UI

AESTHETICS

C++

Tony Van Eerd

C++Now, May 15, 2014

BlackBerry

Overview / Warning

- I have no point.
- No order.
- Nothing new.
- Entertaining?
- Art?
- Poetry?
- Code?

Overview / Warning

- I have no point.
- No order.
- Nothing new.
- Entertaining?
- Art?
- Poetry?
- Code?
- (SUBLIMINAL)

BlackBerry® Z10 smartphone

smile the senses

action reaction cause effect

what are these symbols

Haiku

- 5-7-5 “syllables” (*on*)
- nature/season reference (*kigo*)
- “cutting” (*kiru*)

BlackBerry® Z10 smartphone

Haiku

- 5-7-5 “syllables” (*on*)
- nature/season reference (*kigo*)
- “cutting” (*kiru*)
- ... Constraints

BlackBerry® Z10 smartphone

Constraints?

smile the senses

action reaction cause effect

what are these symbols

Constraints?

smile the senses
action reaction cause effect
what are these symbols

- 5-7-5 “syllables” (*on*)
- nature/season reference (*kigo*)
- “cutting” (*kiru*)

UI Constraints

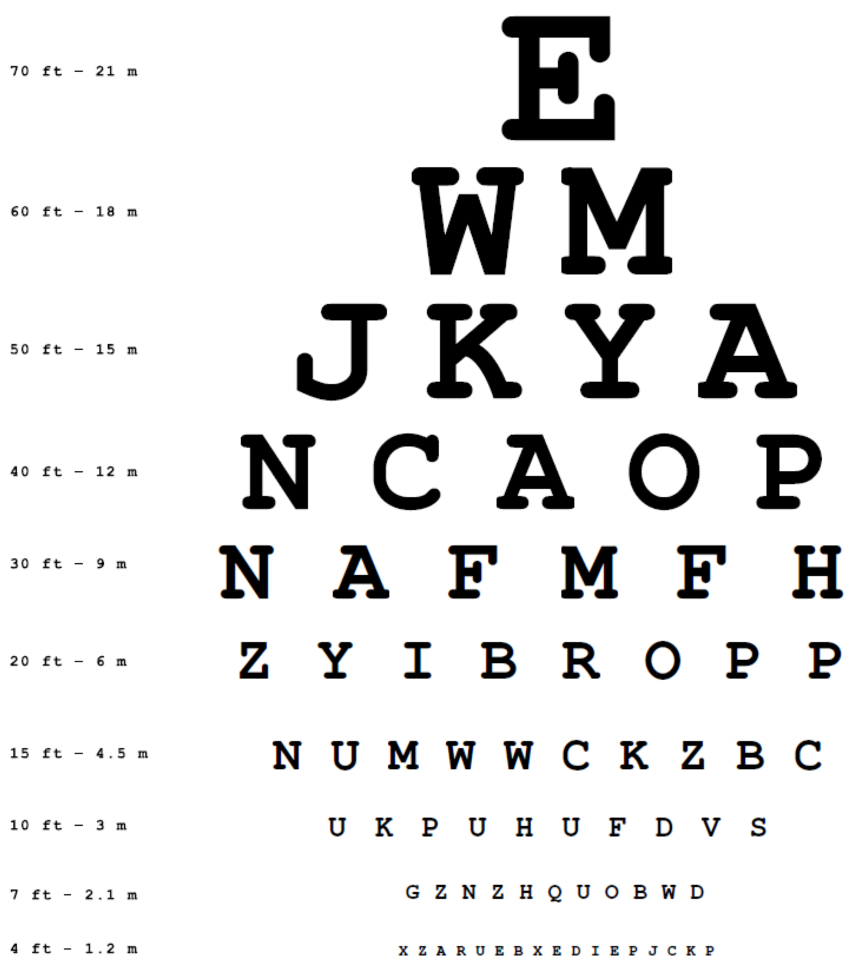
- Display Size – physical/pixels
- Display Aspect / Shape (landscape/portrait/square)
- Display Distance

BlackBerry® Q10 smartphone

UI Constraints

- Display Size – physical/pixels
- Display Aspect / Shape (landscape/portrait/square)
- Display Distance
- … *Visual Acuity*

BlackBerry® Q10 smartphone



70 ft – 21 m	E
60 ft – 18 m	W M
50 ft – 15 m	J K Y A
40 ft – 12 m	N C A O P
30 ft – 9 m	N A F M F H
20 ft – 6 m	Z Y I B R O P P
15 ft – 4.5 m	N U M W W C K Z B C
10 ft – 3 m	U K P U H U F D V S
7 ft – 2.1 m	G Z N Z H Q U O B W D
4 ft – 1.2 m	X Z A R U E B X E D I E P J C K P

UI Constraints

- Display Size – physical/pixels
- Display Aspect (landscape/portrait/square)
- Display Distance
- … *Visual Acuity*
- Input Mechanisms
- Widget-set (pro and con!)
- Consistency
- UI “Police”
- Time!

BlackBerry® Q10 smartphone

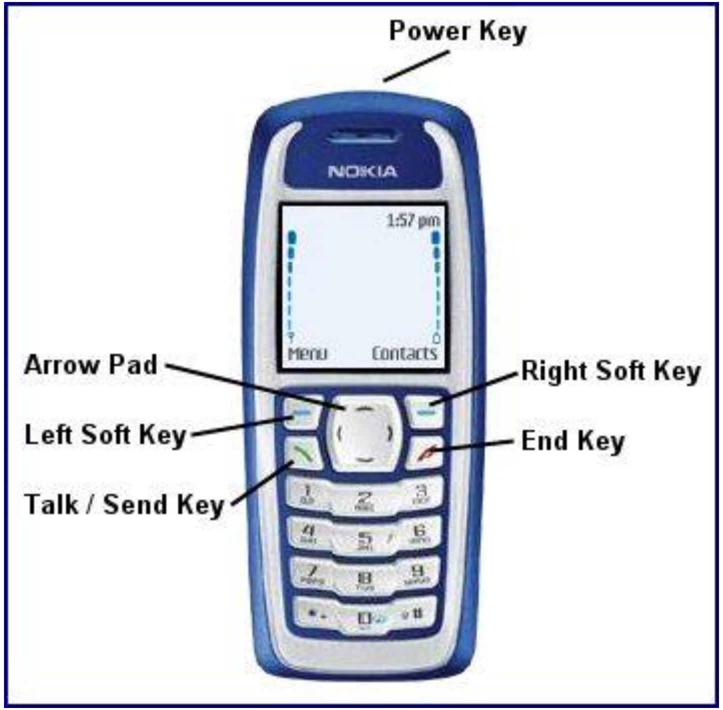
Time…

smile the senses
action reaction cause effect
what are these symbols

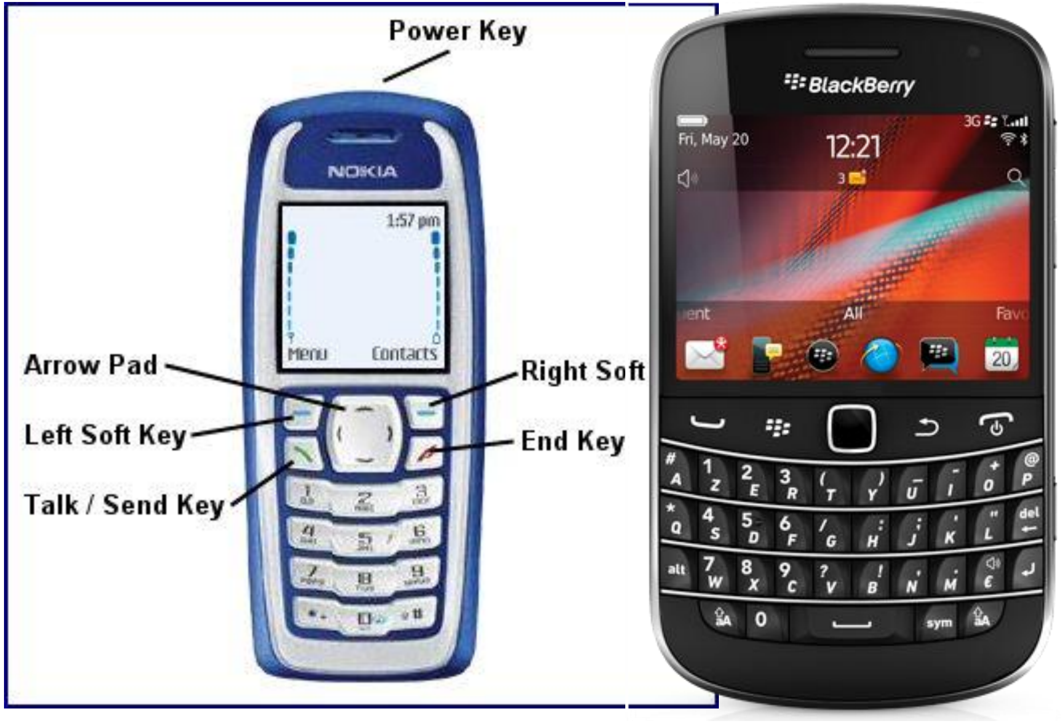
UI Constraints

favourite phone UI?

UI Constraints



UI Constraints



UI Constraints

smile the senses

action reaction cause effect

what are these symbols

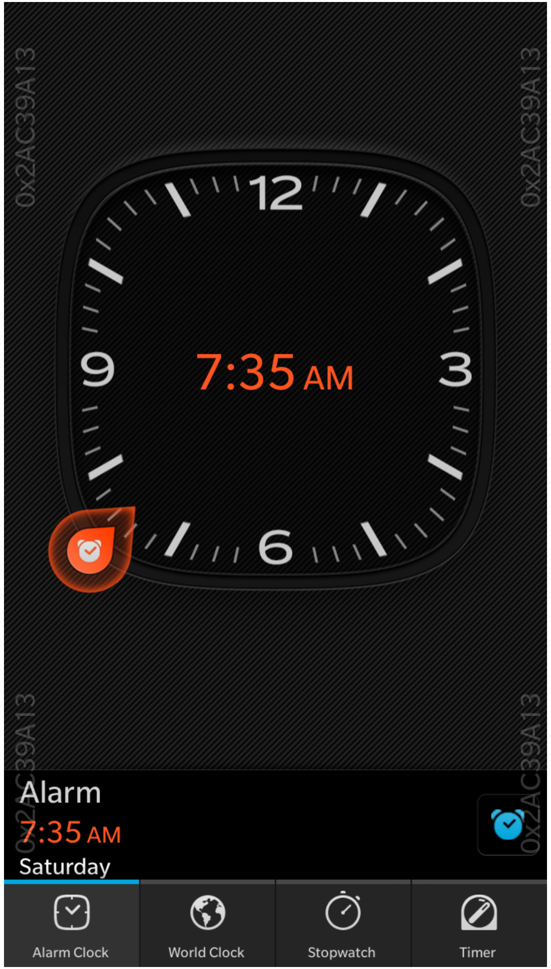
smile the senses

- Text (?)
- Vision
- Sound
- Touch
- Scripting (?)

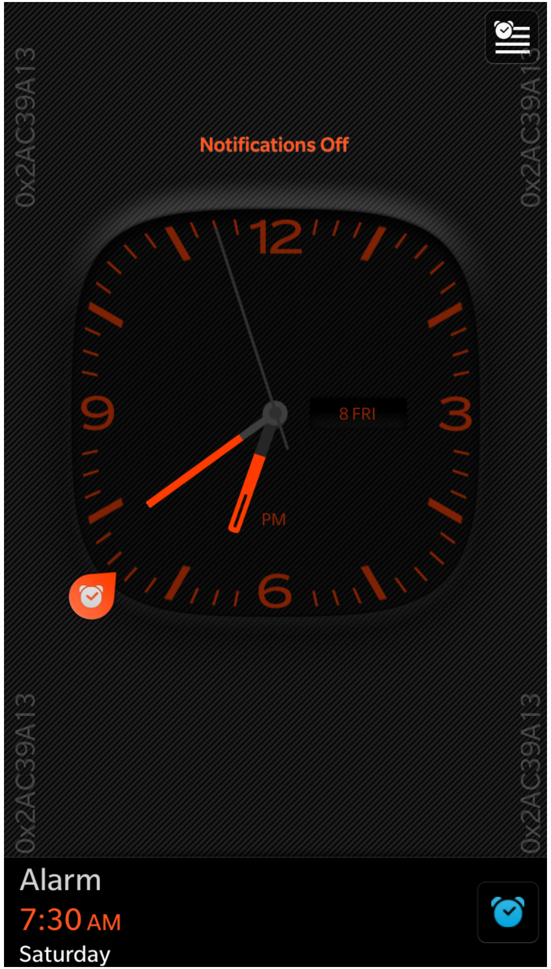
smile the senses

smile the senses

- Aesthetically pleasing
- “Moments of Charm”



Alarm
7:35 AM
Saturday



Notifications Off

Alarm
7:30 AM
Saturday

0x2AC39A13
0x2AC39A13
12
9
3
6
8 FRI
PM
0x2AC39A13
0x2AC39A13

smile the senses

action reaction cause effect

what are these symbols

← Feature Creep

We are hard-wired to understand cause and effect

momo

Difference between Hammer UI and Steering UI?

Difference between Hammer UI and Steering UI?

“Try before you buy”

Difference between Hammer UI and Steering UI?

“Try before you buy”

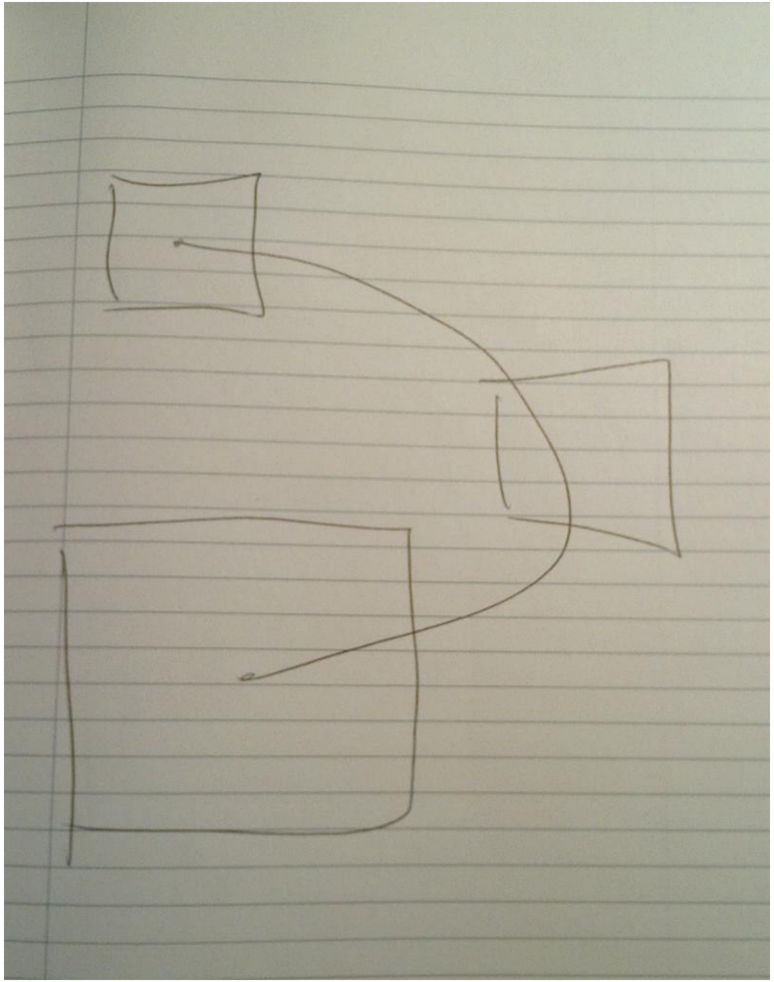
Same?

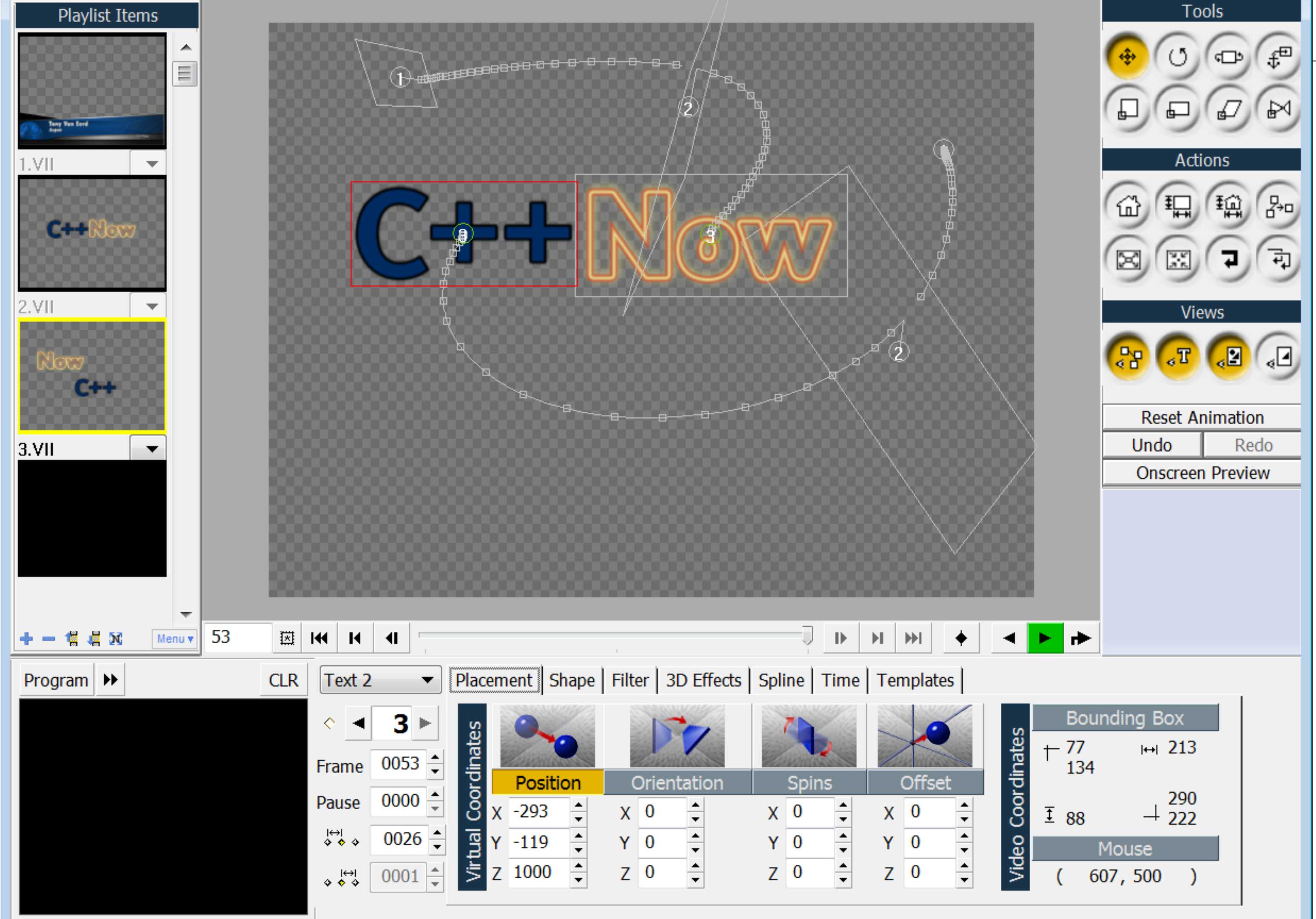
Difference between Hammer UI and Steering UI?

“Try before you buy”

Same?

1 to 1 Mapping.





53

Text 2

3

Frame 0053

Pause 0000

0026

0001

	Position	Orientation	Spins	Offset
X	-293	0	0	0
Y	-119	0	0	0
Z	1000	0	0	0

Bounding Box: 77, 134; 213; 88; 290, 222

Mouse (607, 500)

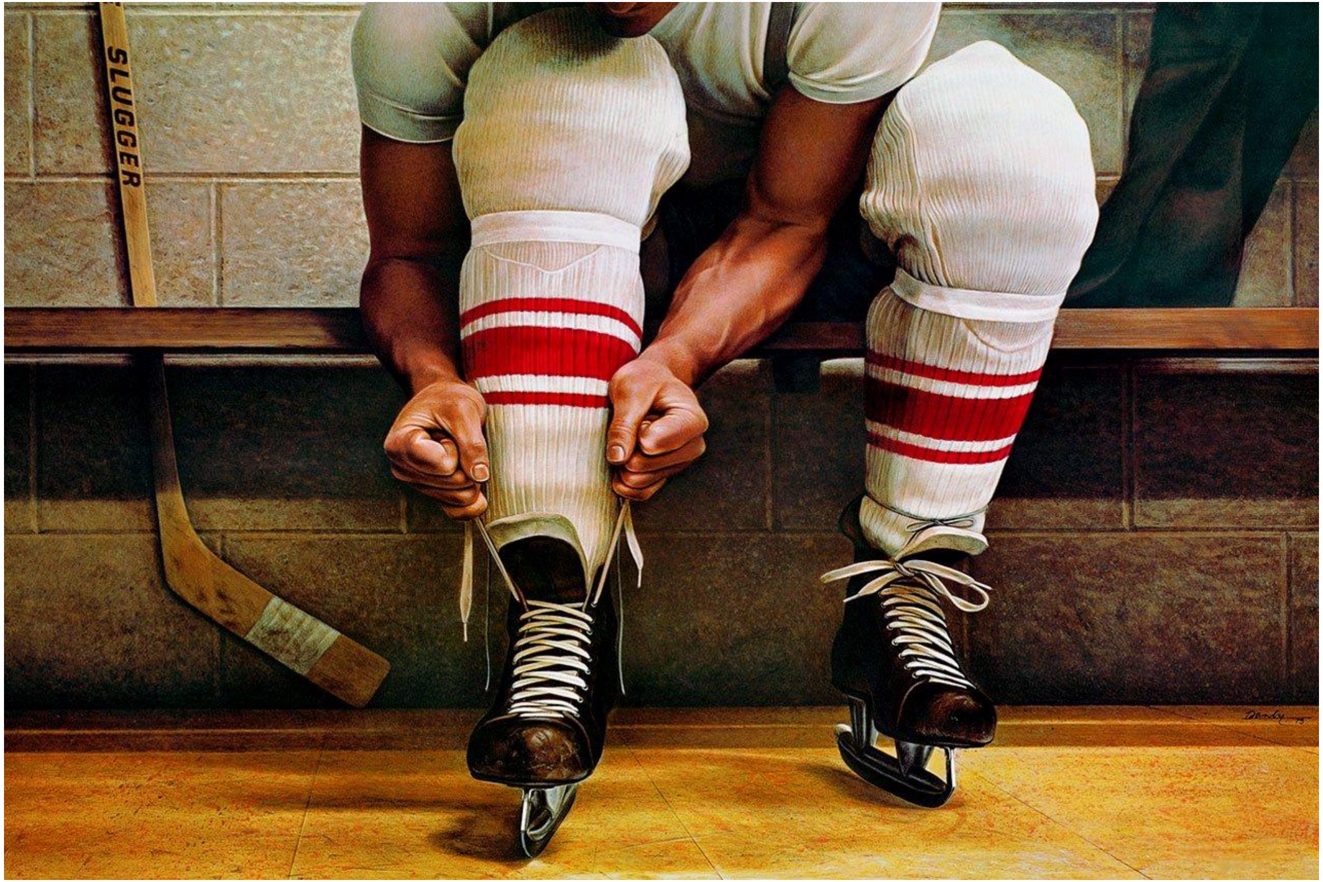
Fx

- Mental Model
- Direct Manipulation
- Action/Reaction, Cause/Effect
- Physics
- Quaternions
- Declarative not Imperative
- (UI History)

Ken Danby

Lacing Up

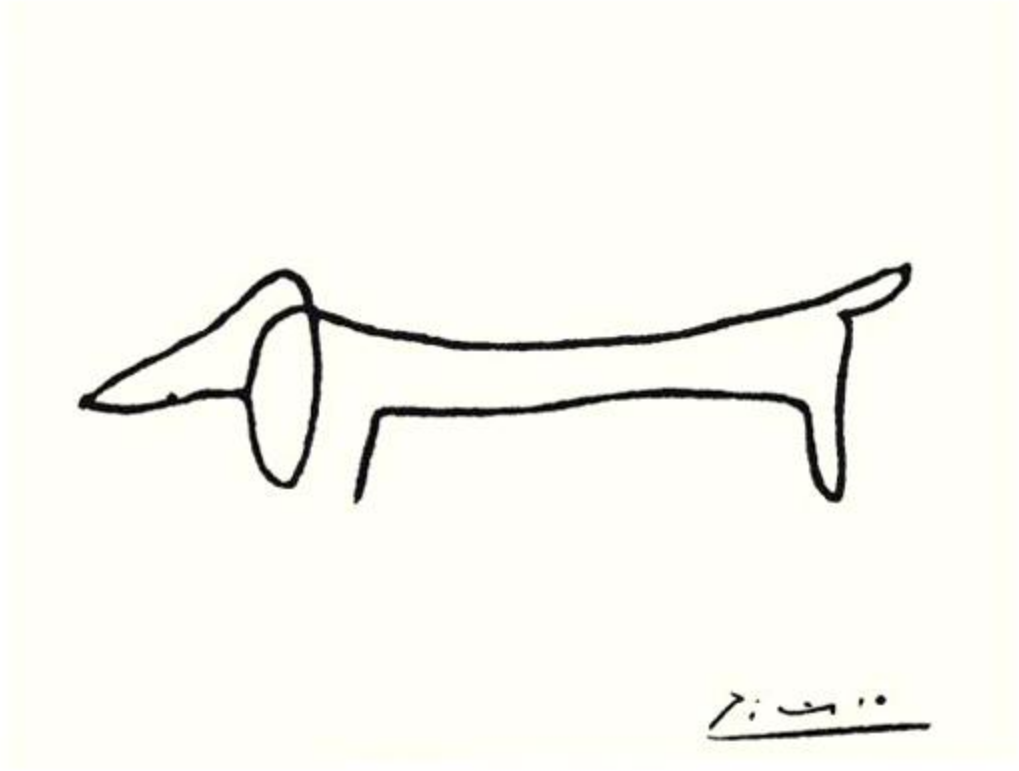
1973



Picasso

Le Chien

(“Lump”)



smile the senses

action reaction cause effect

what are these symbols

smile the senses

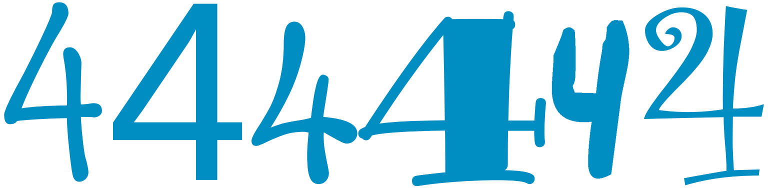
action reaction cause effect

what are these symbols

smile the senses

action reaction cause effect

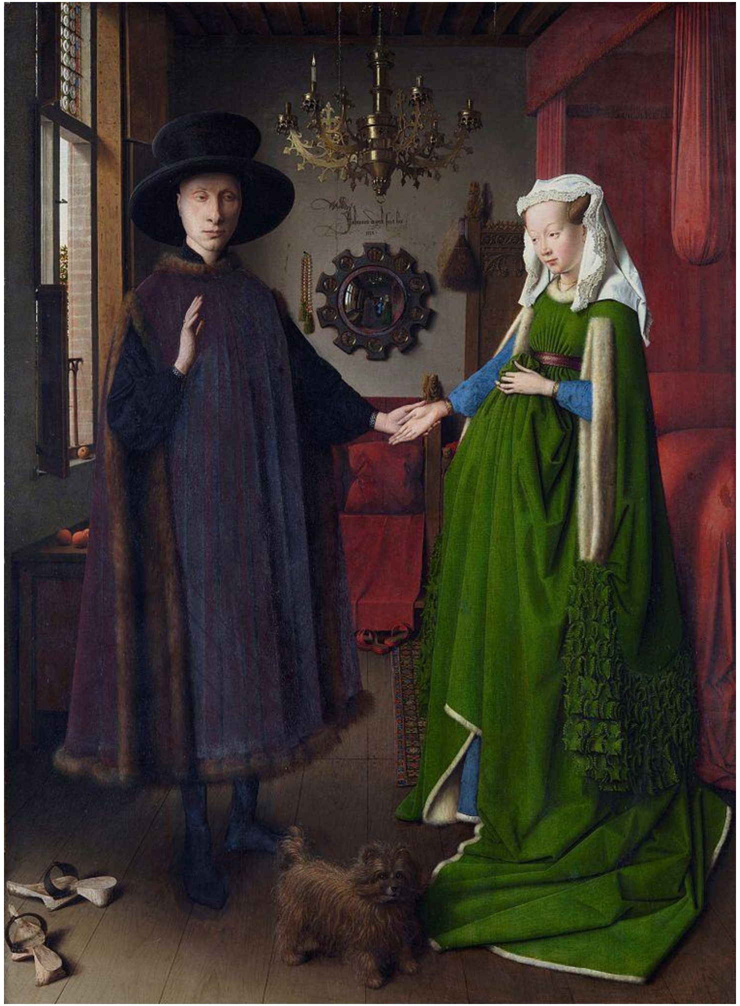
what are these symbols

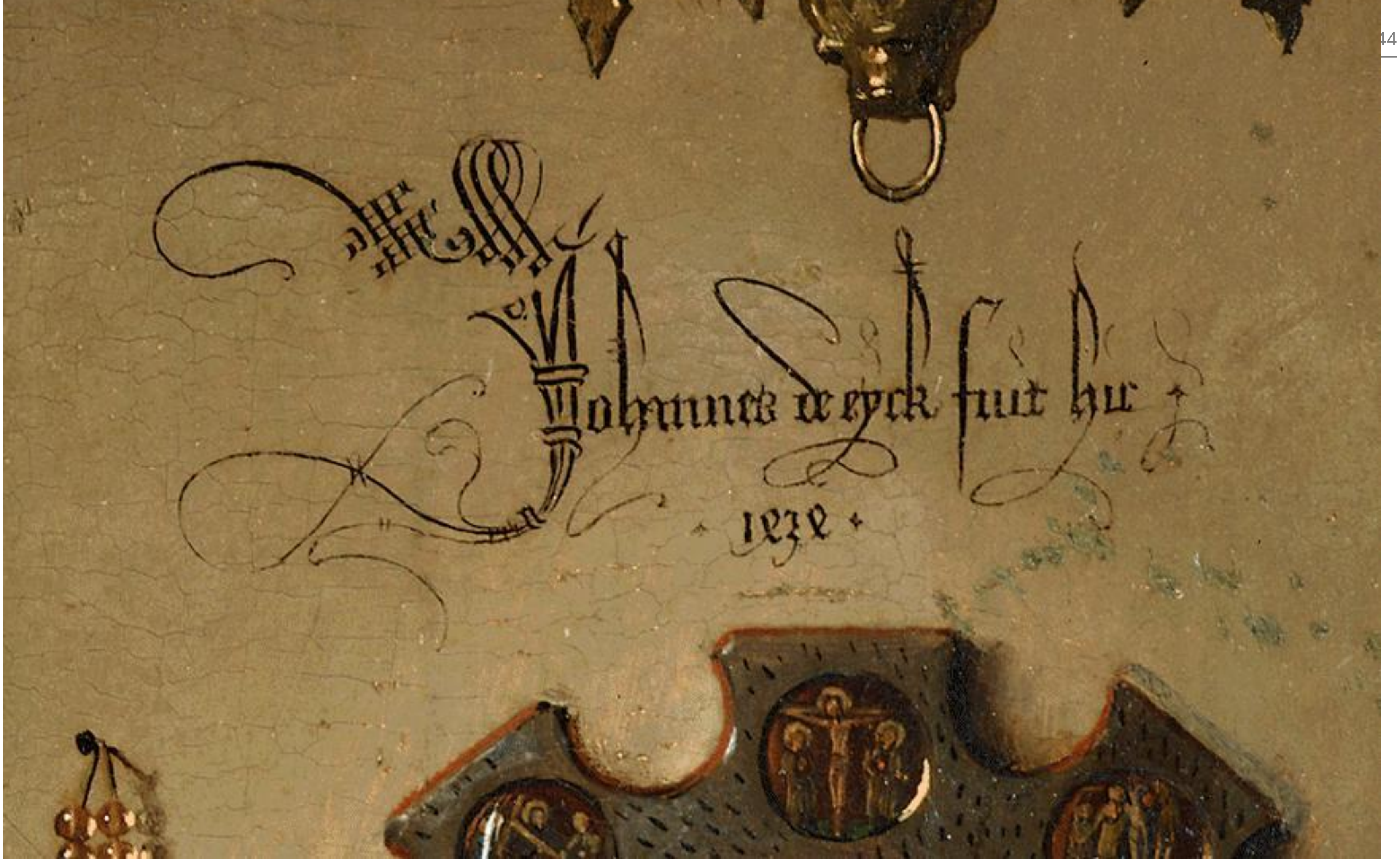


Jan von Eyck

The Arnolfini Portrait

1434

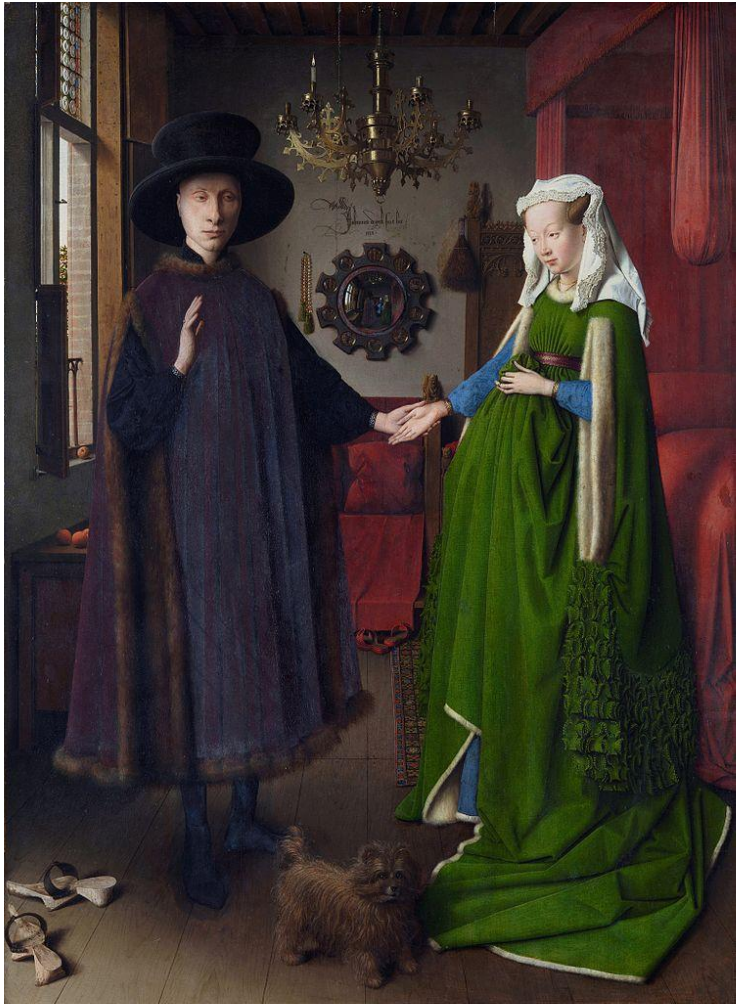


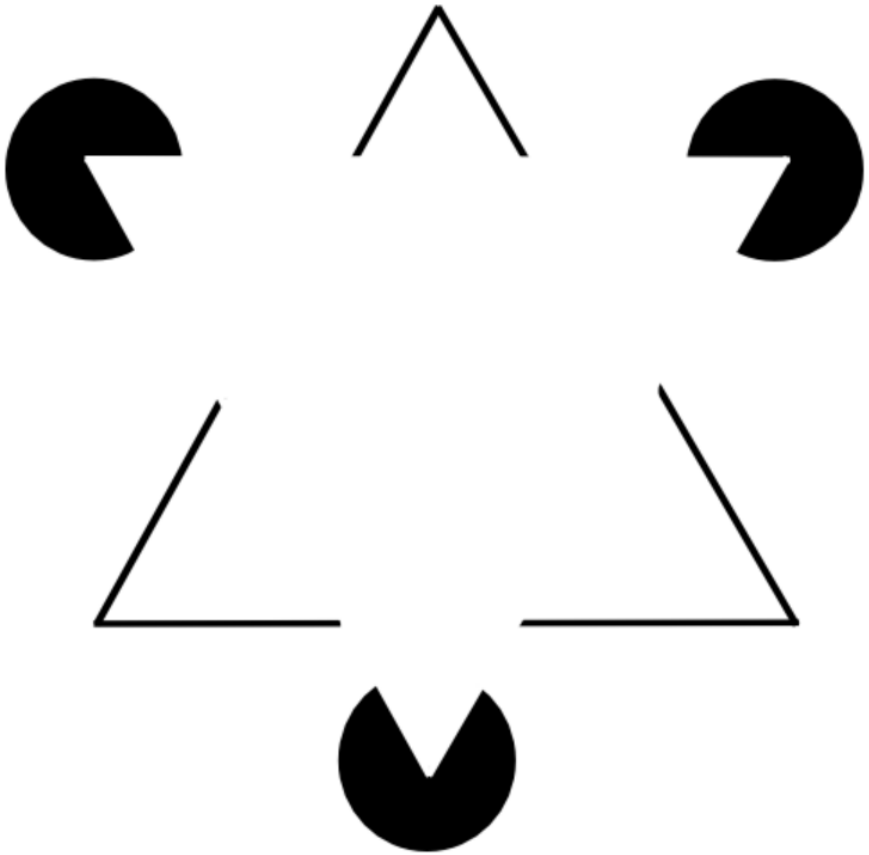

Johannes de eyck fuit hic
1434

Jan von Eyck

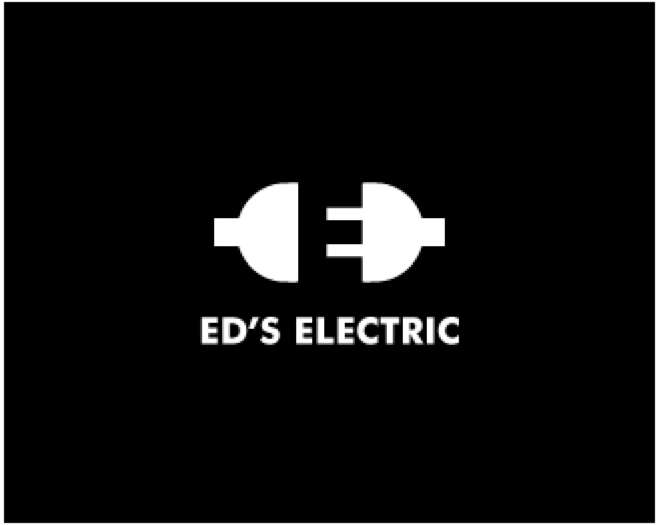
The Arnolfini Portrait

1434

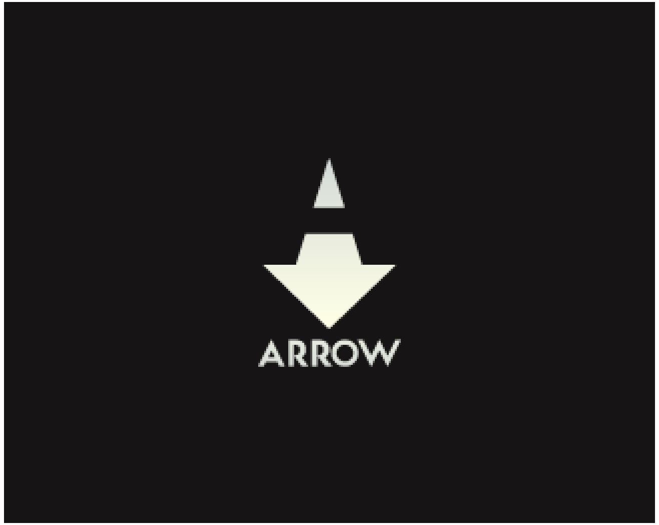




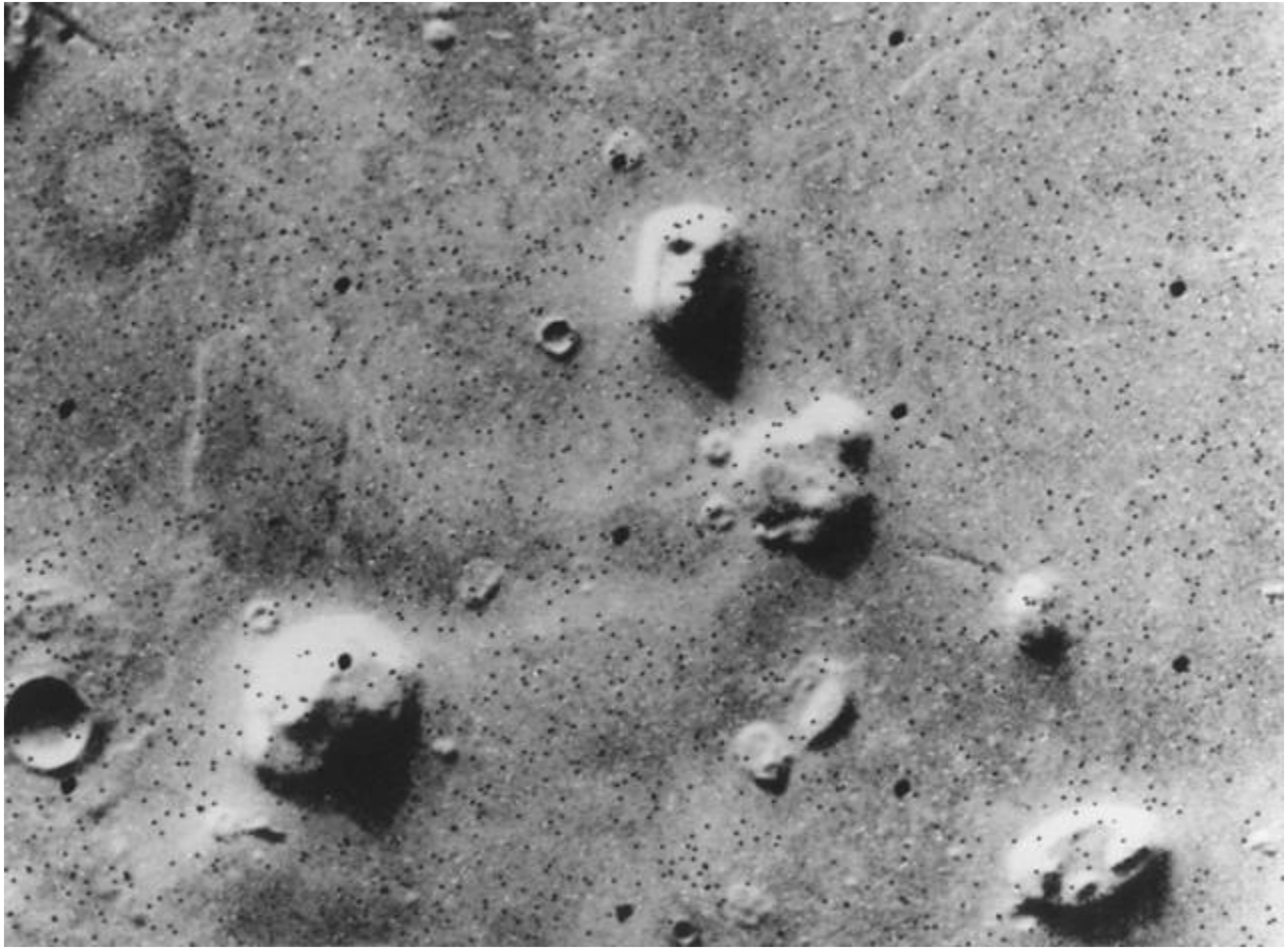
Kanizsa Triangle (1955)

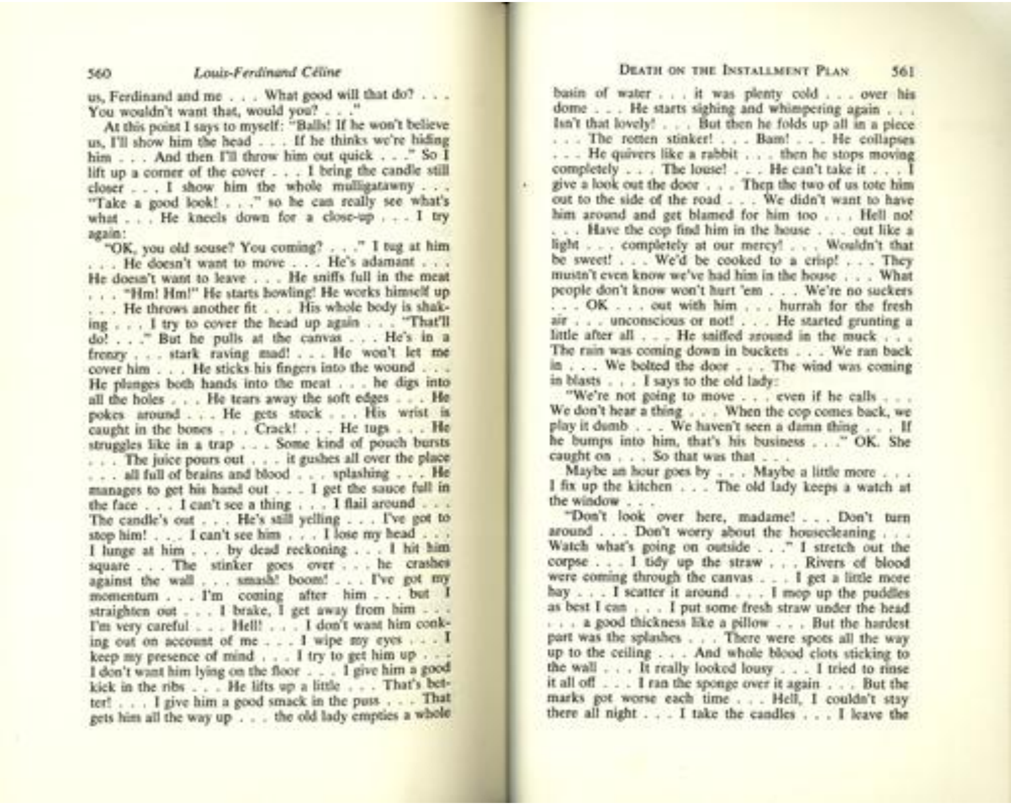


Siah-Design http://logopond.com/gallery/detail/47306



cream5 http://logopond.com/gallery/detail/175601





560 *Louis-Ferdinand Céline*

us, Ferdinand and me . . . What good will that do? . . . You wouldn't want that, would you? . . ."

At this point I says to myself: "Balls! If he won't believe us, I'll show him the head . . . If he thinks we're hiding him . . . And then I'll throw him out quick . . ." So I lift up a corner of the cover . . . I being the candle still closer . . . I show him the whole mulligatawny . . . "Take a good look! . . ." so he can really see what's what . . . He kneels down for a close-up . . . I try again:

"OK, you old souse? You coming? . . ." I tug at him . . . He doesn't want to move . . . He's adamant . . . He doesn't want to leave . . . He sniffs full in the meat . . . "Hm! Hm!" He starts howling! He works himself up . . . He throws another fit . . . His whole body is shaking . . . I try to cover the head up again . . . "That'll do! . . ." But he pulls at the canvas . . . He's in a frenzy . . . stark raving mad! . . . He won't let me cover him . . . He sticks his fingers into the wound . . . He plunges both hands into the meat . . . he digs into all the holes . . . He tears away the soft edges . . . He pokes around . . . He gets stuck . . . His wrist is caught in the bones . . . Crack! . . . He tugs . . . He struggles like in a trap . . . Some kind of pouch bursts . . . The juice pours out . . . it gushes all over the place . . . all full of brains and blood . . . splashing . . . He manages to get his hand out . . . I get the sauce full in the face . . . I can't see a thing . . . I flail around . . . The candle's out . . . He's still yelling . . . I've got to stop him! . . . I can't see him . . . I lose my head . . . I lunge at him . . . by dead reckoning . . . I hit him square . . . The stinker goes over . . . he crashes against the wall . . . smash! boom! . . . I've got my momentum . . . I'm coming after him . . . but I straighten out . . . I brake, I get away from him . . . I'm very careful . . . Hell! . . . I don't want him conking out on account of me . . . I wipe my eyes . . . I keep my presence of mind . . . I try to get him up . . . I don't want him lying on the floor . . . I give him a good kick in the ribs . . . He lifts up a little . . . That's better! . . . I give him a good smack in the puss . . . That gets him all the way up . . . the old lady empties a whole

DEATH ON THE INSTALLMENT PLAN 561

basin of water . . . it was plenty cold . . . over his dome . . . He starts sighing and whimpering again . . . Isn't that lovely! . . . But then he folds up all in a piece . . . The rotten stinker! . . . Bam! . . . He collapses . . . He quivers like a rabbit . . . then he stops moving completely . . . The louse! . . . He can't take it . . . I give a look out the door . . . Then the two of us tote him out to the side of the road . . . We didn't want to have him around and get blamed for him too . . . Hell no! . . . Have the cop find him in the house . . . out like a light . . . completely at our mercy! . . . Wouldn't that be sweet! . . . We'd be cooked to a crisp! . . . They mustn't even know we've had him in the house . . . What people don't know won't hurt 'em . . . We're no suckers . . . OK . . . out with him . . . hurrah for the fresh air . . . unconscious or not! . . . He started grunting a little after all . . . He sniffed around in the muck . . . The rain was coming down in buckets . . . We ran back in . . . We bolted the door . . . The wind was coming in blasts . . . I says to the old lady:

"We're not going to move . . . even if he calls . . . We don't hear a thing . . . When the cop comes back, we play it dumb . . . We haven't seen a damn thing . . . If he bumps into him, that's his business . . ." OK. She caught on . . . So that was that . . .

Maybe an hour goes by . . . Maybe a little more . . . I fix up the kitchen . . . The old lady keeps a watch at the window . . .

"Don't look over here, madame! . . . Don't turn around . . . Don't worry about the housecleaning . . . Watch what's going on outside . . ." I stretch out the corpse . . . I tidy up the straw . . . Rivers of blood were coming through the canvas . . . I get a little more hay . . . I scatter it around . . . I mop up the puddles as best I can . . . I put some fresh straw under the head . . . a good thickness like a pillow . . . But the hardest part was the splashes . . . There were spots all the way up to the ceiling . . . And whole blood clots sticking to the wall . . . It really looked lousy . . . I tried to rinse it all off . . . I ran the sponge over it again . . . But the marks got worse each time . . . Hell, I couldn't stay there all night . . . I take the candles . . . I leave the

Louis-Ferdinand Céline, *Mort à Crédit (Death on the Installment Plan)*. (1936)

Pattern Matching

Pattern Matching

cause and effect

+

pattern matching

Pattern Matching

cause and effect

+

pattern matching

+

substitution

Pattern Matching

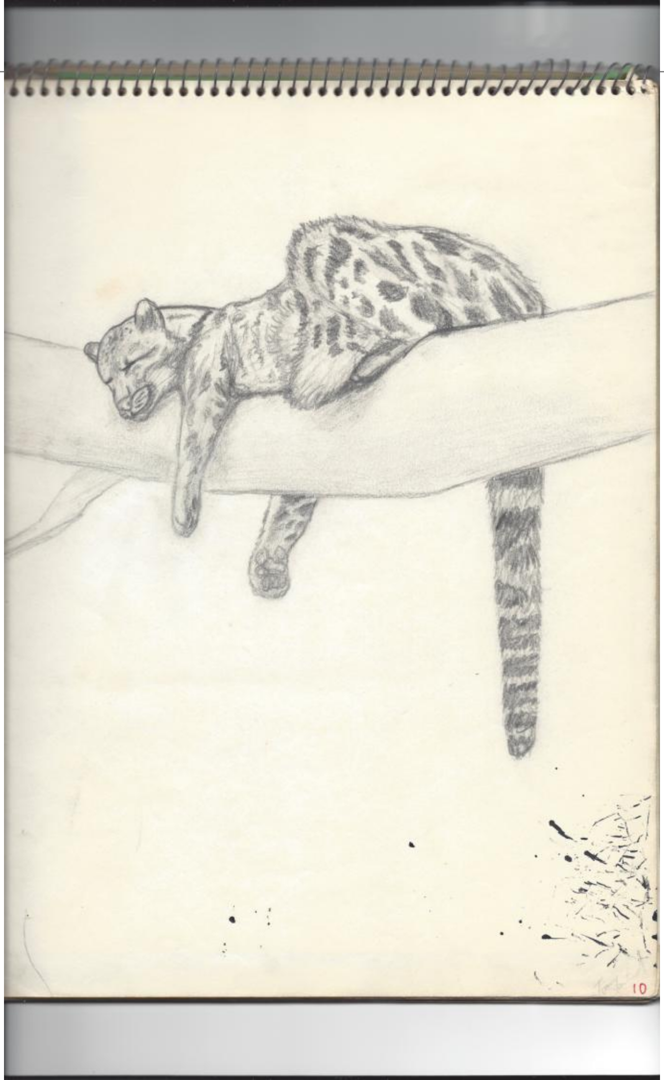
We are hard-wired for

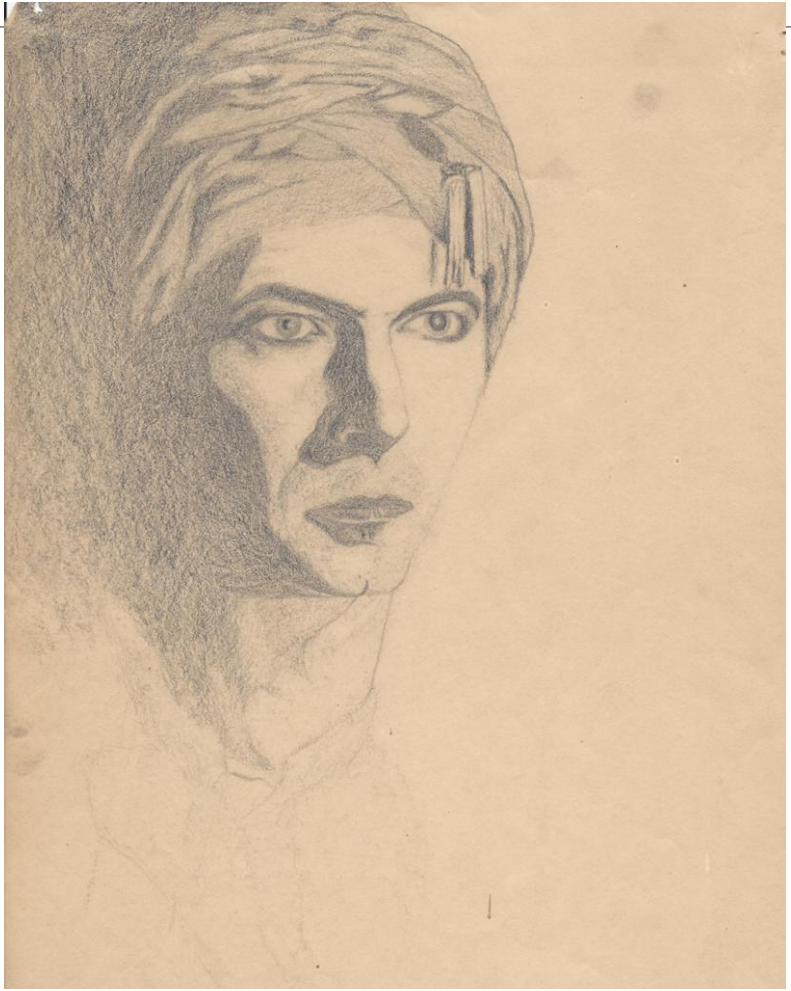
cause and effect

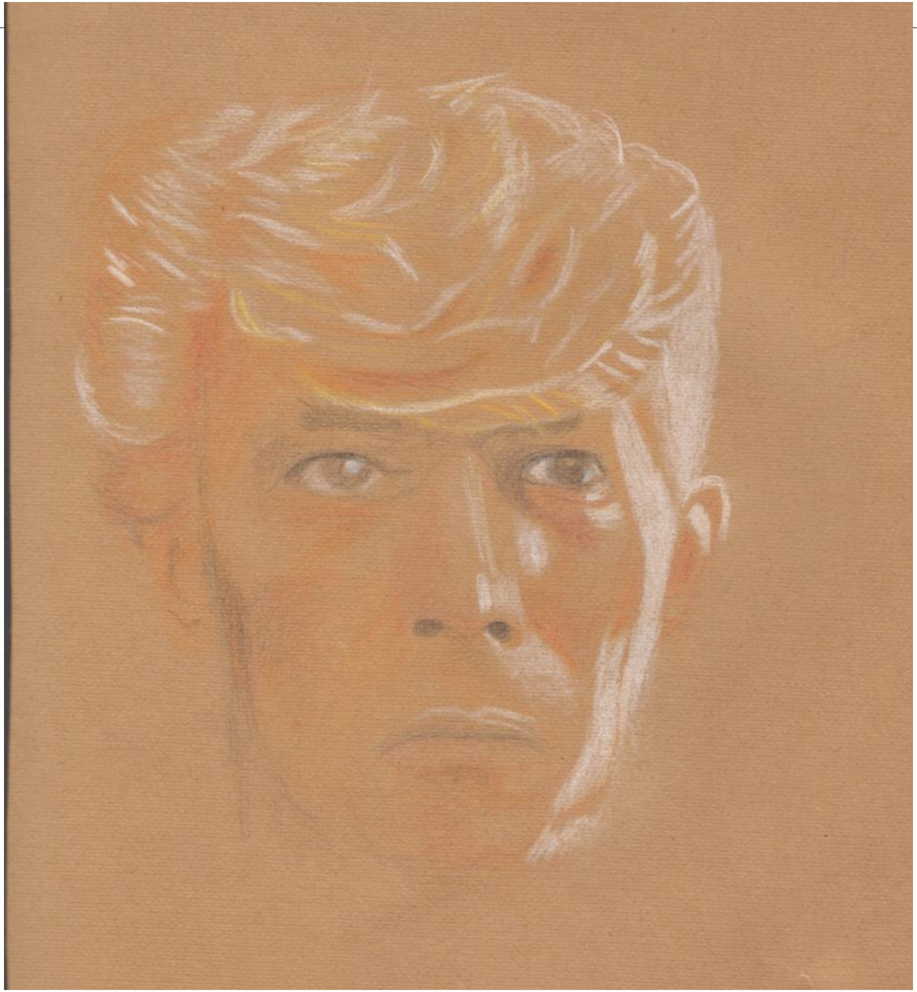
pattern matching

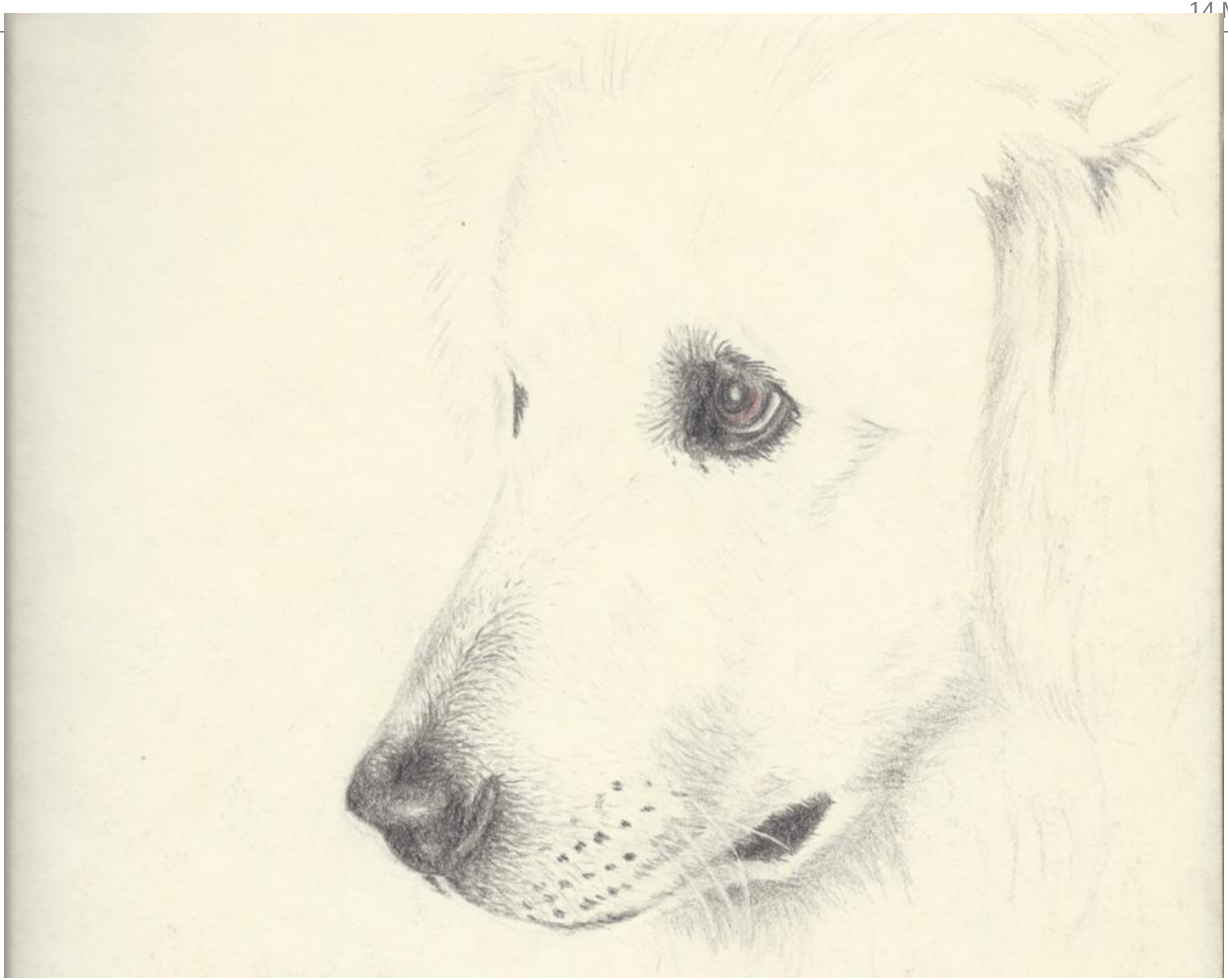
substitution

Negative Space Again… (traditional)


10







http://ericniebler.com

smile the senses

action reaction cause effect

what are these symbols

smile the senses

action reaction cause effect

what are these symbols

```
while (*d++ = *s++)
   ;
```

The Last Button…

The Last Button…

- virtual Button::invokeAction()
- virtual Invokeable::invokeAction() // Button : Invokeable
- member.invokeAction() // Button<T> has a T member.
- (*invokeAction)(theirdata) // C style
- _listener->invokeAction()
- boost/std::function
- boost::signal<>, Qt signal, framework signal
- os/framework_sendmessage(destId, buttonId, actionId, etc)
- os/framework_postmessage(destId, buttonId, actionId, etc) //(async)
- queue a boost::function to a threaded work queue //(async)
- condvar .notify() //(async)

- invokeAction()

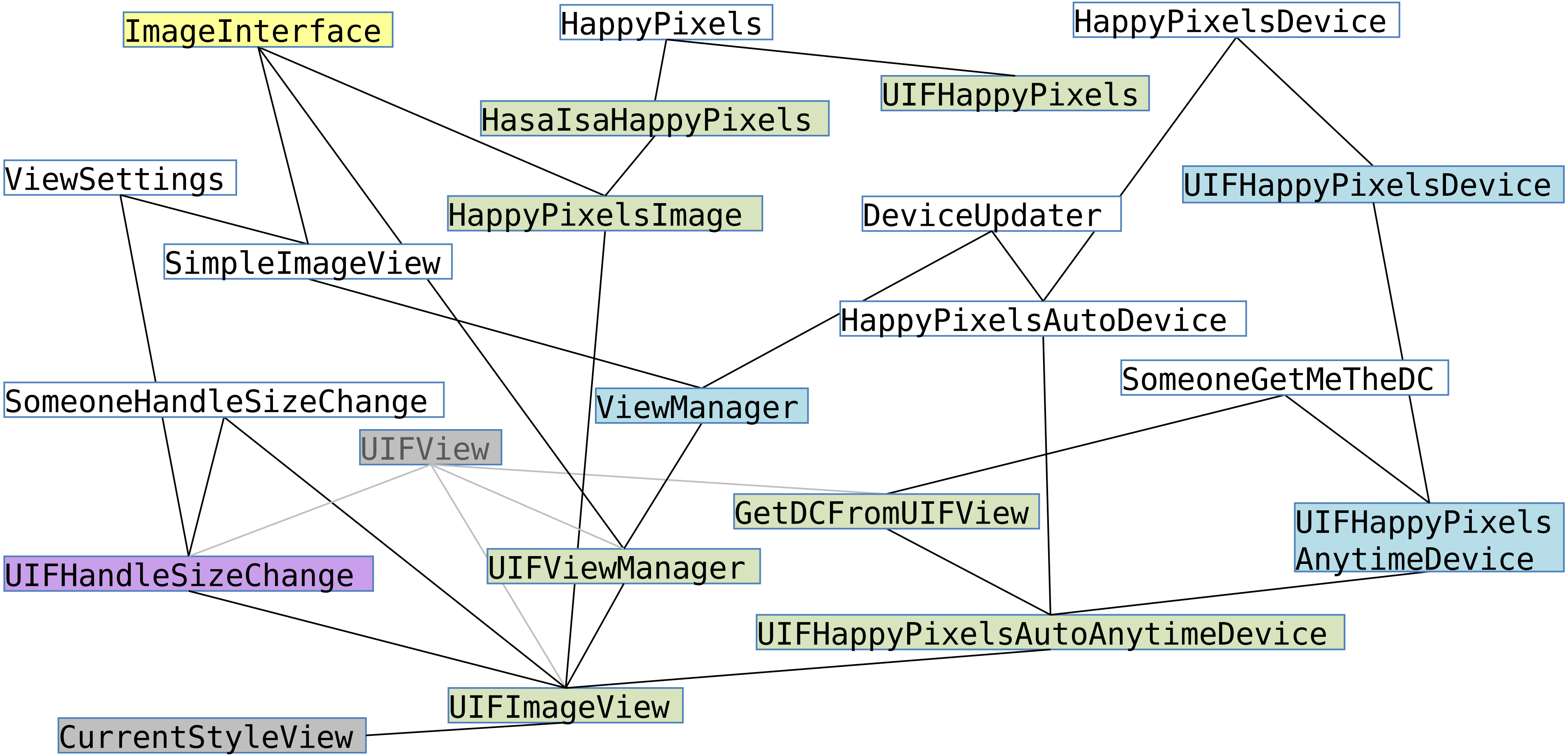
The Last Button…

```
template <typename ActionFramework>
class Button : public ActionFramework   // use CRTP?
{
   ...
   void handleInput(...)
   {
      if (...decide to invoke...)
      {
         invokeAction(*this);
      }
   }
};
```

The Last Button…

```
template <typename ActionFramework>
class Button : public ActionFramework   // use CRTP?
{
    ...
    void handleInput(...)
    {
        if (...decide to invoke...)
        {
            invokeAction(*this);
        }
    }
};
```

← This is not button’s job either


ImageInterface
HappyPixels
HappyPixelsDevice
UIFHappyPixels
HasaIsaHappyPixels
ViewSettings
UIFHappyPixelsDevice
HappyPixelsImage
DeviceUpdater
SimpleImageView
HappyPixelsAutoDevice
SomeoneGetMeTheDC
SomeoneHandleSizeChange
ViewManager
UIFView
GetDCFromUIFView
UIFHappyPixels
AnytimeDevice
UIFHandleSizeChange
UIFViewManager
UIFHappyPixelsAutoAnytimeDevice
UIFImageView
CurrentStyleView

Checkbox::setChecked()...

```
class CheckBox
{
   bool _checked;
public:
   void setChecked(bool checked)
   {
       _checked = checked;
       redraw();
   }
   ...
};
```

Checkbox::setChecked()…

- “The interface is the implementation”

Checkbox::setChecked()…

- “The interface is the implementation”
- Wrong.

Checkbox::setModel(model)…

Model View Controller

- Trygve Reenskaug, Xerox PARC, 1978.
- "A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80“, Glenn Krasner and Stephen Pope (1988, Journal Of Object Oriented Programming).

Model View Controller

- The data.
- Maintains invariants
- Interrelationships between data
- eg Script vs UI

Model View Controller

- The data.
- Maintains invariants
- Interrelationships between data
- **The User’s Mental Model**

Model View Controller

- Displays the data.
- May be Filtered.
- May be multiple views.

Model View Controller

Model View Controller

- Trygve Reenskaug:

“Jim Althoff and others implemented a version of MVC for the Smalltalk-80 class library after I had left Xerox PARC; I was in not involved in this work**. Jim Althoff uses the term Controller somewhat differently from me**.”

Model View Controller

- Handles User Input
- Attempts to Modify Model.

Model View Controller

- Handles User Input
- Attempts to Modify Model.
- Cause-Effect

Model View Controller

The Last Button…

```
template <typename ActionFramework>
class Button : ActionFramework   // use CRTP?
{
   ...
   void handleInput(...)
   {
      if (...decide to invoke...) ← not completely button’s job either
      {
         invokeAction(*this);
      }
   }
};
```

Model

- ASL - Sean Parent …

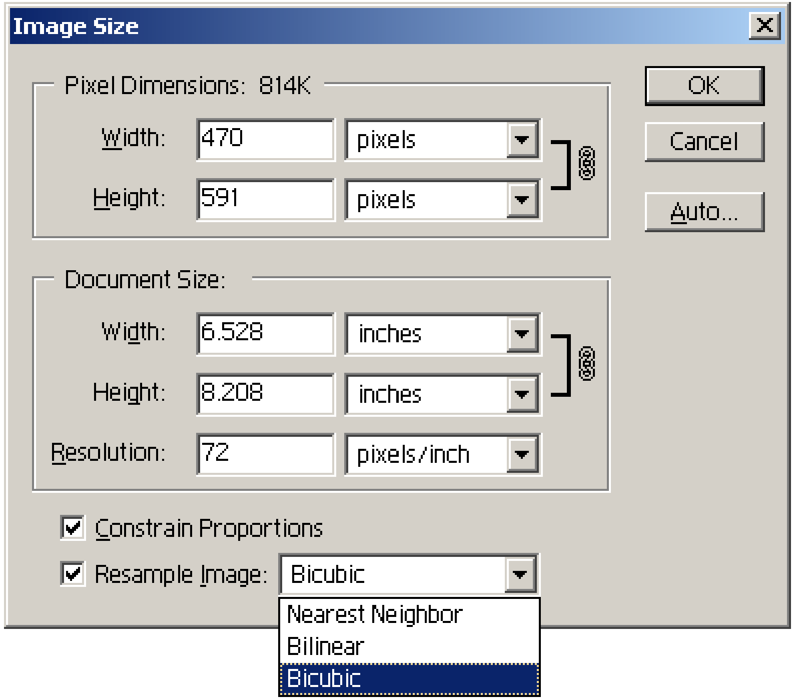


Image Size

Pixel Dimensions: 814K

Width: 470 pixels

Height: 591 pixels

Document Size:

Width: 6.528 inches

Height: 8.208 inches

Resolution: 72 pixels/inch

☑ Constrain Proportions

☑ Resample Image: Bicubic

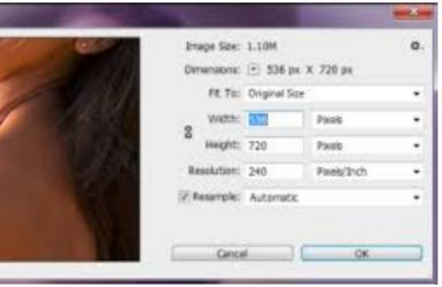


Image Size: 1.10M
Dimensions: 536 px X 720 px
Fit To: Original Size
Width: 536 Pixels
Height: 720 Pixels
Resolution: 240 Pixels/Inch
☑ Resample: Automatic

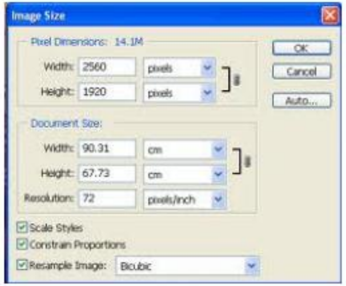


Image Size
Pixel Dimensions: 14.1M
Width: 2560 pixels
Height: 1920 pixels
Document Size:
Width: 90.31 cm
Height: 67.73 cm
Resolution: 72 pixels/inch
☑ Scale Styles
☑ Constrain Proportions
☑ Resample Image: Bicubic

Image Size
Learn more about: Image Size
Pixel Dimensions: 217.6K
Width: 322 pixels
Height: 230 pixels
Document Size:
Scale Styles
☑ Constrain Proportions
Resample Image:
Bicubic (best for smooth gradients)

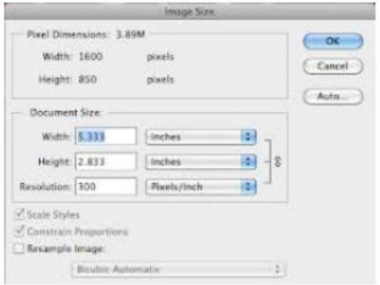


Image Size
Pixel Dimensions: 3.89M
Width: 1600 pixels
Height: 850 pixels
Document Size:
Width: 5.333 Inches
Height: 2.833 Inches
Resolution: 300 Pixels/Inch
☑ Scale Styles
☑ Constrain Proportions
□ Resample Image:
Bicubic Automatic

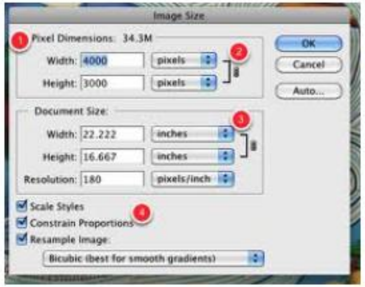


Image Size
Pixel Dimensions: 34.3M
Width: 4000 pixels
Height: 3000 pixels
Document Size:
Width: 22.222 inches
Height: 16.667 inches
Resolution: 180 pixels/inch
☑ Scale Styles
☑ Constrain Proportions
☑ Resample Image:
Bicubic (best for smooth gradients)

Resize and Skew
Resize
By: ● Percentage ○ Pixels
Horizontal: 100
Vertical: 100
☑ Maintain aspect ratio
Skew (Degrees)
Horizontal: 0
Vertical: 0

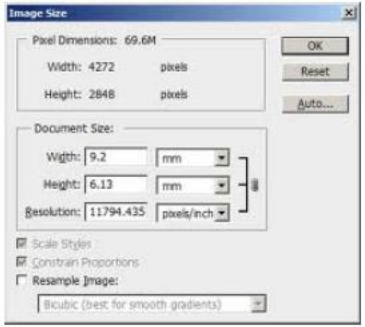


Image Size
Pixel Dimensions: 69.6M
Width: 4272 pixels
Height: 2848 pixels
Document Size:
Width: 9.2 mm
Height: 6.13 mm
Resolution: 11794.435 pixels/inch
☑ Scale Styles
☑ Constrain Proportions
□ Resample Image:
Bicubic (best for smooth gradients)

Export New Files

Image Processor

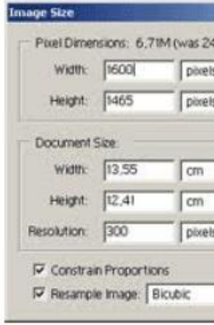


Image Size
Pixel Dimensions: 6,71M (was 2
Width: 1600 pixel
Height: 1465 pixel
Document Size:
Width: 13,55 cm
Height: 12,41 cm
Resolution: 300 pixel
☑ Constrain Proportions
☑ Resample Image: Bicubic

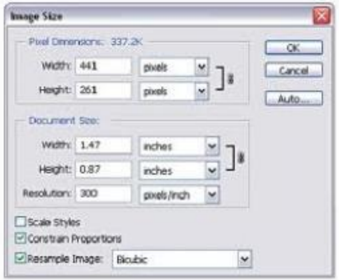


Image Size
Pixel Dimensions: 337.2K
Width: 441 pixels
Height: 261 pixels
Document Size:
Width: 1.47 inches
Height: 0.87 inches
Resolution: 300 pixels/inch
□ Scale Styles
☑ Constrain Proportions
☑ Resample Image: Bicubic

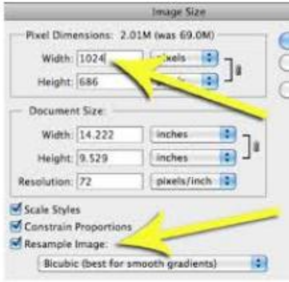
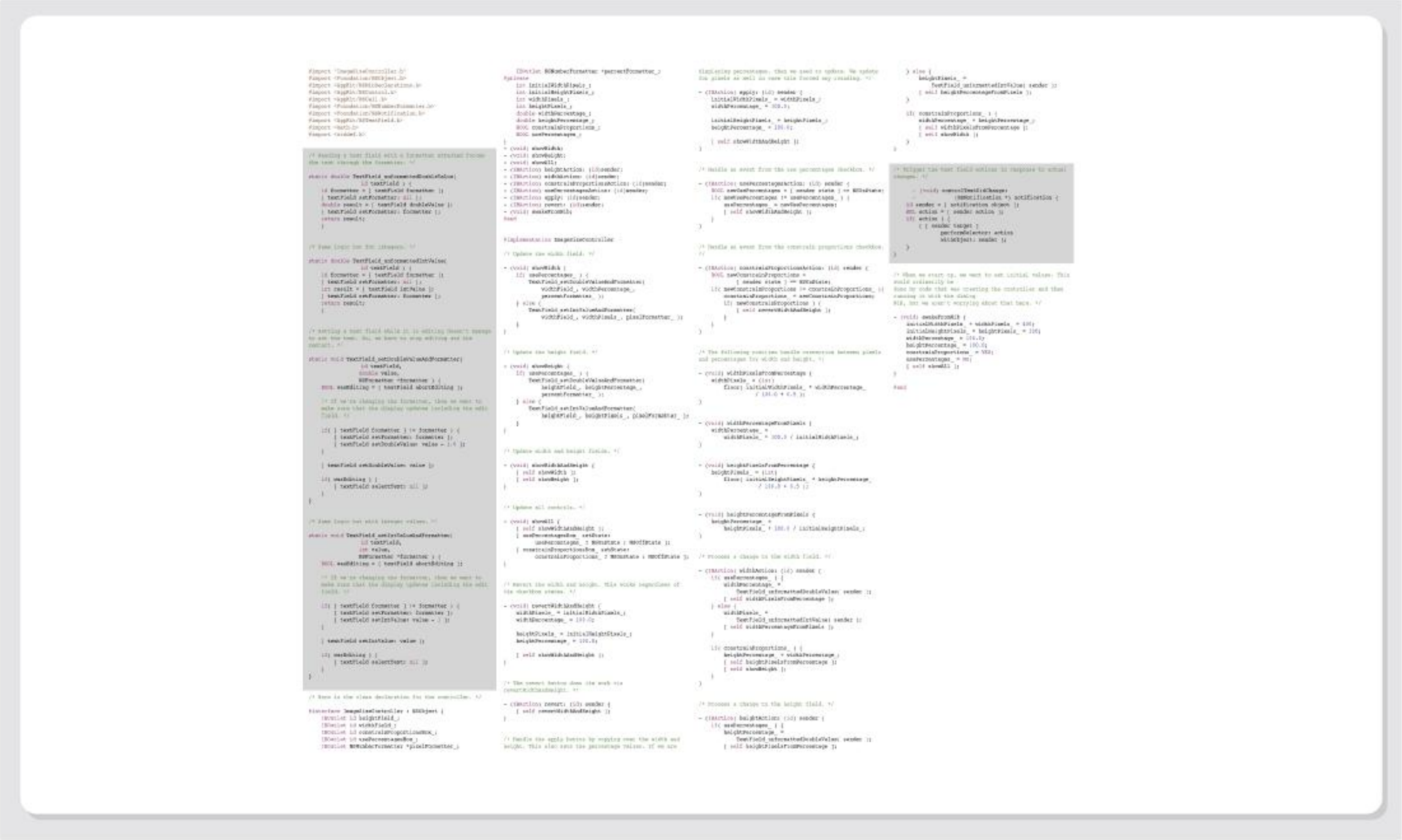


Image Size
Pixel Dimensions: 2.01M (was 69.0M)
Width: 1024 pixels
Height: 686
Document Size:
Width: 14.222 inches
Height: 9.529 inches
Resolution: 72 pixels/inch
☑ Scale Styles
☑ Constrain Proportions
☑ Resample Image:
Bicubic (best for smooth gradients)

Imperative Solution to Mini-Image Size

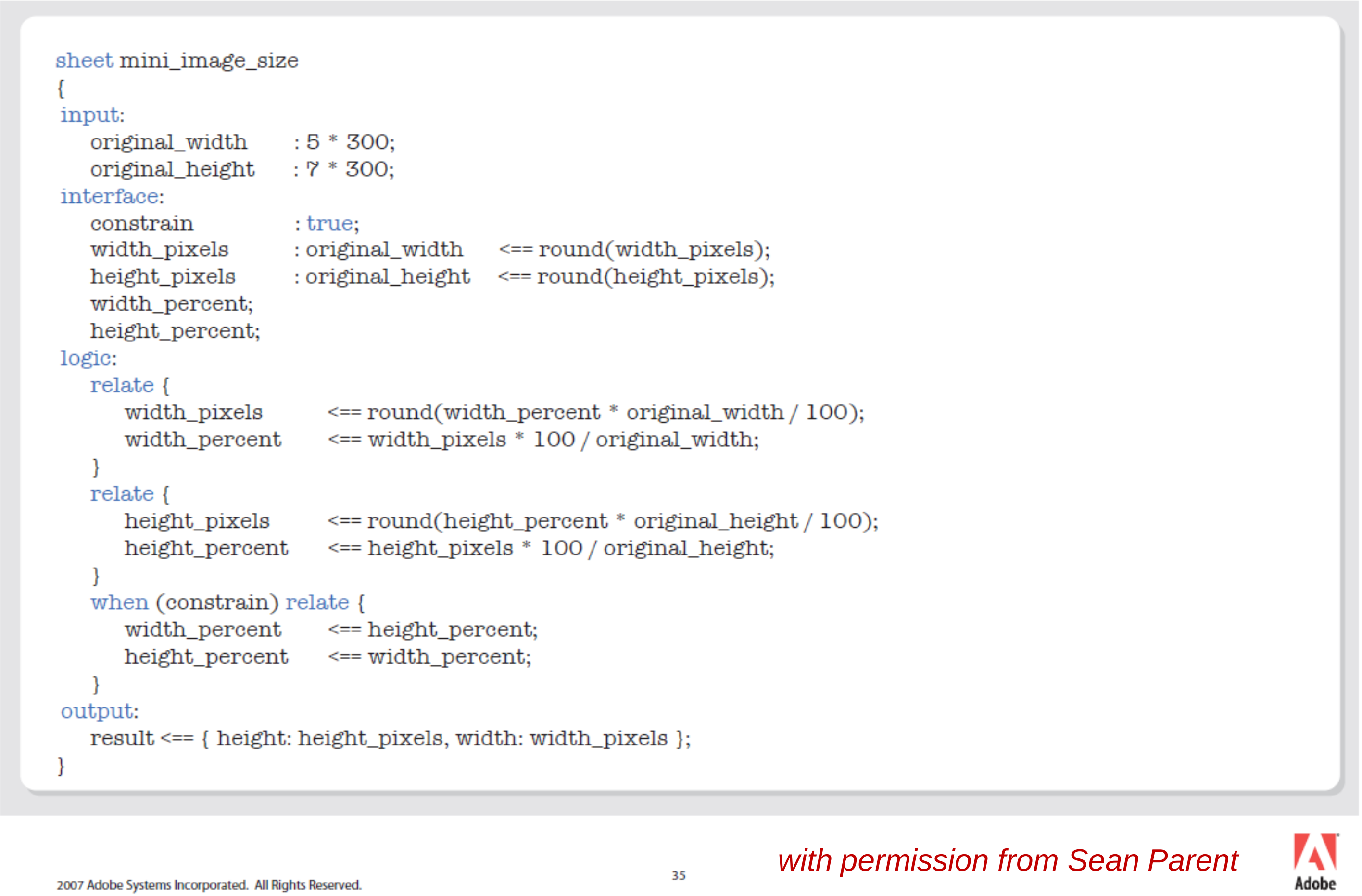


```
[illegible]
```

with permission from Sean Parent

Adobe

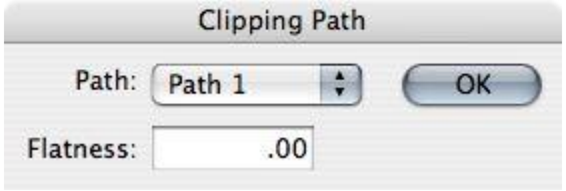
Declarative Solution using the Property Model Library



```
sheet mini_image_size
{
 input:
    original_width       : 5 * 300;
    original_height      : 7 * 300;
 interface:
    constrain            : true;
    width_pixels         : original_width     <== round(width_pixels);
    height_pixels        : original_height    <== round(height_pixels);
    width_percent;
    height_percent;
 logic:
    relate {
        width_pixels       <== round(width_percent * original_width / 100);
        width_percent      <== width_pixels * 100 / original_width;
    }
    relate {
        height_pixels      <== round(height_percent * original_height / 100);
        height_percent     <== height_pixels * 100 / original_height;
    }
    when (constrain) relate {
        width_percent      <== height_percent;
        height_percent     <== width_percent;
    }
 output:
    result <== { height: height_pixels, width: width_pixels };
}
```

with permission from Sean Parent

View

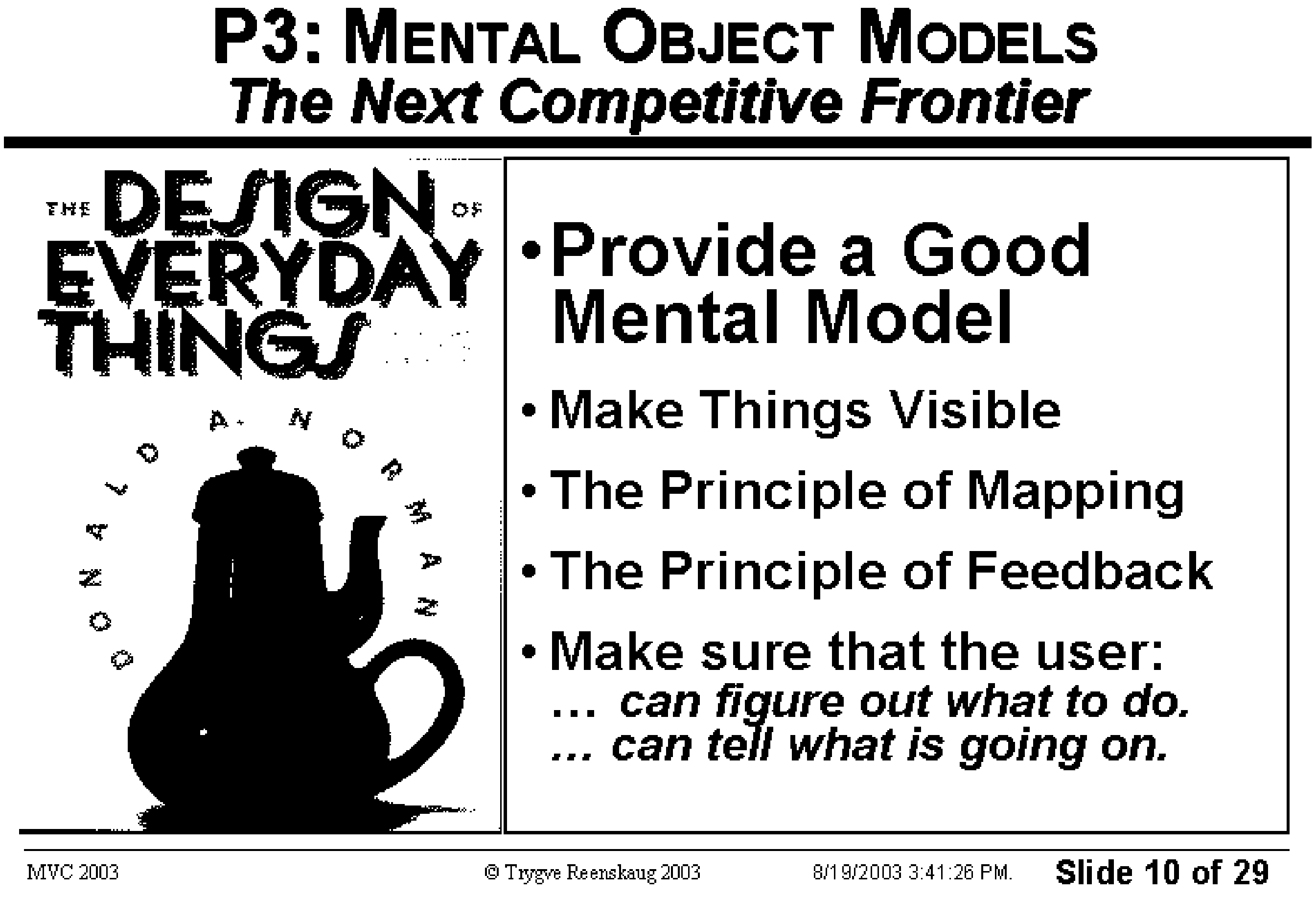


```
view dialog(name: "Clipping Path")
{
    column(child_horizontal: align_fill)
    {
        popup(name: "Path:", bind: @path, items:
        [
            { name: "None", value: empty },
            { name: "Path 1", value: 1 },
            { name: "Path 2", value: 2 }
        ]);
        edit_number(name: "Flatness:", digits: 9, bind: @flatness);
    }
    button(name: "OK", default: true, bind: @result);
}
```

github.com/stlab !

Nothing New

Trygve Reenskaug, 2003…



THE DESIGN OF EVERYDAY THINGS
DONALD A. NORMAN

P3: Mental Object Models

The Next Competitive Frontier

- Provide a Good Mental Model
- Make Things Visible
- The Principle of Mapping
- The Principle of Feedback
- Make sure that the user:

  … *can figure out what to do.*

  … *can tell what is going on.*

MVC 2003 © Trygve Reenskaug 2003 8/19/2003 3:41:26 PM. **Slide 10 of 29**

Nothing New

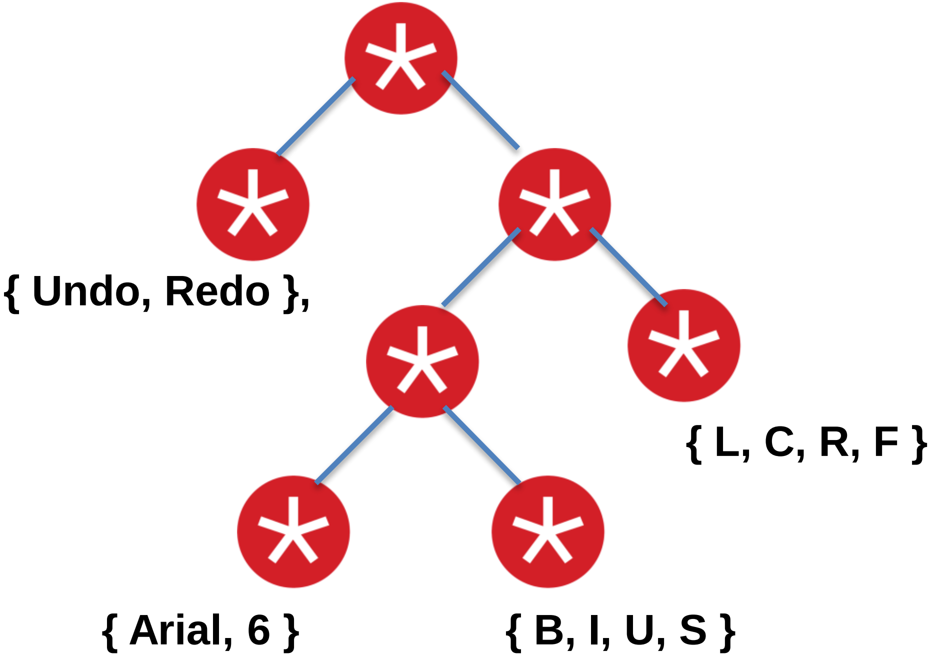
- MVC – Trygve Reenskaug, 1978
- The Design of Everyday Things – Donald A. Norman, 1988

Experimental

Very Experimental

```
return
    WrapGroup() [
      WrapGroup() [
        Button("Undo  <-"),
        Button("Redo  ->")
      ],
      WrapGroup() [  // text settings
        WrapGroup() [ // char settings
          WrapGroup() [  // font
            DropList("Arial", "Helvetica", "Times"),
            DropList("6", "8", "10", "12", "16", "24")
          ],
          WrapGroup() [  // font style (Bold/Italic/Underline/StrikeThrough)
            Button("B"),
            Button("I"),
            Button("U"),
            Button("S")
          ]
        ],
        WrapGroup() [  // paragraph settings (Left/Center/Right/Full)
          Button("L"),
          Button("C"),
          Button("R"),
          Button("F")
        ]
      ]
    ];
```

```
{
    { Undo, Redo },
    {
        {
            { Arial, 6 },
            { B, I, U, S },
        }
        { L, C, R, F }
    }
}
```

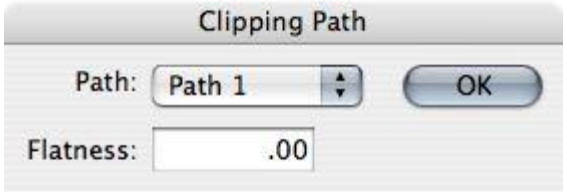


```
class Name : public QObject
{
    Q_PROPERTY(QString firstName READ firstName WRITE setFirstName NOTIFY firstNameChanged FINAL)
    Q_PROPERTY(QString lastName READ lastName WRITE setLastName NOTIFY lastNameChanged FINAL)
    …
};

class JobPosition : public QObject
{
    Q_PROPERTY(QString role READ role WRITE setRole NOTIFY roleChanged FINAL)
    Q_PROPERTY(QString group READ group WRITE setGroup NOTIFY roleChanged FINAL)
    …
};

class Address : public QObject
{
    Q_PROPERTY(QString street READ street WRITE setStreet NOTIFY streetChanged FINAL)
    Q_PROPERTY(QString city READ city WRITE setCity NOTIFY cityChanged FINAL)
    // province, country…
};

class Employee : public QObject
{
    Q_PROPERTY(int employeeId READ employeeId WRITE setEmployeeId NOTIFY employeeIdChanged FINAL)
    Q_PROPERTY(Name * name READ name WRITE setName NOTIFY nameChanged FINAL)
    Q_PROPERTY(JobPosition * position READ position WRITE setPosition NOTIFY positionChanged FINAL)
    Q_PROPERTY(Address * address READ address WRITE setAddress NOTIFY addressChanged FINAL)
};
```

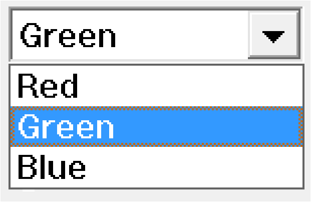


Clipping Path

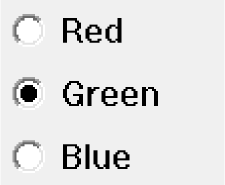
Path: Path 1

Flatness: .00

Label / Entrry “colon” alignment



interchangeable?



smile the senses

action reaction cause effect

what are these symbols

APIs – the other User Interface

APIs – the other User Interface

```
namespace bb {
namespace device {

class WifiDirect {

    WiFiDirectResult::Type requestConnect (const QString & address);
 …
 };
}
}
```

APIs – the other User Interface

From Caller’s point of view:

```
result = WifiDirect::requestConnect (address);
```

APIs – the other User Interface

From Caller’s point of view:

```
result = WifiDirect::requestConnect (address);
```

← iambic pentameter

APIs – the other User Interface

- Constraints
- Consistency
- Patterns vs Misplaced Patterns
- Mental Model
- Mapping
- Cause – Effect
- “can tell what’s going on”
- Substitution
- Versatility

APIs – the other User Interface

- Constraints
- Consistency
- Patterns vs Misplaced Patterns
- Mental Model
- Mapping
- Cause – Effect
- “can tell what’s going on”
- Substitution
- Versatility

- Don’t be Stupid
- Don’t be Clever

MACROS are EVIL

MACROS are EVIL

- Constraints
- Consistency
- Patterns vs Misplaced Patterns
- Mental Model
- Mapping
- Cause – Effect
- “can tell what’s going on”
- Substitution
- Versatility

- Don’t be Stupid
- Don’t be Clever

MACROS are EVIL

- Constraints
- Consistency
- Patterns vs Misplaced Patterns
- Mental Model
- Mapping
- Cause – Effect
- “can tell what’s going on”
- Substitution
- Versatility

- Don’t be Stupid
- Don’t be Clever

Templates?

DSELs?

MACROS are EVIL

- Constraints
- Consistency
- Patterns vs Misplaced Patterns
- Mental Model
- Mapping
- Cause – Effect
- “can tell what’s going on”
- Substitution
- Versatility

- Don’t be Stupid
- Don’t be Clever

Templates?

DSELs?

Cars?

Guidelines

Guidelines

- 130 +
- Meyers, Rule of 3 (5? 0?), …
- Qt
- Style (param order; yes even brackets, etc)
- API centric

Reference Types vs Value Types

- int

Reference Types vs Value Types

- Reference *Semantics*
- Value *Semantics*
- C++ let’s you decide
- Regular Types

- int

Reference Types vs Value Types

- int
- Rect
- string

Reference Types vs Value Types

- int
- Rect
- string
- copy

Reference Types vs Value Types

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- …

Reference Types vs Value Types

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- Math

Reference Types vs Value Types

```
y = x
print g(x)
print g(y)


print k(h(f(x), g(w)), h(f(x), g(w)))

z = h(f(x), g(w))
print k(z, z)
```

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- Math – Substitution/Reasoning

Reference Types vs Value Types

- Object, QObject,…

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- Math

Reference Types vs Value Types

- Object, QObject,…
- java

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- Math

Reference Types vs Value Types

- Object, QObject,…
- java
- non-copyable

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- Math

Reference Types vs Value Types

- Object, QObject,…
- java
- non-copyable
- objects, things – *changeable*

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- Math

Reference Types vs Value Types

- Object, QObject,…
- java
- non-copyable
- objects, things – *changeable*
- signals/slots – *observable*

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- Math

Reference Types vs Value Types

- Object, QObject,…
- java
- non-copyable
- objects, things – *changeable*
- signals/slots – *observable*
- Relationships

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- Math

Reference Types vs Value Types

- Object, QObject,…
- java
- non-copyable
- objects, things – *changeable*
- signals/slots – *observable*
- Relationships

- Steering Wheel Problem

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- Math

momo

Reference Types vs Value Types

- Object, QObject,…
- java
- non-copyable
- objects, things – *changeable*
- signals/slots – *observable*
- Relationships

- Steering Wheel Problem

- int
- Rect

```
y = x
print g(x)
print g(y)


print k(h(f(x), g(w)), h(f(x), g(w)))

z = h(f(x), g(w))
print k(z, z)
```

Reference Types vs Value Types

- Object, QObject,…
- java
- non-copyable
- objects, things – *changeable*
- signals/slots – *observable*
- Relationships

- Steering Wheel Problem

- OH NO!!! Pointers!!!

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- Math

Reference Types vs Value Types

- Object, QObject,…
- java
- non-copyable
- objects, things – *changeable*
- signals/slots – *observable*
- Relationships
- Steering Wheel Problem
- OH NO!!! Pointers!!!

- int
- Rect
- string
- copy
- Sean Parent (Adobe)
- Alex Stepanov (STL)
- Jon Lakos
- Math
- Oh, no pointers.

No Shallow Copies!

- copy or copy not; there is no shallow

No Shallow Copies!

- copy or copy not; there is no shallow

- What does + do?

```
X operator+(X a, X b);
```

No Shallow Copies!

- copy or copy not; there is no shallow

- What does + do?

  X operator+(X a, X b);

- + adds

No Shallow Copies!

- copy or copy not; there is no shallow

- What does + do?

```
X operator+(X a, X b);
```

- + adds

- What does = do?

```
X & operator=(X const & b);
```

No Shallow Copies!

- copy or copy not; there is no shallow

- What does + do?

  X operator+(X a, X b);

- + adds

- What does = do?

  X & operator=(X const & b);

- = copies

Word Aliasing

Word Aliasing

- operator=()

- close(f);
- close(p, q, delta);

- avoidTolls();
- avoidTollsEnabled();
- allowTollsEnabled();

- searchForContacts();
- searchForContactsEnabled();
- includeContacts();
- includeContactsEnabled();
- isIncludeContactsEnabled();

UI Constraints

- Display Size – physical/pixels
- Display Aspect (landscape/portrait/square)
- Display Distance
- … *Visual Acuity*
- Input Mechanisms
- Widget-set (pro and con!)
- Consistency
- UI “Police”
- Time!

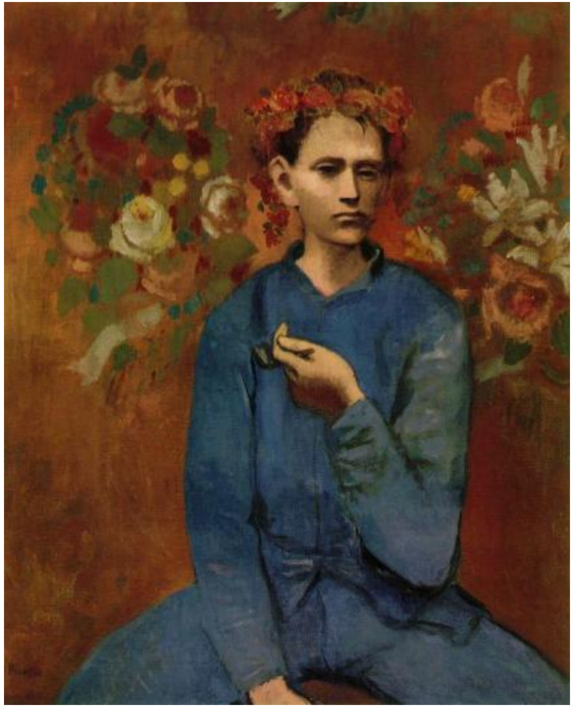
API Constraints

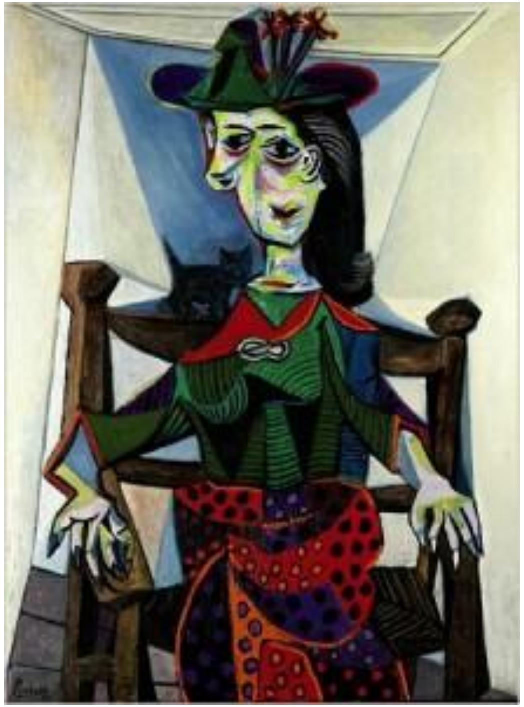
- Memory / CPU
- Language ambiguity/aliasing
- Terminology
- … *Coding Acuity*
- Input Mechanisms (const? ref? ptr?)
- set of types (pro and con!)
- Consistency – Patterns!
- API “Police” (me)
- Time!!!

BlackBerry® Q10 smartphone

Breaking the Rules

1905





1941

“Learn the rules like a pro, so you can break them like an artist.”

-- Pablo Picasso

Breaking the Rules

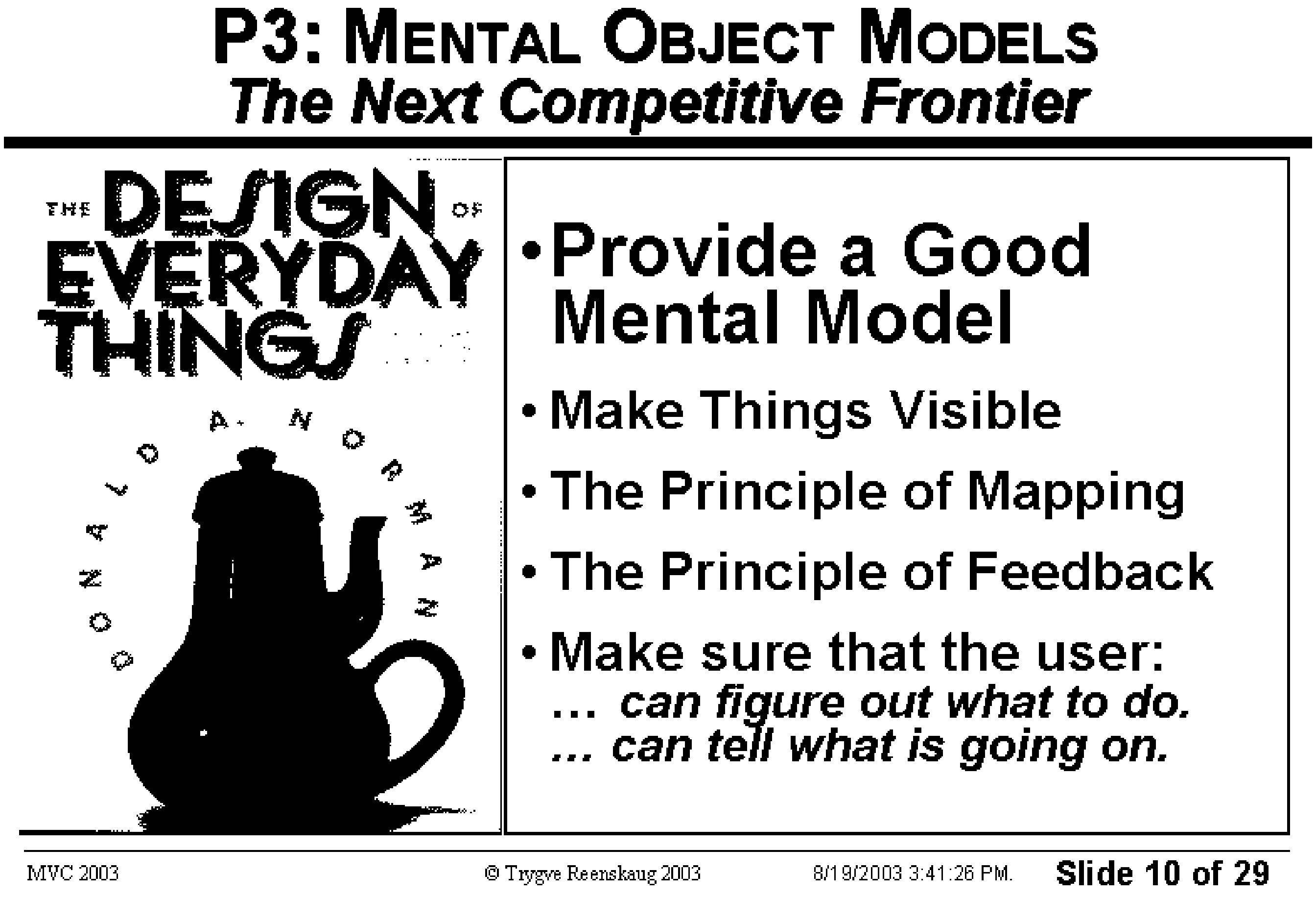
- *Brings focus to the broken rule*
- ... for better or worse

- eg:
- virtual functions in BB10 APIs
- naming – “HappyPixels”
- “is that what you meant to say”
- Pattern or Misplaced Pattern?

API is Art

API is Art

> “I was working on the proof of one of my poems all the morning, and took out a comma. In the afternoon I put it back again.”
>
> -- Oscar Wilde



P3: Mental Object Models

The Next Competitive Frontier

THE DESIGN OF EVERYDAY THINGS
DONALD A. NORMAN

- **Provide a Good Mental Model**
- Make Things Visible
- The Principle of Mapping
- The Principle of Feedback
- Make sure that the user:
  … can figure out what to do.
  … can tell what is going on.

MVC 2003 © Trygve Reenskaug 2003 8/19/2003 3:41:26 PM. **Slide 10 of 29**

smile the senses
action reaction cause effect
what are these symbols