Dispositivos de Entrada

Profa. M. Cristina
Profa. Rosane
março 2006

Dispositivos de Entrada

- Teclado
- Mouse
- *Trackball* e *Spaceball*
- *Joystick*
- Digitalizador (*tablet*)
- *Touch panel*
- *Light pen*
- *Data Glove*
- Voz

Data Glove

3

Data Glove

- Permitem ´pegar´ um objeto virtual
- Conjunto de sensores detectam os movimentos da mão e dos dedos, bem como posição e orientação da mão

4

Scanners 3D

5

Scanners 2D e 3D

- Permitem entrar coordenadas definidas em um espaço 2D e 3D
- Digitalizar desenhos ou objetos
- A partir dos pontos de entrada é reconstruído um modelo da superfície do objeto definido no espaço 3D

6

Dispositivos de Entrada 3D

- Lêem uma posição 3D
- Retornam 3 valores para o programa: tripla (x, y, z)
- Alguns retornam também 3 ângulos de rotação
- Ex. Digitalizador 3D, *spaceball*, *dataglove*

7

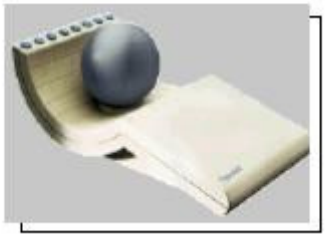
3D Input Devices

SAN DIEGO SUPERCOMPUTER CENTER

A National Center for Computational Science & Engineering

8

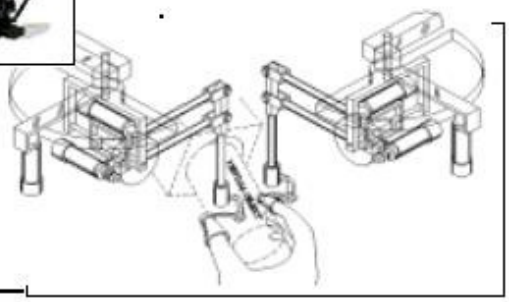
3D Input Devices



SAN DIEGO SUPERCOMPUTER CENTER
A National Center for Computational Science & Engineering

9

Force Feedback in 3D



SAN DIEGO SUPERCOMPUTER CENTER
A National Center for Computational Science & Engineering

10

Realidade Virtual

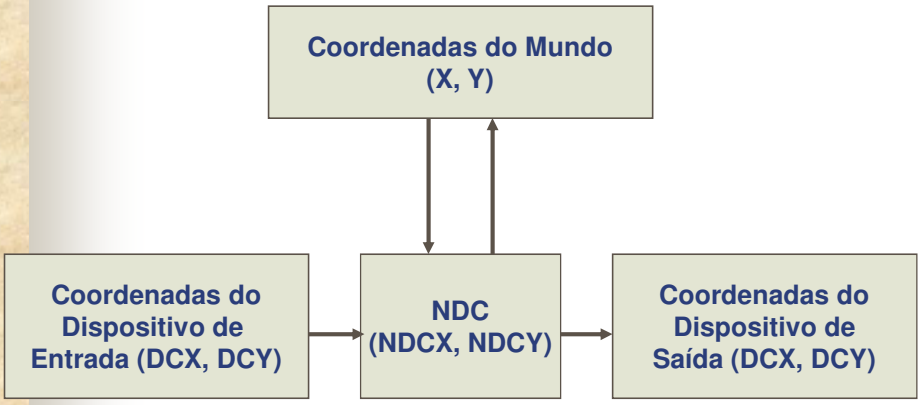
- Ver material na página do GBDI!!

http://www.gbdi.icmc.usp.br/?q=node/360

11

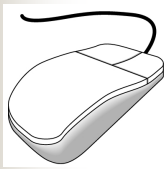
Conversão de Coordenadas

mundo 2D e dispositivos

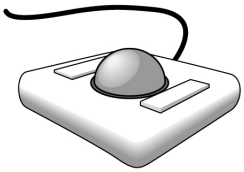


12

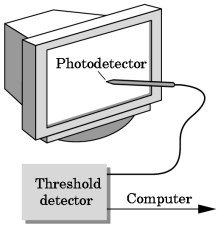
Physical Devices



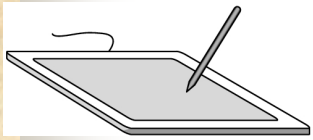
mouse



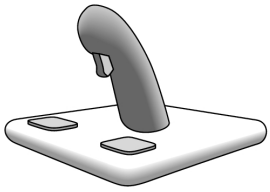
trackball



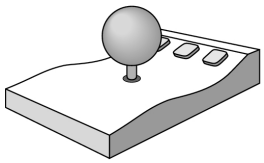
light pen



data tablet



joy stick



space ball

Incremental (Relative) Devices

- Devices such as the data tablet return a position directly to the operating system
- Devices such as the mouse, trackball, and joy stick return incremental inputs (or velocities) to the operating system
  - Must integrate these inputs to obtain an absolute position
    - Rotation of cylinders in mouse
    - Roll of trackball
    - Difficult to obtain absolute position
    - Can get variable sensitivity

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

15

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

16

Dispositivos de Entrada – Tipos Lógicos

- *Choice*
  - retorna uma escolha feita pelo usuário
  - Ex. teclado de funções, seleção de botão
  - Fornece algum tipo de *feedback* sensorial (luz, *clicks*, toque, ...)
- *Keyboard*
  - Retorna teclas com significados específicos
  - Letras, números, ...

17

Dispositivos de Entrada – Tipos Lógicos

- *Valuators*
  - Retorna um valor associado a uma medida
  - Ex. *knobs* (botões)
  - Pode especificar ganho, máximo e mínimo
- *Locators* (posicionadores)
  - Retornam a localização do cursor na tela
  - Ex. mouse, *trackball*, *tablet*,
  - Todos os posicionadores também podem funcionar como *valuators*
  - *Display-to-input ratio* (relação display-entrada)

18

Dispositivos de Entrada – Tipos Lógicos

- *DTI ratio* (“ganho”)
  - quantidade de movimento do cursor na tela dividida pela quantidade de movimento da mão
- Valor alto: velocidade
- Valor baixo: precisão

19

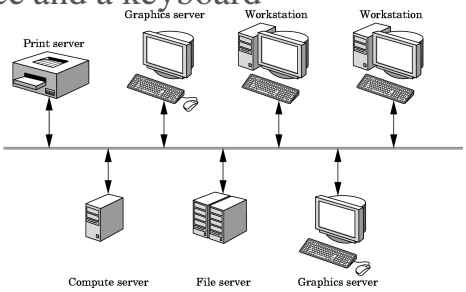
Dispositivos de Entrada – Tipos Lógicos

- Formas de ler um dispositivo de entrada:
  - *Sampling* (amostragem): qual é a entrada nesse momento?
  - *Event-based*: aguarda até que o usuário forneça alguma entrada (execute alguma ação)

20

X Window Input

- 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



21

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

22

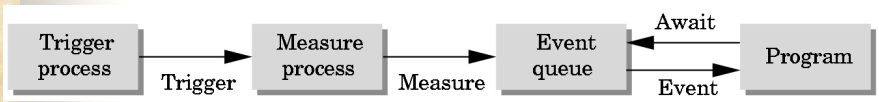
Request Mode

- 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

23

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



24

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

25

Callbacks

- Programming interface for event-driven input
- Define a *callback function* for each type of event the graphics system recognizes
- This user-supplied function is executed when the event occurs
- GLUT example:
  `glutMouseFunc(mymouse)`

mouse callback function

26

GLUT callbacks

GLUT recognizes a subset of the events recognized by any particular window system (Windows, X, Macintosh)

- `glutDisplayFunc`
- `glutMouseFunc`
- `glutReshapeFunc`
- `glutKeyboardFunc`
- `glutIdleFunc`
- `glutMotionFunc, glutPassiveMotionFunc`

27

GLUT Event Loop

- Recall that the last line in `main.c` for a program using GLUT must be

  `glutMainLoop();`

which puts the program in an infinite event loop

- In each pass through the event loop, GLUT
  - looks at the events in the queue
  - for each event in the queue, GLUT executes the appropriate callback function if one is defined
  - if no callback is defined for the event, the event is ignored

28

Bibliografia

- Hearn, D. Baker, M. P. Computer Graphics with OpenGL, Prentice Hall, 2004 (Cap. 2)
- FOLEY, J.D. et al.- Computer Graphics Principles and Practice, Addison-Wesley, 2a. edição, 1990.
- Angel, Edward – Interactive Computer Graphics 4. Ed.