



<ol> <li>Spam Control</li> </ol>	12.	Grade Book
2. Firewall Visualization	13.	MRI Region of Interest Analysis
<ol><li>Lecture Player</li></ol>	14.	6.370 Contest Interface
4. Timeliner IDE	15.	Recitation Assignment
5. Kerberos/AFS Ticket	16.	Sensor Network
Manager	17.	Hotel Food Management
<ol><li>Semantic Web By Example</li></ol>	18.	MIT Course Planner
7. ComicKit	19.	Drink Database
8. Electronic Ballots	20.	Music Theory Helper
9. Rummikub Game	21.	Performance Assessment
10. Airport Information Kiosk	22.	IFC Rush Manager
11. Air Traffic Control		-





## Why Use Events for GUI Input? Console I/O uses blocking procedure calls print ("Enter name:") name = readLine(); print ("Enter phone number:") name = readLine(); System controls the dialogue GUI input uses event handling instead User has much more control over the dialogue (user control and freedom) User can click on almost anything

## Kinds of Input Events

- Raw input events
  - Mouse moved
  - Mouse button pressed or released
  - Key pressed or released
- Translated input events
  - Mouse click or double-click
  - Mouse entered or exited component

6.893 UI Design and Implementation

- Keyboard focus gained or lost
- Character typed

Fall 2003

8



- Mouse position (X,Y)
- Mouse button state
- Modifier key state (Ctrl, Shift, Alt, Meta)
- Timestamp
  - Why is timestamp important?

Fall 2003

6.893 UI Design and Implementation

9

## Event Queue Events are stored in a queue User input tends to be bursty Queue saves application from hard real time constraints (i.e., having to finish handling each event before next one might occur) Mouse moves are coalesced into a single event in queue If application can't keep up, then sketched lines have very few points





