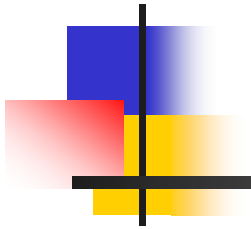
Flex Builder 3

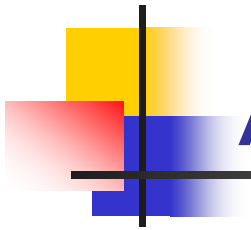
Prise en main
Interface et conteneurs

PureMVC :
http://ressources.mediabox.fr/tutoriaux/flashplatform/programmation/frameworks/puremvc



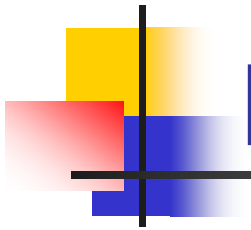
Plan

- Introduction
- Les composants de base
  - Notion de Vue en mxml
- Les contrôles
  - Gestion d'événements en AS
- Le modèle ou gestion de données
  - Modèles de représentation de données
- Enrichissement
  - Définir des conteneurs, personnalisés
- Personnalisation
  - Effets, viewStates, transitions
- 3D
- Applications serveur
  - HTTPServices, Web Services, Bases de données Applications AIR



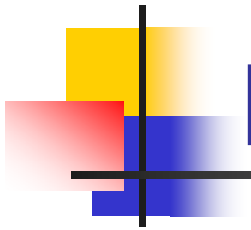
Adobe Flex 3

- C’est quoi ?
  - Adobe Flex 3 est un environnement de développement Open Source multi plateforme (Windows, Linux, Mac…) permettant la création d’Applications Internet Riches (RIA)
- Qu’est ce qu’une RIA ?
  - C’est un concept Macromédia qui désigne une application WEB exécutée localement sur votre PC alors qu’elle a l’habitude d’être exécutée du côté serveur
- Avec les RIA
  - Les calculs sont reportés au niveau client et il n’est plus nécessaire de passer par le serveur
    - ➔ d’où une économie du rafraîchissement du navigateur et des échanges de données (coûteux en temps)



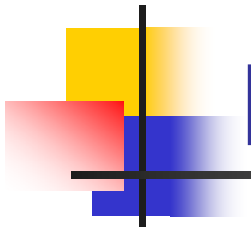
RIA

- Avantages des RIA / aux applis locales
  - Pas nécessaire de les installer, car elles sont directement utilisables via le navigateur Internet
    - Les distributions et les mises à jour sont faites automatiquement
  - Sont multi plateformes
    - En les développant en Flex, on n'a plus besoin de faire plusieurs versions
      - La même peut aller sur toute plateforme
      - Il suffit que l'ordinateur soit équipé d'un Flash Player (aujourd'hui, 78% des machines en sont équipées)
  - Plus besoin d'installer son logiciel sur le Web
    - En laissant l'appli en local, cela la protège des virus…
    - Le navigateur sert de SAS entre l'appli et l'extérieur



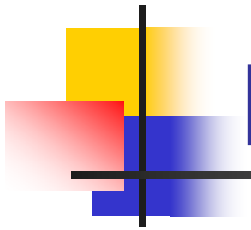
Flex

- Insistons : pourquoi Flex est encore multi plateformes ?
  - Flex donne la possibilité de créer des fichiers .SWF
    - Ce qui permet de pouvoir les jouer partout : il suffit d'avoir Adobe Flash Player
      - ➔ De manière générale, il faut le couple Adobe Flash Player et n'importe quel navigateur
  - Le rendu est identique sur la plupart des navigateurs et des systèmes d'exploitation (Windows, Linux, Mac)
  - Flex permet également de générer des fichiers AIR :
    - applications pouvant être jouées localement sans navigateur
      - ➔ c'est en quelque sorte une machine virtuelle comme Java



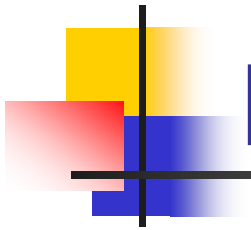
Rich Internet Application

- Intérêt de vous enseigner Flex ?
  - Apprendre plus sur la puissance des bibliothèques de composants et découvrir le moyen de les utiliser dans vos applis Web
  - Connaître des nouvelles ressources pour développer des interfaces sophistiquées
  - Flex permet de faire des IHM suivant une architecture MVC
- Flex propose
  - Une multitude de moyens pour vous connecter à vos sources de données dorsales
    - XML sur HTTP
    - Web services SOAP
    - Protocole d'accès aux objets distants (AMF : Action Message Format) robuste
- Flex propose également
  - Une intégration étroite avec LiveCycle Data Services
  - La prise en charge d'AIR (Adobe Integrated Runtime)



Rich Internet Application

- Récapitulatif
  - Les avantages pour les développeurs
    - Un environnement efficace pour l'exécution du code, du contenu et des communications
    - Un modèle objet puissant et facilement extensible pour l'interactivité
    - Une réutilisation de composants facilitée
    - La possibilité de réaliser des opérations connectées au Web et déconnectées du Web
    - Une distribution facilitée en supprimant les besoins d'installation et de développement spécifiques aux changements de plates-formes

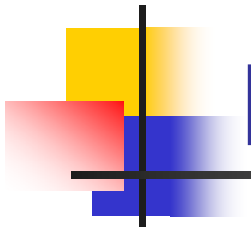


Rich Internet Application

- Récapitulatif (suite)
  - Les avantages pour les utilisateurs
    - L'intégration harmonieuse de contenus riches (vidéo, audio, graphes) de haute fidélité
    - Un meilleur temps de réponse de l'application :
      - transit des données sans besoin de rechargement (présentation d'une partie et le reste en tâche de fond)
    - Une meilleure interactivité avec temps court des réponses
    - Un sentiment du contrôle de la part de l'utilisateur avec les moyens mis à sa disposition pour la visualisation, la personnalisation, etc.

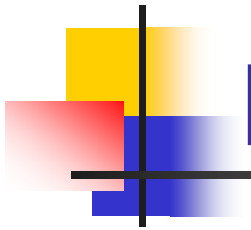
Flex

Prise en main



Prise en main

- L’environnement Flex comprend
  - Une interface graphique de construction de RIA
  - MXML, un langage proche de XML pour décrire des interfaces interactives
  - ActionScript 3.0 un langage de script objet
    - C’est un peu comme JavaScript pour HTML
  - CSS, un langage de style intégré

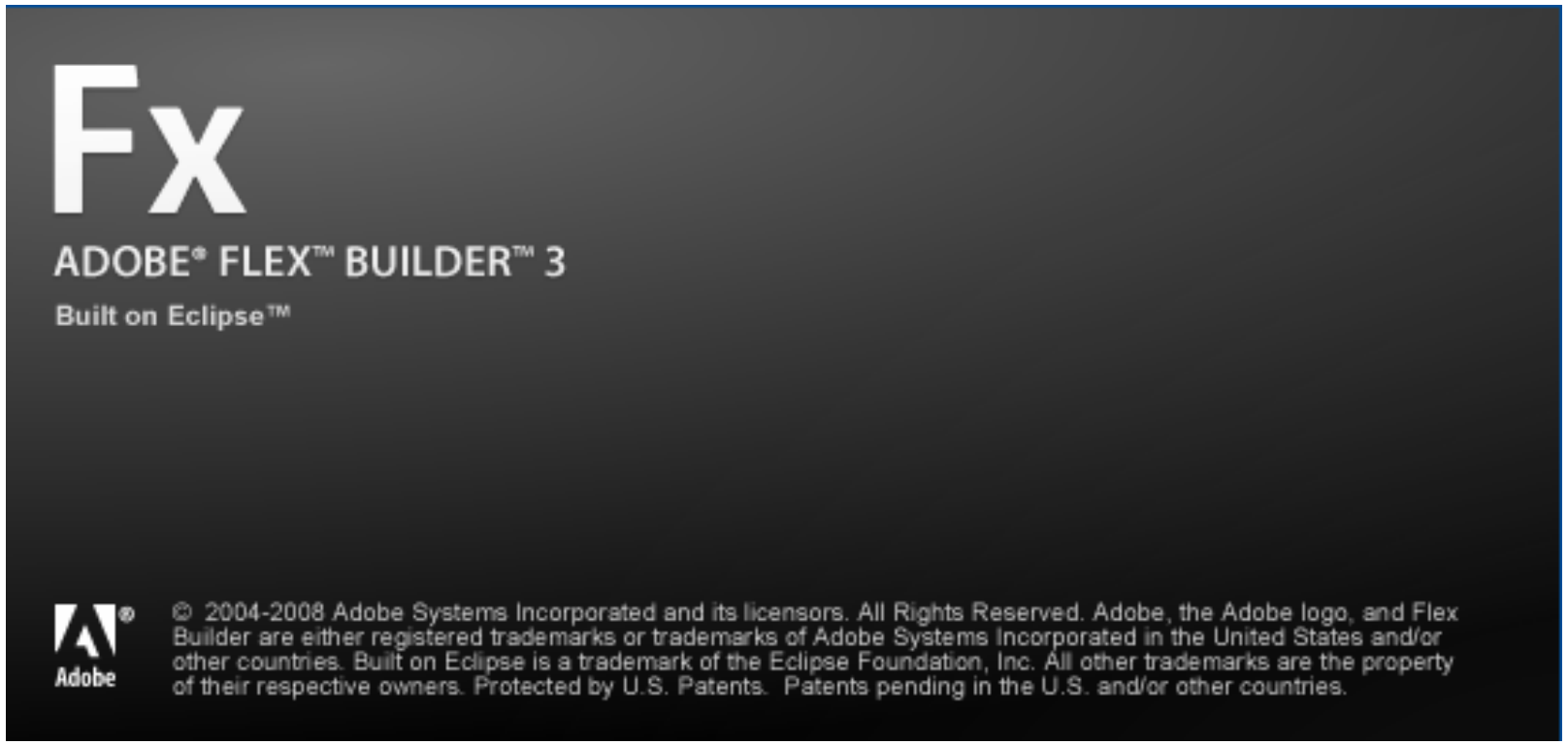


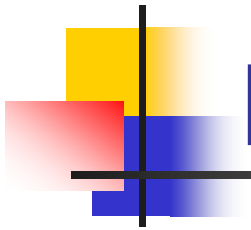
Prise en main

- Développement
  - Flex Builder 3 est l'interface de DEveloppement (IDE) pour Flex 3 (version 3) développée par Adobe
  - Le SDK (Software Development Kit) est disponible gratuitement sur le site d'Adobe
  - Adobe a fait son choix de se baser sur Eclipse pour développer son IDE
    - Flex Builder 3 est disponible sous deux formes différentes :
      - Un plug-in
      - Un package comprenant Eclipse et le plug-in intégré

Prise en main

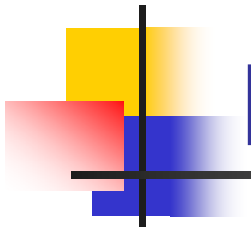
- Développement





Prise en main

- Installer Flex Builder 3
  - Etape 1 : téléchargement :
    - il faut télécharger la version de test (http://www.adobe.com/fr/products/flex/), et s'inscrire à cet endroit (https://freeriatools.adobe.com/flex/) afin d'obtenir les licences gratuites pour étudiants et enseignants. Le numéro de licence sera demandé lors de la première ouverture de Flex Builder 3 sur les PC
  - Etape 2 : dans section Produits, sélectionnez Flex
    - Vous arrivez sur la page dédiée à Flex
    - Cliquez sur Tester Flex Builder
    - Créez un compte …
  - Etape 3 : plusieurs versions sont proposées, récupérez la version standalone pour Mac ou Windows
  - Etape 4 : une fois l'installation terminée, lancez l'exécutable et acceptez la licence



Prise en main

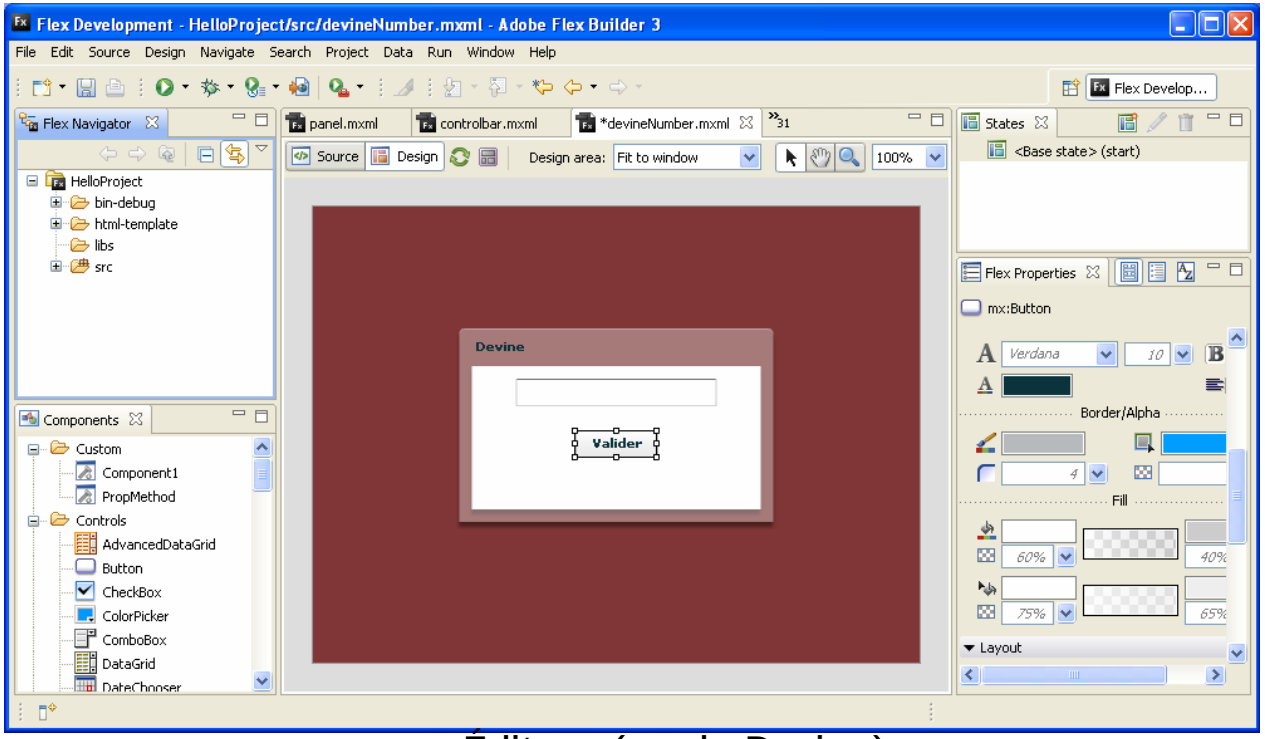
- Installer Flex Builder 3
  - Étape 5 : choisissez le répertoire d'installation:
    Typiquement : C:\Program Files\Flex Builder 3
    - L'écran suivant va vous demander d'installer Flash Player 9 pour les navigateurs installés
    - Cliquez sur Next
  - Étape 6 :
    - au lancement, on demande l'installation de Flash Player (9 ou 10) comme plug-in du navigateur par défaut
    - Choisir le navigateur par défaut et installer ce plug-in qu'on trouve sur Internet
      - Dans l'interface Flex : Cliquer sur Windows >> Préférences >> Web Browser

Prise en main

- Présentation de l'environnement de développement
  - Une fois l'IDE installé, on va essayer de comprendre comment fonctionne le Framework Flex
- Au chargement, on doit voir apparaître l'écran suivant :

Navigateur (Arborescence du projet)

Liste des Composants Flex



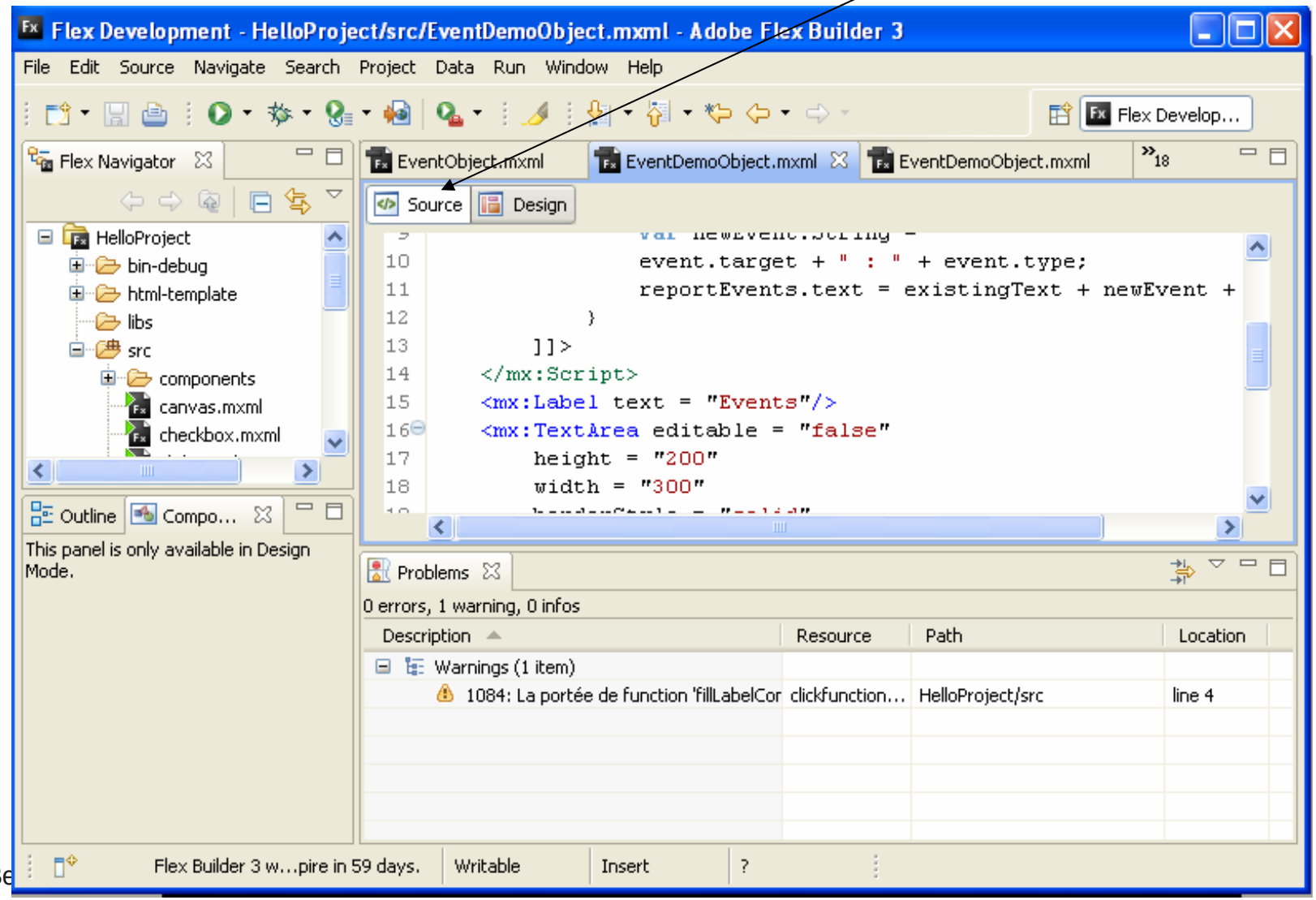
Vue des états

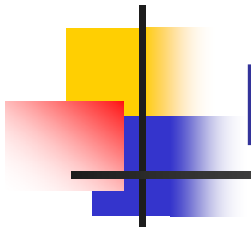
Propriétés du composant sélectionné

Éditeur (mode Design)

Prise en main

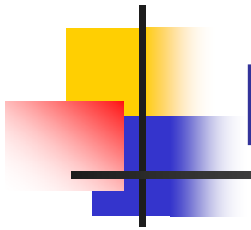
Plateforme Flex : source





Programmation sous Flex

- Explorer mon premier projet
  - Commencez par créer un nouveau projet flex (new>Flex project)
    - Nommez le par exemple "HelloProject"
  - Ouvrez un fichier .mxml
    - File >> New >> MXML Application
    - Une fenêtre s’ouvre avec un code minimal (voir plus loin)

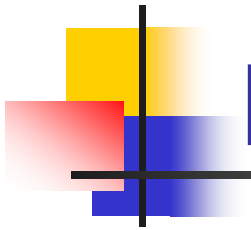


Programmation sous Flex

- En mode Source : premières lignes de votre page .mxml :

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute">
</mx:Application>
```

- Explications
  - La 1ère ligne déclare un document XML
  - mx:Application est la balise de conteneur par défaut, qui indique un attribut xmlns (nameSpace) qui a pour préfixe "mx"
  - layout définit comment sera affiché le conteneur par rapport aux autres
    - Absolute : affichage absolu en donnant X et Y,
    - "horizontal" et "vertical" sont utilisés pour un alignement horizontal (resp. vertical) avec horizontalAlign="center" (resp verticalAlign) par exemple



Programmation sous Flex

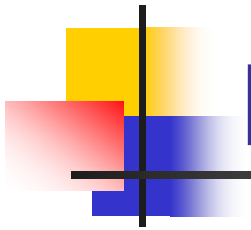
- Première application
  - Insérons un texte "HelloWorld" grâce à la balise :

    <mx:Label text="Hello"/>

    ou

    <mx:Label>
    <mx:text>Hello</mx:text>
    </mx:Label>
  - Les commentaires en MXML sont de la forme :

    <!-- Mon commentaire -->
  - Essayez d'exécuter cette application : Run HelloProject
    - Un navigateur s'ouvre avec le texte "Hello"
  - On peut préciser ce navigateur : window >> Preferences >> General >>Network connection >> Web Browser
    - Exemple : HelloWorld.mxml

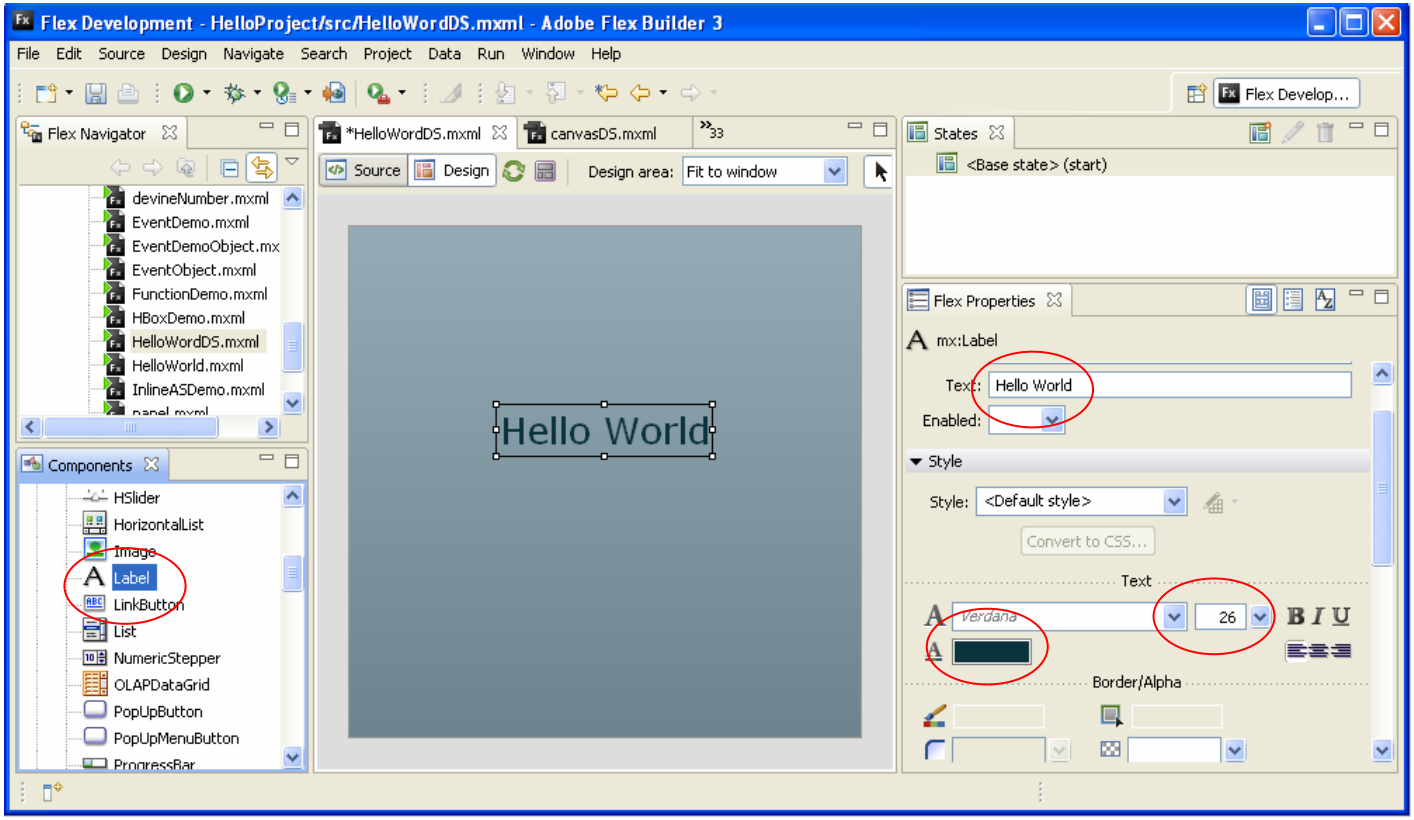


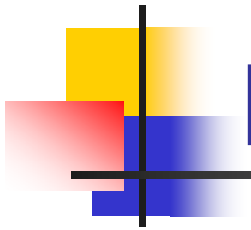
Programmation sous Flex

- En mode Design : HelloWorldDS.mxml
  - Créer une application MXML
  - Placez vous en mode Design
  - Allez dans fenêtre Components (controls) :
    - Sélectionnez “Label” dans la partie Outline
    - Glissez-Déposez le sur l’espace de travail
    - Allez dans la fenêtre Flex Properties
    - Editez le texte dans la zone après Text:
    - Choisissez le style…

Programmation sous Flex

- Interface :





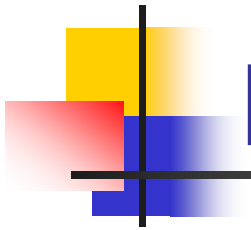
Programmation sous Flex

- Compiler une application Flex
  - En appuyant sur la flèche verte, Flex Builder 3 se charge de compiler votre application et de la lancer dans le navigateur par défaut
  - Les étapes
    - Etape 1 :
      - Transforme les balises MXML en classe ActionScript 3
    - Etape 2 :
      - Le compilateur crée du code instanciant cette classe qui sera finalement compilée en fichier SWF (format standard du Flash Player)
  - Le SDK permet de faire ces opérations de manière transparente, mais sinon, on peut utiliser des commandes en ligne et lancer le compilateur de MXML : Mxmlc
  - 3 fichiers sont générés dans le répertoire/bin de votre projet :
    - Une version standard pour la mise en production
    - Une version Debug
    - Une version Profile

Création de la Vue

ou

les composants de Flex

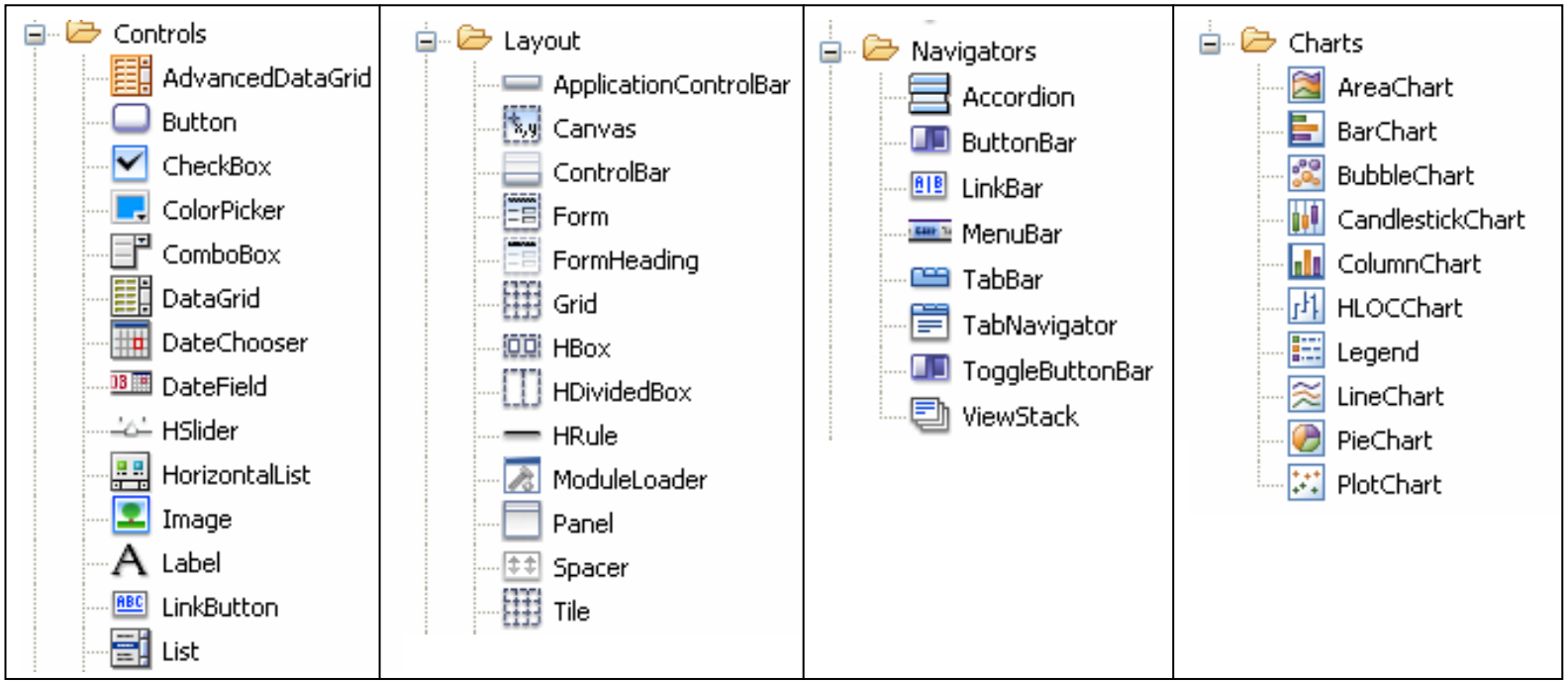


Les composants de Flex

- Flex propose 3 types de composants
  1. Les composants de contrôle : Controls
     - typiques de l'interface, deux types :
       - les contrôles basiques : labels, boutons, checkbox, …
       - les data provider : datagrid pour structurer les données reçues
  2. Les composants de mise en page : Layout et Navigator
     - de type conteneur, servant à accueillir des contrôles :
       - les agencer de diverses manières (Layout)
       - ou gérer leur affichage (Navigator)
  3. Les composants de type Chart
     - permettent d'afficher des graphes (si le plug-in est là)

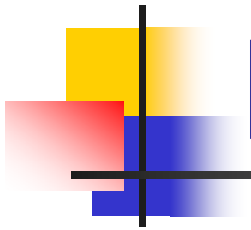
Les composants de Flex

- Ces composants sont
  - visibles en mode design
  - représentées par des balises spécifiques en MXML
  - correspondent à des classes ActionScript



Les composants de mise en page

Layout ou Containers



Les containers

- Layout ou contrôleurs d'agencement
  - Représentent des surfaces rectangulaires pouvant accueillir des composants ou d'autres conteneurs
  - Permettent de structurer hiérarchiquement l'application (mise en page)
- Hbox, Vbox
  - Permettent d'afficher des éléments horizontalement ou verticalement au sein du conteneur parent
  - Exemple
    - publier des photos de vacances en miniature
  - Ces conteneurs vont pouvoir organiser la façon dont ces images vont s'afficher les unes par rapport aux autres

Les containers

Layout ou contrôleurs d'agencement

- *Exemple : HBoxDemo.mxml*
  - *Les controls sont ici des TextInput*

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:HBox>
    <mx:TextInput width="100"/>
    <mx:TextInput width="100"/>
    <mx:TextInput width="100"/>
  </mx:HBox>
</mx:Application>
```

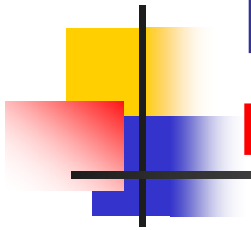
- *Exemple : VBoxDemo.mxml*
  - *Les containers sont : Panel et VBox*
  - *Ils contiennent des boutons et une ComboBox*

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
   <mx:Panel title="VBox Container Example" height="75%" width="75%"
             paddingTop="10" paddingLeft="10" paddingRight="10"
             paddingBottom="10">

             <mx:Label width="100%" color="blue" text="A VBox container
             with vertically aligned children."/>

             <mx:VBox borderStyle="solid" paddingTop="10"
             paddingBottom="10"  paddingLeft="10" paddingRight="10">
                      <mx:Button label="Button 1"/>
                      <mx:Button label="Button 2"/>
                      <mx:Button label="Button 3"/>
                      <mx:ComboBox/>
             </mx:VBox>

   </mx:Panel>
</mx:Application>
```

Les containers

Layout ou contrôleurs d'agencement

- Le refaire en mode Design
  1. Sélectionner le composant HBox, le faire glisser-déposer sur l'éditeur, préciser 50% pour la taille, un rectangle noir apparaît
  2. Sélectionner VBox et poser-le dans la HBox, préciser une largeur de 50% et laisser la valeur par défaut de la hauteur. Répéter l'opération, on remarque que la seconde VBox se place à droite de la première. C'est le résultat attendu, vu qu'un HBox place ses contrôles enfants horizontalement
  3. Sélectionner ensuite des composants et placer les dans les VBox créées : Panel avec un titre dans la VBox de droite, et plusieurs boutons dans la VBox de gauche. On peut remarquer que les éléments sont placés verticalement



Les containers

Layout ou contrôleurs d'agencement

- Retour sur le Panel
  - inclut
    - Une barre de titre, un message de status
    - Une bordure et une zone de contenu pour y insérer les autres composants
  - Permet de disposer les éléments de 3 manières
    - Vertical (par défaut), horizontal et absolute

Les containers

Layout ou contrôleurs d'agencement

- Exemple : panel.mxml

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
   xmlns="*">
    <mx:Panel title="Employee List" borderColor="#00FF00"
            backgroundColor="#FF0000">
            <mx:Label text="Employee 1"/>
            <mx:Label text="Employee 2"/>
            <mx:Label text="Employee 3"/>
            <mx:Button label="Submit"/>
    </mx:Panel>
</mx:Application>
```

Les containers

Layout ou contrôleurs d'agencement

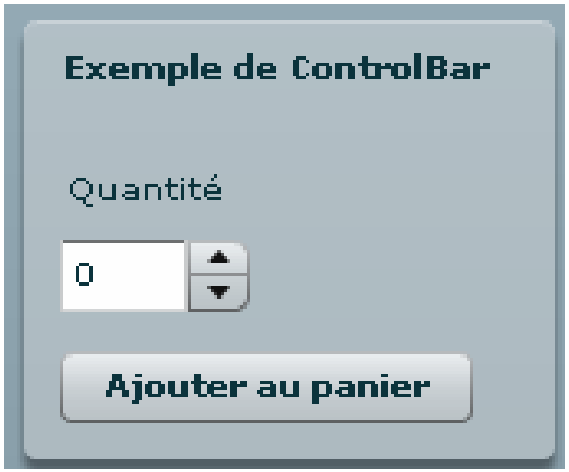
- Panel : mode Design : panelDS.mxml
  - Aller dans Layout
  - Créer un panel
  - Aller dans Propriétés
    - Lui donner un titre
    - Régler sa couleur, taille…
    - On peut également le positionner : en mettant 0, 0 on le centre dans l'écran…
  - Mettre un accordéon (par ex.)
  - On peut en mettre un deuxième (en cliquant sur le bouton +)
    - Les objets que l'on pourra créer dans cet accordéon seront dévoilés en cliquant sur le titre
    - Ici : deux labels

Les containers

Layout ou contrôleurs d'agencement

- ControlBar : controlbar.mxml
  - S'utilise pour arrimer une barre d'outils à un conteneur Panel ou un TitleWindow
  - Il peut agir comme un conteneur HBox ou VBox , selon l'attribut direction (par défaut Horizontal)
  - Exemple :

```
<mx:ControlBar direction="vertical">
        <mx:Label text="In the Control Bar"/>
        <mx:HBox>
                <mx:Label text="Quantity" />
                <mx:NumericStepper id="myNS"/>
                <mx:Button label="Add to Cart"/>
        </mx:HBox>
</mx:ControlBar>
```

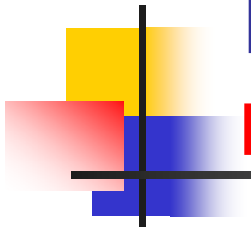


Les containers

Layout ou contrôleurs d'agencement

- La barre de contrôle : applicationControlBar.mxml
  - Permet de contenir des composants fournissant des commandes de navigation et d'application, comme des menus
    - Sous-classe de ControlBar
  - Contient généralement :
    - Un contrôle MenuBar qui reçoit des textes structurés (dataProvider) en menus
  - Exemple :

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
   layout="absolute">
   <mx:ApplicationControlBar x="79" y="47" width="262">
            <mx:MenuBar dataProvider="menu1"></mx:MenuBar>
            <mx:MenuBar dataProvider="menu2"></mx:MenuBar>
   </mx:ApplicationControlBar>
</mx:Application>
```



Les containers

Layout ou contrôleurs d'agencement

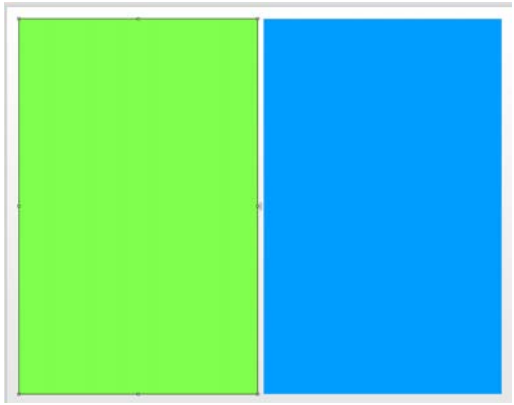
- HDivideBox, VDivideBox
  - Disposent leurs enfants de la même manière que dans une VBox ou HBox
  - Placent automatiquement un séparateur entre chaque composant enfant
    - Ce séparateur permet de redimensionner la surface des documents allouée à chaque enfant grâce à la souris, pendant l'exécution
  - L'attribut « direct » peut avoir la valeur « horizontal » ou « vertical »

Les containers

Layout ou contrôleurs d'agencement

- Exemple : HdivideBox.mxml
  - Ici, les fils sont des VBox

```
<mx:HDividedBox id="hdivbox" width="100%" height="100%">
   <!-- children -->
   <mx:VBox backgroundColor="haloGreen" width="100%"
            height="100%">
   </mx:VBox>
   <!-- children -->
   <mx:VBox backgroundColor="haloBlue" width="100%" height="100%">
   </mx:VBox>
</mx:HDividedBox>
```



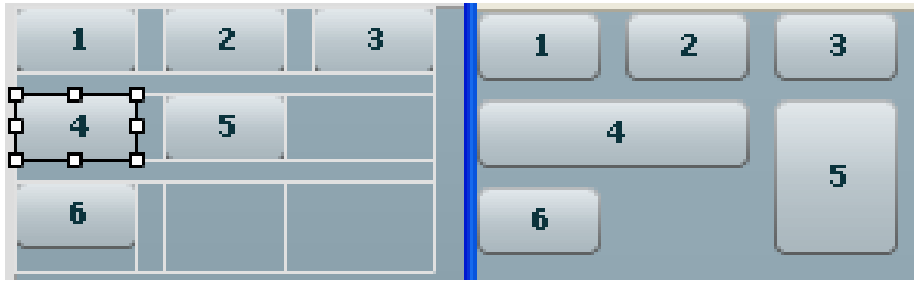
Les containers

Layout ou contrôleurs d'agencement

- Grid
  - Permet d'agencer les composants dans un tableau dont le nombre de cellules est variable, à la manière d'un tableau HTML
  - Création en mode source :
    1. Définir la balise <mx:Grid>
    2. Ajouter des lignes <mx:GridRow>
    3. Ajouter des cellules <mx:GridItem> :
       - on n'est pas obligé d'avoir le même nombre de cellules
       - on peut mettre des conteneurs dans chaque cellule

Exemple : grid.mxml

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute">
<mx:Grid>
    <mx:GridRow>
     <mx:GridItem>
                <mx:Button label="1"/>
     </mx:GridItem>
     <mx:GridItem>
                …
    <mx:GridRow>
     <!-- Fusion des colonnes 1 et 2 -->
     <mx:GridItem colSpan="2">
                <mx:Button label="4" width="100%"/>
     </mx:GridItem>
    …
```



Les containers

Layout ou contrôleurs d'agencement

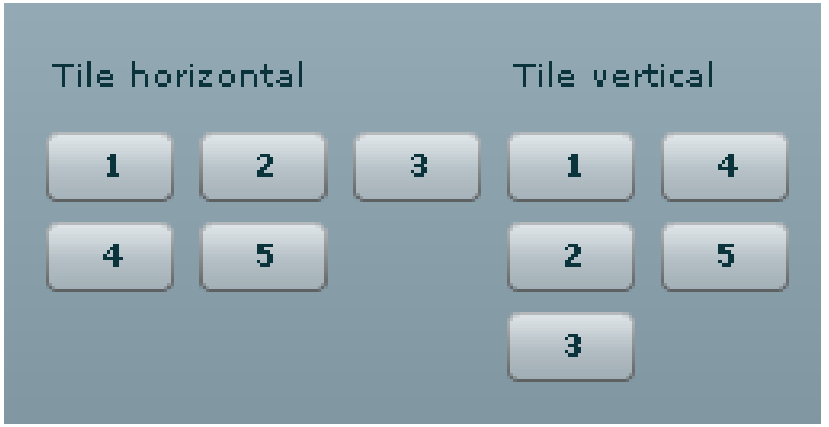
- Tile : tile.mxml
  - Une autre sorte de grille qui agence ses enfants sur plusieurs lignes et colonnes automatiquement
  - La propriété *direction* détermine l'agencement horizontal (par défaut) ou vertical des contrôles enfants
  - Concrètement, si on définit un Tile de 16 enfants, il fera un rangement de 4 x 4, si on définit 13 enfants, il fera toujours un rangement de 4x4 mais les 3 derniers seront absents

Les containers

Layout ou contrôleurs d'agencement

- Tile : tile.mxml

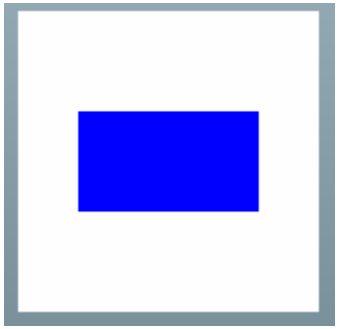
```
<mx:Tile x="39" y="48" direction="horizontal">
    <mx:Button x="39" y="78" label="1"/>
    <mx:Button x="39" y="108" label="2"/>
    <mx:Button x="112" y="48" label="3"/>
    <mx:Button x="112" y="78" label="4"/>
    <mx:Button x="112" y="108" label="5"/>
</mx:Tile>
<mx:Label x="39" y="22" text="Tile horizontal"/>
<mx:Tile x="183" y="48" direction="vertical" width="93" height="84">
    <mx:Button label="1"/>
    <mx:Button label="2"/>
    <mx:Button label="3"/>
    <mx:Button label="4"/>
    <mx:Button label="5"/>
</mx:Tile>
```



Les containers

Layout ou contrôleurs d'agencement

- Canvas
  - Définit une surface rectangulaire dans laquelle on peut placer librement les conteneurs
  - Contrairement aux autres conteneurs, Canvas n'agence pas les enfants
  - On doit préciser la position des enfants en utilisant une position absolue ou basée sur des contraintes
  - Pour utiliser la position absolue, utiliser les propriétés x et y
  - Pour utiliser les contraintes, spécifier la taille des ancres et leur côté

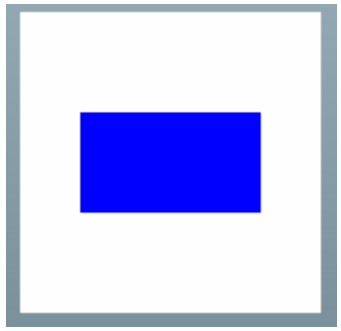


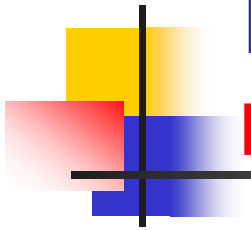
Les containers

Layout ou contrôleurs d'agencement

- Canvas : canvas.mxml

```
<mx:Canvas width="150" height="150" backgroundColor= "#FFFFFF">
    <mx:Box left="30" right="30" y="50" height="50"
            backgroundColor="blue">
    </mx:Box>
</mx:Canvas>
```





Les containers

Layout ou contrôleurs d'agencement

- Redimensionnement d'un composant
  - Plusieurs possibilités offertes
    - Taille par défaut
      - C'est ce qui se produit quand on ne précise rien
    - Taille absolue
      - On précise explicitement la taille
    - Taille relative
      - On précise la taille d'un enfant par rapport à la taille de son conteneur en %
    - Taille par contrainte
      - On précise la taille d'un enfant par rapport à la taille de son conteneur

Les containers

Layout ou contrôleurs d'agencement

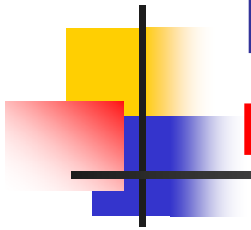
- Exemple : redimension.mxml

```
<mx:Panel  layout="vertical" title="Taille par défaut" horizontalAlign=
    "center">
    <mx:DateChooser showToday="true"/>
    <mx:CheckBox label="activer paramètre 1"/>
    <mx:CheckBox label="activer paramètre 2"/>
    <mx:Button  label="Appliquer"/>
</mx:Panel>
<mx:Panel width="100%" height="100%" layout="absolute" title="Taille
    relative">
    <mx:Button label="Button"/>
</mx:Panel>
<mx:Panel width="250" height="200" layout="horizontal" title="Taille
    absolue">
    <mx:Button label="Button"/>
</mx:Panel>
```

Les containers

Layout ou contrôleurs d'agencement

- Exemple : redimension.mxml



Les containers

Layout ou contrôleurs d'agencement

- Positionnement absolu et contraintes
  - Lorsque le conteneur a la propriété layout à vertical ou horizontal, il ne gère pas la position des contrôles enfants
    - Flex les aligne verticalement ou horizontalement
  - Cependant, on peut spécifier la valeur absolue à cette propriété, dès lors, on peut positionner les contrôles comme bon nous semble
  - Le positionnement absolu ne concerne que trois types de conteneurs :
    - Application, Canvas et Panel

Les containers

Layout ou contrôleurs d'agencement

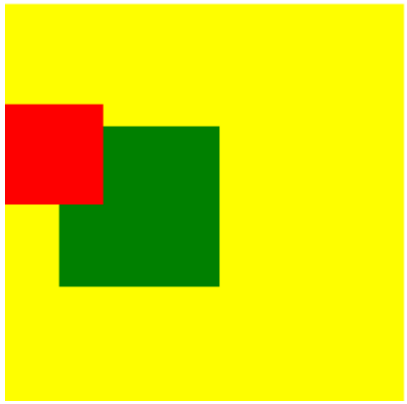
- Exemple : positionnement.mxml

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    layout="absolute" backgroundGradientColors="[#FFFFFF, #FFFFFF]">
    <mx:Canvas x="41" y="44" width="200" height="200"
        backgroundColor="yellow">
    <mx:VBox x="28" y="61" width="80" height="80"
        backgroundColor="green">
    </mx:VBox>
    <mx:VBox width="50" height="50" x="0" y="50"
        backgroundColor="red">
    </mx:VBox>
</mx:Canvas>
```

Les containers

Layout ou contrôleurs d'agencement

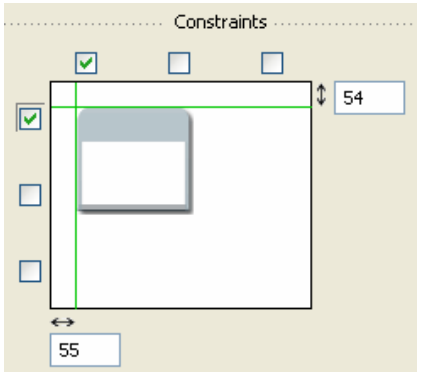
- Exemple : positionnement.mxml



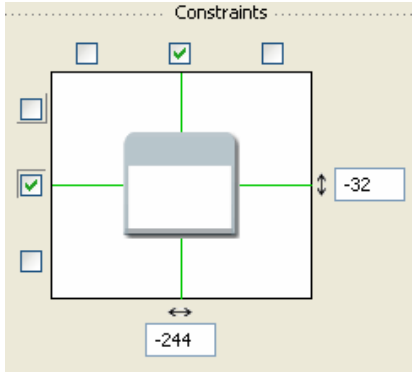
Les containers

Layout ou contrôleurs d'agencement

- Utiliser les contraintes
  - On peut utiliser le système de contraintes afin de gérer la position et la taille des composants dans un conteneur supportant la mise en forme absolue
  - Les contraintes sont des ancres qui permettent de maintenir la disposition des contrôles dans le conteneur lorsque la taille de celui-ci est variable
  - Ces contraintes permettent de donner une distance minimale entre le contrôle et l'un des bords du conteneur ou de l'axe de symétrie



Contraindre la position



Contraindre la symétrie

Les containers

Layout ou contrôleurs d'agencement

- Utiliser des contraintes améliorées
  - dans les conteneurs autorisant un positionnement absolu (Canvas, Panel, Module et Application)
  - Fonctionnement
    - Les contraintes améliorées divisent l'espace du conteneur en lignes et colonnes qui peuvent être ensuite utilisées pour positionner d'autres composants par rapport aux limites de ces lignes, colonnes
    - Les zones horizontales se définissent avec
      - la classe ConstraintRow
    - et les zones verticales,
      - avec la classe ConstraintColumn

Les containers

Layout ou contrôleurs d'agencement

- Exemple : constraints_C_L.mxml

```
<mx:constraintColumns>
    <mx:ConstraintColumn id="col1" width="33%"/>
    <mx:ConstraintColumn id="col2" width="33%"/>
    <mx:ConstraintColumn id="col3" width="33%"/>
</mx:constraintColumns>
<mx:constraintRows>
    <mx:ConstraintRow id="row1" height="50%"/>
    <mx:ConstraintRow id="row2" height="50%"/>
</mx:constraintRows>
<mx:TextArea id="ta1" text="TextArea 1" left="col1:5" right="col1:5" top="row1:10"
bottom="row1:20" backgroundColor="yellow" fontSize="20"/>
<mx:TextArea id="ta2" text="TextArea 2" left="col1:5" right="col2:5" top="row2:10"
bottom="row2:10" backgroundColor="white" fontSize="20"/>
<mx:TextArea id="ta3" text="TextArea 3" left="col3:5" right="col3:5" top="row1:10"
bottom="row2:10" backgroundColor="green" fontSize="20"/>
```

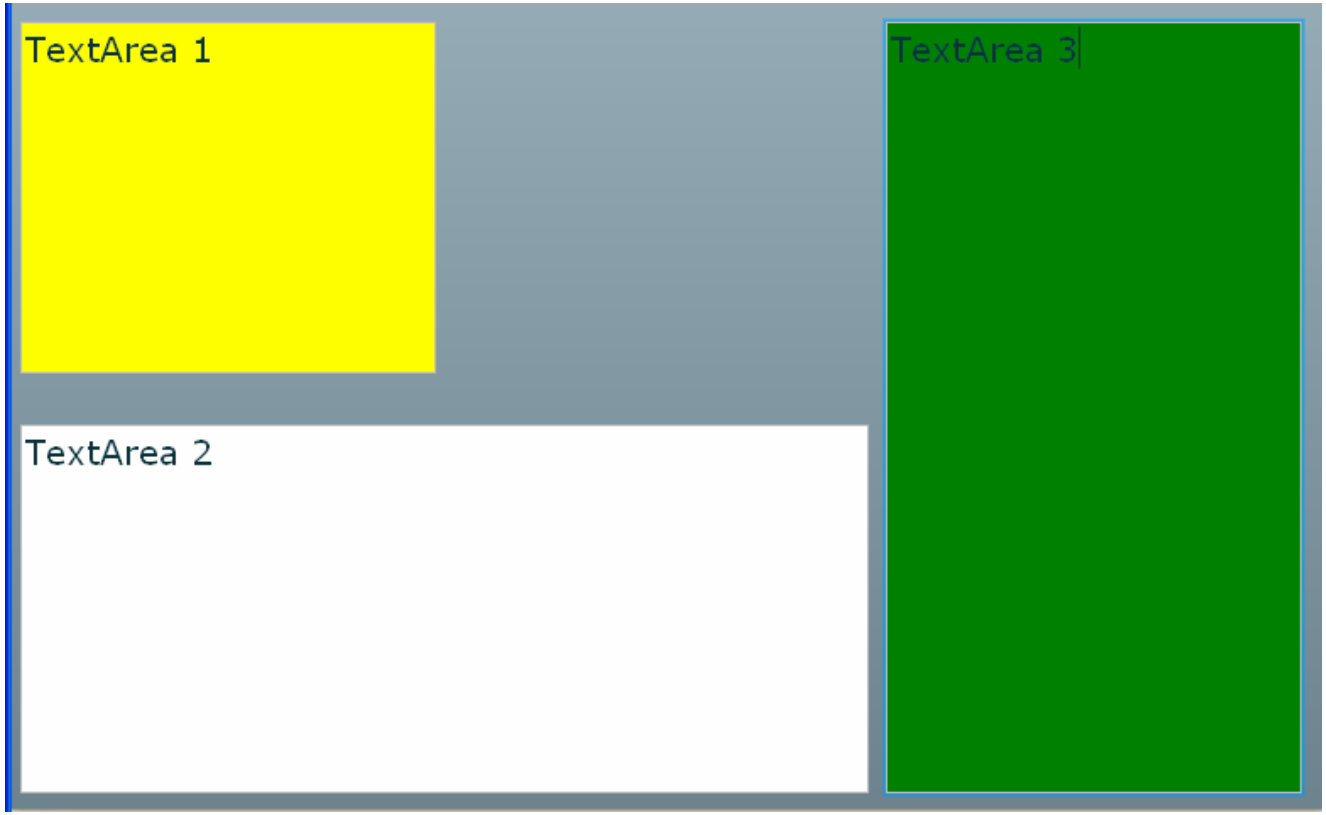
On divise l'espace en 3 colonnes

On divise l'espace en 2 lignes

Les containers

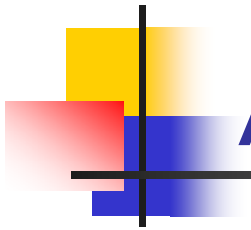
Layout ou contrôleurs d'agencement

- Exemple : constraints_C_L.mxml



Exercice

- Énoncé : Cours1-Exo/annuaire.mxml
  - Réaliser une interface d'annuaire pour gérer les contacts
  - Pour l'instant avec des boutons non interactifs
  - Réfléchir uniquement sur les conteneurs

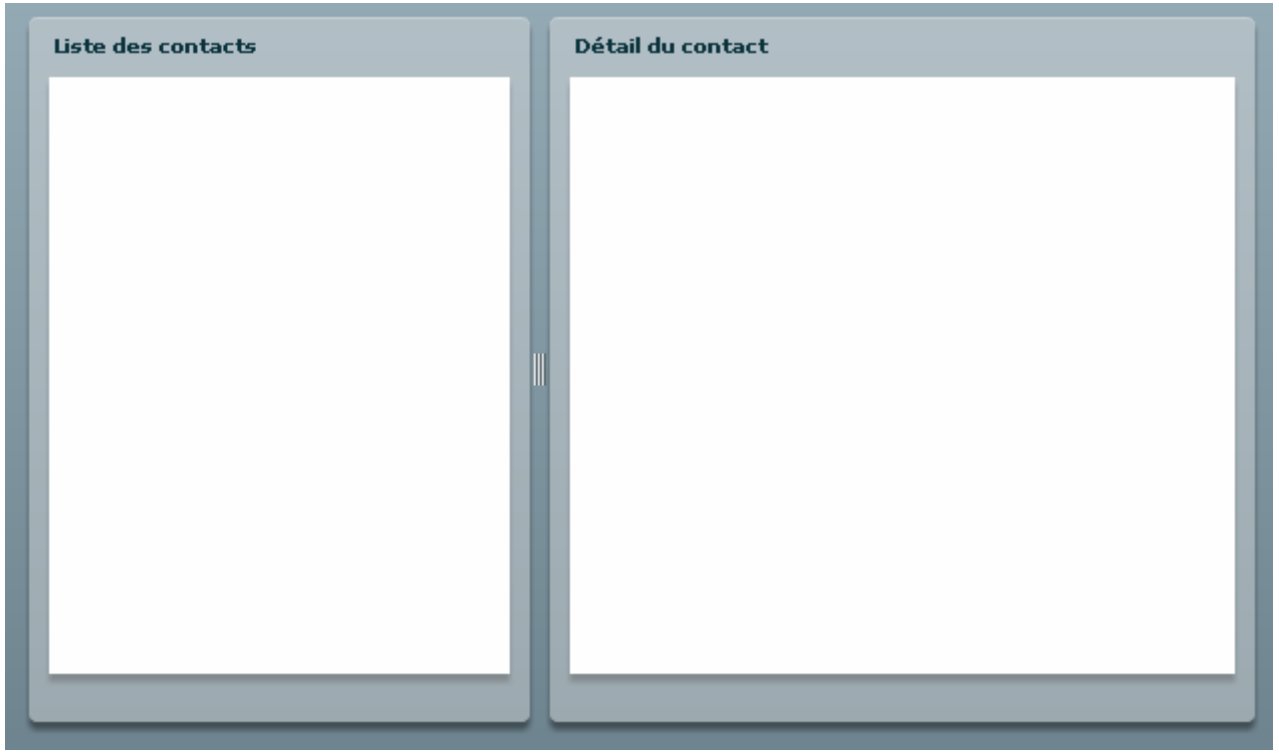


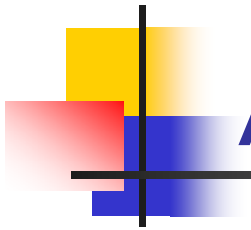
Application

- Les étapes
  1. Mettre en place les conteneurs
     - Créer un nouveau projet à nommer : annuaire (utiliser l'emplacement par défaut)
     - Créer une application mxml appelée : annuaire.mxml
     - Redimensionner le conteneur Application à 650x400
     - Définir le layout en vertical au lieu d'absolute
     - Déposer une HDivideBox avec height et width à 100%
     - Déposer 2 Panels dans la HDivideBox avec une largeur par défaut et une hauteur à 100%
     - Ajouter un titre à chacun
     - Ajouter dans chaque Panel un ControlBar

Application

1. Mettre en place les conteneurs



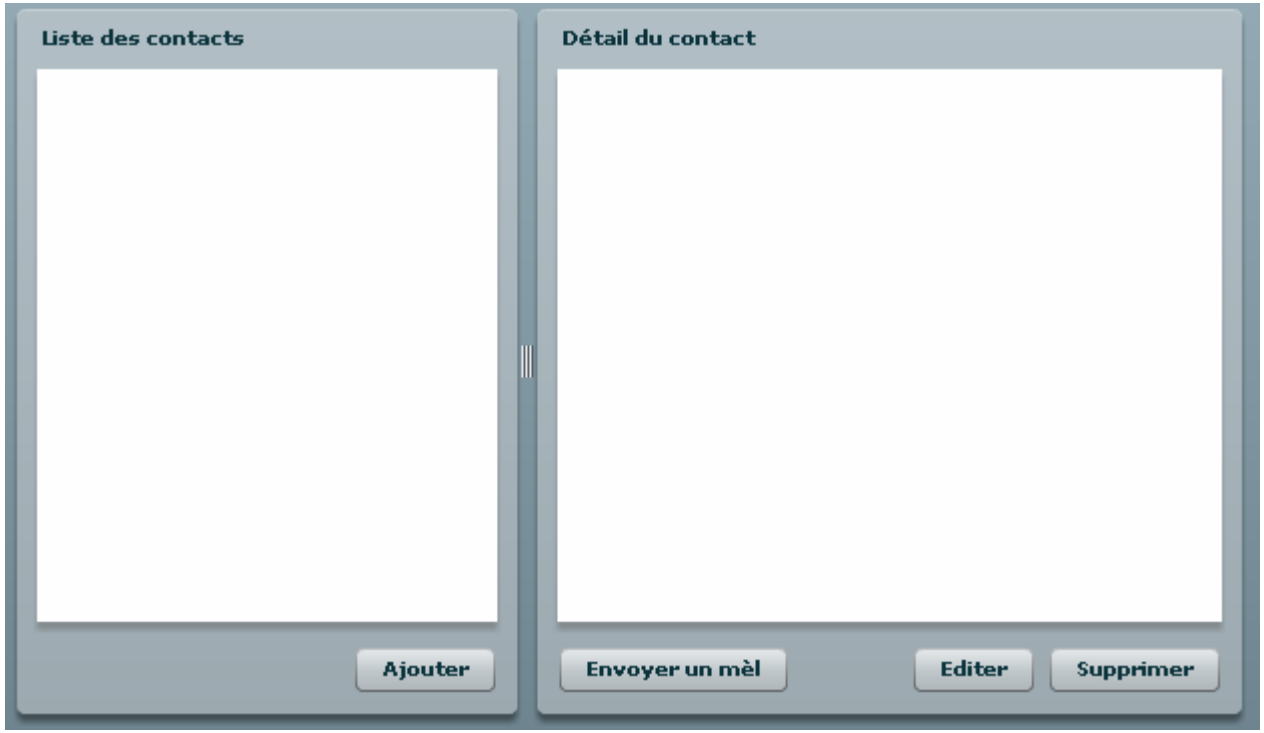


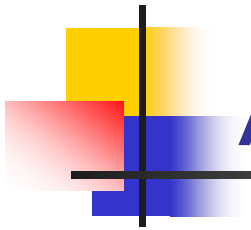
Application

2. Ajouter les contrôles simples en mode Design
   - Déposer un Button dans le premier ControlBar et 3 Button dans le second
   - Les placer correctement
     - Sélectionner le ControlBar de gauche et dans le panel Common, attribuer la valeur right à la propriété Horizontal align
     - Dans le ControlBar de droite, placer le conteneur Spacer entre les deux premiers boutons et attribuer lui une largeur de 100%

Application

2. Ajouter les contrôles simples en mode Design

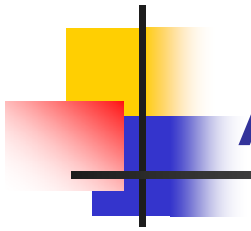




Application

2. Ajouter les contrôles simples en mode Design
   - Compléter le Panel de gauche
     - Définir le Layout du Panel en vertical
     - Placer une HBox de larg 100%
     - Ajouter en dessous une DataGrid qui prend l'espace disponible
     - Passer en mode source et modifier le code pour supprimer la colonne supplémentaire et renommer les titres des colonnes
     - Le code doit ressembler à ceci

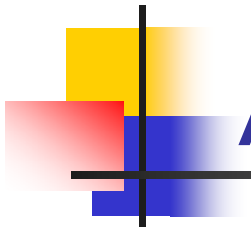
```
<mx:DataGrid width="100%" height="100%">
<mx:columns>
<mx:DataGridColumn headerText="Prénom" dataField="firstName"/>
<mx:DataGridColumn headerText="Nom" dataField="lastName"/>
</mx:columns>
</mx:DataGrid>
```



Application

2. Ajouter les contrôles simples en mode Design
   - Compléter le Panel de gauche
     - Ajouter des valeurs par le code suivant dans le tag de la DataGrid

```
<mx:dataProvider>
<mx:ArrayCollection>
<mx:Object firstName="Blandine"
    lastName="Dupont"/>
<mx:Object firstName="Jean" lastName="Frochen"/>
</mx:ArrayCollection>
</mx:dataProvider>
```



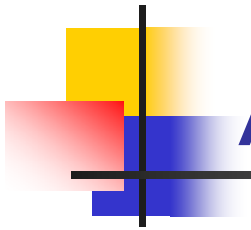
Application

2. Ajouter les contrôles simples en mode Design
   - En mode source toujours, ajouter une HBox dans le Panel de gauche avec les contrôles TextInput et Image

```
<mx:HBox x="0" y="0" width="100%" height="100%">
<mx:TextInput width="100%" height="100%"/>
<mx:Image source="http://www.philatelie.free.fr/img/search.gif"
    width="25" height="25" />
</mx:HBox>
```

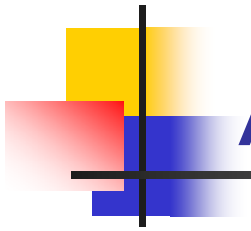
   - Ou télécharger l'image dans un répertoire à créer : images, puis écrire

```
<mx:Image width="25" height="25"
    source="@Embed(source='images/search.gif')"/>
```



Application

2. Ajouter les contrôles simples en mode Design
   - Rajouter une marge car la photo est collée au Panel:
     - Sélectionner pour Panel la vue Alphabetical View de Flex Properties et rechercher paddingBottom et paddingLeft, préciser une valeur de 5 pixels aux quatre marges



Application

2. Ajouter les contrôles simples en mode Design
   - Compléter le Panel de droite
     - Sélectionner les contrôles Label et positionner les dans le Panel en mode Design
     - Le layout absolute permet de placer vos contrôles comme bon vous semble
     - Rajouter l'image du contact dans le dossier images
     - Rajouter un contrôle Image et spécifier le chemin par Embed
     - Rajouter des marges…
     - Enfin, rajouter au-dessus de la HDividedBox un Label en gras et avec une taille de 18 pixels afin d'intituler cette première interface : mon Annuaire !