Plan for today
- small group exercises
  - Complexity Families
- last week reading Q&A
  - paper: Polaris
- this week reading Q&A
  - chap: Manipulate Interactive, Multiple Views, paper: Scalable Insets
- reminder: post-class office hours
  - if you want discussion of your project proposal feedback ASAP
    - faster than waiting for my written comments

Upcoming
- next week (W10): reading week, no class, no readings, no async discussion
  - work on projects!
- week after (W11)
  - light: async reading/discussion (note updated web page)
  - 1 reading: Ch. 13: Reduct
  - due Thu 3pm: project updates
    - in class: post-update meetings with Tamara
    - small group exercises
    - first A critiques B; then B critiques A
    - read other team’s written update before class
    - in class: mini-lecture
    - in class: post-update meetings with Tamara

How to handle complexity: 1 previous strategy
- derive new data to show within view
  - why: find extreme values, trends
  - change view over time
  - facet across multiple views

Idiom: Change parameters
- widgets and controls
  - sliders, buttons, radio buttons, checkboxes, dropdowns/comboboxes
- pros
  - clear affordances, self-documenting (with labels)
- cons
  - uses screen space
- design choices
  - separated vs interleaved
  - controls & canvas

Idiom: Change order/arrangement
- what: simple table
- how: data-driven reordering by selecting column
- why: find extreme values, trends

Upcoming
- week after that (W12)
  - async last week reading / discussion
  - Ch. 14: Designed Focus/Context
    - last week of readings / discussion
    - last week of classes (W13)
    - faster than waiting for my written comments

How to handle complexity: 1 previous strategy + 2 more

Idiom: Change alignment
- stacked bars
  - easy to compare
  - first segment
total bar
  - align to different ...
  - aggregation level, what is filtered...
  - interaction entails change
  - powerful & flexible

Idiom: Re-encode
- derive new data to show within view
  - why: find correlations between attributes
  - system: DataStripes
    - stacked bars
    - easy to compare
    - first segment
total bar
  - align to different segment
    - supports flexible comparison

Idiom: Reorder
- what: table with many attributes
- why: find correlations between attributes
  - system: LineUp
    - aligned segments
    - supports flexible comparison

Idiom: Change order/arrangement
- what: simple bar
- how: data-driven reordering by selecting column
- why: find extreme values, trends

Idiom: Change parameters
- widgets and controls
  - sliders, buttons, radio buttons, checkboxes, dropdowns/comboboxes
- pros
  - clear affordances, self-documenting (with labels)
- cons
  - uses screen space
- design choices
  - separated vs interleaved
  - controls & canvas

Visualization Analysis & Design
Interactive Views (Ch 11/12)

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner
Navigate: Changing viewpoint/visibility

- change viewpoint
  - changes which items are visible within view
- camera metaphor
  - pan/translate/scroll
- pan/translate/scroll
- constrained navigation
- unconstrained navigation

Selection

- selection: basic operation for most interaction
- design choices
  - how many selection types?
- interaction technology

Highlighting

- highlight change visual encoding for selection targets
- visual feedback closely tied to but separable from selection
- changes which items are visible within view

Manipulate

- animated transition
  - network drilldown/rollup
- change viewpoint
  - changes which items are visible within view

Interaction technology

- what do you design for?
- mouse & keyboard on desktop?
- large screens, hover, multiple dots
- touch interaction on mobile?
- small screens, no hover, just tap
- gestures from video / sensors?
- ergonomic reality vs movie bombst
- eye tracking?

Idiom: Animated transitions - visual encoding change

- smooth transition from one state to another
  - alternative to jump cuts, supports item tracking
  - best case for animation
  - staging to reduce cognitive load

Idiom: Animated transition - tree detail

- animated transition
  - network drilldown/rollup

Idiom: Animated transition + constrained navigation

- example: geographic map
  - simple zoom, only viewpoint changes, shapes preserved

Navigate: Changing viewpoint/visibility

- change viewpoint
  - changes which items are visible within view
- camera metaphor
  - pan/translate/scroll

Navigate: Unconstrained vs constrained

- unconstrained navigation
  - easy to implement for designer
  - hard to control for user
  - easy to overemphasize/underemphasize
- constrained navigation
  - typically uses animated transitions
  - trajectory automatically computed based on selection
  - just click; selection ends up framed nicely in final viewpoint

Navigate: Changing viewpoint/visibility

- how: navigate page by scrolling (panning down)
- pros:
  - familiar & intuitive, from standard web browsing
  - linear (only up/down) vs possible overload of click-based interface choices
- cons:
  - full-screen mode may lack affordances
  - scrolling/panning without direct access
  - unexpected behaviour
  - continuous control for discrete steps

Interaction benefits

- interaction pros
  - major advantage of computer-based vs paper-based visualization
    - flexible, powerful, intuitive
    - exploratory data analysis: change as you go during analysis process
    - fluid task switching: different visual encodings support different tasks
    - animated transitions provide excellent support
    - empirical evidence that animated transitions help people stay oriented

Interaction limitations

- interaction has a time cost
  - sometimes minor, sometimes significant
  - degrades to human-powered search in worst case
- remembering previous state imposes cognitive load
- controls may take screen real estate
  - or invisible functionality may be difficult to discover (lack of affordances)
- users may not interact as planned by designer
  - NYTimes logs show ~90% don’t interact beyond scroll (Ansch, 2016)
Interactive Views (Ch 11/12)

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner

Visualization Analysis & Design

How to handle complexity: 1 previous strategy + 2 more

- Derive
- Manipulate
- Juxtapose
- Change
- Partition
- Navigate
- Superimpose
- Select
- Multiple views
- Coordinate Multiple Side By Side Views
- Embed

Facet

Juxtapose benefit

Different interaction idiom

- encoding: same or different
- data: subset shared
- navigation: shared
- unidirectional vs bidirectional linking
- other differences
- (window size)

Facet

Juxtapose views: tradeoffs

- Juxtapose costs
- display area
- 2 views side by side; each has only half the area of one view

- Juxtapose benefits
- cognitive load: eyes vs memory
- lower cognitive load: more eyes between 2 views
- higher cognitive load: compare single changing view to memory of previous state

Linked views: Directionality

- unidirectional vs bidirectional linking
- bidirectional almost always better!

Small multiples

- encoding: same
- data: none shared
- items or attributes
- ex: stock prices for different companies

How?

- Encode Manipulate Facet Reduce
- Arrange
- Map
- Change
- Select
- Navigate
- Superimpose

Why?

- What?
- How?
- Why?

Linked highlighting

- see how regions contiguous in one view are distributed within another
- powerful and pervasive interaction idiom

- encoding: different
- data: all shared
- all items shared
- different attributes across the views
- aka brushing and linking

Linked views: Directionality

- unidirectional vs bidirectional linking
- bidirectional almost always better!
### View coordination: Design choices

<table>
<thead>
<tr>
<th>Data</th>
<th>All</th>
<th>Sub</th>
<th>Main</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Idiom: Reorderable lists

- Set views
  - easy navigation
  - content visible in other views
- many views in one vs free movement
- open research questions

### System: Improvise

### Facet

- Juxtapose
- Partition
- Superimpose

### Partition into views

- how to divide data between views
- split into regions by attributes
- encodes association between items using spatial proximity
- other methods have major implications for what patterns are visible

### Partitioning: Recursive subdivision

- different encoding for second-level regions
- choropleth maps
- superimposed layers
- layer: set of objects spread out over region
  - each set is visually distinguishable group
  - entire view
- design choices
  - how many layers, how to distinguish?
  - concurrency with different, non-overlapping channels
  - two layers achievable, three with careful design
  - small or dynamic from many possible?

### Static visual layering

- foreground layer: roads
- hue, size distinguishes man from mirror
- high luminance contrast from background
- background layer: regions
- desaturated colors for water, parks, land areas
- user can selectively focus attention

### Superimposing limits (static)

- superimpose within same frame
  - color code by year
- partitioning
  - split by site, rows are barley varieties
- main-effects ordering
  - derive value of median for group
  - order rows within view by variety median
  - order views themselves by site median

### Dynamic visual layering

- interactive, based on selection
- one-hop neighbour highlighting
  - click (heavyweight) hover (fast)