Cours 6:
Design, rappel - programmation GUI

Anastasia.Bezerianos@lri.fr

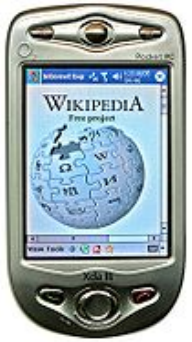
(partie de la présentation basée sur des transparents de Michel Beaudouin-Lafon et de Saul Greenberg)

interfaces graphiques

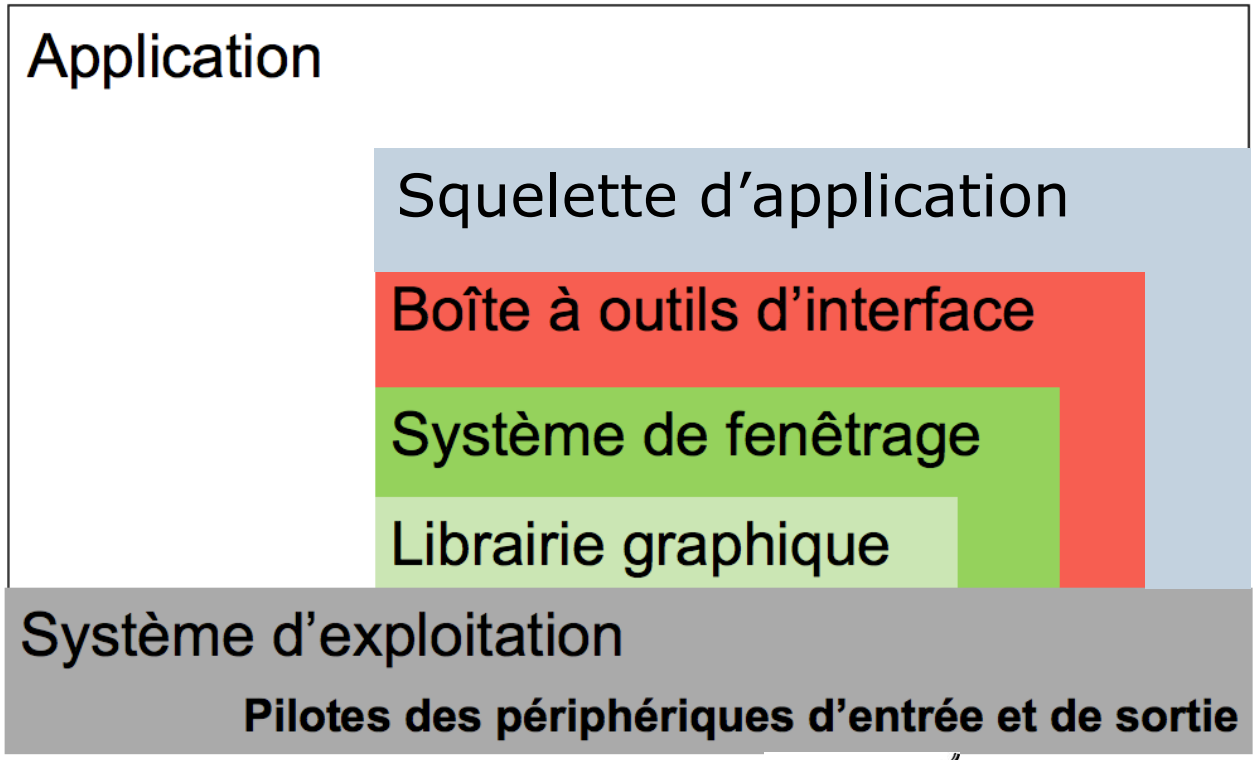
L'interaction graphique : les entrées sont spécifiées directement à partir des sorties (Graphical User Interfaces – **GUI**s)

Une périphérique d'entrée spécifie dans une commande une position à l'écran qui désigne un objet déjà affiché par le système (*pointage*).

Elle est familière dans le monde physique, donc le succès de ces interfaces



couches logicielles



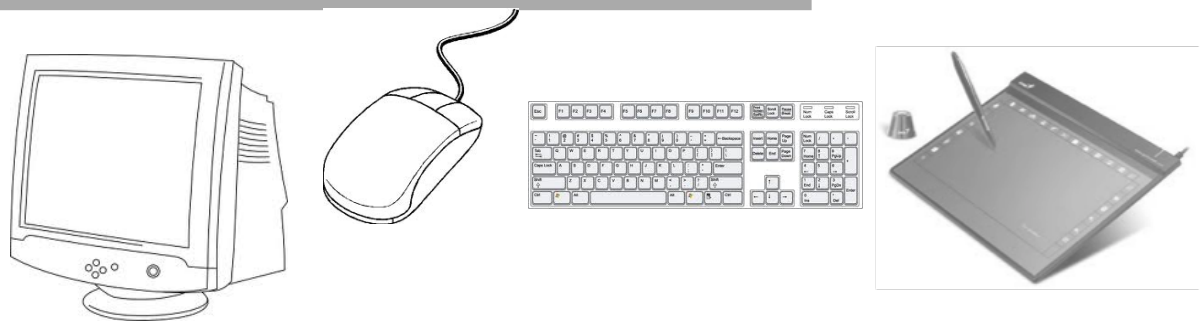
MacApp

Java Swing, Qt, GTK+, MFC, Cocoa

X Windows

GDI+, Quartz, GTK+/Xlib, OpenGL

Windows, Mac OS X, Linux



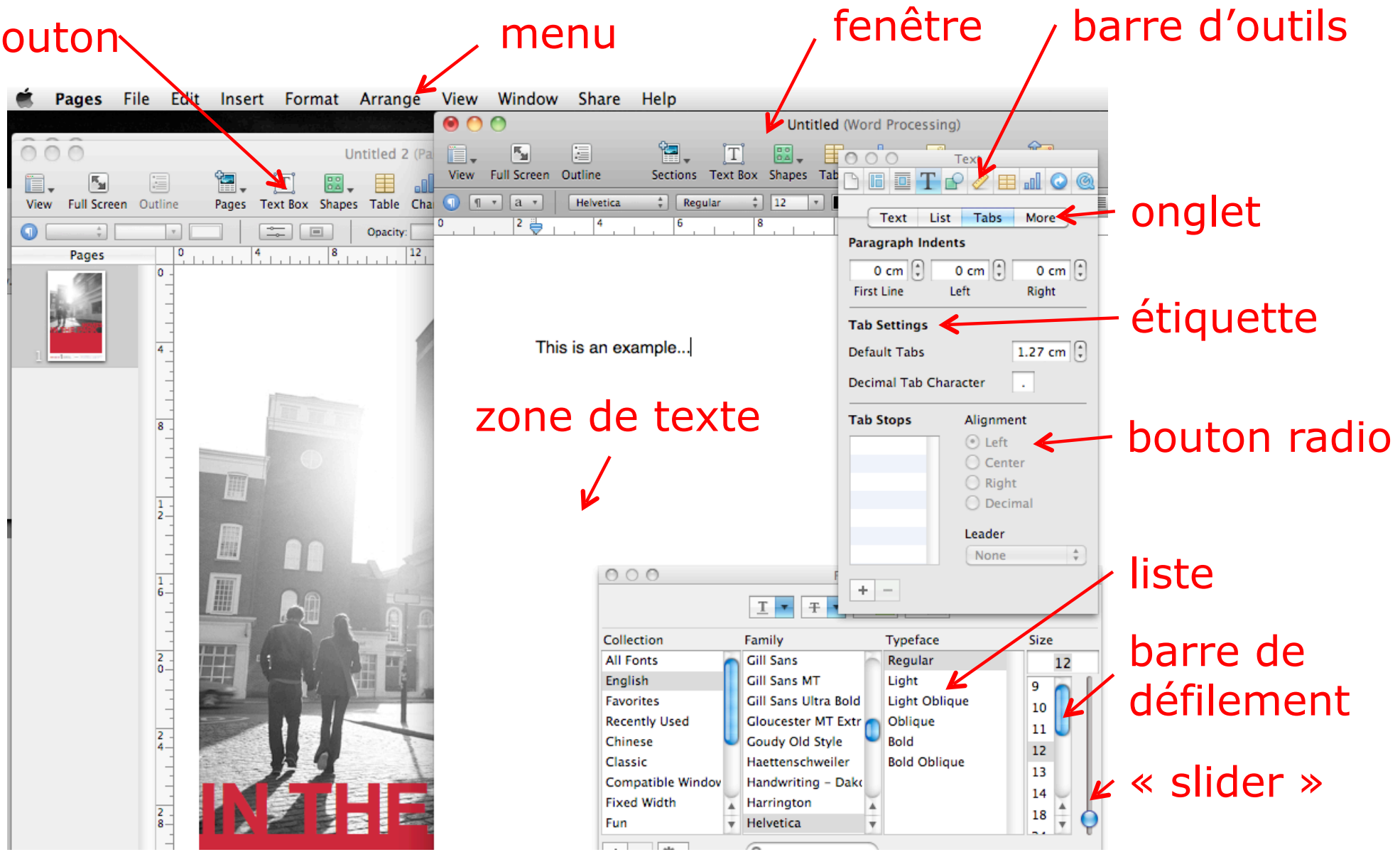
boîte à outils d'interface

bibliothèque d'objets interactifs (les « widgets »)
que l'on assemble pour construire l'interface

fonctionnalités pour faciliter la programmation
d'applications graphiques interactives

Windows : MFC, Windows Forms (.NET)
Mac OS X : Cocoa
Unix/Linux : Motif, QT, GTK+
Multiplateforme : Java AWT/Swing

les « widgets » (window gadget)

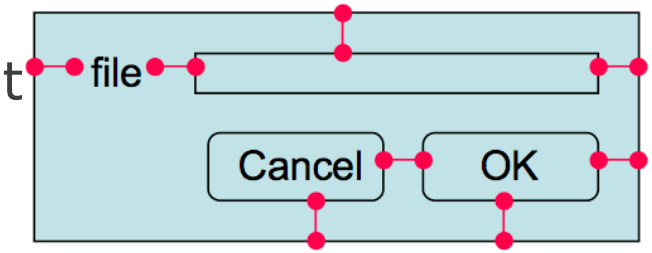


placement de widgets

Boîtes à outilles contrôlent
le placement des widgets :

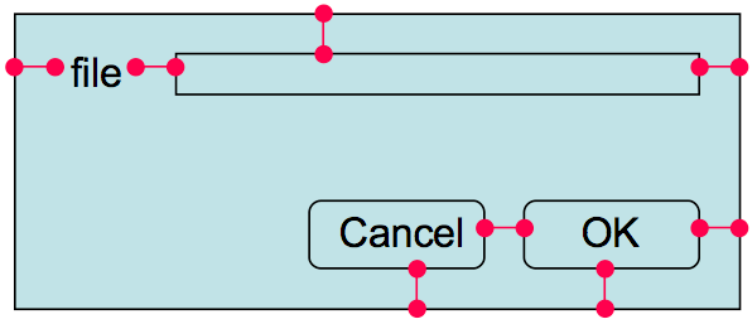
- il faut être indépendant de la taille des widgets

(menu au moins égale à son plus large item,
en changement de taille la barre de défilement
et le texte s'ajustent)



- gestionnaires de géométrie,

dans le widgets composés



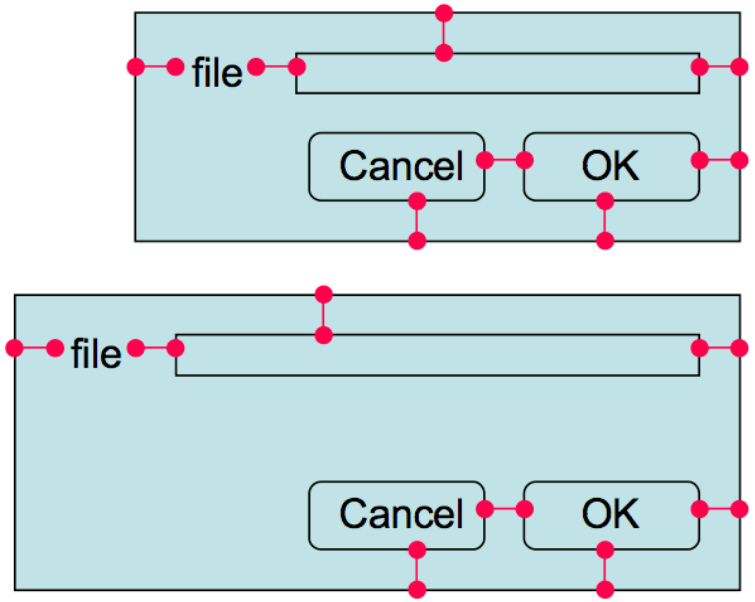
placement de widgets

règles générales

- imbrication géométrique d'un widget fils dans son parent
- contrôle par le parent du placement de ses fils

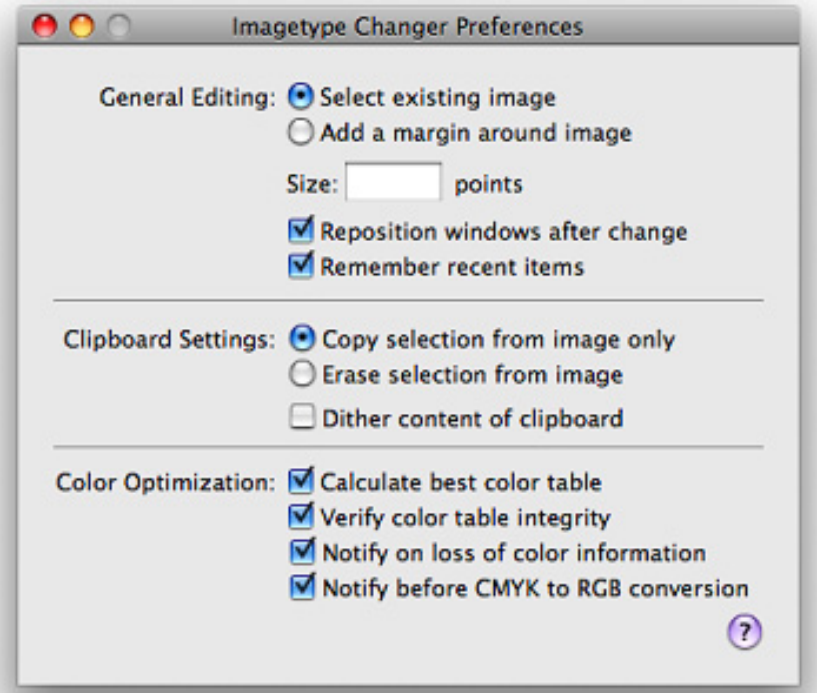
algorithme de placement

- taille naturelle de chaque fils
- taille et position finales imposées par le parent
- contraintes : grille, formulaire, etc.



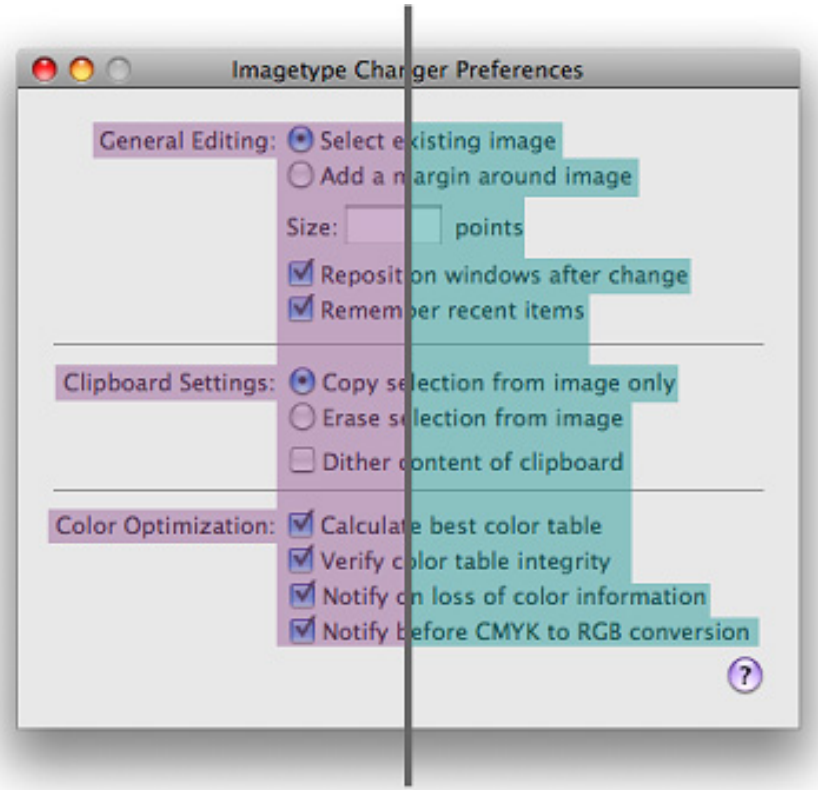
Placement des widgets (design)

guides de placement (Mac OS X)



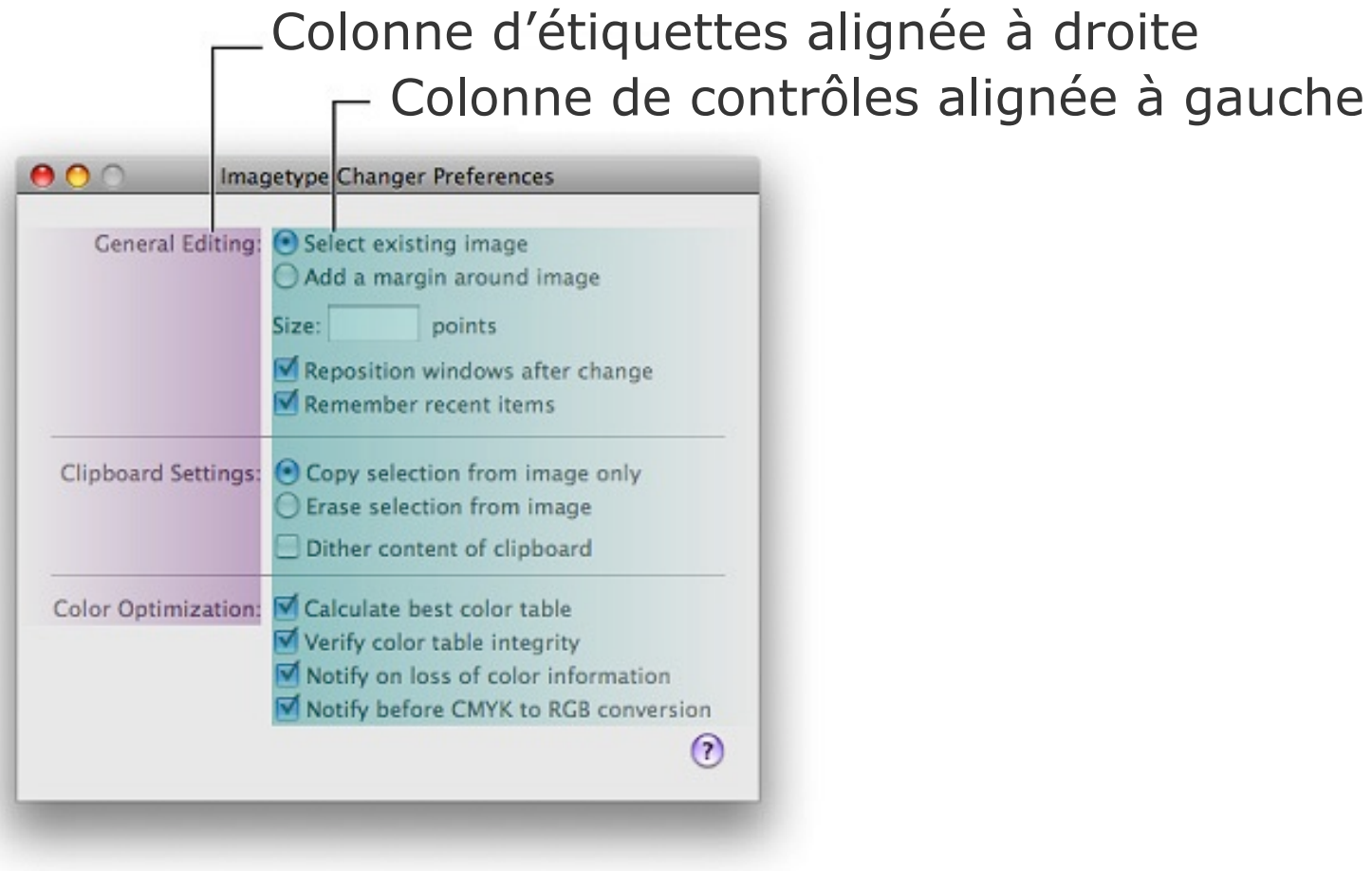
guides de placement (Mac OS X)

« **Center-equalization** » : équilibre visuelle du contenu d'un composant, à droite et à gauche de la médiane



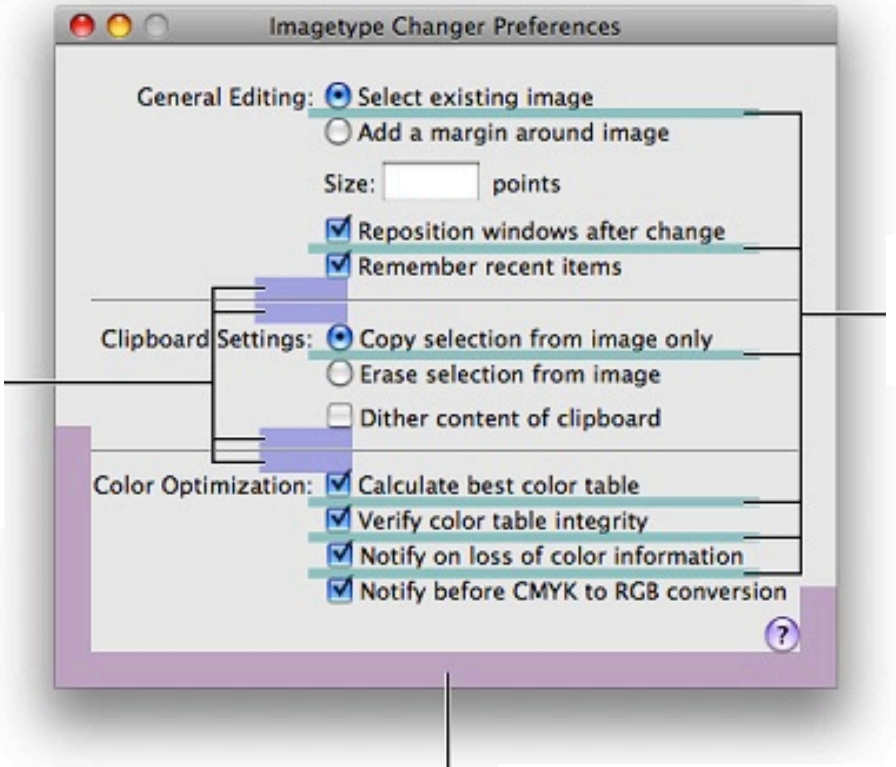
guides de placement (Mac OS X)

Alignement



guides de placement (Mac OS X)

Espacement



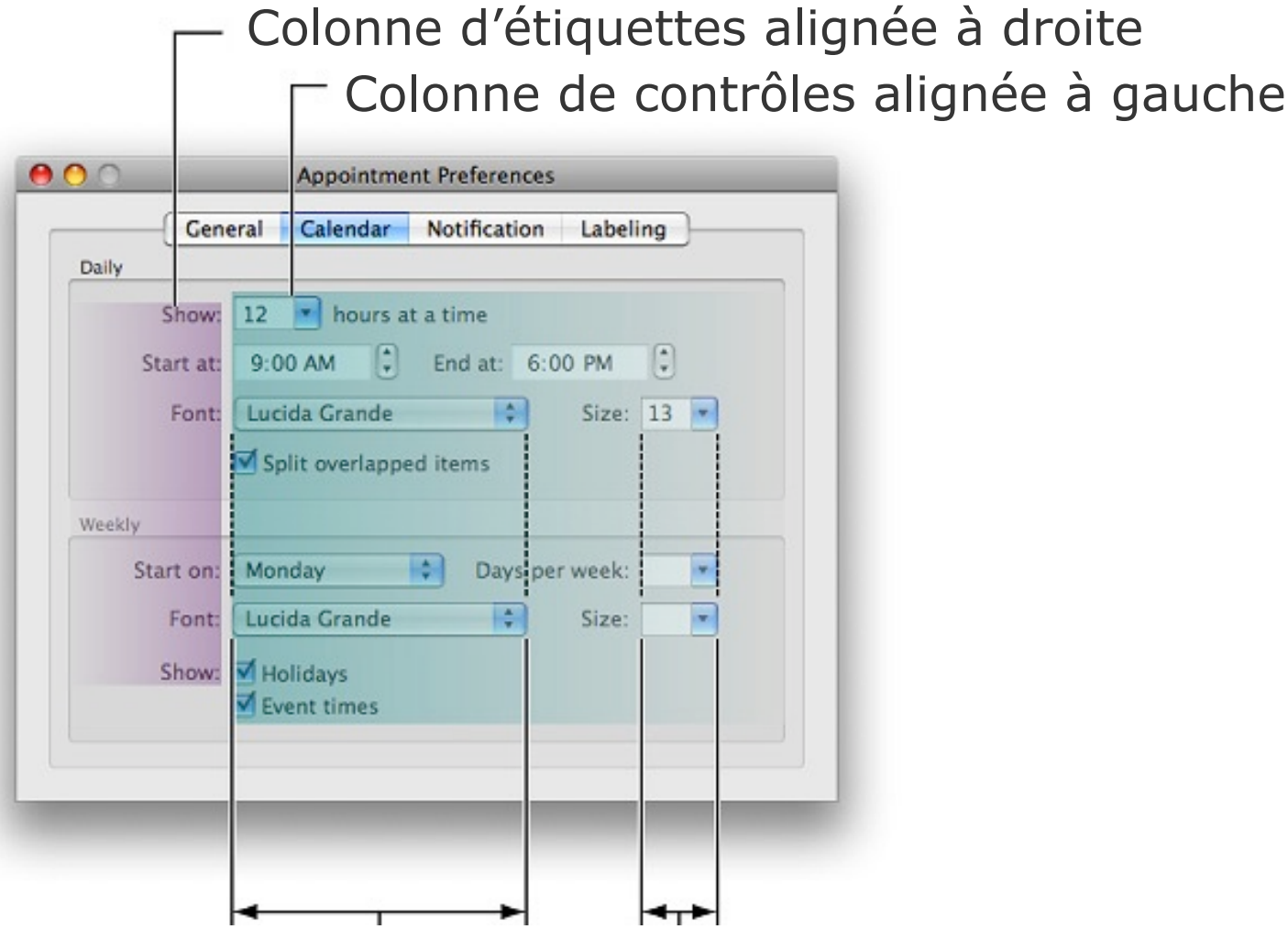
Même hauteur avant et après les lignes de séparation

Même espace entre des contrôles groupés

Même espace dans tous les trois côtés

guides de placement (Mac OS X)

Alignement et cohérence

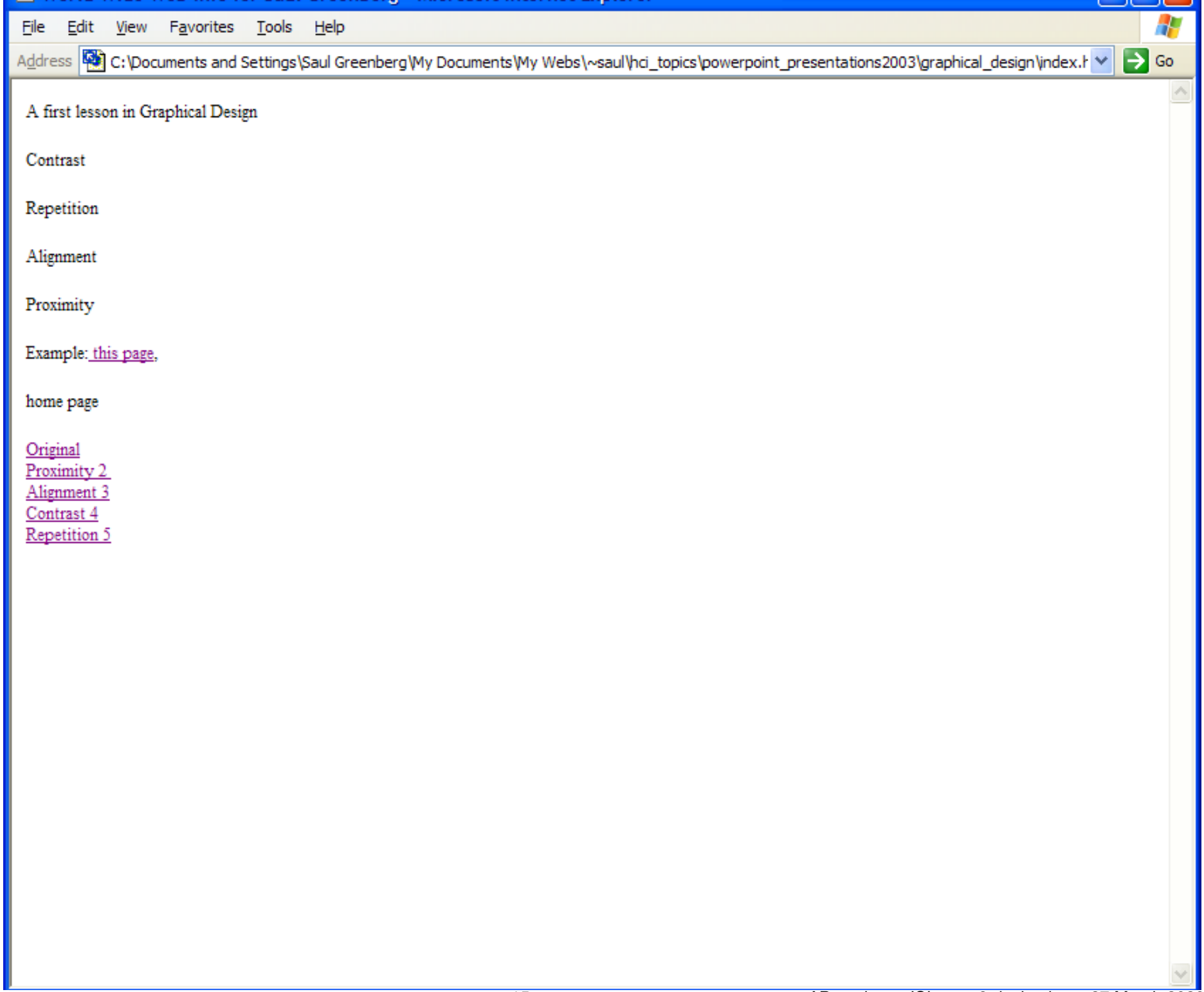


CRAP

contraste, répétition, alignement, proximité

Major sources: Designing Visual Interfaces, Mullet & Sano, Prentice Hall / Robin Williams Non-Designers Design Book, Peachpit Press

Slide deck by Saul Greenberg. Permission is granted to use this for non-commercial purposes as long as general credit to Saul Greenberg is clearly maintained.
Warning: some material in this deck is used from other sources without permission. Credit to the original source is given if it is known.


File Edit View Favorites Tools Help
Address C:\Documents and Settings\Saul Greenberg\My Documents\My Webs\~saul\hci_topics\powerpoint_presentations2003\graphical_design\index.h
Go
A first lesson in Graphical Design
Contrast
Repetition
Alignment
Proximity
Example: this page,
home page
Original
Proximity 2
Alignment 3
Contrast 4
Repetition 5

Good Design Is As Easy as 1-2-3

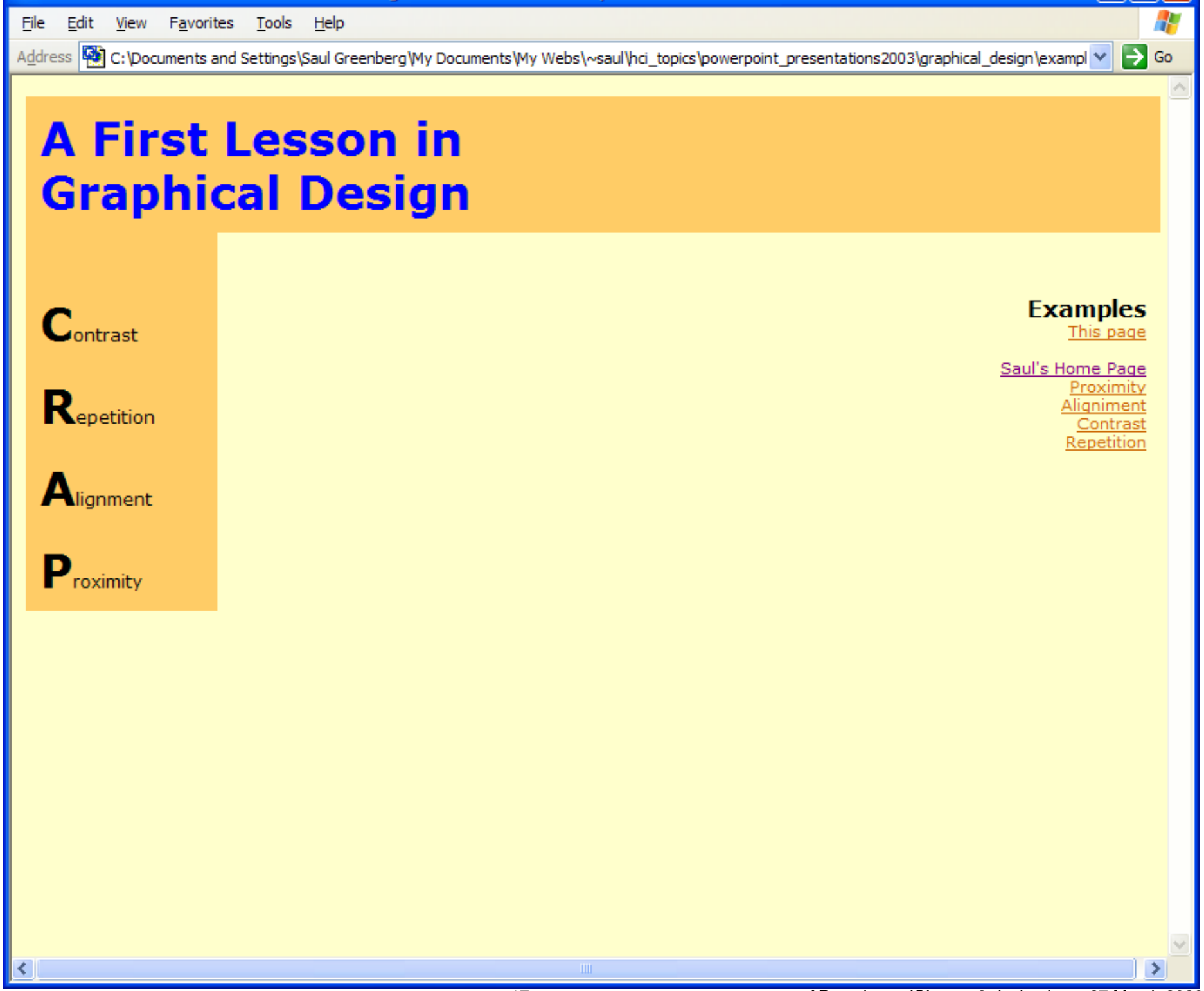
1. Learn the principles.
They're simpler than you might think.
2. Recognize when you're not using them.
Put it into words -- name the problem.
3. Apply the principles.
You'll be amazed.

Good design
is as easy as . . .

1 **Learn the principles.**
They're simpler than you might think.

2 **Recognize when you're not using them.**
Put it into words—name the problem.

3 **Apply the principles.**
You'll be amazed.


File Edit View Favorites Tools Help
Address C:\Documents and Settings\Saul Greenberg\My Documents\My Webs\~saul\hci_topics\powerpoint_presentations2003\graphical_design\exampl
Go
A First Lesson in Graphical Design
Contrast
Repetition
Alignment
Proximity
Examples
This page
Saul's Home Page
Proximity
Alignment
Contrast
Repetition

CRAP

- **C**ontraste
- **R**épétition
- **A**lignement
- **P**roximité

Robin Williams Non-Designers Design Book, Peachpit Press

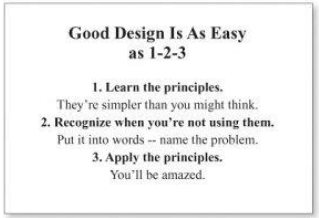
CRAP

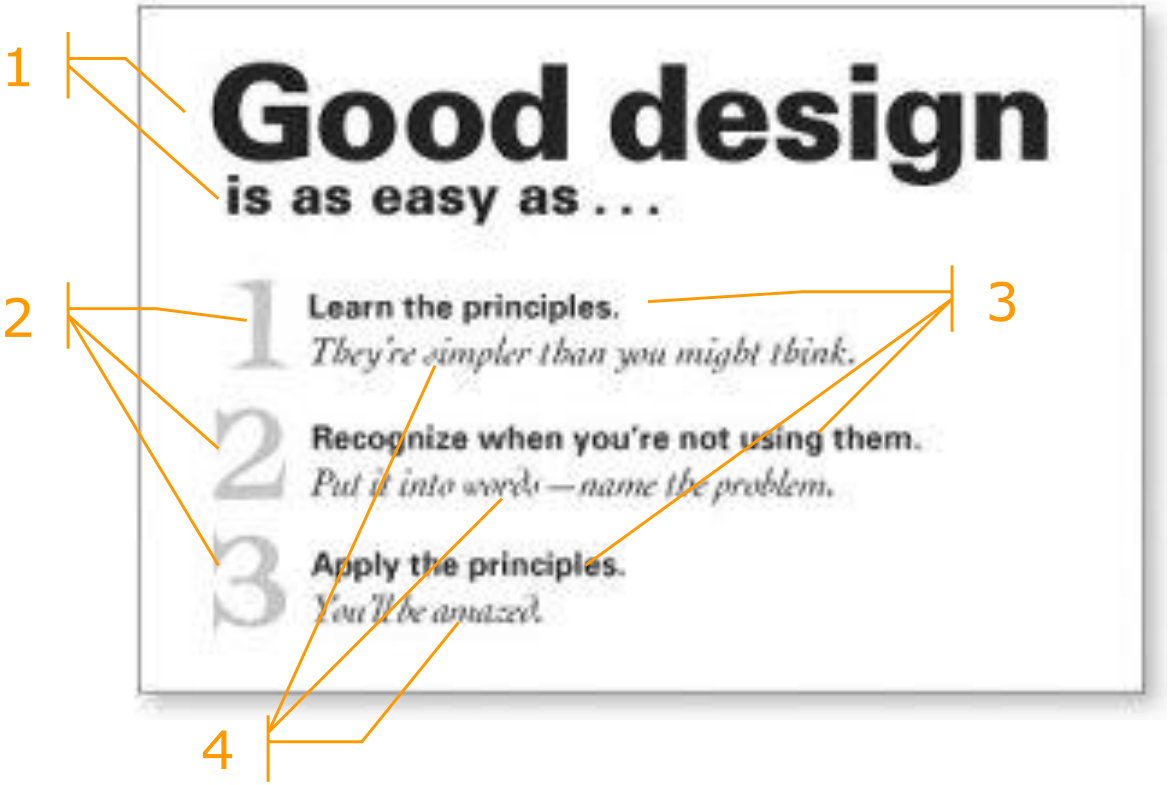
- **Contraste**
  - Faire des choses différentes différents
  - Maitre en évidence les élém. dominantes
  - Faire élém. moins importants moins visib.
  - Créer un dynamisme
- **Répétition**
- **Alignement**
- **Proximité**

Robin Williams Non-Designers Design Book, Peachpit Press

CRAP

- **C**ontraste
- **Répétition**
  Conception repetée au long de l'interface
  Consistance
  Créer unité
- **A**lignement
- **P**roximité





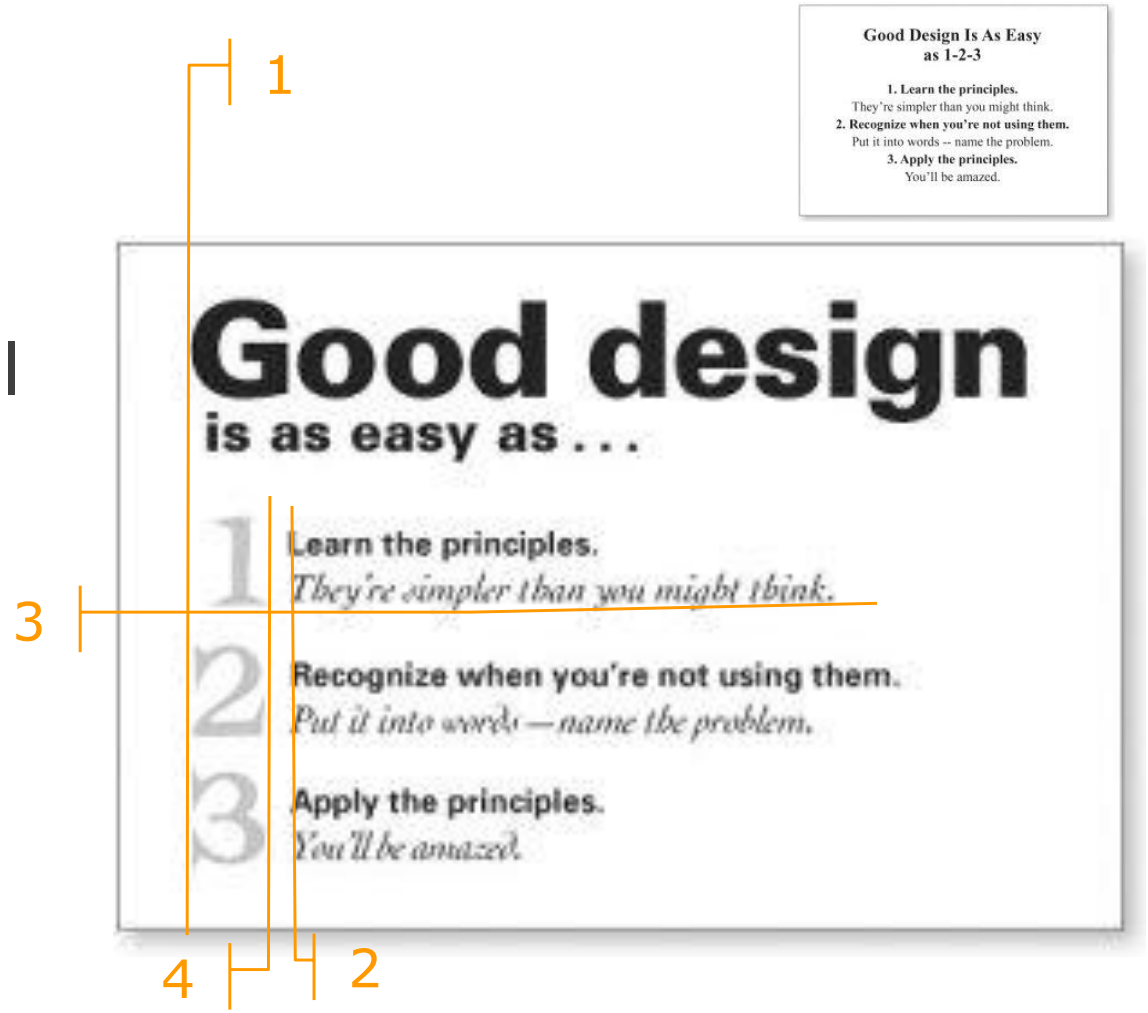
Robin Williams Non-Designers Design Book, Peachpit Press

CRAP

- **C**ontraste
- **R**épétition
- **A**lignement

  Créer un flux visuel

  Connecter élém.
- **P**roximité



Robin Williams Non-Designers Design Book, Peachpit Press

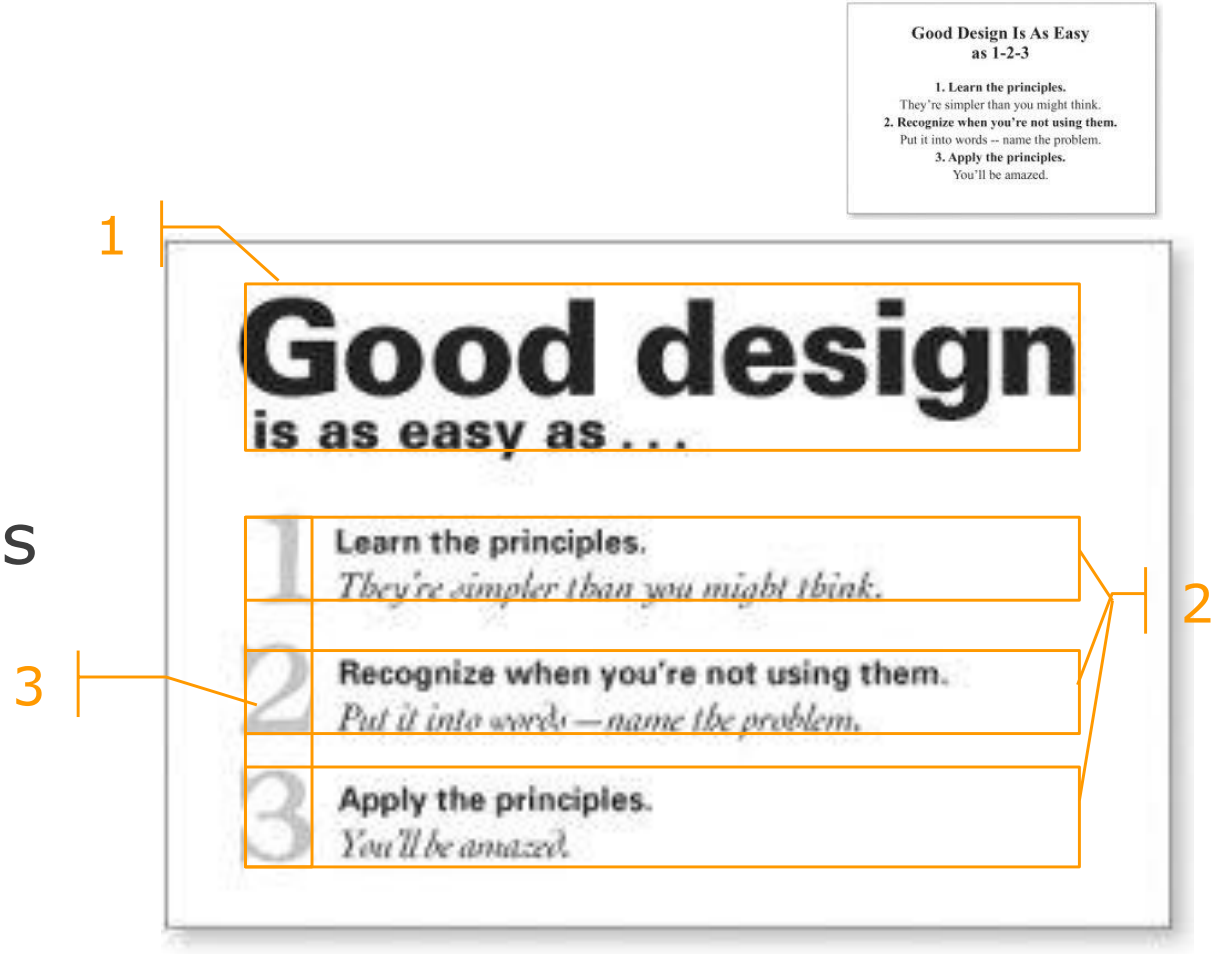
CRAP

- **C**ontraste
- **R**épétition
- **A**lignement
- **P**roximité

  Groupes évidentes

  Indépendants

  séparées



Robin Williams Non-Designers Design Book, Peachpit Press

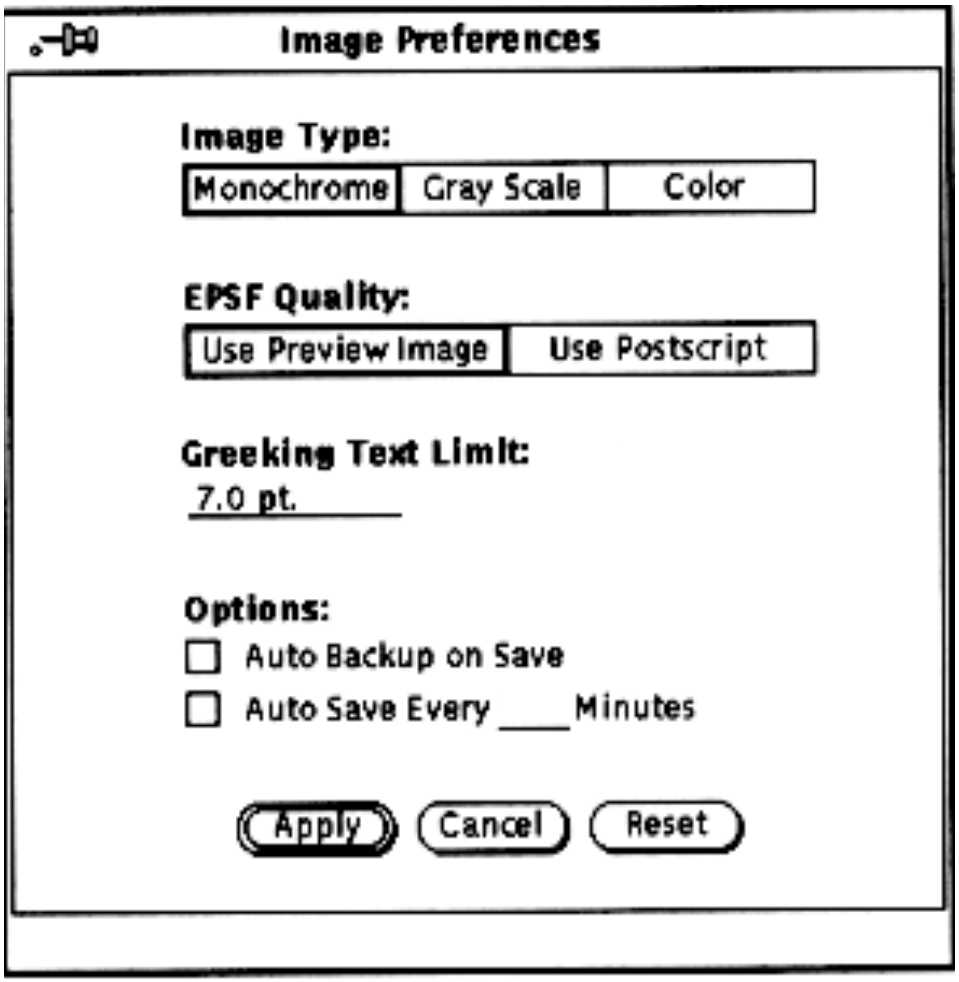
Qu'est-ce que tu vois d'abord?

- CRAP donne des indices sur la façon de lire le graphique

Robin Williams Non-Designers Design Book, Peachpit Press

Qu'est-ce que tu vois d'abord?

- Ici contraste mais proximité pas bien utilisé
  - Structure ambigüe
  - Groupes ambigües

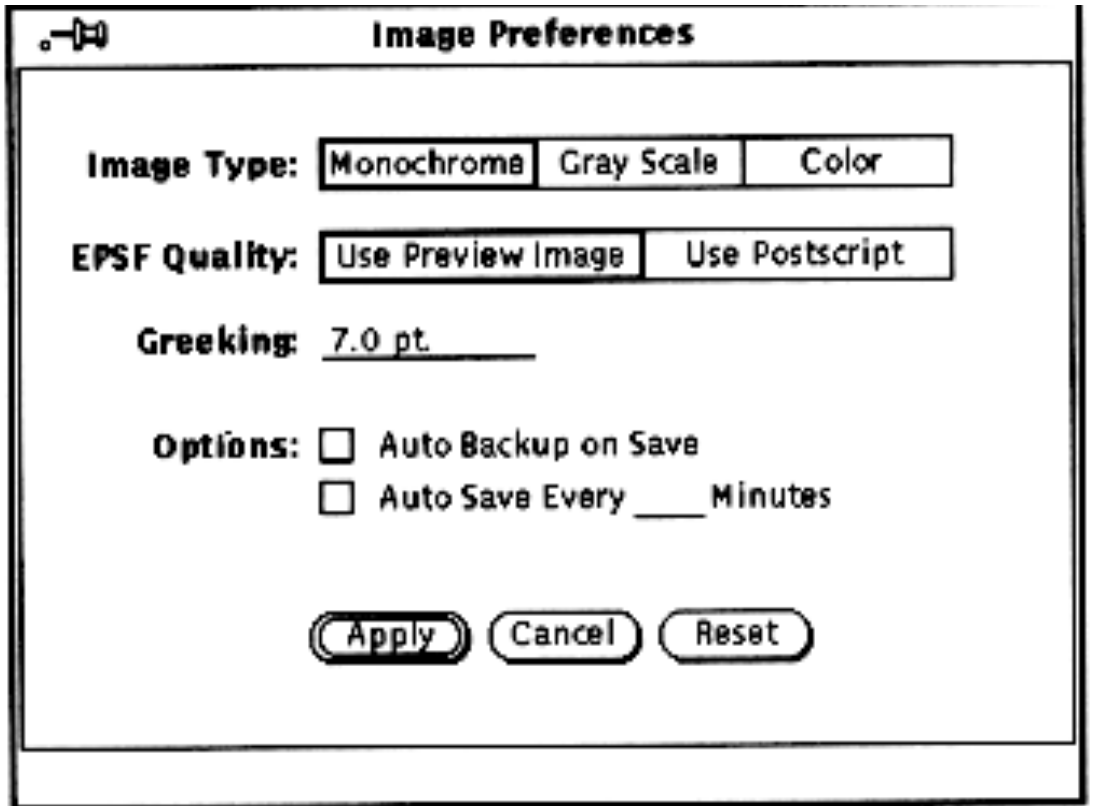


✗

Robin Williams Non-Designers Design Book, Peachpit Press

Qu'est-ce que tu vois d'abord?

- Proximité (séparation gauche/droit)
  - Structure claire

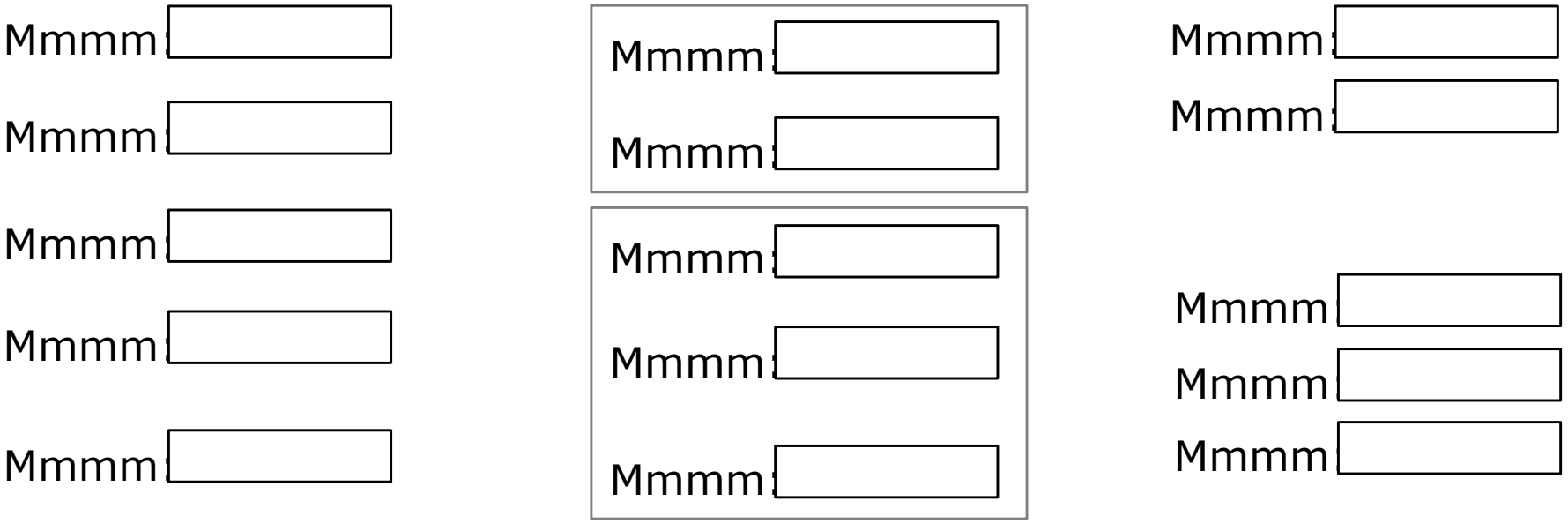


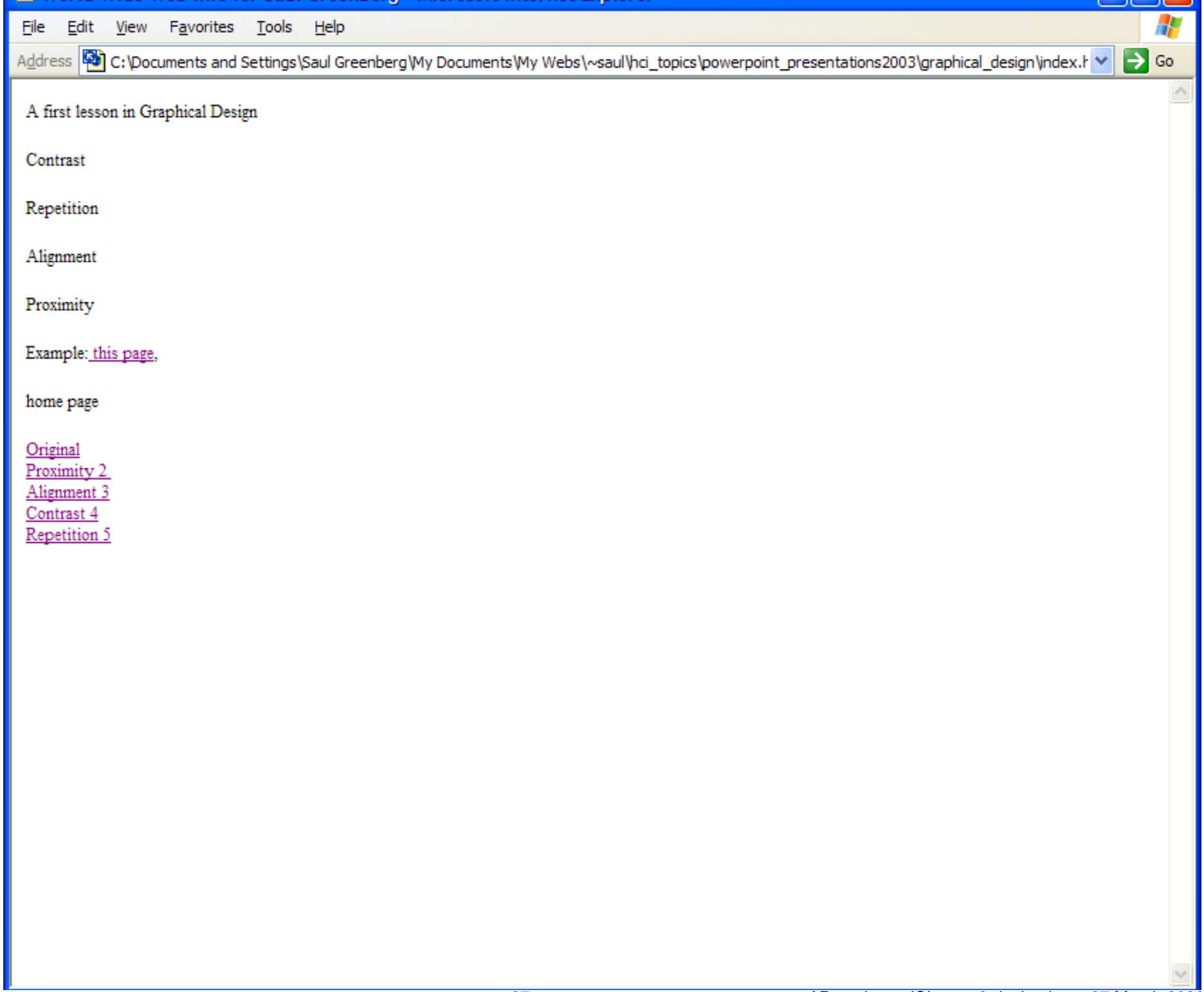
✓

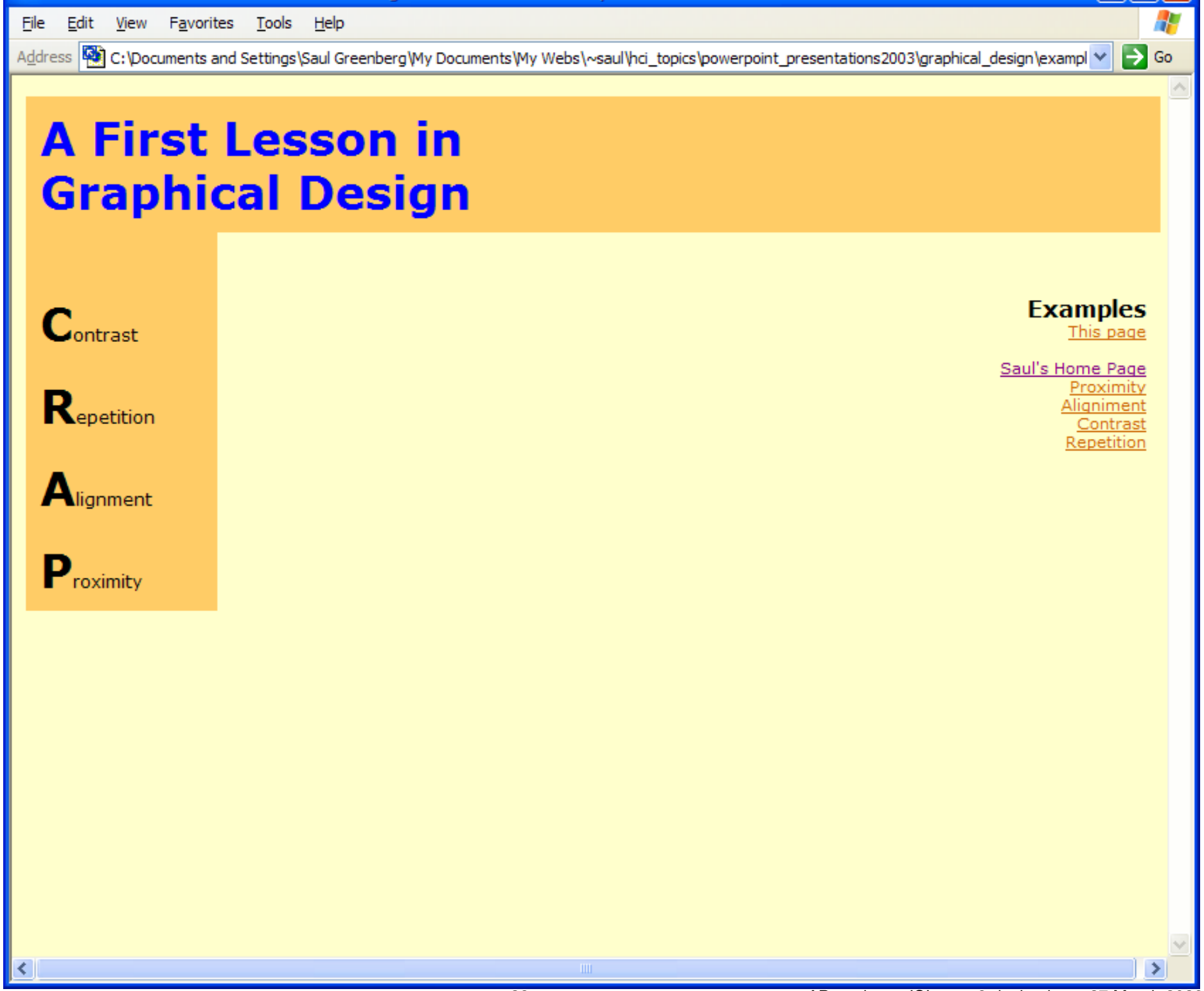
Robin Williams Non-Designers Design Book, Peachpit Press

Qu'est-ce que tu vois d'abord?

- La puissance de la proximité
  - Alignement
  - Structure explicite peut être améliorée
  - Espace blanc




File Edit View Favorites Tools Help
Address C:\Documents and Settings\Saul Greenberg\My Documents\My Webs\~saul\hci_topics\powerpoint_presentations2003\graphical_design\index.h
Go
A first lesson in Graphical Design
Contrast
Repetition
Alignment
Proximity
Example: this page,
home page
Original
Proximity 2
Alignment 3
Contrast 4
Repetition 5


File Edit View Favorites Tools Help
Address C:\Documents and Settings\Saul Greenberg\My Documents\My Webs\~saul\hci_topics\powerpoint_presentations2003\graphical_design\exampl
Go
A First Lesson in Graphical Design
Contrast
Repetition
Alignment
Proximity
Examples
This page
Saul's Home Page
Proximity
Alignment
Contrast
Repetition

File Edit View Favorites Tools Help

Address C:\Documents and Settings\Saul Greenberg\My Documents\My Webs\~saul\hci_topics\powerpoint_presentations2003\graphical_design\versior Go

GroupLab
The University of Calgary

Saul Greenberg GroupLab Dept Computer Science University of Calgary

Saul Greenberg, Professor
Human-Computer Interaction &
Computer Supported Cooperative Work
Dept. of Computer Science
University of Calgary
Calgary, Alberta
CANADA T2N 1N4
Phone: +1 403 220-6087
Fax: +1 403 284-4707
Email: saul@cpsc.ucalgary.ca

Research

GroupLab project describes research by my group

Publications by our group; most available in HTML, PDF, and postscript

Project snapshots describes select projects done in Grouplab

Grouplab software repository

Grouplab people

Graduate Students

I have a few openings for MSc and PhD students who are interested in Human Computer Interaction and / or Computer Supported Cooperative Work. Some research and project ideas honors and graduate students

Courses offered this year

CPSC 481: Foundations and Principles of Human Computer Intera

Original

File Edit View Favorites Tools Help

Address C:\Documents and Settings\Saul Greenberg\My Documents\My Webs\~saul\hci_topics\powerpoint_presentations2003\graphical_design\versior Go

GroupLab
The University of Calgary

Saul Greenberg GroupLab Dept Computer Science University of Calgary

Saul Greenberg, Professor
Human-Computer Interaction &
Computer Supported Cooperative Work

Dept. of Computer Science
University of Calgary
Calgary, Alberta
CANADA T2N 1N4

Phone: +1 403 220-6087
Fax: +1 403 284-4707
Email: saul@cpsc.ucalgary.ca

Research

GroupLab project describes research by my group
Publications by our group; most available in HTML, PDF, and postscript
Project snapshots describes select projects done in Grouplab
Grouplab software repository
Grouplab people

Graduate Students

I have a few openings for MSc and PhD students who are interested in Human Computer Interaction and / or Computer Supported Cooperative Work. Some research and project ideas honors and graduate students

Courses offered this year

CPSC 481: Foundations and Principles of Human Computer Intera
CPSC 581: Human Computer Interaction II: Interaction Desig
CPSC 601.13: Computer Supported Cooperative Work

Proximité

File Edit View Favorites Tools Help

Address C:\Documents and Settings\Saul Greenberg\My Documents\My Webs\~saul\hci_topics\powerpoint_presentations2003\graphical_design\version Go

Saul Greenberg GroupLab Dept Computer Science University of Calgary

GroupLab
The University of Calgary

Saul Greenberg, Professor
Human-Computer Interaction &
Computer Supported Cooperative Work

Dept. of Computer Science
University of Calgary
Calgary, Alberta
CANADA T2N 1N4

Phone: +1 403 220-6087
Fax: +1 403 284-4707
Email: saul@cpsc.ucalgary.ca

Research
GroupLab project describes research by my group
Publications by our group; most available in HTML, PDF, and postscript
Project snapshots describes select projects done in Grouplab
Grouplab software repository
Grouplab people

Graduate Students
I have a few openings for MSc and PhD students who are interested in Human Computer Interaction and / or Computer Supported Cooperative Work. Some research and project ideas honors and graduate students

Courses offered this year
CPSC 481: Foundations and Principles of Human Computer Interaction
CPSC 581: Human Computer Interaction II: Interaction Design
CPSC 601.13: Computer Supported Cooperative Work

Previous Years:
CPSC 681: Research Methodologies in Human Computer Interaction
CPSC 699: Research Methodology for Computer Science (old!)
CPSC 601.48: Special Topics: Heuristic Evaluation

Alignement

File Edit View Favorites Tools Help

Address C:\Documents and Settings\Saul Greenberg\My Documents\My Webs\~saul\hci_topics\powerpoint_presentations2003\graphical_design\version Go

GroupLab
The University of Calgary

Saul Greenberg GroupLab Dept Computer Science University of Calgary

Saul Greenberg
Professor

Human-Computer Interaction &
Computer Supported Cooperative Work

Dept. of Computer Science
University of Calgary
Calgary, Alberta
CANADA T2N 1N4

Phone: +1 403 220-6087
Fax: +1 403 284-4707
Email: saul@cpsc.ucalgary.ca

Graduate Students

Research Ideas. I have a few openings for MSc and PhD students who are interested in Human Computer Interaction and / or Computer Supported Cooperative Work.

Courses offered this year

CPSC 481: Foundations and Principles of Human Computer Interaction
CPSC 581: Human Computer Interaction II: Interaction Design
CPSC 601.13: Computer Supported Cooperative Work

Previous Years

CPSC 681: Research Methodologies in Human Computer Interaction
CPSC 699: Research Methodology for Computer Science (old!)
CPSC 601.48: Special Topics: Heuristic Evaluation
CPSC 601.56: Advanced Topics in HCI: Media Spaces and Casual Interaction
SENG 609.05: Graphical User Interfaces: Design and Usability
SENG 609.06: Special Topics in Human Computer Interaction
Ego alert: My entry on U Calgary's 'Great Teachers" Web Site

Administration

Ethics Committee for research with human subjects; I am the chair

Last updated: March 20, 1867

Contraste

File Edit View Favorites Tools Help

Address C:\Documents and Settings\Saul Greenberg\My Documents\My Webs\~saul\hci_topics\powerpoint_presentations2003\graphical_design\versior Go

Saul Greenberg GroupLab Dept Computer Science University of Calgary

GroupLab
The University of Calgary

Saul Greenberg
Professor

Human-Computer Interaction &
Computer Supported Cooperative Work

Dept. of Computer Science
University of Calgary
Calgary, Alberta
CANADA T2N 1N4

Phone: +1 403 220-6087
Fax: +1 403 284-4707
Email: saul@cpsc.ucalgary.ca

Graduate Students — Research Ideas I have a few openings for MSc and PhD students who are interested in Human Computer Interaction and / or Computer Supported Cooperative Work.

Courses offered this year
- CPSC 481 Foundations and Principles of Human Computer Interaction
- CPSC 581 Human Computer Interaction II: Interaction Design
- CPSC 601.13 Computer Supported Cooperative Work

Previous Years
- CPSC 681 Research Methodologies in Human Computer Interaction
- CPSC 699 Research Methodology for Computer Science (old!)
- CPSC 601.48 Special Topics: Heuristic Evaluation
- CPSC 601.56 Advanced Topics in HCI: Media Spaces and Casual Interaction
- SENG 609.05 Graphical User Interfaces: Design and Usability
- SENG 609.06 Special Topics in Human Computer Interaction
- Ego alert My entry on U Calgary's 'Great Teachers" Web Site

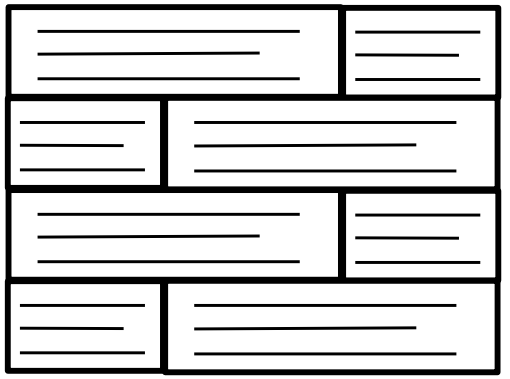
Administration — Ethics Committee for research with human subjec

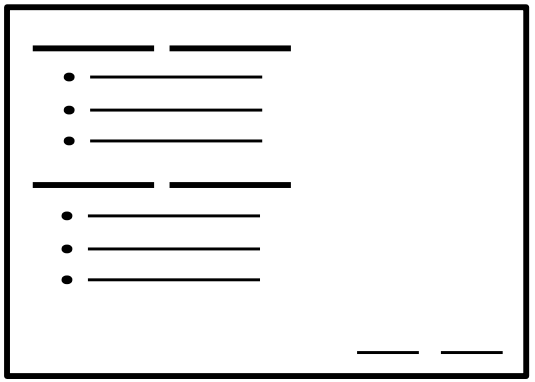
Last updated: March 20, 1867

Répétition

Qu'est-ce que tu vois d'abord?

- Boîtes ne montre pas la structure
  - Utilisez CRAP





Robin Williams Non-Designers Design Book, Peachpit Press

(rappelle) Mémoire et apprentissage

Guides:

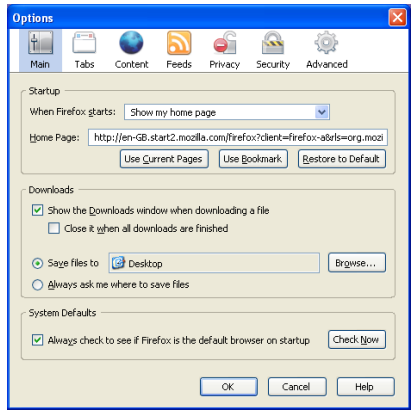
Éviter des procédures complexes (risque d'interférence)

Rappel et reconnaissance (faciliter les deux)
mais la reconnaissance est préférée

ex. menus, icônes, listes de sélection

Placement consistent

ex. boutons « OK » / « Cancel »



Regrouper les informations liées (Gestalt)

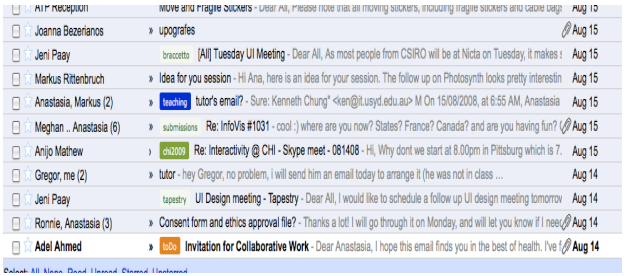
ex. onglets, sous-menus

Aider à l'apprentissage associatif

ex. Aide, Conseils

Aider les associations et contexte

ex. couleurs, des étiquettes, métadonnées temporelles



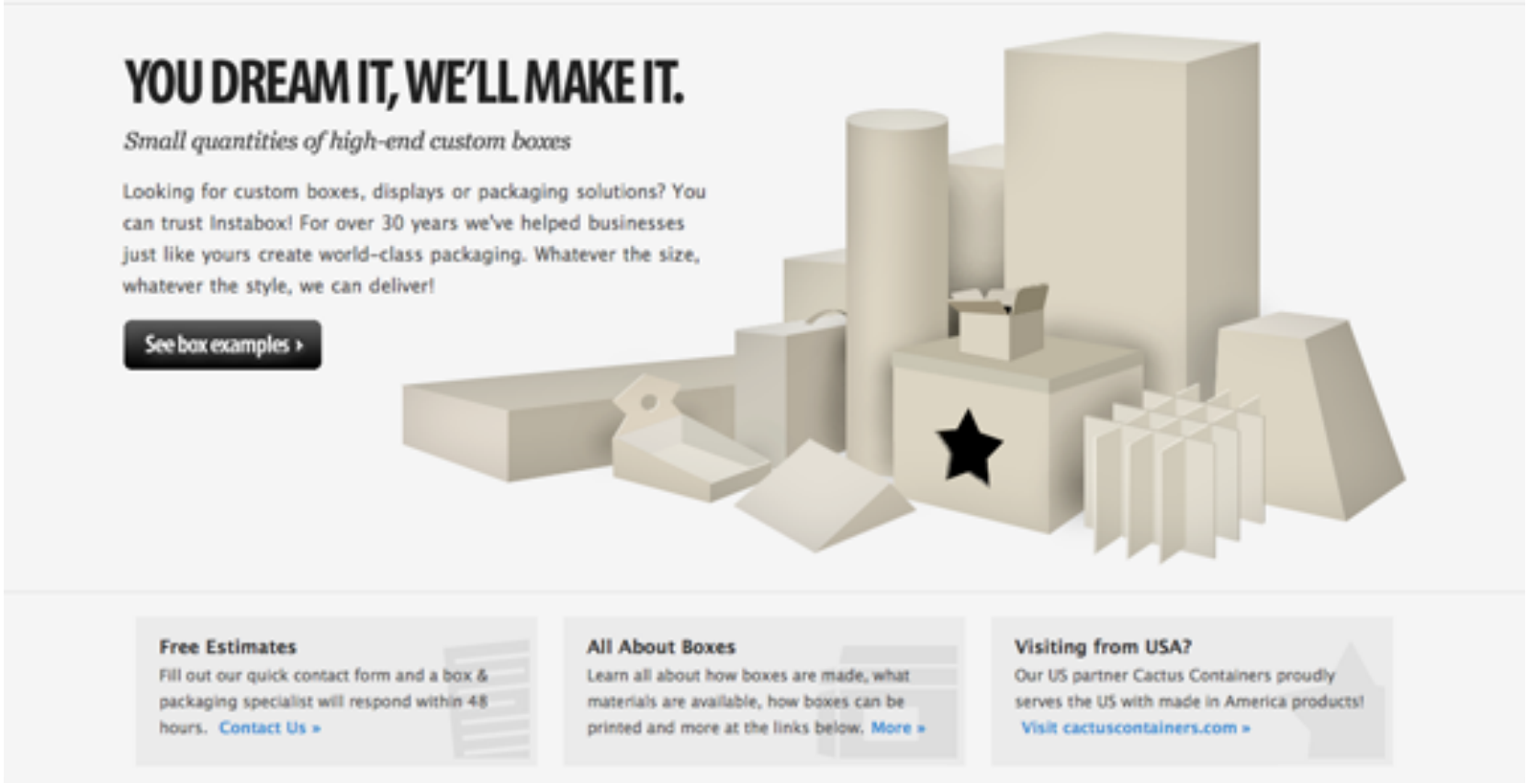
CRAP: Contrast

with images

http://invoicemachine.com/home

CRAP: Contrast

with images



http://www.instabox.com/

CRAP: Contrast

with color

http://fatburgr.com/

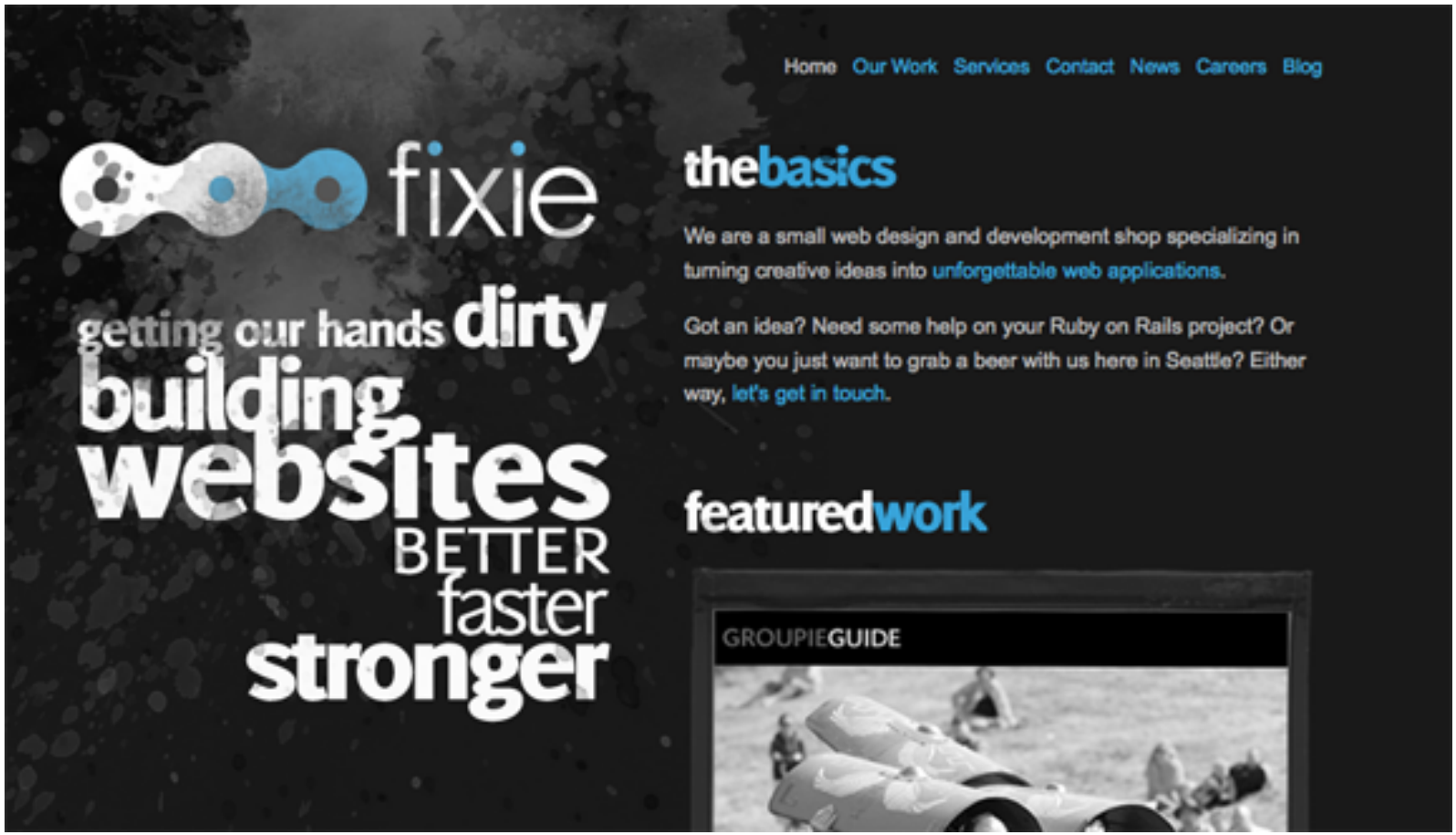
CRAP: Contrast

with color

http://ilovetypography.com/

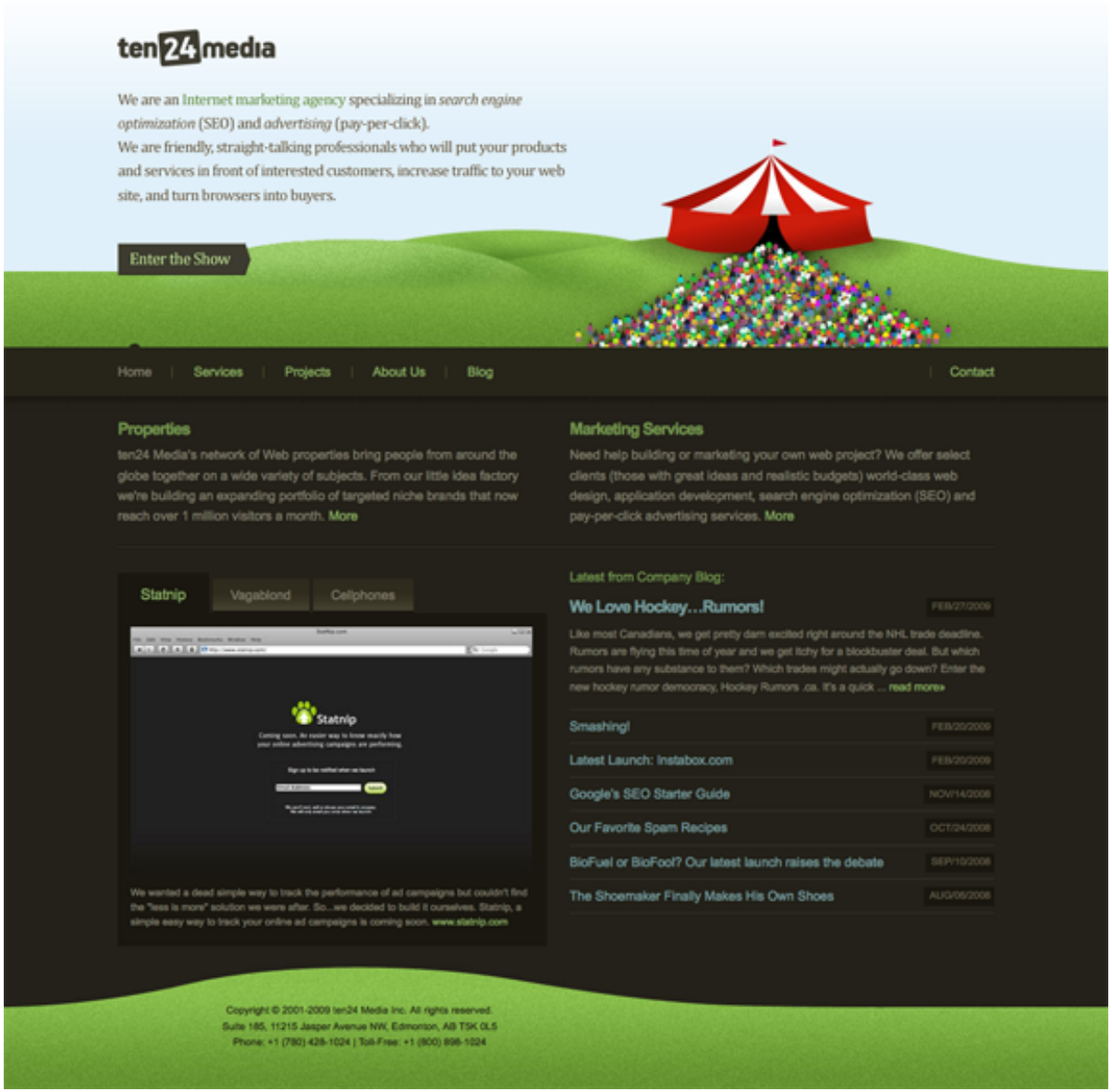
CRAP: Contrast

with fonts



http://fixieconsulting.com/

CRAP: Repetition



http://www.1024media.com/

CRAP: Repetition

http://silverbackapp.com/

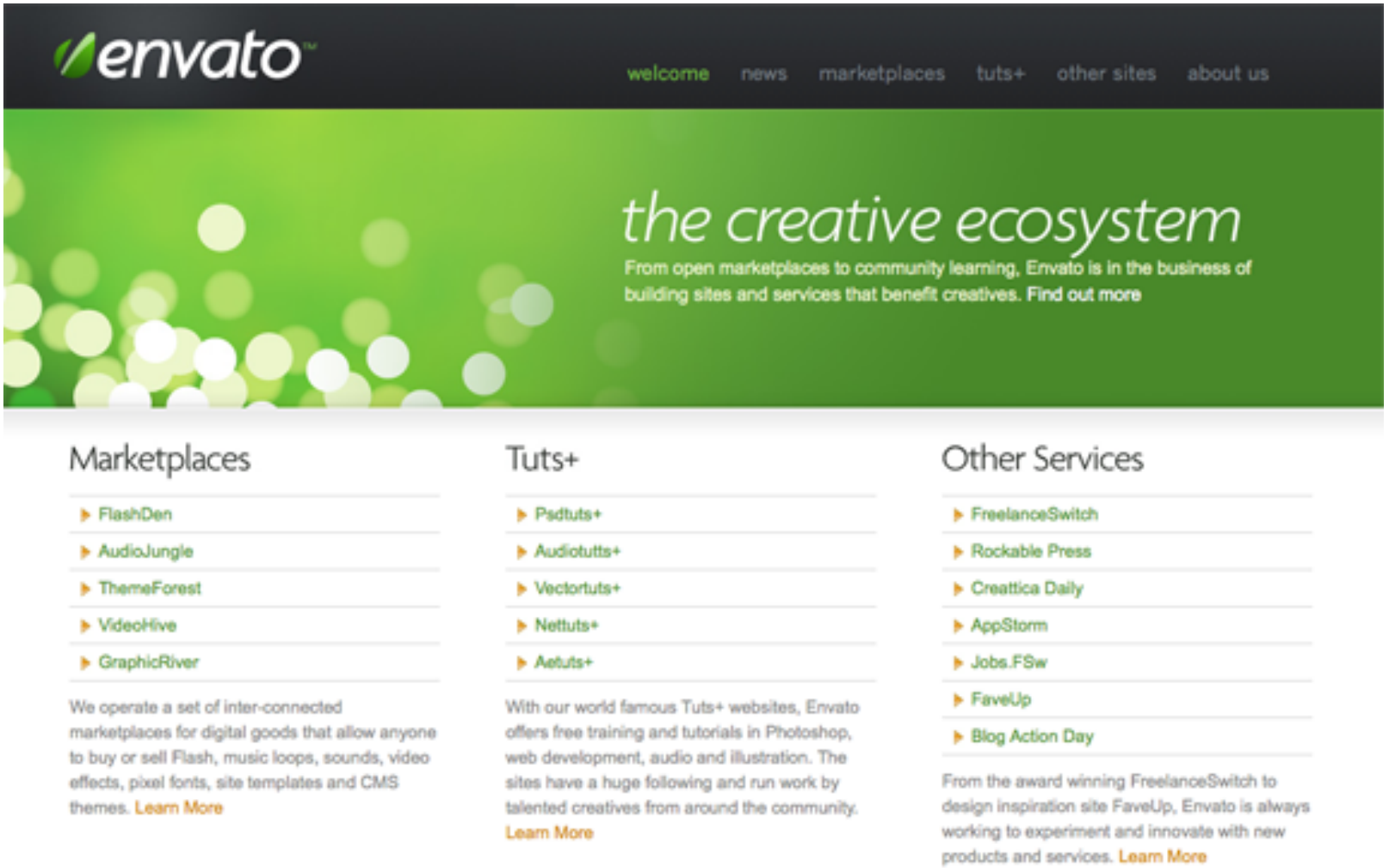
CRAP: Alignment

http://www.blackestate.co.nz/

CRAP: Alignment

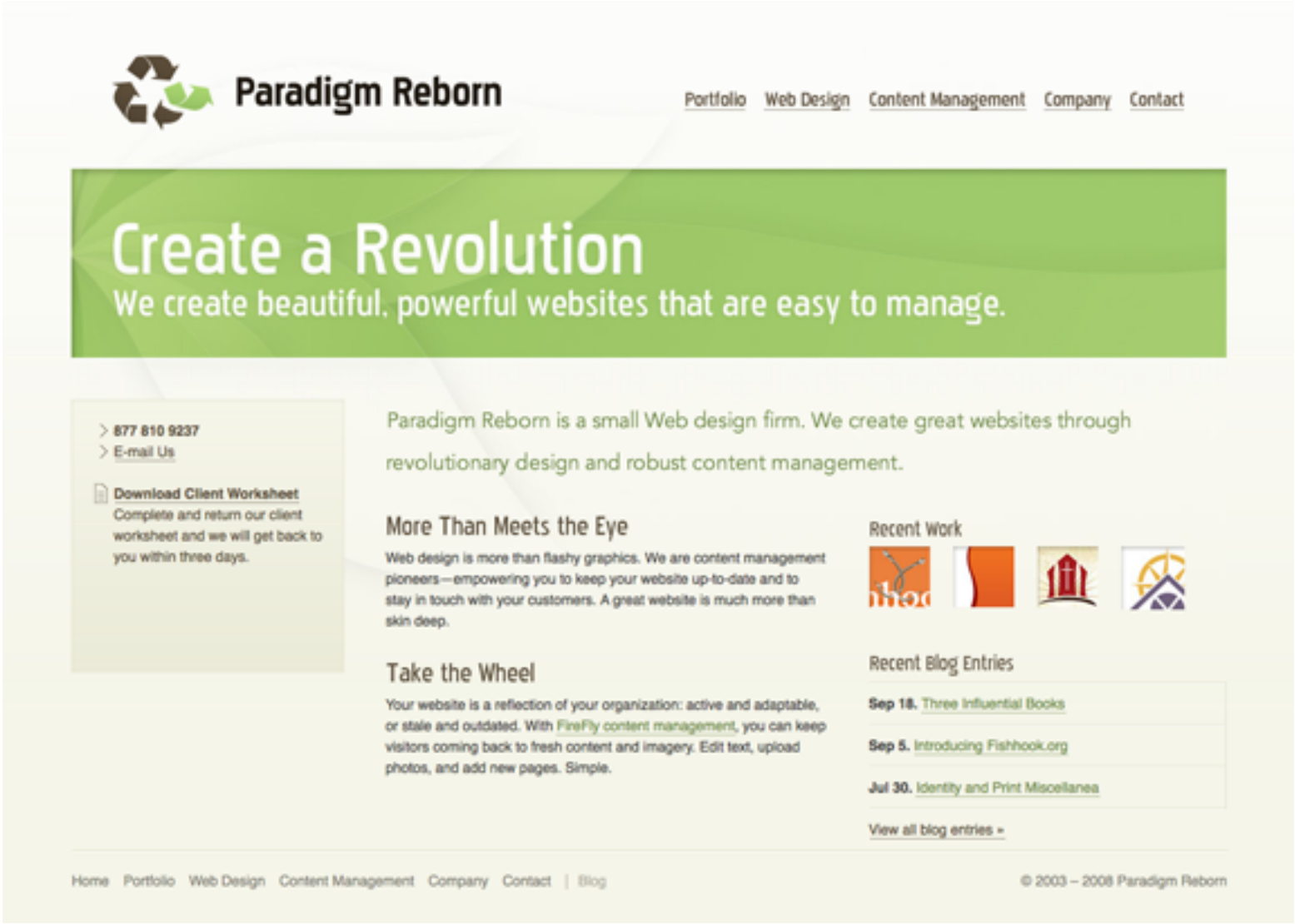
http://www.alistapart.com/

CRAP: Proximity



http://envato.com/

CRAP: Proximity



http://www.createarevolution.com/

facettes d'un widget

présentation

- apparence graphique

comportement

- réactions aux actions de l'utilisateur

interfaces d'application :

notifications de changement d'état

Bouton:

cadre avec un nom à l'intérieur
« enfoncement » ou inversion video lorsque l'on clique dessus
fonction appelée lorsque le bouton est cliqué

facettes d'un widget

présentation

- apparence graphique

comportement

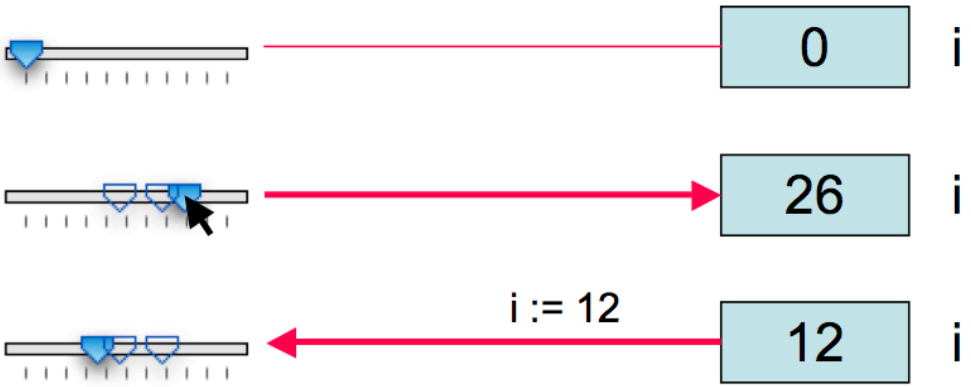
- réactions aux actions de l'utilisateur

interfaces d'application :
notifications de changement d'état

- fonctions de rappel (« callbacks »)
- variables actives (Tcl/Tk)
- envoi de message (Qt)

variables actives

lien bi-directionnel entre une variable d'état du widget et une variable de l'application



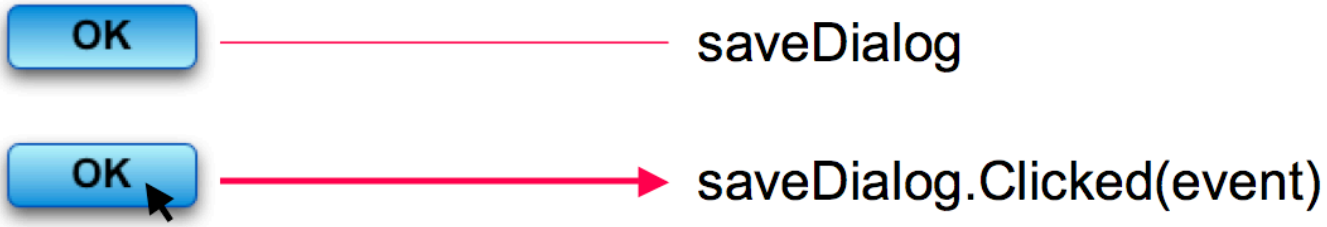

```
main (){
    int i = 0;
    …
    /* widget */
    nc = CreateSlider (…);
    /* var active */
    SetIntegerActiveVariable (nc, &i);
    …
}
```

problèmes

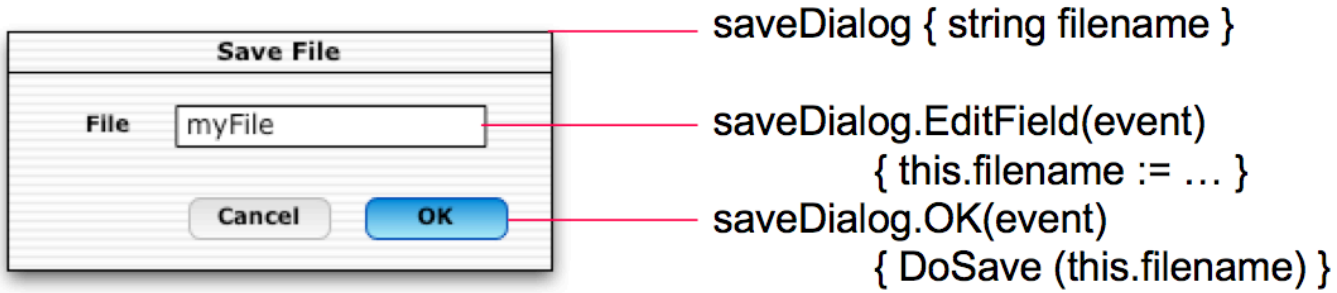
- limité aux types simples
- lien de retour peut être coûteux

envoi de message

association d'un objet à un widget et de méthodes de l'objet aux changements d'état



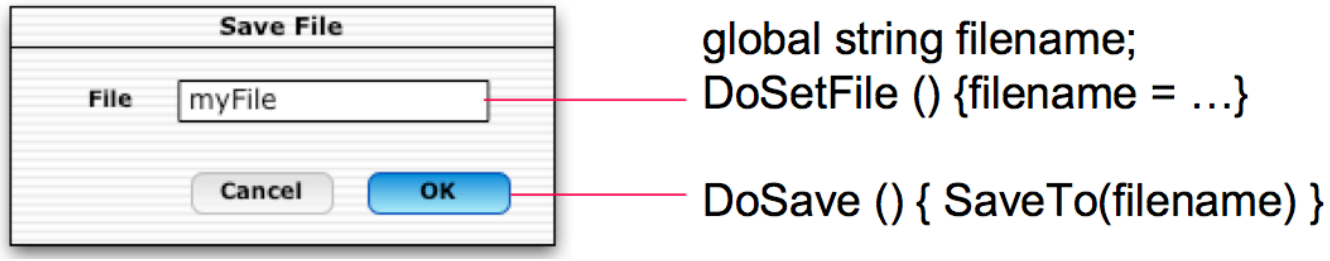
meilleure encapsulation



fonctions de rappel

Enregistrement lors de la création du widget

Appel lors l'activation du widget

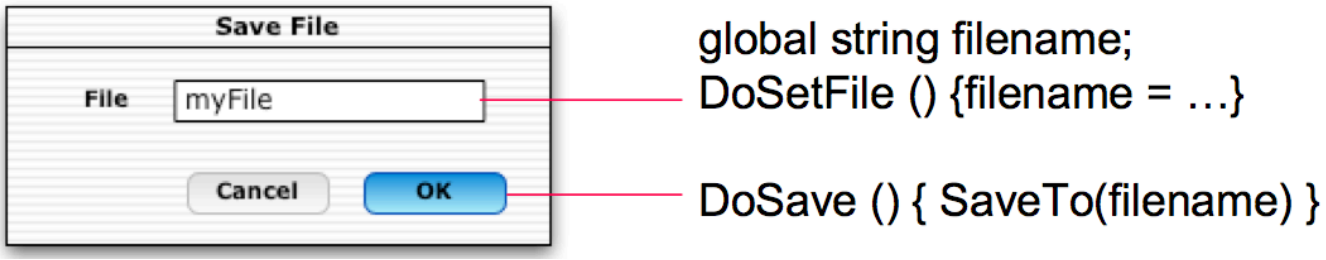


fonctions de rappel

Problème : spaghetti des callbacks

Partage d'état entre plusieurs callbacks par:

- variables globales
  - Trop dans une application réelle
- arbre des widgets : la fonction de rappel est appelée en lui passant le widget qui l'a déclenché
  - Fragile si l'on change la structure, insuffisante pour d'autres données pas associés aux widgets
- « jeton » (token) : donnée enregistrée avec la callback, passée automatiquement au moment de l'appel



fonctions de rappel

```
/* fonction de rappel */
void DoSave (Widget w, void* data) {
     /* récupérer le nom de fichier */
      filename = (char**) data;
     /* appeler la fonction de l'application */
     SaveTo (filename);
     /* fermer la boîte de dialogue */
     CloseWinfow (getParent(getParent(w)));
}

/* programme principal */
main () {
     /* variable contenant le nom du fichier */
     char* filename = "";
     …
     /* créer le widgets et lui associer sa callback */
     ok = CreateButton (....);
     RegisterCallback (ok, DoSave, (void*) &filename);
     …
     /* boucle de traitement des événements */
     MainLoop ();
}
```

programmation événementielle

Événement : Un objet (ou message) qui représente l'interaction d'un utilisateur avec un composant d'interface graphique; peut être « traité » pour créer des composants interactifs

programmation événementielle

Événement : Un objet (ou message) qui représente l'interaction d'un utilisateur avec un composant d'interface graphique; peut être « traité » pour créer des composants interactifs

Types d'événements de l'interface utilisateur :

Souris: déplacer / glisser / cliquez sur le bouton (pression / relâchement), …

Clavier: appuyer sur une touche / relâcher, parfois avec touche de modification (shift / contrôle / alt), …

Écran tactile: toucher doigt / glisser, …

Fenêtre: redimensionner / minimiser / restaurer / fermer, …

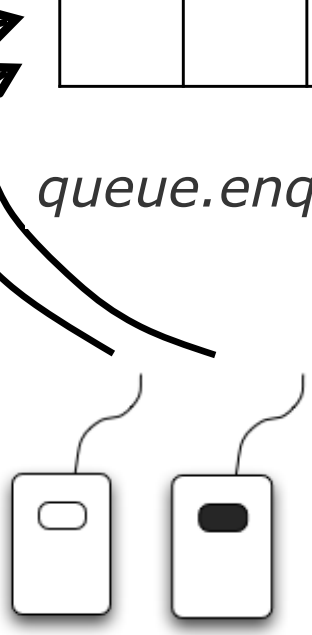
(variation de fonctions de rappel)

programmation événementielle

```
while active
  if queue is not empty
     event <- queue.dequeue()
     source <- findSource(event)
     source.processEvent(event)
  end if
end while
```

File d'attente *(queue)*

queue.enqueue(event)



Animations : horloge comme source d'évènements
« tick » -> évènement -> progrès

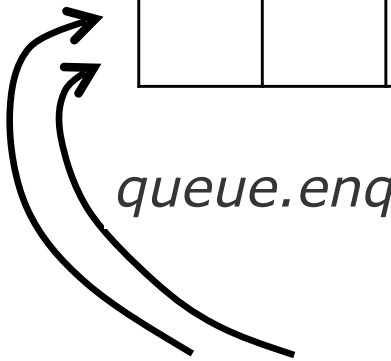
programmation événementielle

```
while active
  if queue is not empty
     event <- queue.dequeue()
     source <- findSource(event)
     source.processEvent(event)
  end if
end while
```

Source :
Mouse Click

File d'attente *(queue)*

queue.enqueue(event)



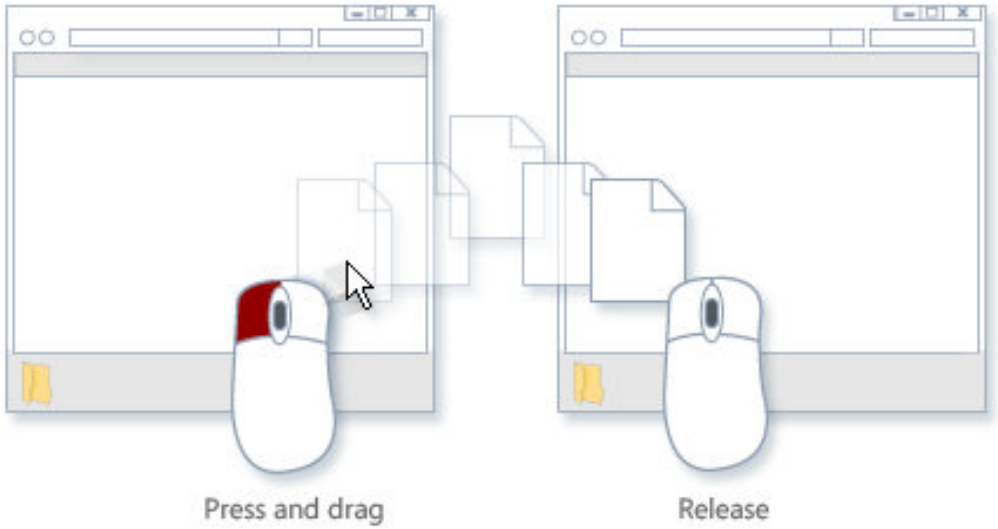
Cible :
Bouton
« Cancel »

```
processEvent(event)
   target <- FindTarget (event)
   if (target ≠ NULL)
    target.processEvent(event)
```

Animations : horloge comme source d'évènements
« tick » -> évènement -> progrès

« drag-and-drop »

Quels sont les « widgets » affectés ?
Quels sont les événements?



Exercice : comment décrire cette interaction avec un « event listener » ?