INTERACTING WITH GRAPHICS

CMPT 381

Outline

- Working with paths
  - Bounds, Distance to object, Nearest point, Object intersection, Inside/Outside
- Transformations
  - Translation, Scaling, Rotation
- User interactions
  - Handles, Object creation, Selection, Transformation, Snapping, Grouping

Doing it yourself

- In an application-defined graphical workspace:
  - There are no UI layers to help you determine:
    - Which object was clicked on (“hit detection”)
    - Where objects should be drawn (e.g. arrows between objects)
- What do toolkits provide?
  - JavaFX / Android Canvas:
    - Basic drawing commands
    - Matrix transforms: translate, rotate, scale
  - JavaFX Shape API:
    - Inside/outside, bounding box, intersection check
    - Node properties: location, rotation, scale

Path equations

- Bounds
- Distance from a point to an object
- Nearest point on an object
- Object intersection
- Inside/Outside

Bounding box

- Why do we need bounding boxes?
  - Checking inside/outside of a rectangle is easy
- If convex hull property true:
  - Use the min and max of the control points
  - For a line:
    - left = min(x1,x2)
    - right = max(x1,x2)
    - top = min(y1,y2)
    - bottom = max(y1,y2)
- If not true?
  - E.g. circle

Distance from point to object

- Why:
  - Is a mouse click close enough to select the object?
- Use implicit equation for the object
  - for a line:
    - $Line(x,y) = Ax + By + C$
    - For points on the line, returns 0
    - For points left/above the line, returns negative
    - For points right/below the line, returns positive
  - for a circle:
    - $Dist(x,y) = sqrt((x - x_C)^2 + (y - y_C)^2)$ - radius

Nearest point on an object

- Why?
  - E.g. drawing arrows between objects
- Use implicit equation again
  - For line:
    - Nearest(x) = x – A * Line(x,y)
    - Nearest(y) = y – B * Line(x,y)

Intersection of objects

- Why
  - Not frequent, but sometimes important
- Lines
  - Linear equations
  - Algebraic
- Circles, arcs, ellipses
  - Quadratic equations
  - Algebraic
- Rotated ellipses, splines, curves
  - Difficult
  - requires numerical methods

Inside/Outside

- For selection of filled objects
- For general shapes
  - inside/outside is difficult
  - Use bounding boxes first
  - Determine which objects must be checked in more detail
- For simple shapes
  - use basic equations
- Java Graphics2D Shape API

Transformations

- Translation
- Scaling
- Rotation

- Each transformation uses a pair of equations
  - maps (x,y) to (x’,y’)
- Each transformation uses coefficients to determine magnitude of the change
  - Amount of movement, scaling, or rotation
- Matrix versions for these (and more) transformations

Translation

- Moves an object from one location to another
- Adds a distance to x and y

  $x' = x + dx$

  $y' = y + dy$

Scaling

- Changes the size of an object
- Multiplies x and y by scale factor

  x’ = x * sx

  y’ = y * sy
- Note
  - Scaling is done relative to the origin
    - Objects not at the origin will move as well
  - What does a negative scale factor do?

Rotation

- Rotates an object counterclockwise
- Adds a distance to x and y

  $x' = \cos(a) * x - \sin(a) * y$

  $y' = \sin(a) * x + \cos(a) * y$
- Note
  - Again, rotates around the origin

Combining transformations

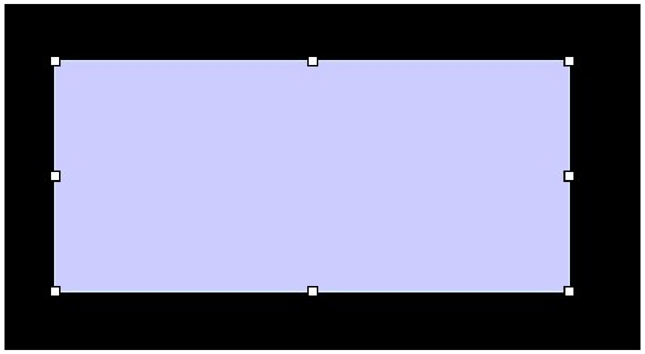
- Often need multiple transformations
  - E.g. in-place scaling/rotation
    - Translate to origin
    - Scale or rotate
    - Translate back to starting position
- Homogeneous coordinates
  - Represent all transformations in similar format
  - There are matrix representations for each transform
  - Matrix multiplication is the combination mechanism
    - Also allows undo of any transformation

User interactions

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

Handles

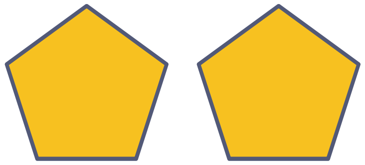
- Basic interaction paradigm
  - Select-and-manipulate
- MVC view of graphical shapes
  - Model stores control points of the shape
  - View can make control points visible to user
    - “selection handles”
- Allows for interactive manipulation



Interactive object creation

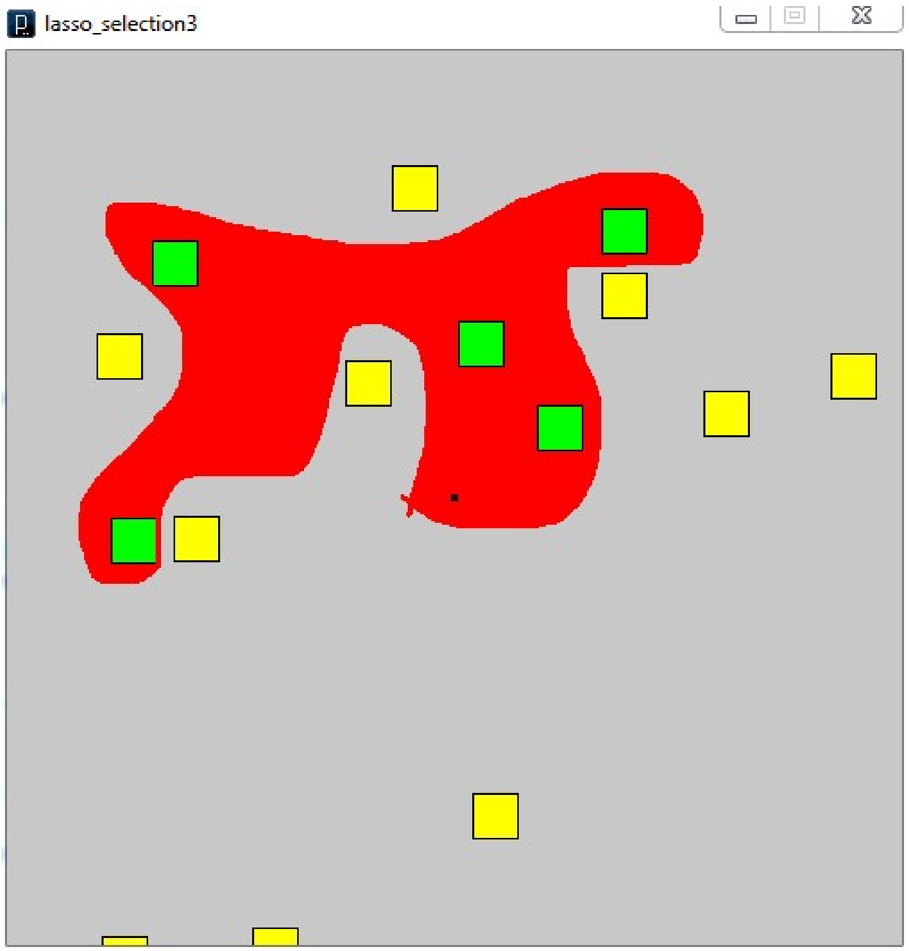
- Provide feedback during creation
  - Improves accuracy and provides preview
- Two approaches
  - If object does not depend on other objects
    - E.g. rectangle in drawing editor
    - Create object on initial mouseDown
    - Then proceed as if resizing
  - If object has dependencies
    - E.g. edge in graph editor
    - Create temporary representation for the object
    - Then check validity on mouseUp
    - Where in MVC is this temporary object kept?

Selection



- Traditional interaction rules (for some platforms):
  - mouseDown selects
  - mouseDown+mouseMove drags
  - an object stays selected until:
    - Another object is selected
    - mouseDown on no object occurs
- Multiple selection:
  - Discrete:
    - Add objects to the selected list with modifier key
  - “Rubber-band” area:
    - Initiated by clicking on background and dragging
    - Requires visual feedback
    - Requires intersection testing on mouseUp

Lasso selection



User-controlled transformations

- Translation
  - Done by dragging on shape area (not handle)
- Scaling
  - Type 1: by dragging control handle
    - Recalculate size based on new control points
  - Type 2: by specific amount (e.g. 200%)
    - Must use scale transformation (in-place)
- Rotation
  - How to differentiate from dragging or scaling
  - What point to rotate around?
    - Again, must translate→rotate→translate

Snapping

- Many graphical tasks involve precise positioning
  - However, difficult to position the mouse cursor accurately
- Snapping
  - Rounds off mouse position to a more appropriate place
  - Snap to grid:
    - Simple localization of position to a rectilinear grid
    - Grid can have arbitrary resolution
    - Some intersections can lie between grid points
  - Snap to object:
    - Localize positions to nearest intersection
    - Uses basic equations discussed earlier

Grouping

- Allows manipulation of composite objects
  - Some objects need grouping for correct transformation
- View organized into a hierarchy
  - Assists with selection, transformation, duplication