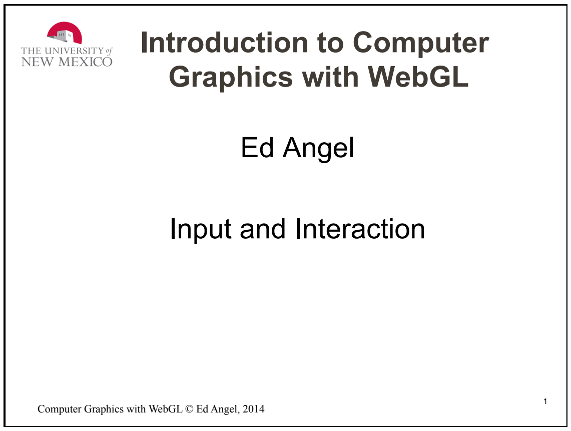


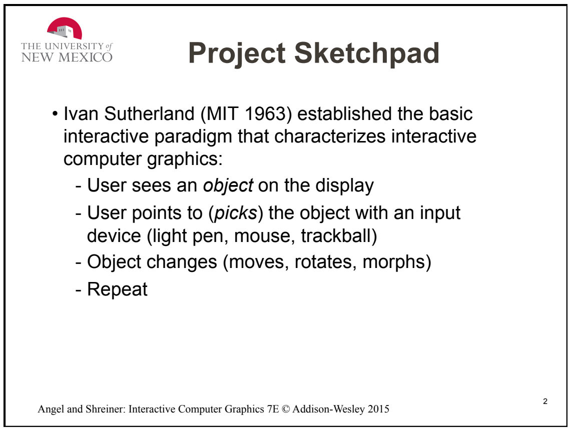
Introduction to Computer Graphics with WebGL

Ed Angel

Input and Interaction

Computer Graphics with WebGL © Ed Angel, 2014

1

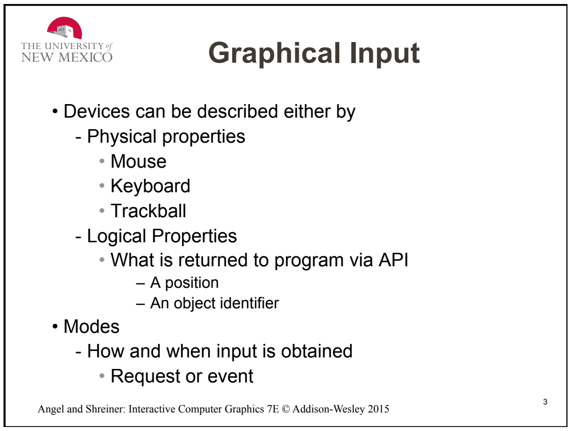


Project Sketchpad

- Ivan Sutherland (MIT 1963) established the basic interactive paradigm that characterizes interactive computer graphics:
  - User sees an *object* on the display
  - User points to (*picks*) the object with an input device (light pen, mouse, trackball)
  - Object changes (moves, rotates, morphs)
  - Repeat

Angel and Shreiner: Interactive Computer Graphics 7E © Addison-Wesley 2015

2

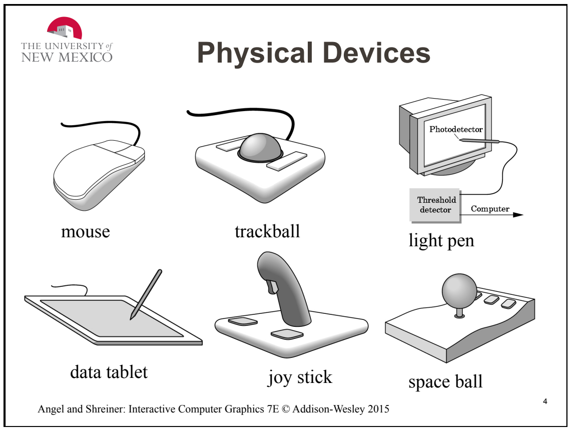


Graphical Input

- Devices can be described either by
  - Physical properties
    - Mouse
    - Keyboard
    - Trackball
  - Logical Properties
    - What is returned to program via API
      - A position
      - An object identifier
- Modes
  - How and when input is obtained
    - Request or event

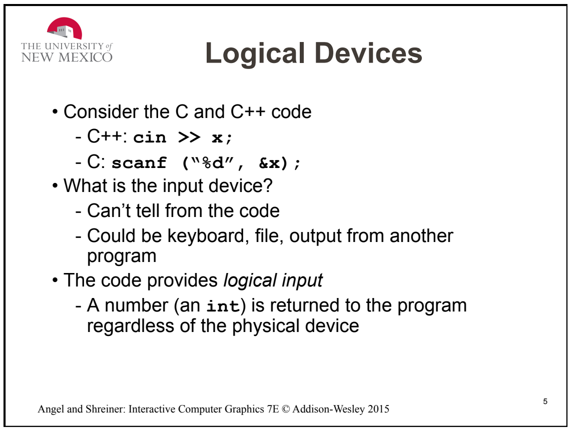
Angel and Shreiner: Interactive Computer Graphics 7E © Addison-Wesley 2015

3



Physical Devices

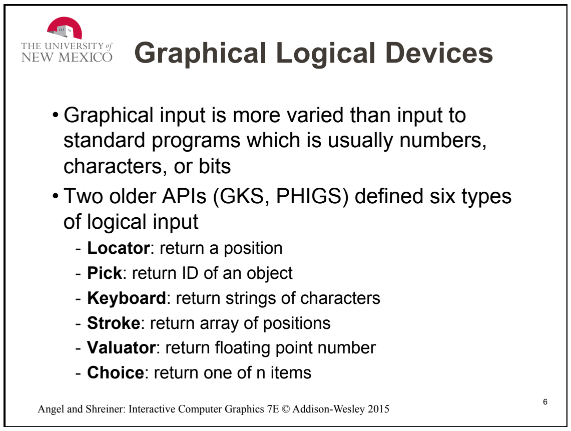
Angel and Shreiner: Interactive Computer Graphics 7E © Addison-Wesley 2015 4



Logical Devices

- Consider the C and C++ code
  - C++: `cin >> x;`
  - C: `scanf ("%d", &x);`
- What is the input device?
  - Can't tell from the code
  - Could be keyboard, file, output from another program
- The code provides *logical input*
  - A number (an `int`) is returned to the program regardless of the physical device

Angel and Shreiner: Interactive Computer Graphics 7E © Addison-Wesley 2015 5



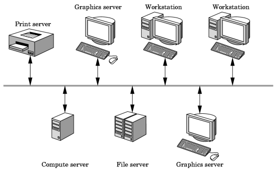
Graphical Logical Devices

- Graphical input is more varied than input to standard programs which is usually numbers, characters, or bits
- Two older APIs (GKS, PHIGS) defined six types of logical input
  - **Locator**: return a position
  - **Pick**: return ID of an object
  - **Keyboard**: return strings of characters
  - **Stroke**: return array of positions
  - **Valuator**: return floating point number
  - **Choice**: return one of n items

Angel and Shreiner: Interactive Computer Graphics 7E © Addison-Wesley 2015 6

Client Server Systems

- The X Window System introduced a client-server model for a network of workstations
  - **Client**: OpenGL program
  - **Graphics Server**: bitmap display with a pointing device and a keyboard

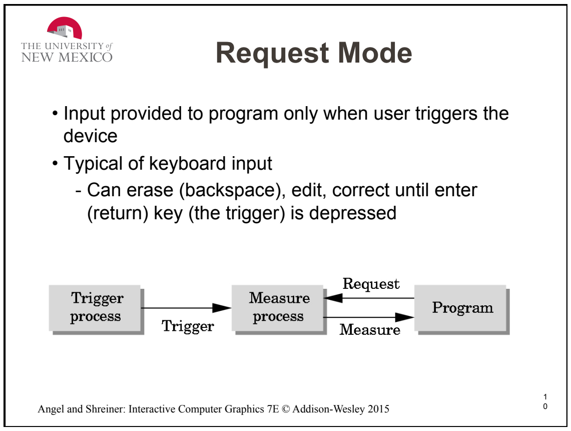


Browser Client Server

- Server: location of URL
  - can be remote or local
- Client side: anything run in your browser
- We generally write client side code
  - Client side JS
  - Client code cannot affect server side directly
  - We can send data to server but data is processed there by server side code

Input Modes

- Input devices contain a *trigger* which can be used to send a signal to the operating system
  - Button on mouse
  - Pressing or releasing a key
- When triggered, input devices return information (their *measure*) to the system
  - Mouse returns position information
  - Keyboard returns ASCII code



Request Mode

- Input provided to program only when user triggers the device
- Typical of keyboard input
  - Can erase (backspace), edit, correct until enter (return) key (the trigger) is depressed

Trigger process

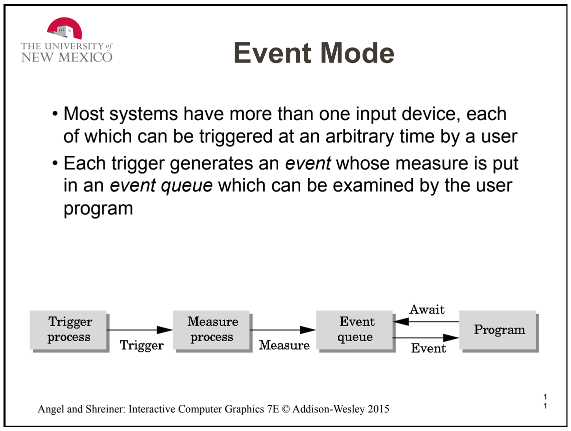
Trigger

Measure process

Request

Measure

Program



Event Mode

- Most systems have more than one input device, each of which can be triggered at an arbitrary time by a user
- Each trigger generates an *event* whose measure is put in an *event queue* which can be examined by the user program

Trigger process

Trigger

Measure process

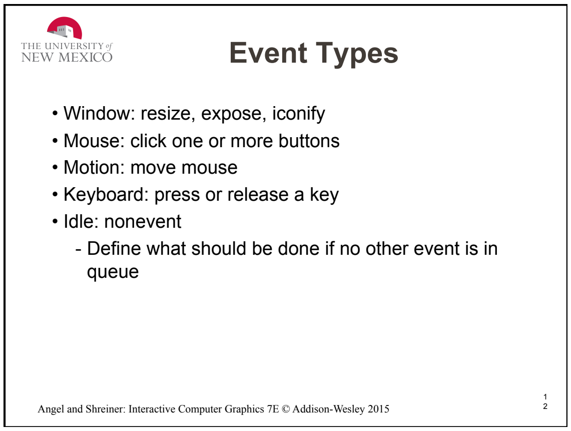
Measure

Event queue

Await

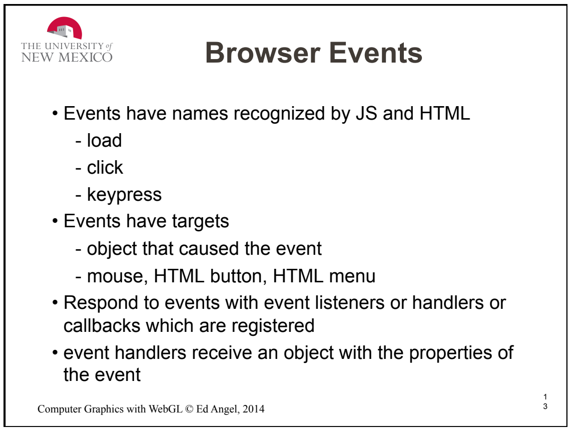
Event

Program



Event Types

- Window: resize, expose, iconify
- Mouse: click one or more buttons
- Motion: move mouse
- Keyboard: press or release a key
- Idle: nonevent
  - Define what should be done if no other event is in queue



Browser Events

- Events have names recognized by JS and HTML
  - load
  - click
  - keypress
- Events have targets
  - object that caused the event
  - mouse, HTML button, HTML menu
- Respond to events with event listeners or handlers or callbacks which are registered
- event handlers receive an object with the properties of the event