
      >
        <Animated.View
          style={[
            styles.pinchableImage,
            {
              transform: [{ perspective: 1 },
              { scale: this.scale }],
            },
          ]}
        ></Animated.View>
      </PinchGestureHandler>
    );
  }
```



[16] React Native Gesture Handler

What do these mean for interactive systems?

Example Research System: Figaro

17

[17] YouTube

HACK 3, INTEGRATE Milestone 2 Q&A

Timeline of Events

INTEGRATE Milestone 2

→ Due Wednesday: **Conceptual Design:** Define RQ, delineate contribution(s), outline plan

HACK 3

→ Due Wednesday: **Update:** Demo or present as far as you have gotten

→ Due Monday: **Final deliverable:** Demo video, report

LEARN

→ Lighter reading load