Enlightenment Developer Day 2017

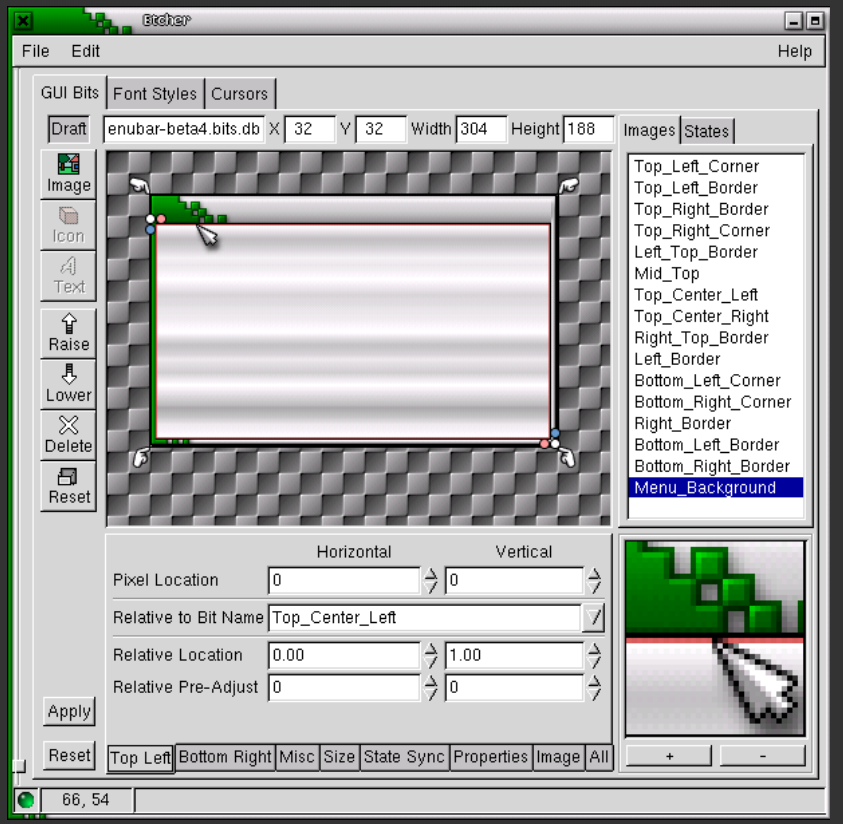
Time to try making themes easier

Carsten Haitzler
<raster@rasterman.com
<c.haitzler@samsung.com>

SOME BACKGROUND

Ebits + Etcher

- Ebits was a file with a single group with N layers
  - 4 images (normal, clicked, hilighted, disabled)
  - rel1/2 to + relative + absolute offset
  - Fill (scale or tile) setup
- Had full GUI editor (GTK+) at start – no text source
- Far too limited → gave birth to Edje
  - No animation
  - States only changed image not rel1/2 etc.



Edje

- Replaced Ebits
- Tired of writing animators just to animate from A to B
  - Edje got tweening/transitions as a result
  - Layout (rel1/2 etc.) was part of the state (description)
- Became the runtime abstraction for GUI design objects
  - Flexible runtime changes allowed with themes
  - Convenient vs. C to do the same

Temporary means “hang about forever”

- EDC was a tmp solution until a GUI tool could be built
  - Well… we all know how long tmp solutions last now
  - Was not considered from human maintainability POV
    - It was temporary!
    - Was meant to make it easy to make a GUI editor not text
      - Planned editor using EFL once enough worked
        - That took a while…

Small - HAHAHAHA

- Started small
  - Expected maybe 50 or 100 groups in EDJ files
    - We now have 1570 in EFL’s default theme
      - ~700 are manually defined
  - Expected it to be “very little work” to make a theme
    - Now it is an insane amount of work
      - Complexity
      - Size
      - Permutations of states

TIME TO CHANGE

Time to change

- Reduce the size of groups to make it less work
  - Use more powerful features and concepts in Edje
    - Break up widgets into standard elements
    - Recycle elements between widgets a lot
    - Some features need fixing to be used sensibly
  - Result
    - Sample simple flat button
      - From 384 to 277 lines of EDC (28% down)
      - Used 2 sub groups (_icon_wrapper and _label)
      - Used box (table works too – less broken)
      - Used parts aliases
      - Used programs to forward signals from child groups/items

Need improvements

- It’d cut down size of check and radio even more
  - Share button logic/layout
- If box and table were not broken…
- If I could also define a deeper tree
- Add flags/fields to min size calc child groups
  - In group part, box & table children
  - Currently this is totally broken
    - Table can allow setting of min size
      - Box fails to do this
    - Neither actually calcs min size from child content
      - Like Elm/Evas box/table do with min size hints

Deeper tree changes

- Define swallows within a tree rather than use group parts
  - Group parts (or table/box items) fix the child used
  - Allow this to be flexible and include a group
    - Define specific swallows to be groups of the child group
    - Add specific items to tables/boxes within the child group
      - Define the tree of these items too explicitly
    - Etc. recursively (define at any depth in the tree)

Common elements

- Expected people would make #defined macro libraries
  - Also #include segments too
  - Problems for runtime changes
  - Not self-contained
- Define a much smaller set of building blocks
  - “clickable box” - button background
  - “check” - just the tick/checkbox
  - “radio” - just the round on/off selector
  - “close/del” - just the cross
  - “arrows” - up/down/left/right etc.
  - “selected item” - toolbar items, ibar icons ...

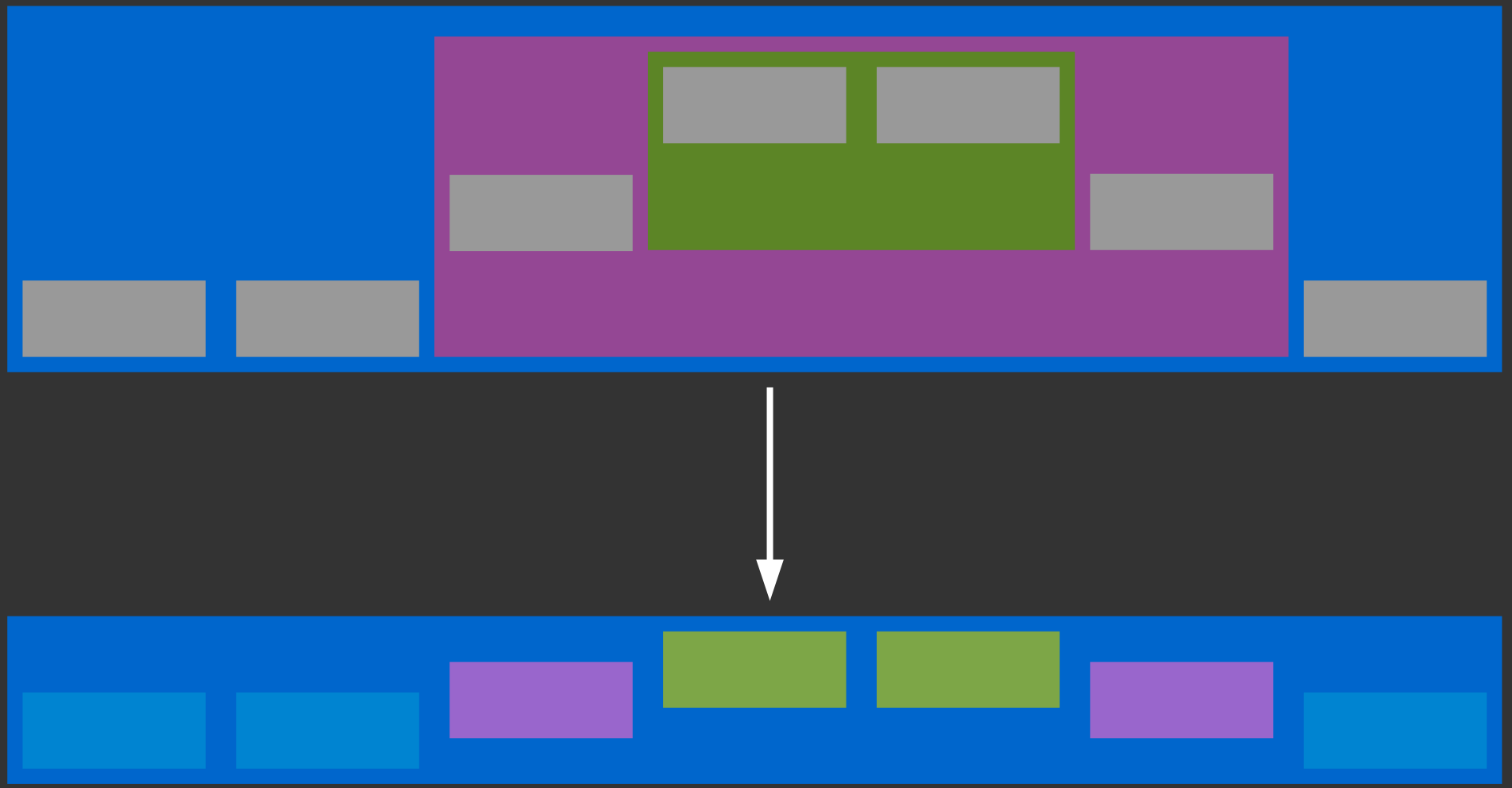
Could re-use them from elsewhere

- Allow default EDJ layout to source user EDJ files
  - Default EDJ file can provide broad layout etc.
    - User EDJ can override and/or provide above building blocks
  - Need to provide search path to Edje at runtime
- Makes making a simple theme far easier
  - Just override a few dozen building blocks
    - Not every single widget
  - Still allows full override widget/group by group for detailing

CAN OPTIMIZE

Optimize the deeper tree

- Later flatten child smart objects (box, table, edje)
  - Merge into parent (efficiency)



More optimized signal mapping

- Add signal aliases so child groups/items
  - Makes children signals appear in parent as if there
  - Currently you must use a program
    - Listen to the table/box/group child and emit another signal
- Less EDC code
- Less memory footprint for mapping vs. whole program

Some usability improvements

- Table names are currently 1D addressed
  - Always
    - **`table_or_box_part[N]:child_part_name`**
  - Offer 2D naming/addressing option instead of 1D
    - **`name[4]:yyy`** → **`name[2,3]:yyy`**

Guesstimates

- Default theme without building blocks…
  - Probably < ½ the size
- Building blocks…
  - Probably about ~40 at most
- Building custom styles for widgets…
  - SO MUCH EASIER
    - Just provide a custom building block not whole widget theme
      - e.g. custom “pushed button” edje

This is orthogonal/in addition to color, text, size classes

What to do

- Optimize later, improve usability now
  - Add Edje-internal group file fallback mechanism
    - Drive it from Elm theme
  - Start defining + using building blocks in default theme
  - Fix containering (box/table) bugs/features
  - Formalize building block naming and usage
    - Reviewed/used by people

Result?

Result?

PROFIT

DISCUSS