CS 349 — User Interfaces

Kevin James

Winter 2015

Contents

1 Vector UI

We can use multiplicative vectors to translate, shift, or rotate our objects. The important vectors are as follows:

- the **translation matrix** is $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$
- the **scaling matrix** is $\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- the **rotation matrix** is $\begin{bmatrix} \cos\Theta & \sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

2 Widgets

A **widget** is a generic name for a part of an interface with its own behaviour. They tend to have their own appearance, their own purpose, and can be pretty much anything (ie. a scrollbar, a button, a textbox...). A **logical input device** is a graphical component defined by a function rather than by what it looks like. Each devices transmits a set of primitives:

- locator: an (x,y)-position
- pick: identifies a displayed object
- choice: selects from a set of alternatives
- valuator: inputs a value
- string: inputs a string of characters
- stroke: inputs a sequence of positions

The primitives are abstracted away so that widgets do not need to handle multiple inputs types, eg. keyboards and voice-recognition.

A widget may be considered a logical input device with an appearance. A logical *button* device can generate a "pushed" event, though it may look like a button, be a simple keyboard shortcut, etc.

There exist three types of wisgets: simple, container, and abstract model widgets. We can further characterize them by the model they manipulate, the events they generate, and the properties affecting their appearance.

Goals of a widget toolkit:

- complete: GUI designers should have everything they need
- consistent: user sees a consistent look and feel and developers have constistent usage paradigms

- customizable: developer can reasonably extend functionality to meet the needs of an application