SELECTING OBJECTS

CMPT 381

Context: adding user interactions

- Handles
- Interactive object creation
- Selection
- User-controlled transformation
- Snapping
- Grouping

Types of selection

- Single selection
  - Select overlapped object
  - Clearing the selection
- Multiple selection
  - Add to / subtract from existing selection
  - Rubber-band rectangle selection
  - Lasso selection
  - Control key “select all”

Single selection

- Need a way to select a single object in the model
- List traversal
  - Press a key (e.g., Tab) to iteratively select each item
- Filtering
  - User sets filters or search criteria to determine selection
- Pointer-based selection
  - User clicks on item of interest

Pointer-based selection

- Principles of “direct manipulation”:
  - continuous representation of objects of interest
  - actions and feedback that are:
    - Rapid
    - Reversible
    - Incremental

Pointer-based selection

- On pointer event (e.g., mouse up):
  - Iterate through each item in model
  - Check whether item contains pointer location
  - Return first item that matches
- "Contains" for different shapes:
  - Line or curve: threshold distance from line
  - Filled shape: inside/outside test

Selecting an overlapped object

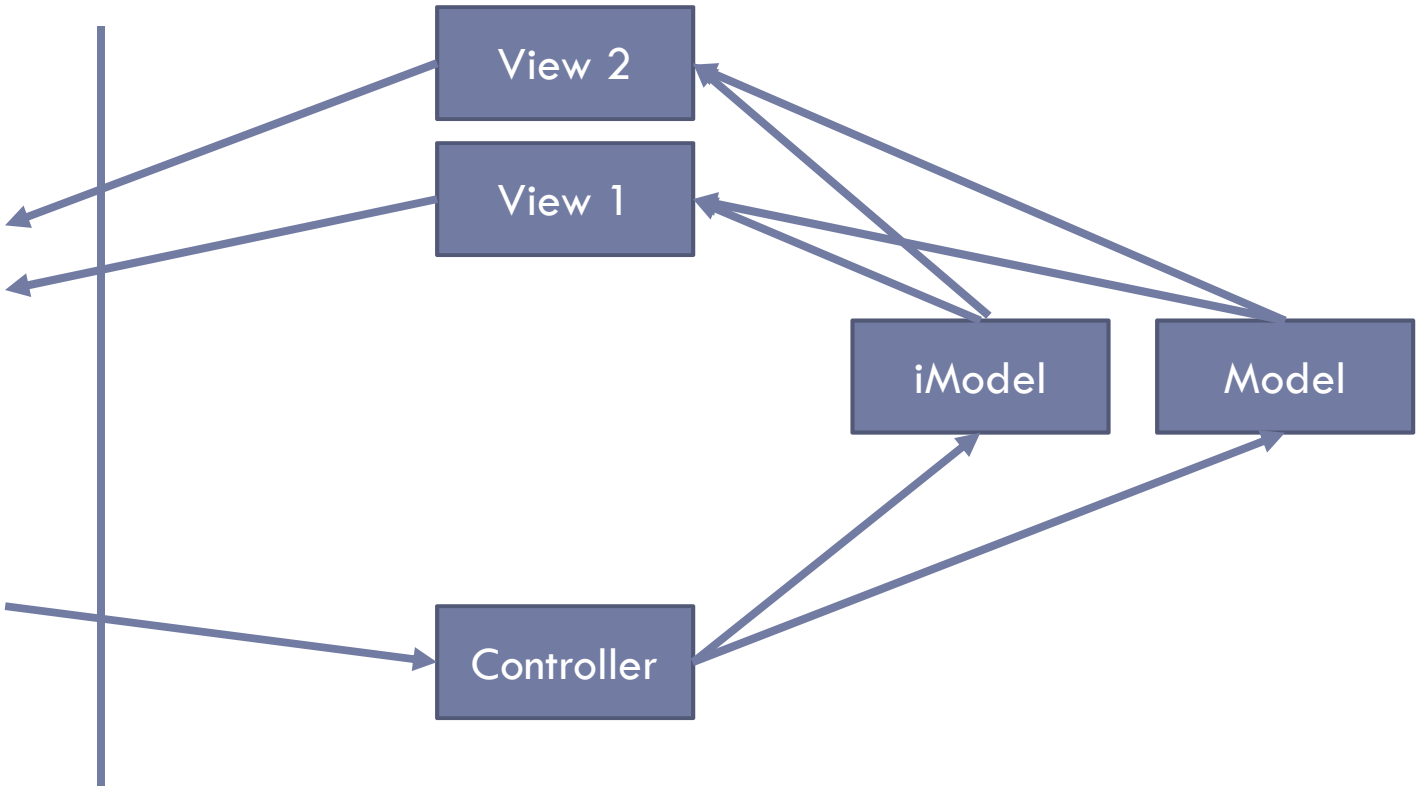
- Select object "on top" in the View, not just first hit
- Model objects can store Z-order (stacking order)
  - View draws items in this order
  - Information about which is on top (lowest/highest Z value)
- On pointer event (e.g., mouse up):
  - Iterate through each item in model
    - Check whether item contains pointer location
      - Check whether this item has a higher Z value than the current candidate
      - If so, set current candidate to point to item
  - Return current candidate

Storing the selection

- Usually stored as a reference to model object
- Where should the selection be stored?
  - In the View?
  - In the Model?
- Will the selection be needed by other components?
  - E.g., other Views
  - Views should not talk to each other

Interaction Models

- A way to store data needed for Views, but that is not really appropriate as part of the Model



Visual representation of the selection

- The user needs feedback that they have successfully selected an item, and which one
- Visual effects to indicate selection (in View):
  - Change object colour (but colour may encode meaning)
  - Object border width or colour
  - Other visual effect (glow, lifting, shadow, etc.)
  - Object handles (if object can be manipulated)
  - Bring object to top of Z-order

Clearing the selection

- Clicking on background
  - Semantics = “select nothing”
  - Is this action already taken (e.g., for object creation)?
- Explicit “clear selection” command
  - E.g., Photoshop / GIMP

Single selection

- Typical interaction rules:
  - mouseDown selects
  - Selection brings to top
  - mouseDown+mouseMove drags
  - An object stays selected until:
    - Another object is selected (see multiple select)
    - mouseDown on no object occurs

Multiple selection

- How to select multiple objects?
- List traversal
  - Modifier key to extend selection (e.g., Shift-Click)
- Filtering
  - User sets filters or search criteria to determine selection
- Pointer-based selection
  - How to specify which objects are selected
  - Need mechanism for manipulating the selection set

Add to / subtract from selection

- Builds on single-select mechanism (click to select)
- Uses modifier control to add to / subtract from selection set
  - Shift-click (Macintosh)
  - Control-click (Windows)
  - Many others possible (e.g., right-click)

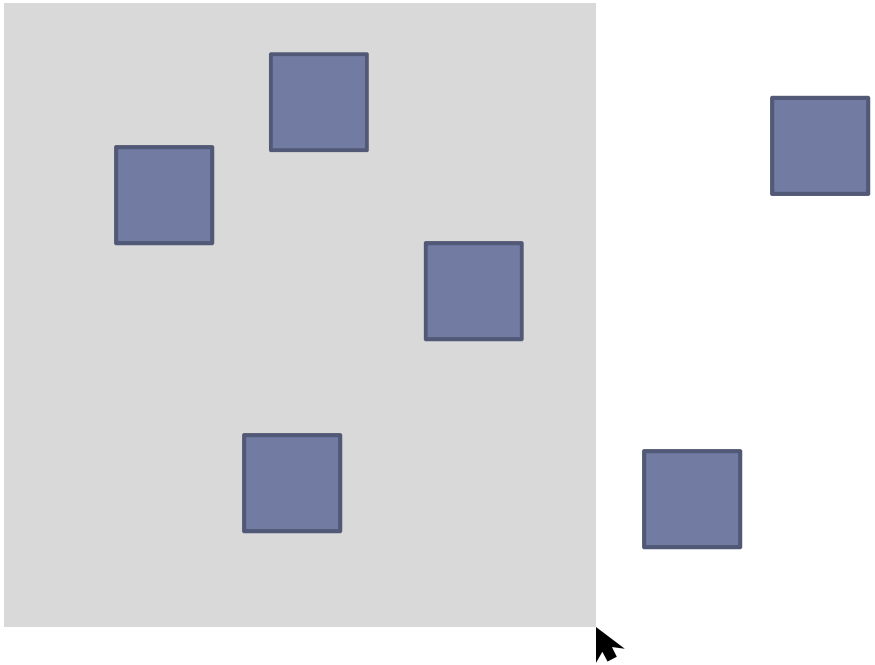
Add to / subtract from selection

- Builds on single-select mechanism (click to select)
- Uses modifier control to alter selection set
  - E.g., Shift-click (Macintosh), Control-click (Windows)
- Algorithm similar to single selection
  - If modifier is active:
    - Check whether item is in the selection set
      - If so, remove item
      - If not, add item

Storing multiple selections

- Collection class to store multiple model items
- Same decision about where to locate the selection

Rubber-band rectangle selection



Rubber-band rectangle selection

- Interactive creation of selection rectangle
  - Click on empty space and drag
- All items contained within the rectangle are selected
  - Completely contained, or partially contained?
    - Bounding box intersects?
    - Control points inside?
  - Requires variation on “contains” method
- Check selection on mouseUp or interactively

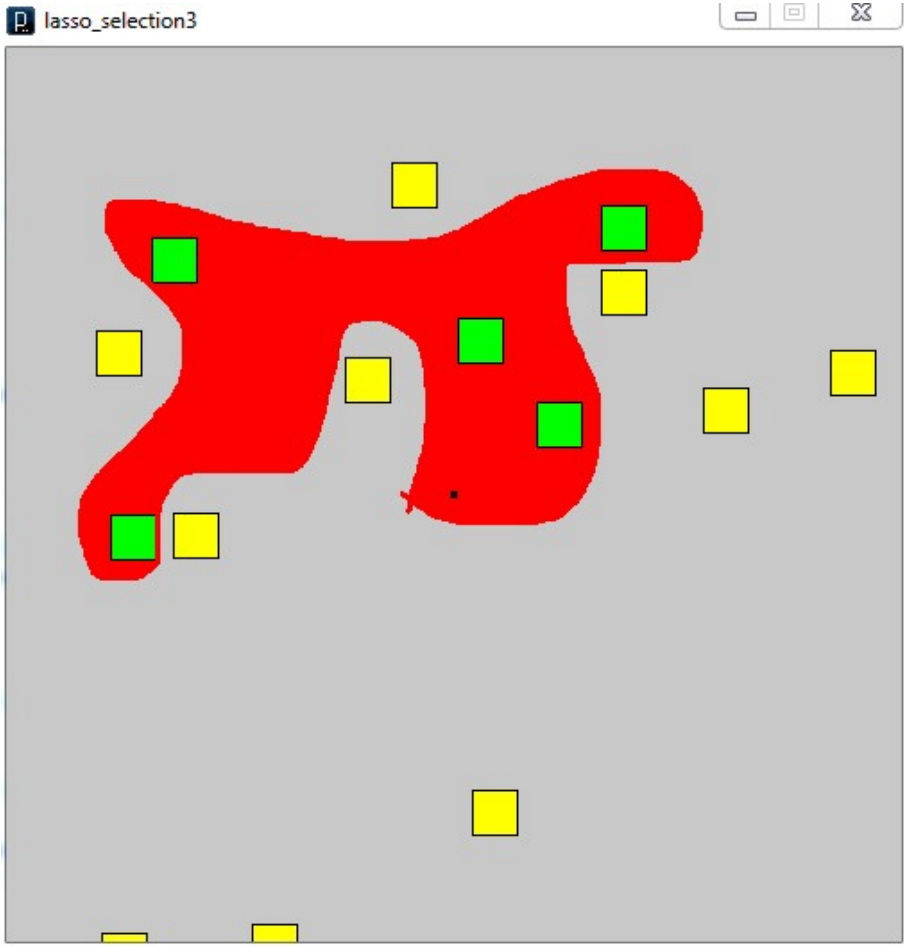
Rubber-band rectangle selection

- If selection rectangle is non-zero in size
  - Clear current selection set
  - Iterate through model items
    - Determine whether item is inside rectangle
    - If so, add item to selection set

Rubber-band plus add/subtract

- The two techniques can be combined
- Modifier-click to add/subtract individual items
- Modifier-rubber-band to add/subtract multiple
  - Second rectangle select may add some items and remove others with a single action
- Need change to storage:
  - Current selection set
  - Adjustment selection set
    - For each item in adjustment, add/remove from current

Lasso selection



Lasso selection

- Problem with rectangle selection is that desired objects do not always fit in a rectangle
- Lasso selection allows user to draw a line to specify the selection set
- How to determine “contains” for arbitrary lasso?

Lasso selection

- Create a bitmap the size of the View
- Draw the lasso on the bitmap
- Close the lasso
- Flood-fill the lasso with red
- For each item in the model:
  - Check colour of bitmap pixel for item's principal points
  - If all checked pixels are red, item is inside lasso

Combined rectangle/lasso selection

- Saund and Lank, UIST 2003
- If mouse goes down and up without moving more than a few pixels, it is a click
- If mouse goes down and then up some distance from the down point, it is a rectangle selection
- If mouse goes down and moves some distance but goes up close to down point, it is a lasso selection
- Show feedback for both rectangle and lasso until it becomes clear which is happening