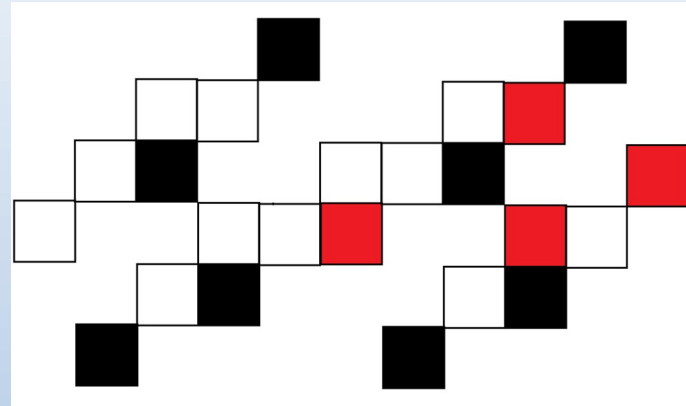


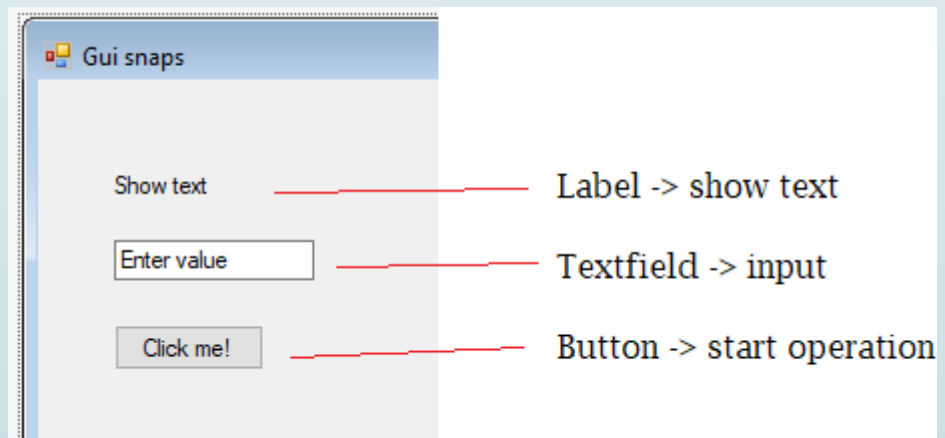
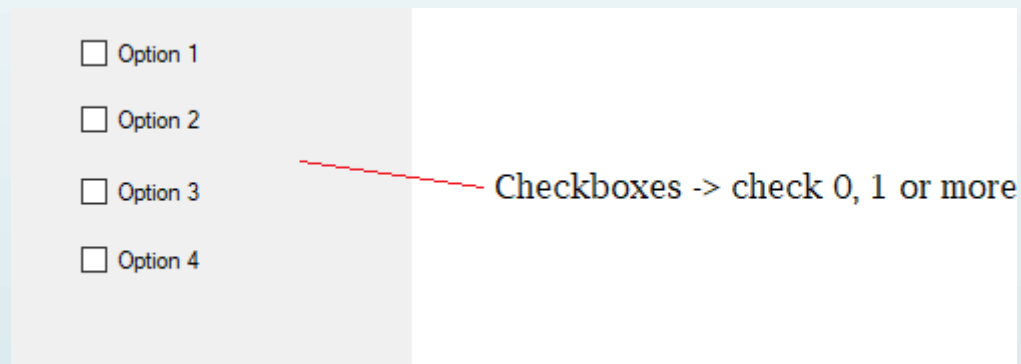
GUI & Usability Practical Hints



By
Adam Higherstein

Usability Snaps

Basic controls





☐ Choice 1

☐ Choice 2

☐ Choice 3

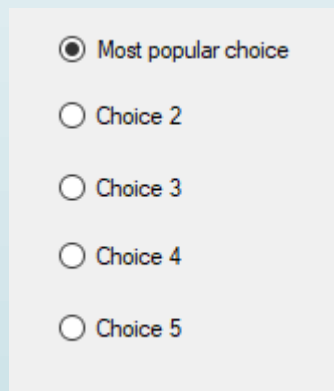
☐ Choice 4

☐ Choice 5

Radiobuttons -> check 0 or 1

When using checkboxes or radiobuttons, user does not need to type anything.

With radiobutton group the most popular option is added to the 1. place and checked.



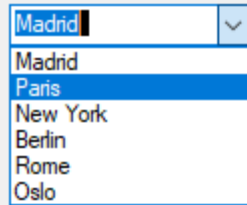
☒ Most popular choice

☐ Choice 2

☐ Choice 3

☐ Choice 4

☐ Choice 5

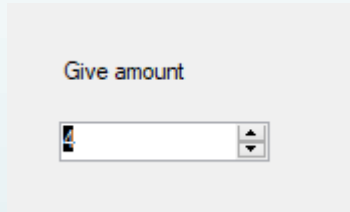


Listboxes -> when there are several options

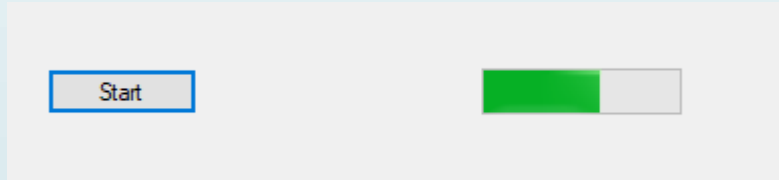
Many lines text ...

Textarea controls -> a lot of text

Steppers: limits can be defined and the step, user does not need to type anything and we avoid wrong values

A screenshot of a user interface element. It features a light gray rectangular background. At the top, the text "Give amount" is displayed in a small, dark font. Below the text is a numeric stepper control, which consists of a small square icon on the left, a white rectangular input field in the center, and a vertical column of four small arrows (up and down) on the right for incrementing and decrementing the value.

Progressbar tells the situation

A screenshot of a user interface element. It features a light gray rectangular background. On the left side, there is a rectangular button with a blue border and the word "Start" in the center. To the right of the button is a horizontal progress bar. The progress bar is a rectangle divided into two parts: the left portion is filled with a solid green color, and the right portion is a light gray color, indicating the current progress level.

Toggling (used in above example)

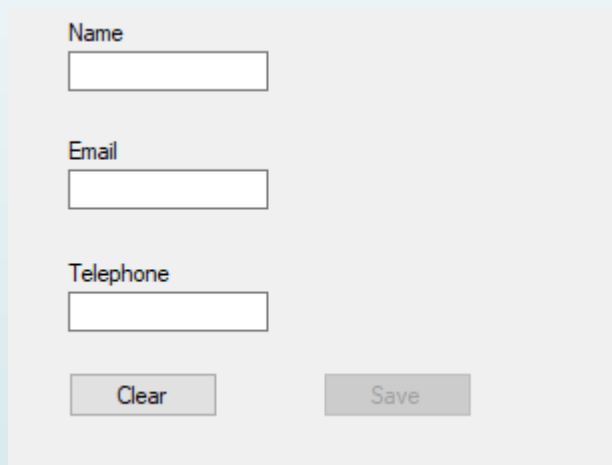
Use same control to e.g. start and stop timer:

```
private void button1_Click(object sender, EventArgs e)
{
    timer1.Enabled = !timer1.Enabled;

    if (timer1.Enabled == true)
        button1.Text = "Stop";
    else
        button1.Text = "Start";
}
```

Enabled or not

Example: when user has typed values to all 3 textboxes, Save buttons becomes enabled:

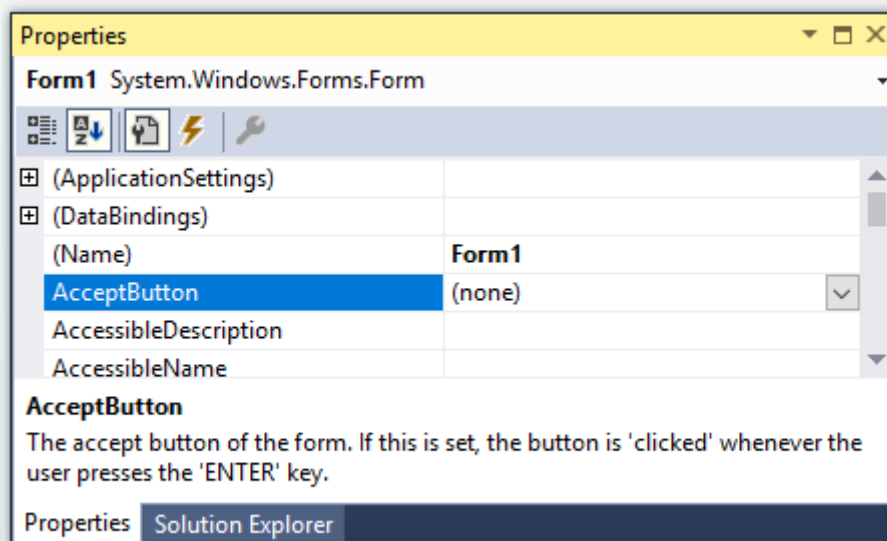


The image shows a light gray rectangular form. Inside the form, there are three text input fields stacked vertically. The first field is labeled "Name", the second "Email", and the third "Telephone". Below these fields are two buttons: "Clear" on the left and "Save" on the right. The "Save" button is disabled, indicated by its gray color and faded text.

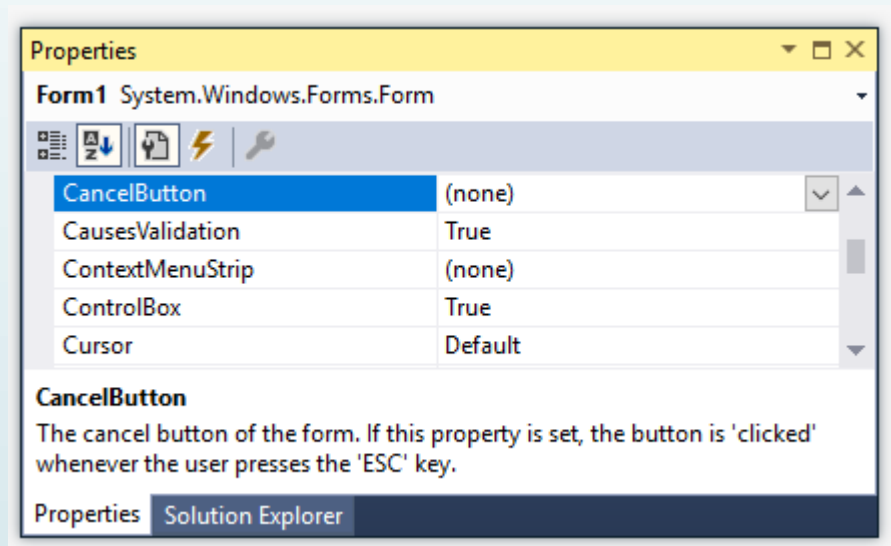
Name	<input type="text"/>
Email	<input type="text"/>
Telephone	<input type="text"/>
Clear	Save

Remember also

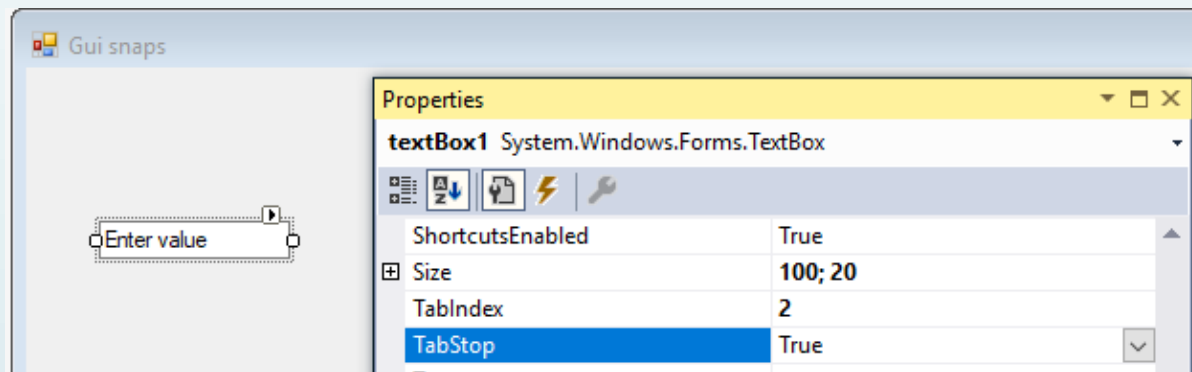
AcceptButton



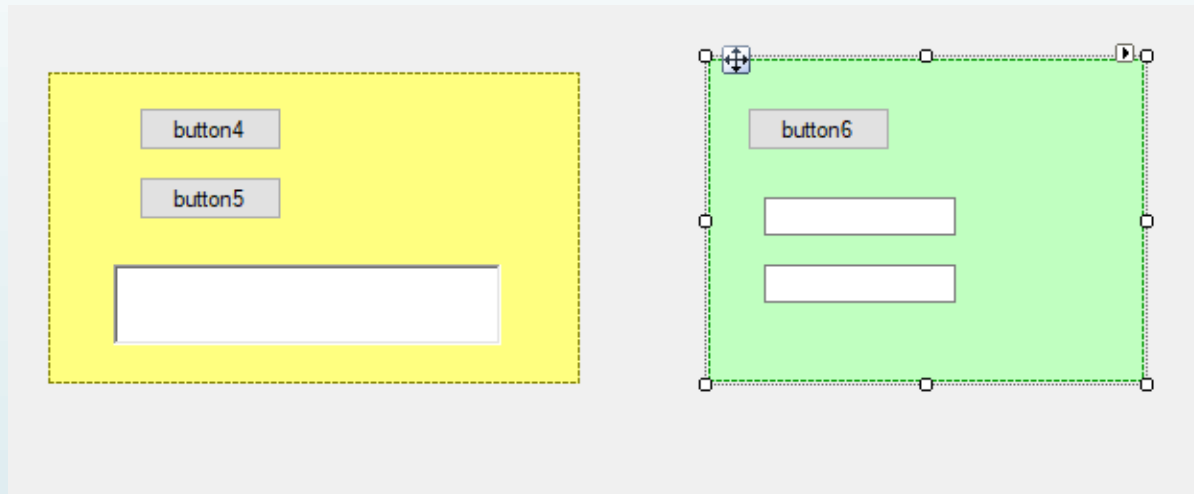
And CancelButton



Tabulator may be also used by some users: check tab stops:



With e.g. panels you can separate parts of to window:



Panels can be hidden and moved, too.

Some tools allow masked textfields (there can be predefined patterns in use):

Input Mask

Select a predefined mask description from the list below or select Custom to define a custom mask.


Mask Description	Data Format	Validating Type
Numeric (5-digits)	12345	Int32
Phone number	(574) 555-0123	(none)
Phone number no area code	555-0123	(none)
Short date	12/11/2003	DateTime
Short date and time (US)	12/11/2003 11:20	DateTime
Social security number	000-00-1234	(none)
Time (European/Military)	23:20	DateTime
Time (US)	11:20	DateTime
Zip Code	98052-6399	(none)
<Custom>		(none)

Mask:

Preview:

☒ Use ValidatingType

OK Cancel



Tel. nr

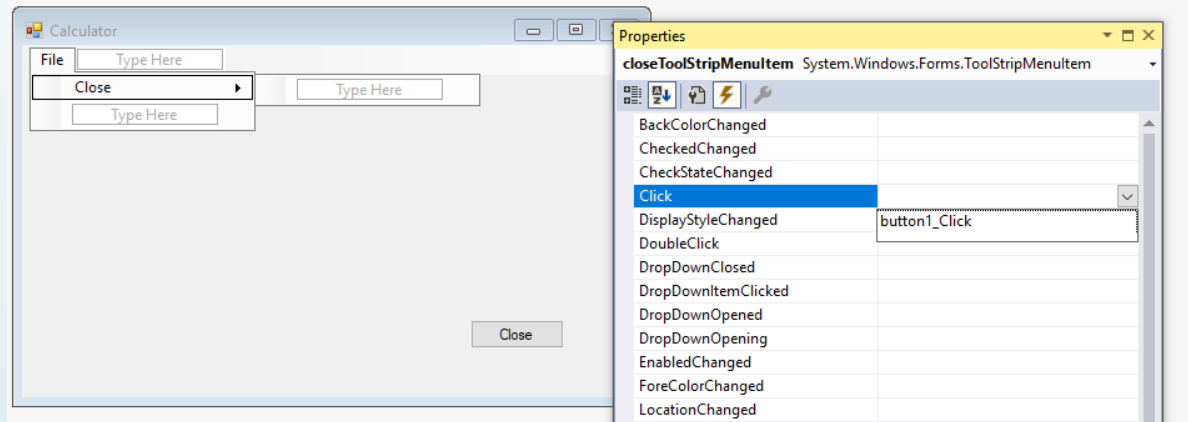
(333) 332-1234

Even this helps to get the right format but time saving is near zero.

When we have several choices we can use radiobuttons, checkboxes and lists: we save filling time and avoid typing errors.

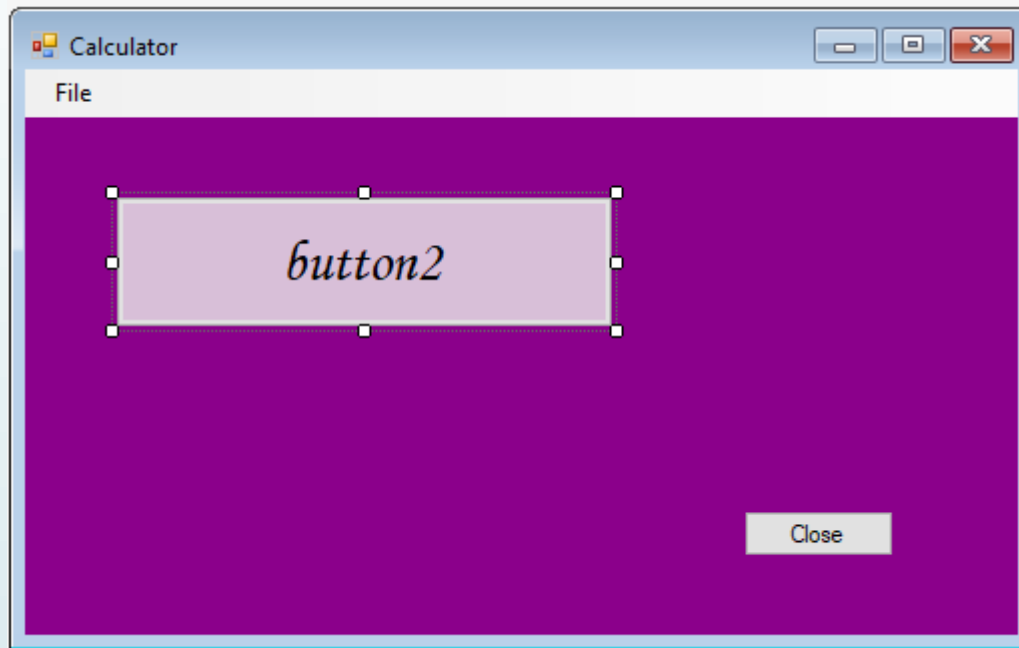
Lists can also seen as checklists: when we see the item, we remember to take it into account.

If several events (concerning more than one control) causes same operation, you can combine events: *this event causes this event (that is already handled)*:



```
private void button1_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Check also colours, fonts, font sizes, contrast and so on!



Avoid weird appearances ...

Tell user that some process is going on: using led, progress bar, ...

Ask confirmation when user is making important decisions.

About websites

Plan:

What do you offer?

Who is your customer, visitor? What does he/she want?

Analyze pages: do all pages have visitors

Are the most important things on right places? Gather feedback -> update

Show the menu all the time.

Tell user where he/she is (the path)

Animations and/or jokes on a website?

They may be really annoying!!



Same thing with advertisements and popup windows.

AND

Who reads the same joke several times?

About GOMS

GOMS =Goals, Operators, Methods, Selection keystroke-level model

: <http://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/printer/goms.html>

$$T_{\text{Execute}} = T_K + T_P + T_H + T_D + T_M + T_R$$

- **K (Keystroke)** =0.2
- **P (Pointing)** =1.1
- **H (Homing)**
- **D (Drawing)**
- **M (Mentally)**
- **R (Response)**

Example: calculate this using calculator

121 * 13/4.

Trial one

GOMS =Goals, Operators, Methods, Selection keystroke-level model
: <http://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/printer/goms.html>

$$T_{\text{Execute}} = T_K + T_P + T_H + T_D + T_M + T_R$$

- **K (Keystroke)** =0.2
- **P (Pointing)** =1.1
- **H (Homing)**
- **D (Drawing)**
- **M (Mentally)**
- **R (Response)**

We got 11,2 seconds.

With mouse only:

H[keyboard]	0.4
M3K[word]	1.7 (3x0.5 +0.2)
H[mouse]	0.4
<i>P[field]</i>	1.1
<i>K[mouse]</i>	0.2
H[keyboard]	0.4
M2K[word]	1.2
H[mouse]	0.4
<i>P[field]</i>	1.1
<i>K[mouse]</i>	0.2
H[keyboard]	0.4
M2K[word]	0.7
H[mouse]	0.4
<i>P[field]</i>	1.1
<i>K[mouse]</i>	0.2
	9.9

Another way to express calculation

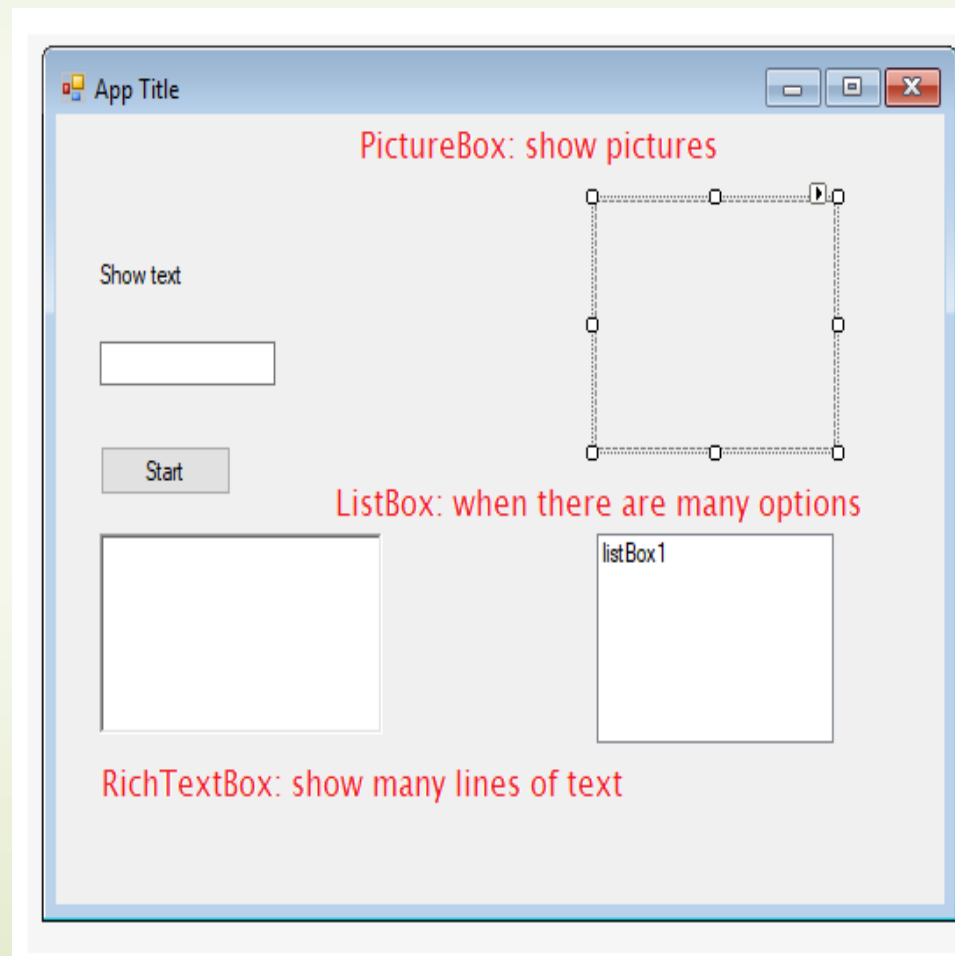
GOAL: CALCULATE-SENTENCE	
GOAL: ENTER-NUMBER ...repeat for numbers 121	
[select**:	GOAL: CLICK-NUMBER
	MOVE-CURSOR-TO-NUMBER 1.10
	CLICK-MOUSE-BUTTON 0.20
GOAL:	ENTER-NUMBER
	PRESS-NUMBER-KEY
VERIFY-NUMBER	1.35
GOAL: PRESS-MULTIPLIER-KEY	
	MOVE-CURSOR-TO-KEY 1.10
	CLICK-MOUSE-BUTTON 0.20
	VERIFY-KEY-PRESS 1.35
GOAL: ENTER-NUMBER ...repeat for numbers 13	
[select**:	GOAL: CLICK-NUMBER
	MOVE-CURSOR-TO-NUMBER 1.10
	CLICK-MOUSE-BUTTON 0.20
GOAL:	ENTER-NUMBER
	PRESS-NUMBER-KEY
VERIFY-NUMBER	1.35
...	
GOAL: ENTER-NUMBER ... for number 4	
[select**:	GOAL: CLICK-NUMBER
	MOVE-CURSOR-TO-NUMBER 1.10
	CLICK-MOUSE-BUTTON 0.20
GOAL:	ENTER-NUMBER
	PRESS-NUMBER-KEY
VERIFY-NUMBER	1.35
GOAL: PRESS-EQUALS-KEY	
	MOVE-CURSOR-TO-KEY 1.10
	CLICK-MOUSE-BUTTON 0.20
	VERIFY-KEY-PRESS 1.35
TOTAL TIME PREDICTED	19.8

Other Usability things to consider

There are many other things, too, that make gui efficient:

- enabled and disabled controls and menus
- distances between controls
- tool tips
- sub menus
- color code
- locations of controls (eye tracking test can be used)
- default options (most popular) re-selected
- font selections
- tabbed windows
- obligatory controls
- and so on

Short repetition...



App Title

RadioButtons

☐ Choose only one of these

☐ radioButton1

☐ radioButton2

☐ radioButton3

☐ radioButton4

CheckBoxes

Choose one or more

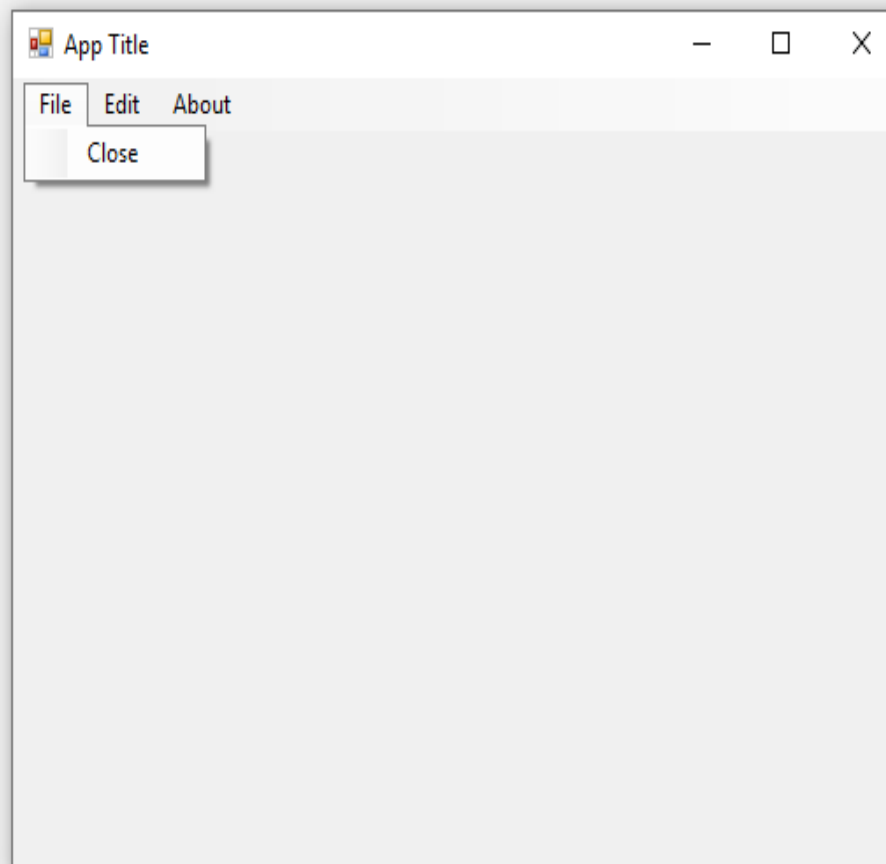
☐ checkBox1

☐ checkBox2

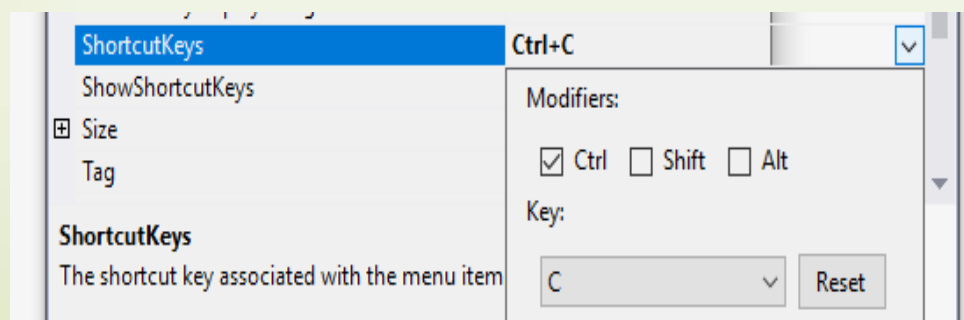
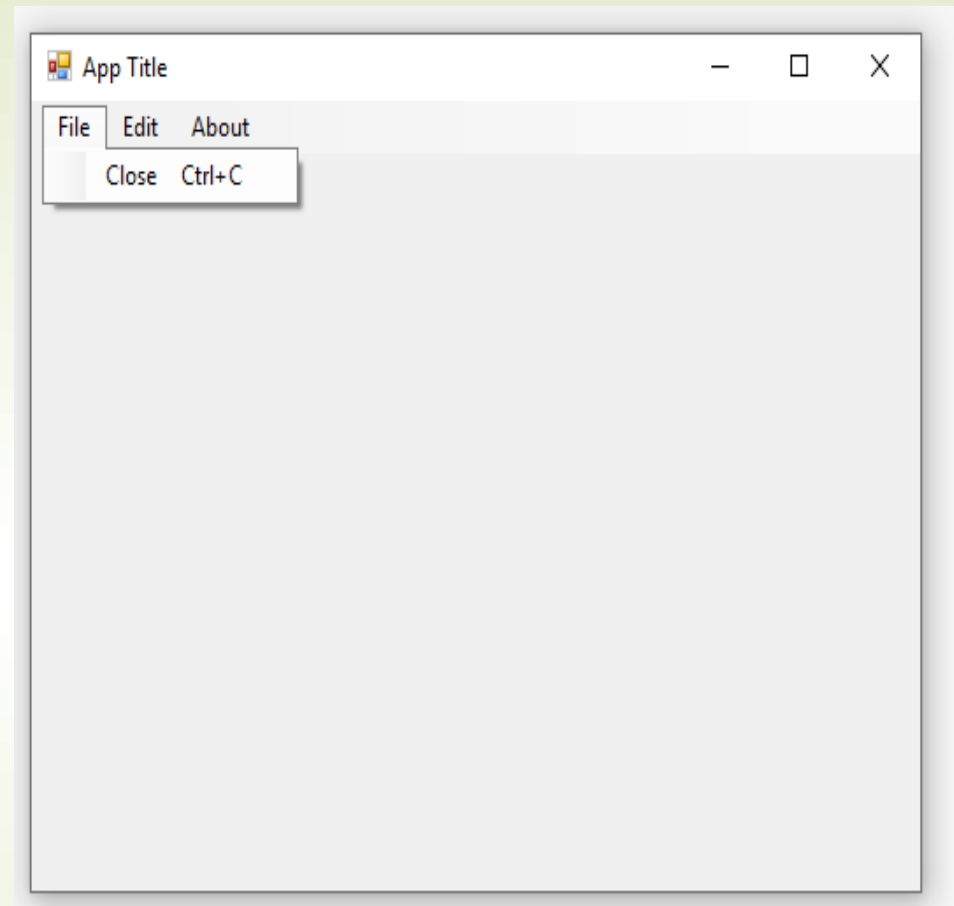
☐ checkBox3

☐ checkBox4

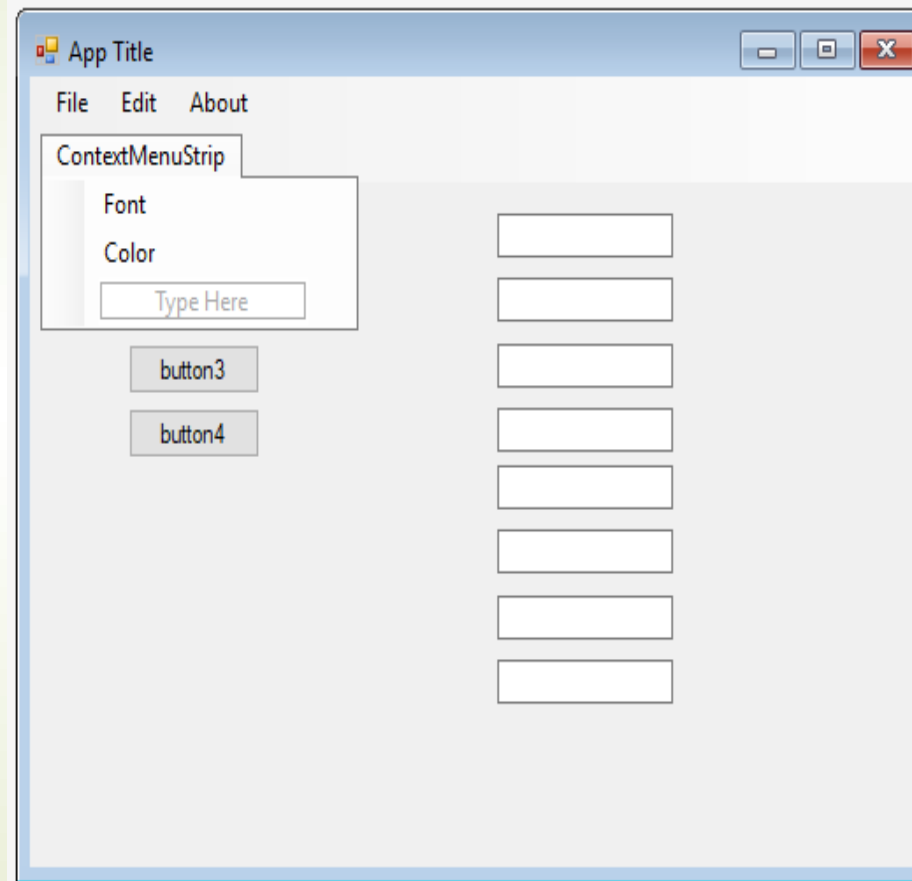
Use menus, follow
Microsoft style



Add
shortcut
keys, too



Add
Context
menu to
help
user



Not good!

A window titled "App Title" with a menu bar containing "File", "Edit", and "About". The "File" menu is highlighted. The form contains five input fields and four buttons. The labels and input fields are on the left, and the buttons are on the right.

Label	Input Field	Button
Name	<input type="text"/>	button1
Address	<input type="text"/>	button2
Telephone nr	<input type="text"/>	button3
Email	<input type="text"/>	button4
Native language	<input type="text"/>	

Better!

A window titled "App Title" with a menu bar containing "File", "Edit", and "About". The form contains five input fields and four buttons. The labels and input fields are on the left, and the buttons are on the right. The layout is more organized than the first window.

Label	Input Field	Button
Name	<input type="text"/>	button1
Address	<input type="text"/>	button2
Telephone nr	<input type="text"/>	button3
Email	<input type="text"/>	button4
Native language	<input type="text"/>	

Not good!

A window titled "App Title" with a menu bar containing "File", "Edit", and "About". The "File" menu is highlighted. The form contains five input fields and four buttons. The labels and their corresponding input fields are: "Name" (short), "Address" (medium), "Telephone nr" (short), "Email" (medium), and "Native language" (short). The buttons are labeled "button1", "button2", "button3", and "button4" and are arranged vertically on the right side of the form.

Better!

A window titled "App Title" with a menu bar containing "File", "Edit", and "About". The form contains five input fields and four buttons. The labels and their corresponding input fields are: "Name" (short), "Address" (medium), "Telephone nr" (short), "Email" (medium), and "Native language" (short). The buttons are labeled "button1", "button2", "button3", and "button4" and are arranged vertically on the right side of the form. A red arrow points from the "Not good!" label to the "App Title" window title bar.

Not good: many
fonts, many sizes

App Title

File Edit About

Name

Address

Telephone nr

Email

Native language

button1
button2
button3
button4

Better: 1 font, 1
size!

App Title

File Edit About

Name

Address

Telephone nr

Email

Native language

button1
button2
button3
button4

Not good: colourful, many colors

App Title

File Edit About

Name

Address

Telephone nr

Email

Native language

button1

button2

button3

button4

Better: 1-2 colours,
light background
color

App Title

File Edit About

Name

Address

Telephone nr

Email

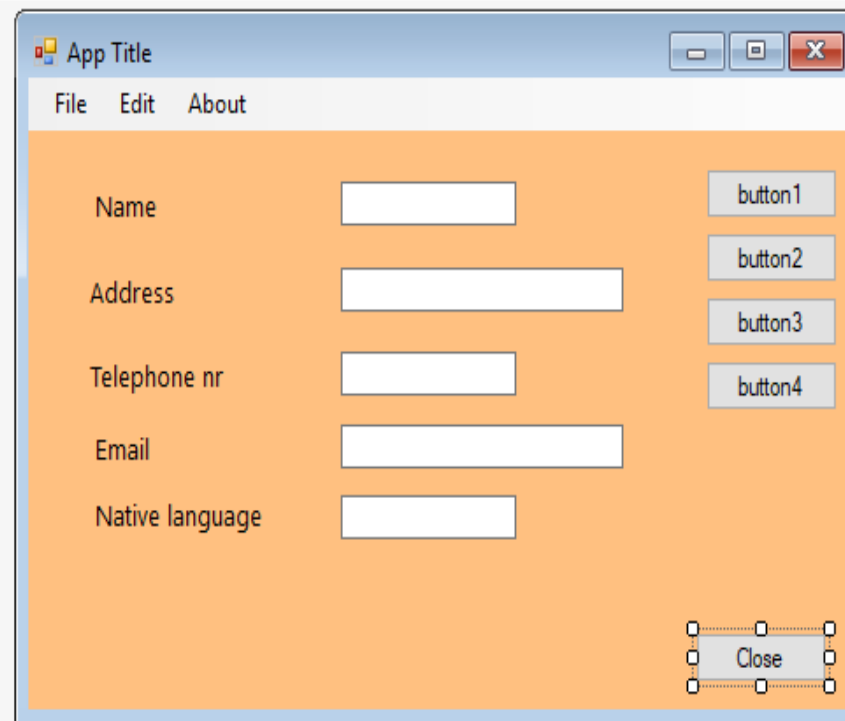
Native language

button1

button2

button3

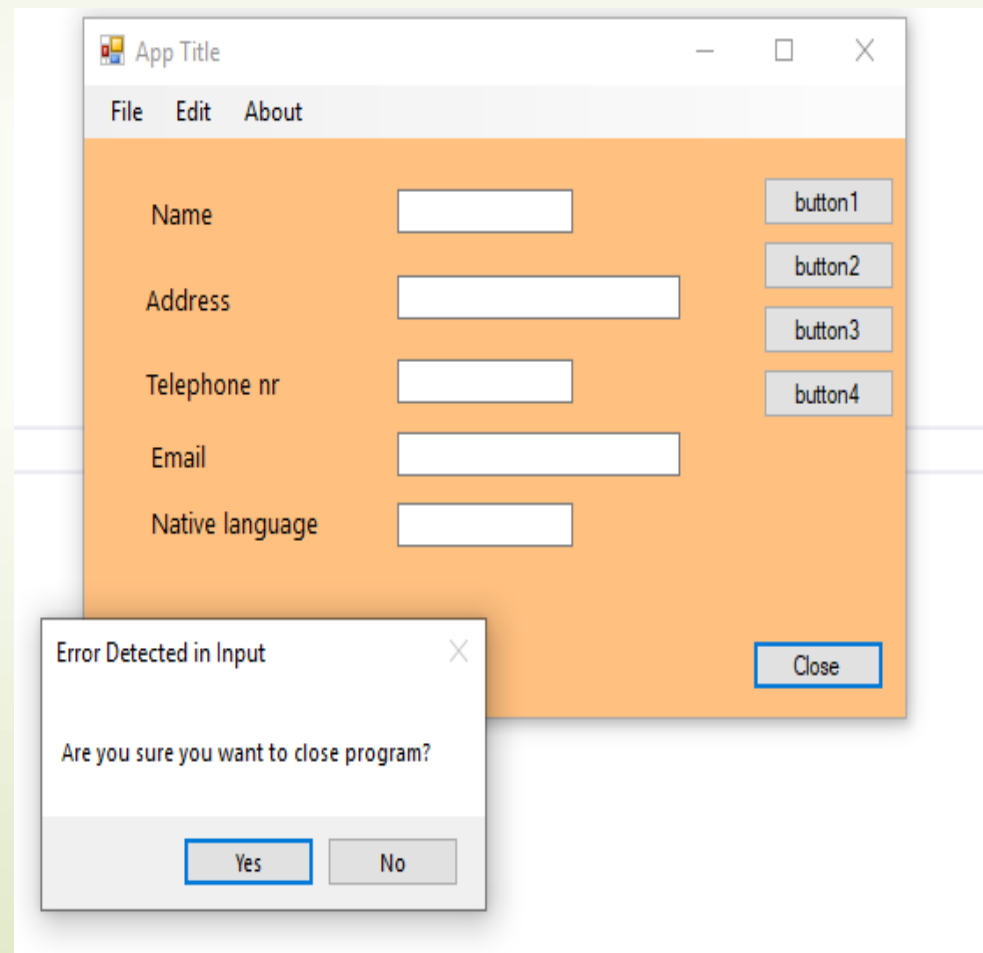
button4



Close button: classic place, outside,
not near other controls

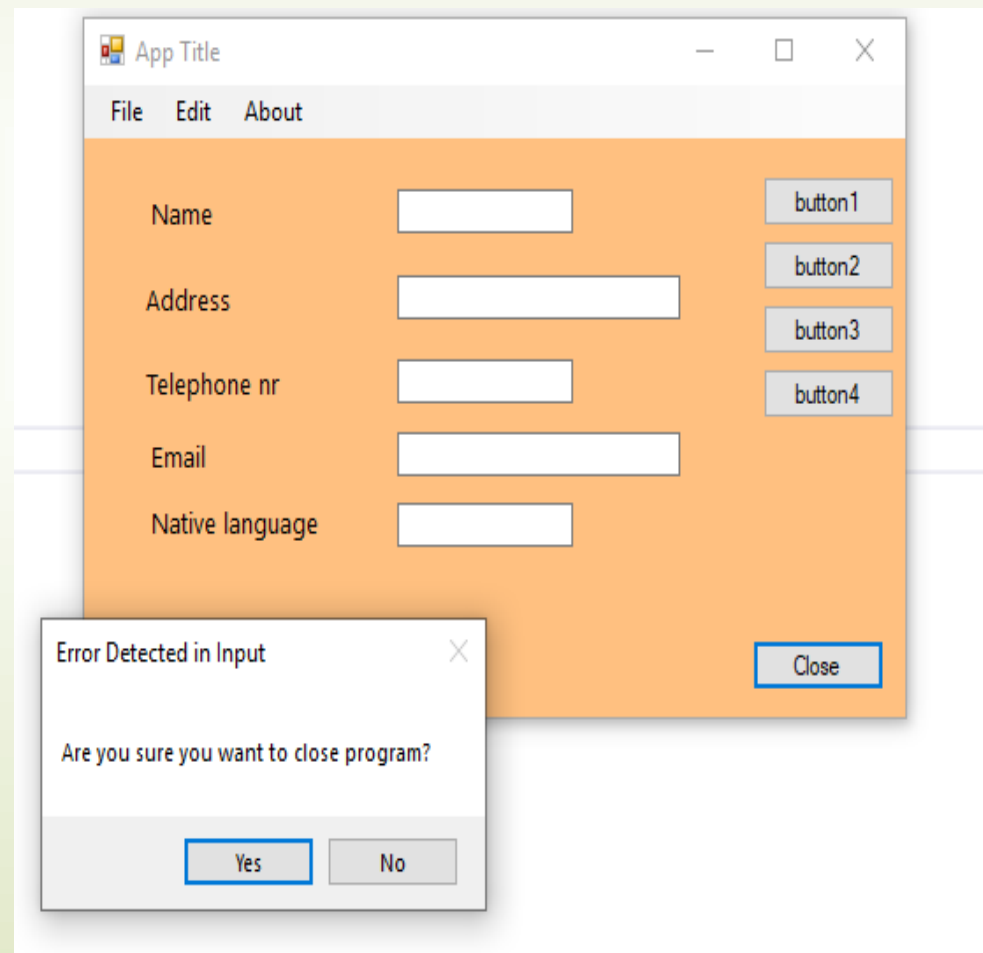
Gui – main principles

Important or dangerous activities: asking for confirmation is good to have!



Gui – main principles

Important or dangerous activities: asking for confirmation is good to have!



Not good: order of controls is not logical!

App Title

File Edit About

First name

Telephone nr

Email

Surname

Your feedback:

Send Close

Better: after first name comes surname

App Title

File Edit About

First name

Surname

Email

Telephone nr

Your feedback:

Send Close

When app is started,
focus is in the first
control

Change on
background color
tells user that current
control has focus

App Title

File Edit About

First name

Surname

Email

Telephone nr

Your feedback:

Send Close

Show clearly what
info is obligatory

App Title

File Edit About

First name
[required]

Surname
[required]

Email
[required]

Telephone nr

Your feedback:
[required]

Send Close

The most common option is first and it is by default checked

App Title

What is your native language?

☒ Finnish

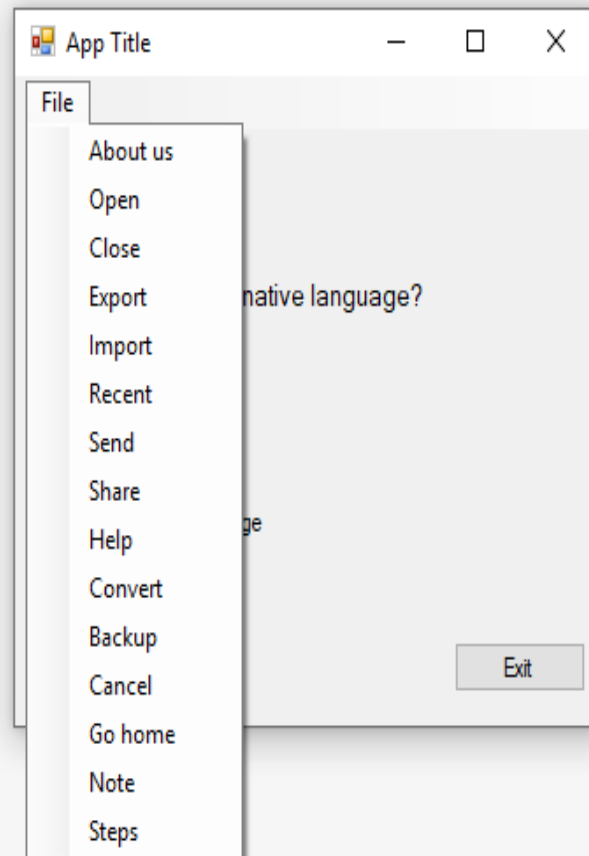
☐ Swedish

☐ English

☐ Other language

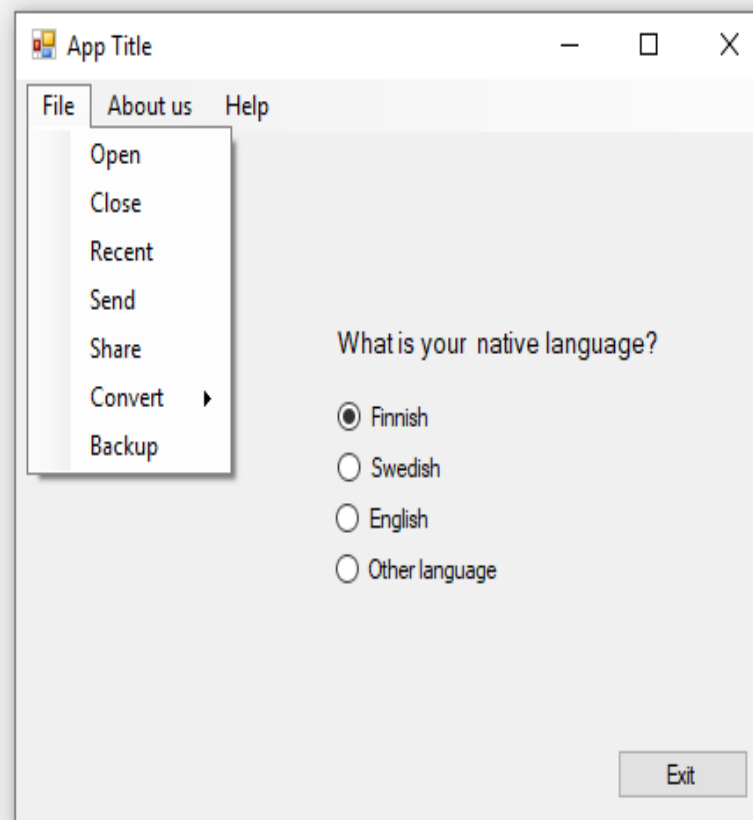
Design menus

Too long
menu

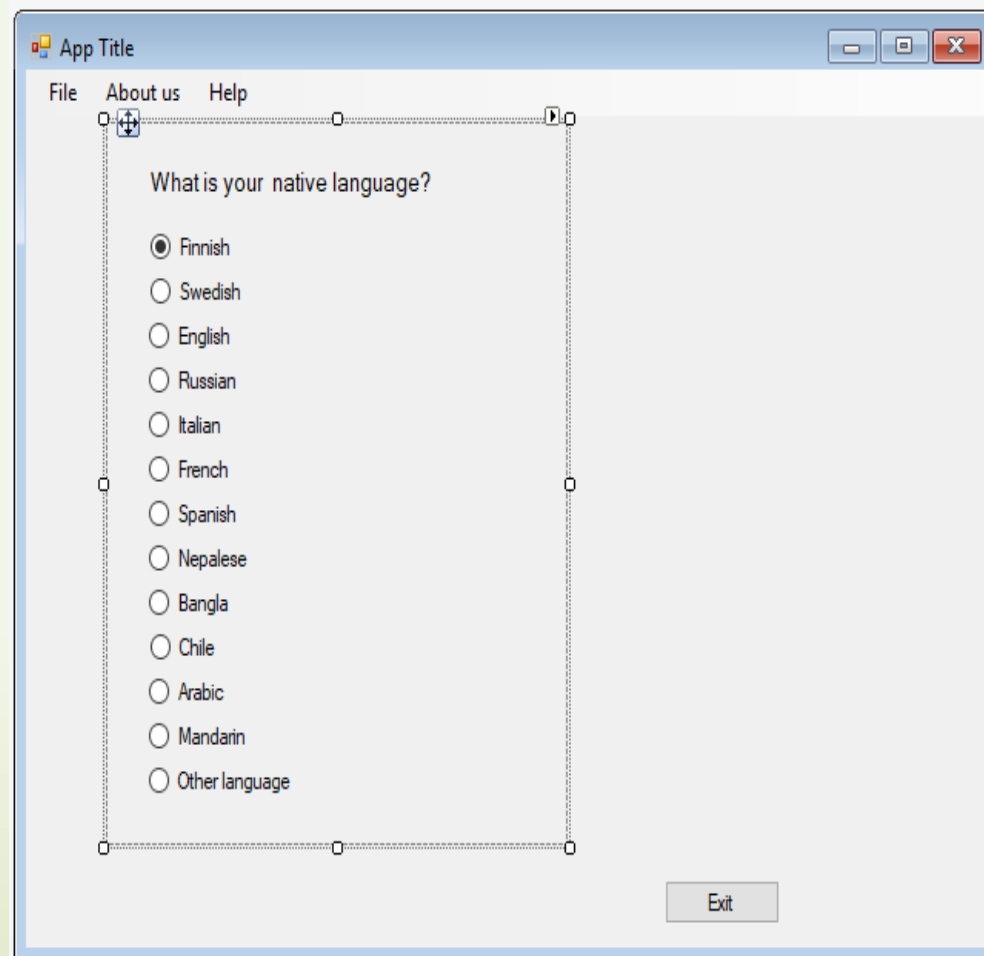


Design menus

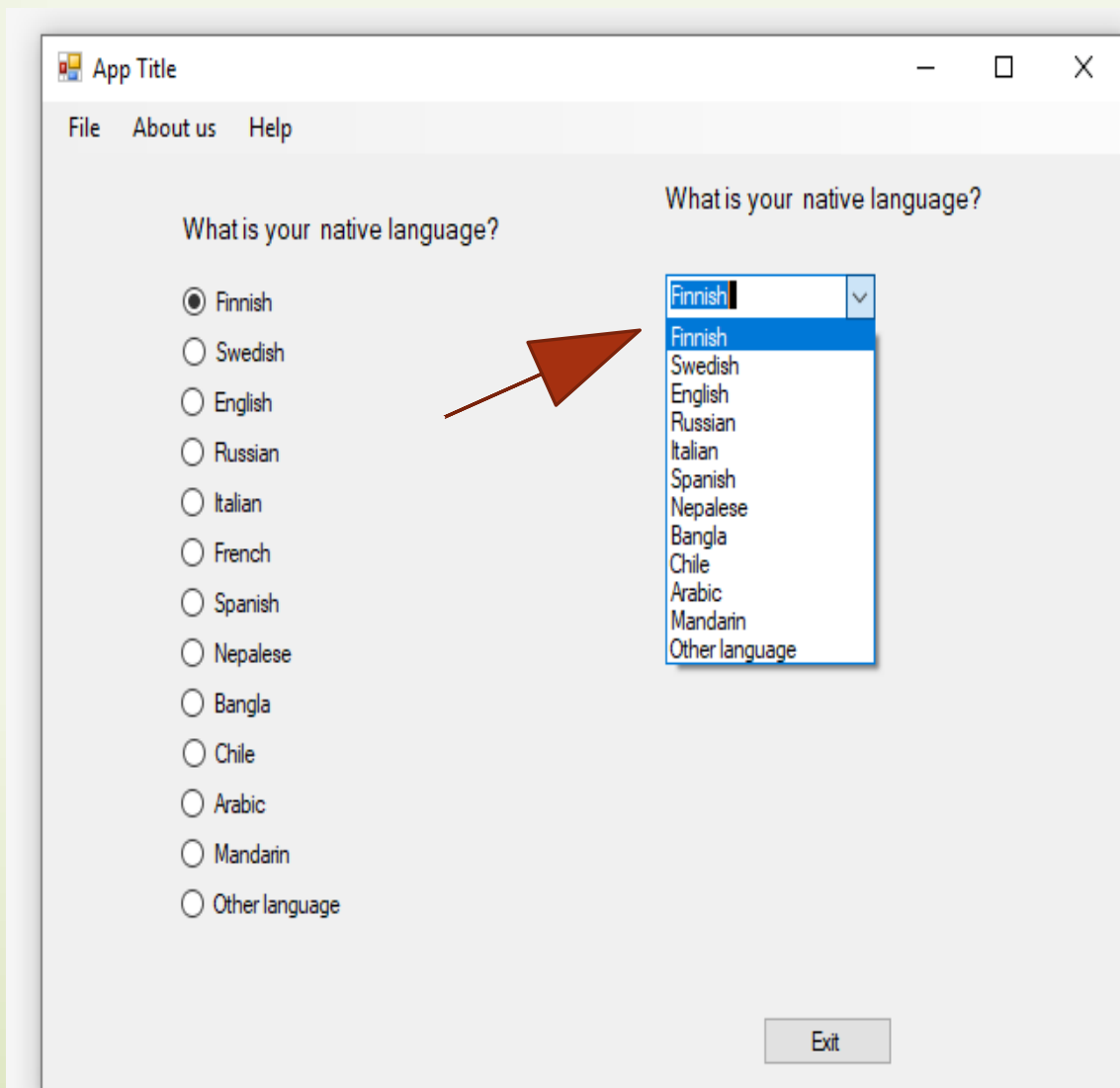
Better menu



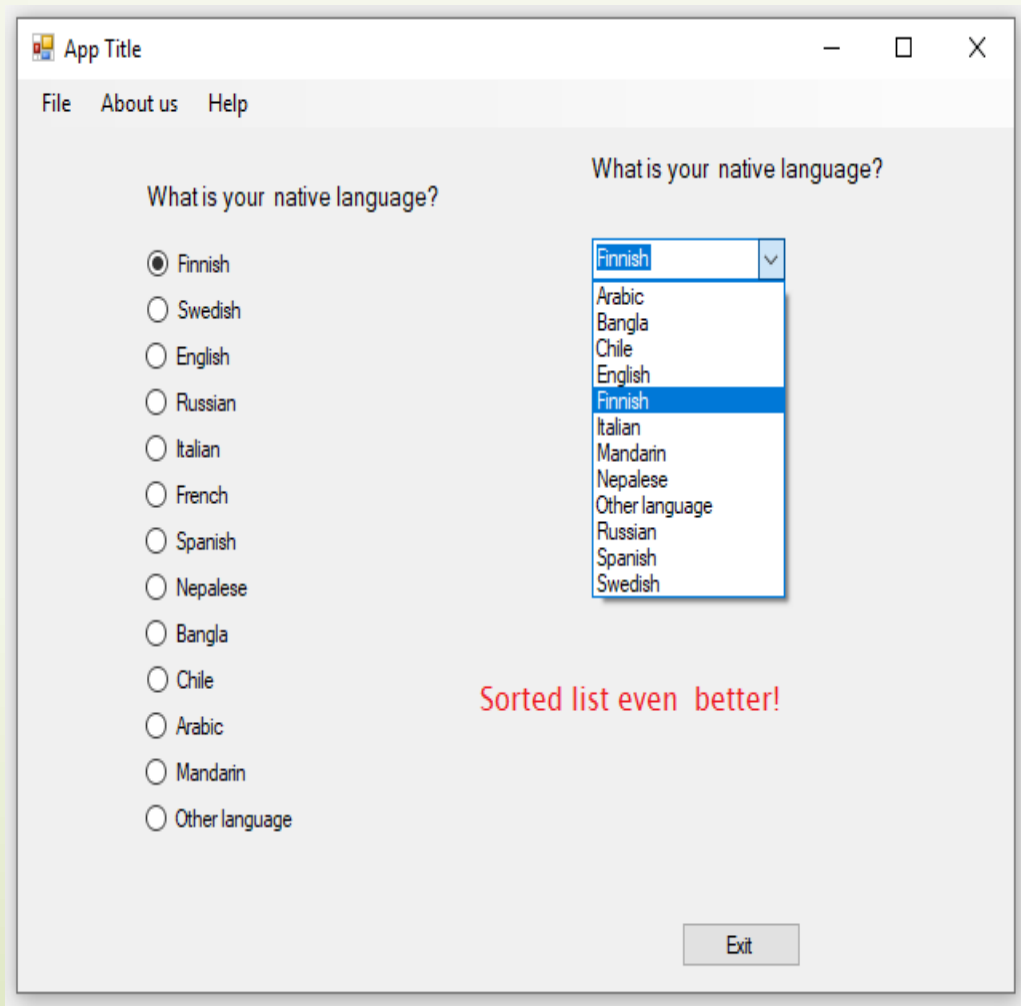
Radiobuttons: too many options



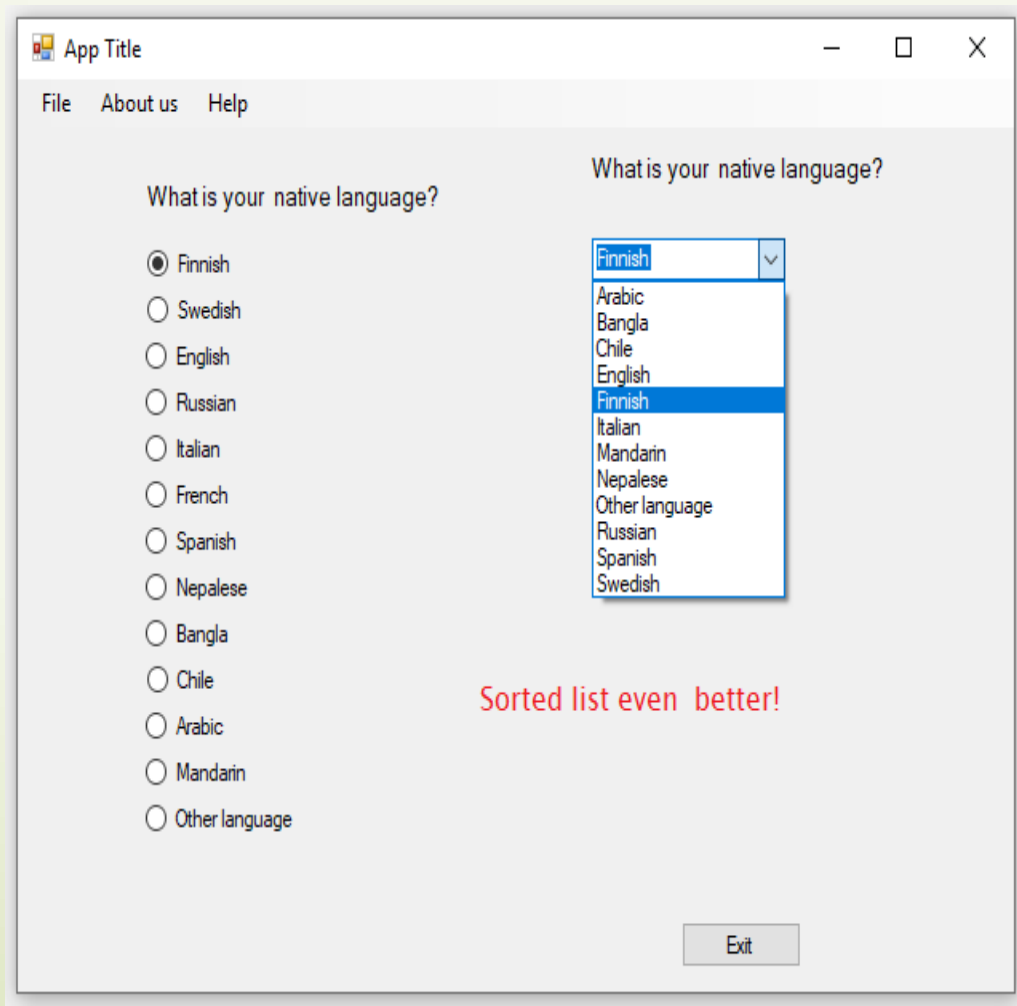
Radiobuttons: better with a list control



Radiobuttons: better with a list control



Radiobuttons: better with a list control



Too many controls, no clear groups, cognitive load is higher

The image shows a Windows application window titled "Form1". The window contains a collection of user interface controls that are not visually grouped, leading to a high cognitive load. The controls are arranged as follows:

- Top Row:** Four text input fields followed by a dropdown menu.
- Second Row:** Three text input fields followed by a large, empty rectangular box.
- Third Row:** Three text input fields.
- Fourth Row:** One text input field, followed by "button4", and then "button5".
- Fifth Row:** "button1" followed by three empty spaces.
- Sixth Row:** Three text input fields followed by "button2".
- Seventh Row:** Three text input fields.
- Bottom Right:** "button3" is located in the bottom right corner of the form area.

The window has a standard Windows title bar with minimize, maximize, and close buttons. The overall layout is disorganized, with no visual cues to group related controls together.

Easier to use:

The image shows a Windows Forms application window titled "Form1" with standard minimize, maximize, and close buttons. The form contains four group boxes, each with a move handle (a small square with a cross) in its top-left corner:

- groupBox2** (top-left): Contains four empty text boxes stacked vertically and a button labeled "button1" at the bottom.
- groupBox3** (top-middle): Contains four empty text boxes stacked vertically and a button labeled "button4" at the bottom.
- groupBox4** (top-right): Contains three empty text boxes stacked vertically, a large empty rectangular area to the right of the text boxes, a button labeled "button5" at the bottom left, and a dropdown menu at the bottom right.
- groupBox1** (bottom): Contains two rows of three empty text boxes each, a button labeled "button2" to the right of the text boxes, and a button labeled "button3" located outside the group box at the bottom right of the form.

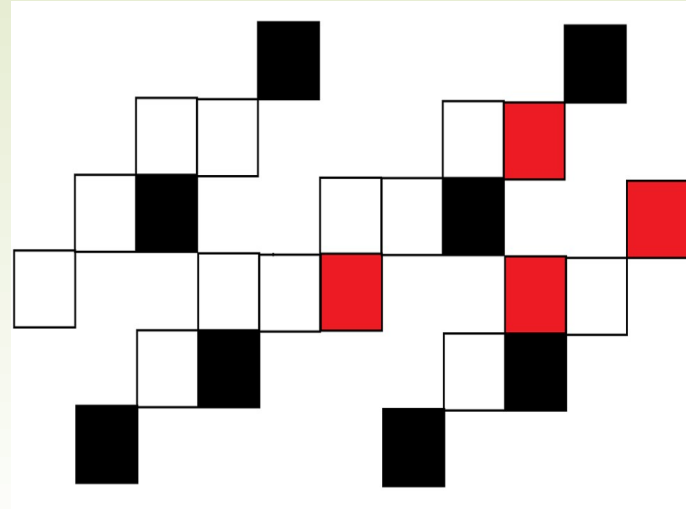
Maybe even better and easier
to use:

The screenshot shows a Windows Forms application window titled "Form1". The window contains the following UI elements:

- groupBox2**: A light gray group box containing four text boxes stacked vertically and a button labeled "button1" at the bottom.
- groupBox3**: A blue group box containing three text boxes stacked vertically and a button labeled "button4" at the bottom.
- groupBox4**: A light gray group box containing three text boxes stacked vertically, a large empty rectangular area, and a button labeled "button5" at the bottom.
- groupBox1**: A large blue group box at the bottom containing a grid of six text boxes (two rows of three) and a button labeled "button2" on the right side.
- button3**: A button labeled "button3" located at the bottom right of the window.

Gui – cognitive laws

Laws of similarity,
pragnanz, proximity,
continuity, and
closure.

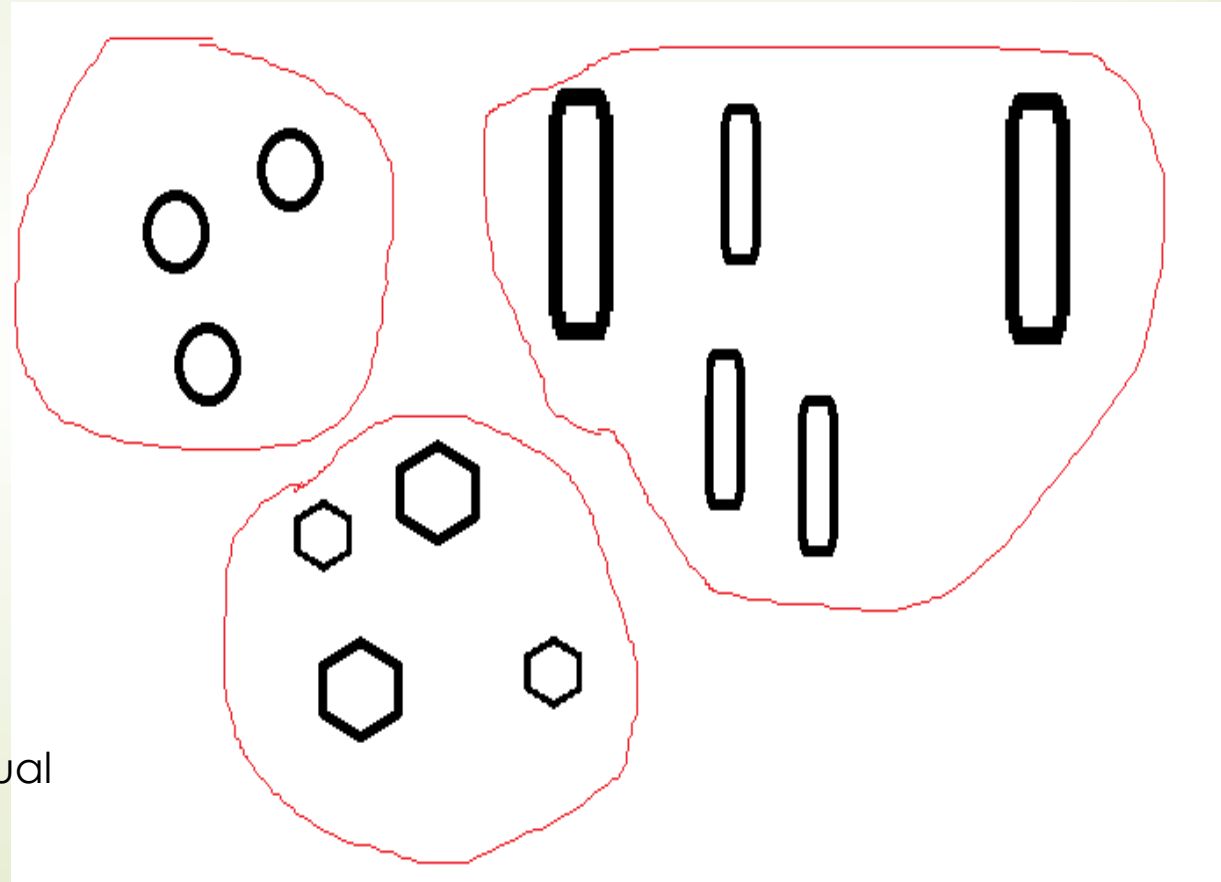


Similarity

Gui – cognitive laws

Similar items are grouped together: they seem to be seen as one group

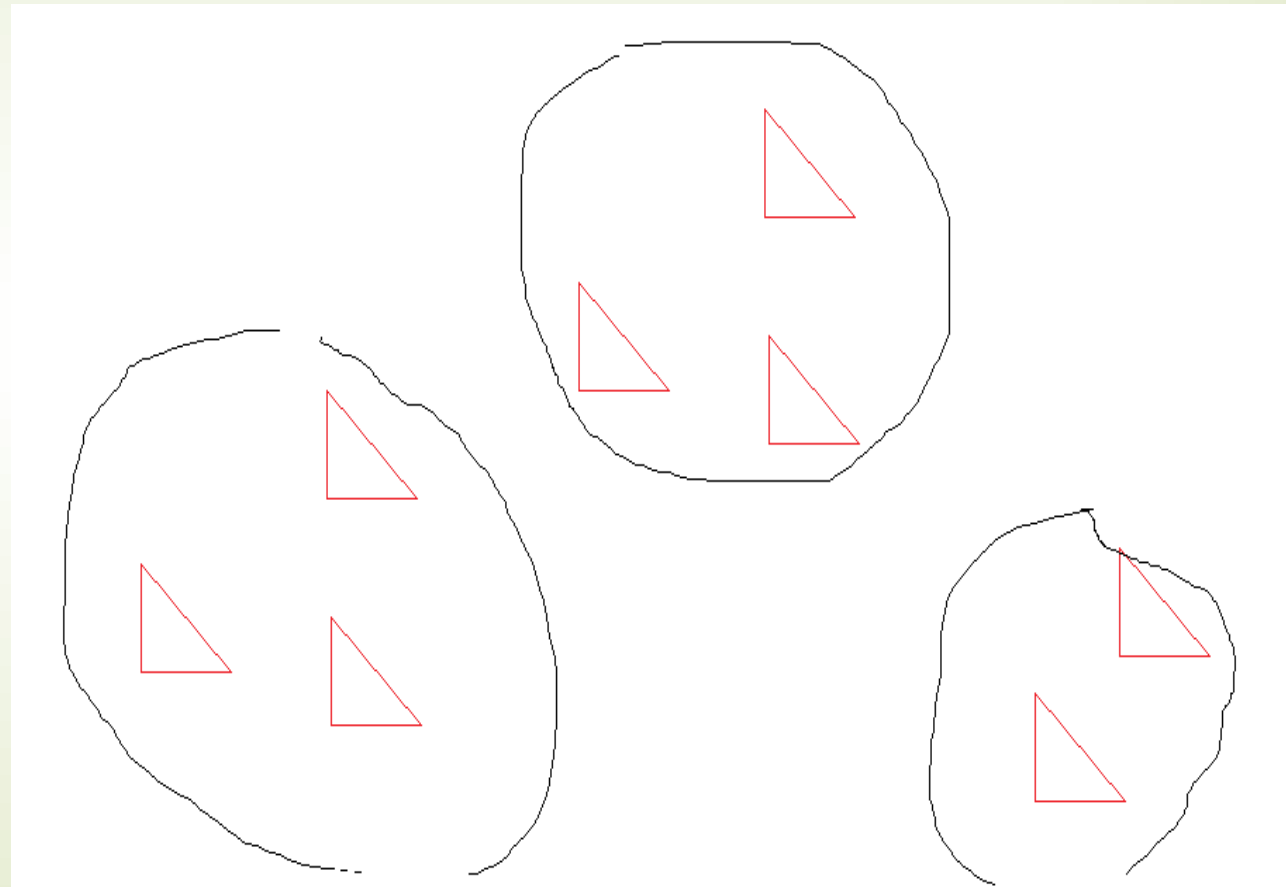
Items are not seen as individual shapes...



Gui – cognitive laws

Objects near
each other
tend to be
viewed as a
group

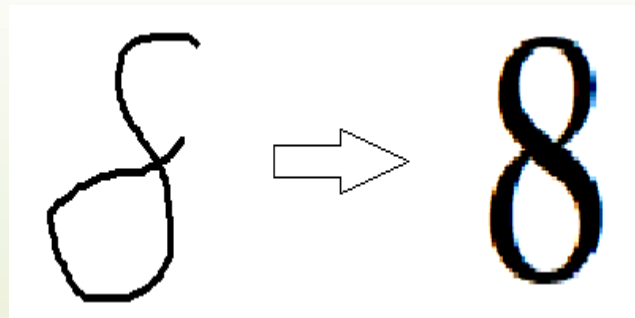
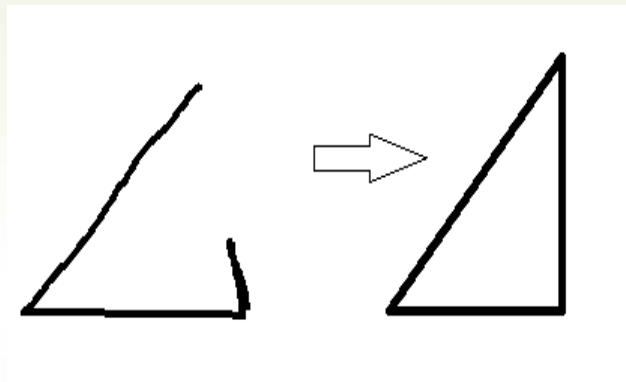
Proximity



Gui – cognitive laws

Objects or
groups of
different shapes
are viewed as
some real, good
figure ..
(are
completed...)

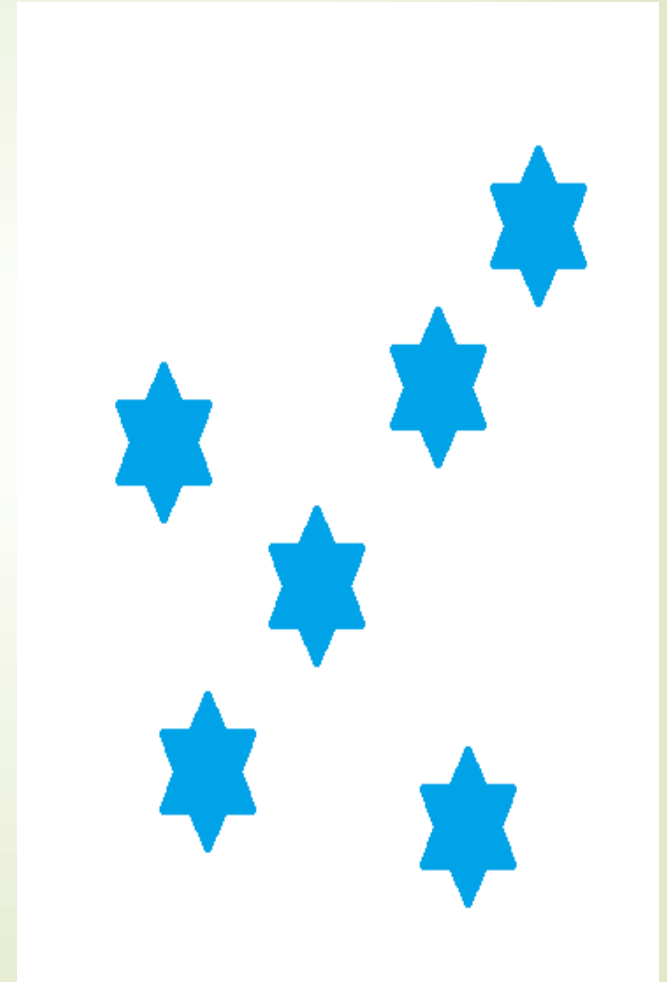
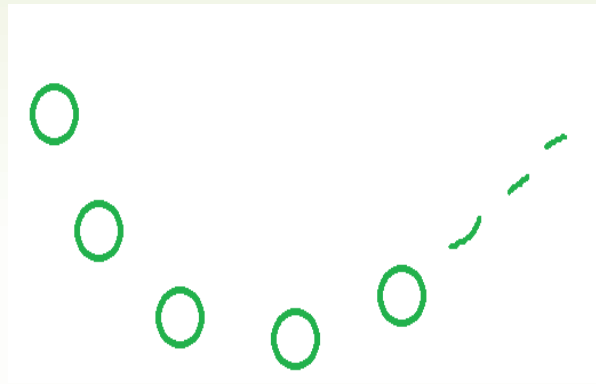
Pragnanz



Gui – cognitive laws

Objects are
viewed to format
a continuing
shape (line,
curve..)

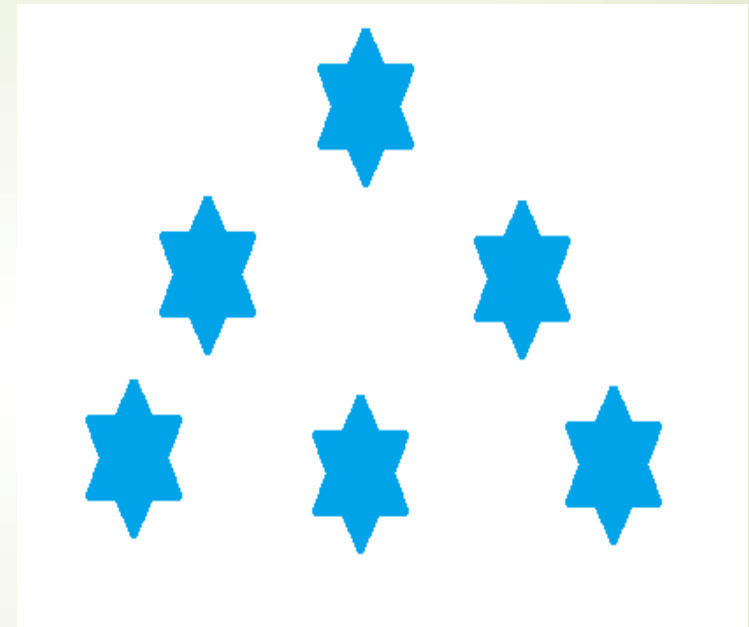
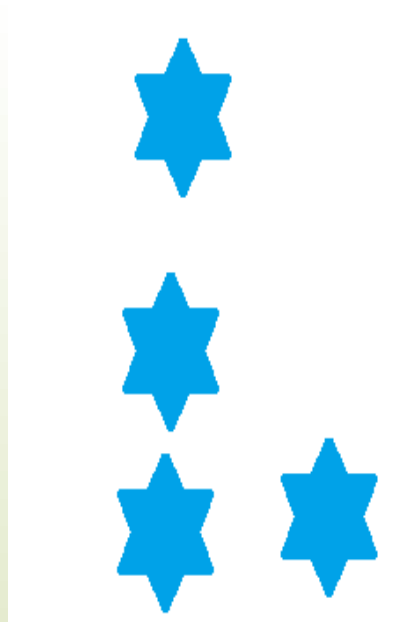
Continuity



Closure

Gui – cognitive laws

Objects are
viewed
as one shape if
there are “seen”
boundaries



Gui

UI:s are everywhere:
examples!



Gui

UI:s are everywhere:
examples!



Gui

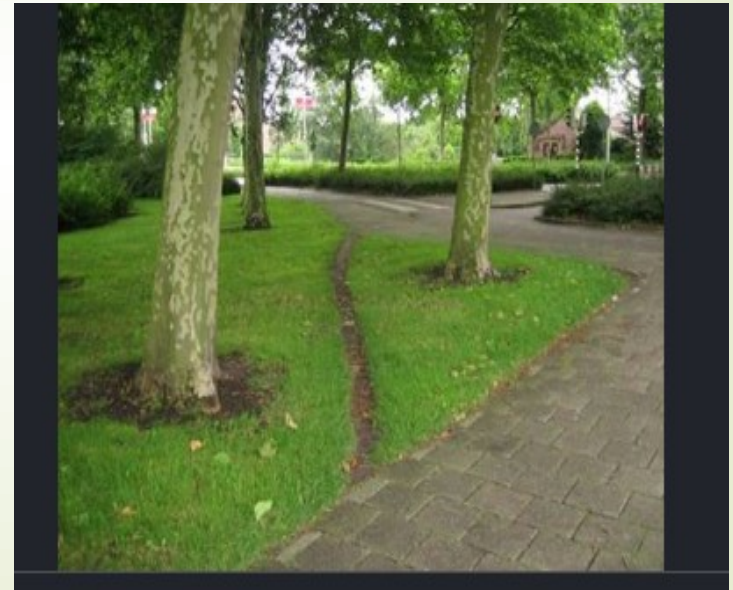
UI:s are everywhere:
examples!



Gui

UI:s are everywhere:
examples!
Even new ways are found!

Time is
money!



<http%3A%2F%2Fwww.steve-wheeler.co.uk%2F2017%2F05%2Fdesire-lines.html&psig=AOvVaw1z-JzHthLpb8ZBaZ53Vxyh&ust=1580495596642282>

Study usability

Famous guru: Jakob Nielsen

nngroup.com/articles/usability-101-introduction-to-usability/

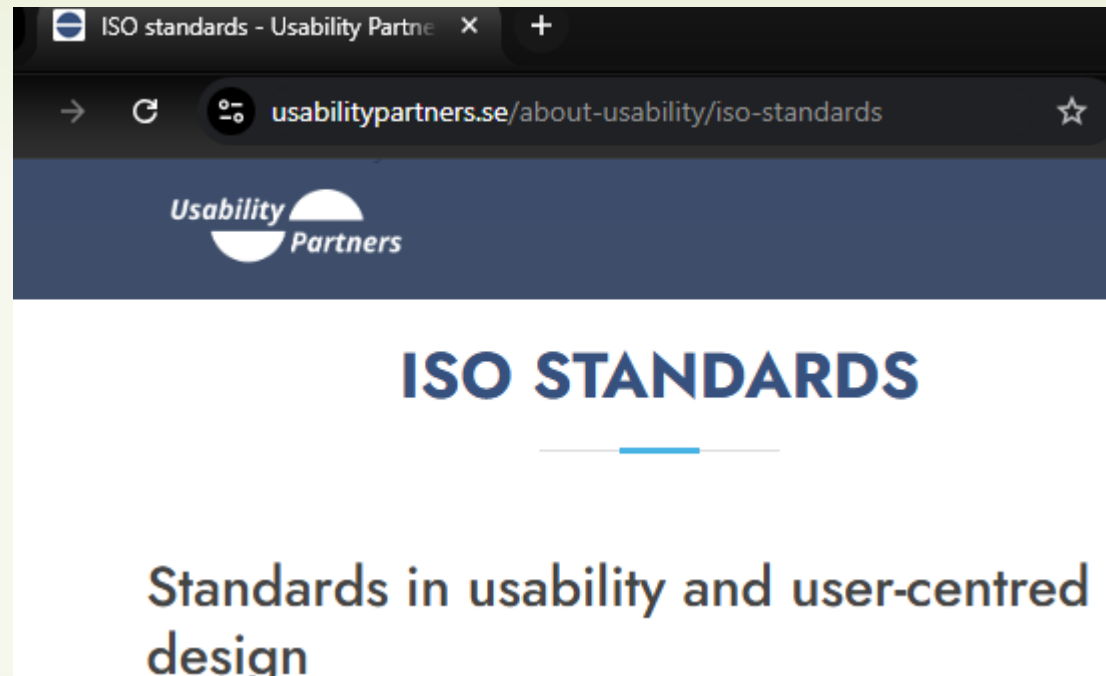
Usability 101: Introduction to Usability



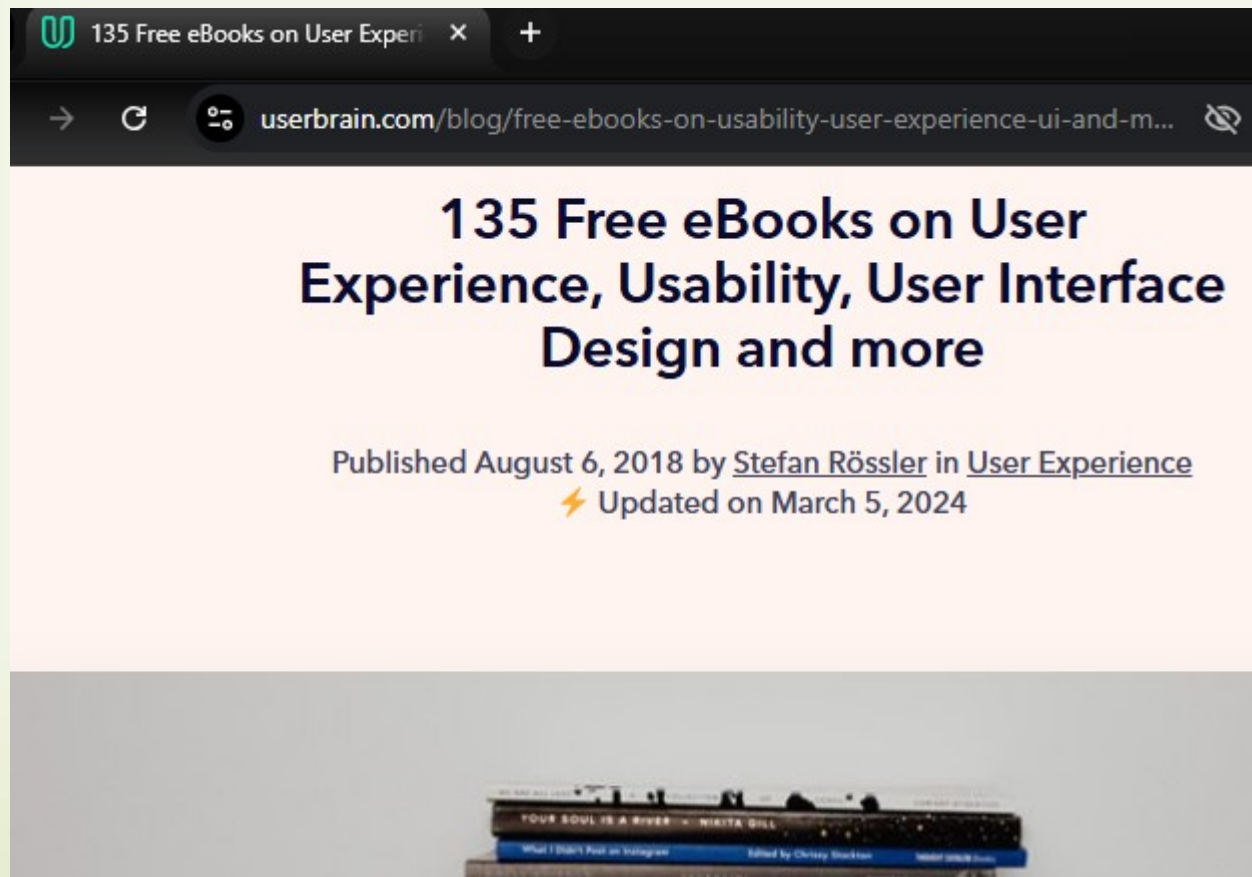
Jakob Nielsen

January 3, 2012

There are also ISO standards



There are also many good books about Usability and UX, even eBooks!!



Thank you!

The eBook that is presented here, is free!

You can see the link in the description part...

Feedback is welcome!

Because of shadowed letters, text above is more difficult to read :)

But when you put energy for reading, you remember better :)