

3. Input and Interaction

Example interfaces:

- Sketchpad – Sutherland
- Sketch – Brown University, J. Hughes
- Teddy – Siggraph 99

Note: interaction is system dependent, hence in GLUT.

physical devices: mouse, trackball, keyboard (direct or relative positioning), data tablets, lightpen, joystick, spaceball (6 dof)

logical device = API program

- String (keyboard)
- Locator (world coordinates, pointing device)
- Pick (locate+return identifier of object, OpenGL: select)
- Choice (*widget*: menus, scroll bars, graphical buttons)
- Dial (slidebars, dial)
- Stroke (cut and paste)

Event-driven input

1. Mouse Event, mouse up, mouse down:


```

      glutMouse(mouseCallBackFunc)
      mouseCallBackFunc(button, state, x,y)
      registered callback:
      if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
      exit(); 3squarepaint
      
```
2. Window Event


```

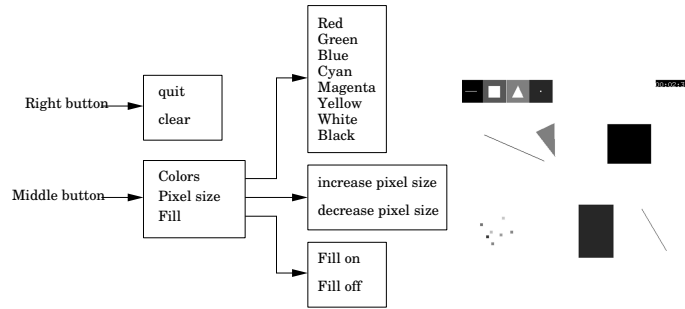
      Resize (aspect ratio? redraw? set in myReshape() )
      
```
3. keyboard Event glutKeyboardFunc()


```

      glutPostRedisplay() – execute glutDisplayFunc at next possibility
      glutIdleFunc()
      double buffering; glutSwapBuffers(); glutInitDisplayMode (GLUT_DOUBLE);
      
```

Menus, Picking

pop-up menus: link entry to mouse button + link entry to callback `glutCreateMenu()`, `glutAddMenuEntry()`, `glutAttachMenu()`



- HW1: Picking by *bounding rectangles*
- picking by color
- **3D Picking** is tricky since the graphics pipeline has to be reversed!
 Basic idea: draw scene twice. First draw into an item buffer that stores for each pixel the names of each object (item) that overlaps the pixel (as well as depth).
 Then draw the scene into the framebuffer.
selection a hit list of objects overlapping a clipping region near the cursor is returned. `OPENGL/3select` <http://www.lighthouse3d.com/opengl/picking/>