

Choice of devices



- **Absolute locators: one-to-one mapping from device to cursor**
 - e.g. tablet
 - faster & easier
 - fixed range therefore less accurate
- **Relative locators: map device movement to rate of change of cursor**
 - e.g. joystick
 - harder motor skills
 - range is infinite

What about the mouse?



- **Absolute?**

- **Relative?**

- **Third type: “clutched absolute”**

- **absolute mapping within a range, but device can then be clutched and the range moved**

- **good compromise**

- **trackballs also in this category**

Fitts' Law

- Direct relationship to distance
- Inverse relationship to size
- distance/size = “difficulty”
- expert use
- avoid device swapping

$$\text{Time} = A + B \cdot \log_2(\text{Dist}/\text{Size} + 0.5)$$

Case study: menus



■ Fixed location

■ Pulldown

■ Popup

■ Pie

Beating Fitts' Law



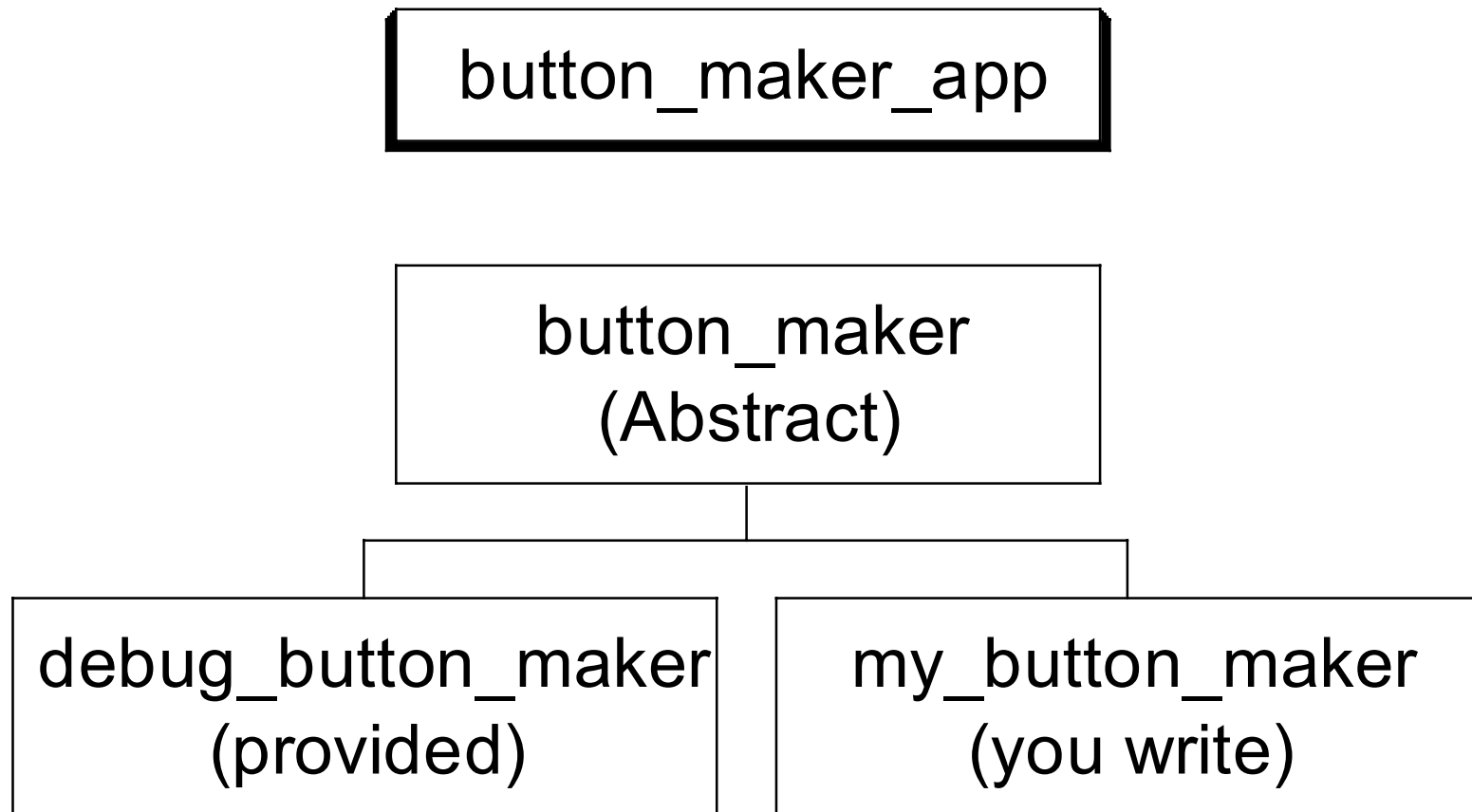
- **Map all movements to nearest legal or useful position**
 - **snap to grid**
 - **snap to object**
 - **snap to feature**
 - **dynamic semantics**

Programming Assignment



- **Button palette for UNIX commands**
- **Build interface based on command-line arguments**
- **Interface is row or column of buttons**
- **Write in Java/Swing**

Class Hierarchy



Your Java program

- **Provided:** `button_maker_app.java`,
`debug_button_maker.java`
- **You write:** `my_button_maker.java`
- **Add** `/usr/local/jdk1.2-01/bin/` **and**
`/usr/local/jdk1.2/bin/` **to your path**
- **Compile:** `javac [-g -deprecate] *.java`
- **Run:** `java button_maker_app [args]`