

Laborator 4

Amplasarea componentelor folosind entitati de tip LayoutManager

Notiunea de LayoutManager

Ce este un LayoutManager?

Un layout manager este o entitate utilizata pentru incapsularea unui algoritm de positionare si dimensionare a componentelor grafice dintr-o aplicatie GUI. Astfel are loc o separare intre componentele dezvoltate pentru a implementa logica GUI si codul dezvoltat pentru administrarea grafica a aplicatiei GUI. Astfel un algoritm de dimensionare si amplasare a componentelor grafice, inclus intr-o componenta LayoutManager, poate fi testat independent de aplicatie si poate fi reutilizate pentru oricate GUI care necesita comportamentul dictat conform cu algoritmul implementat.

Din perspectiva Java un LayoutManager este o interfata care descrie modul in care comunica un Container si administratorul de amplasare a componentelor – layout manager. Acest protocol implica metode pentru:

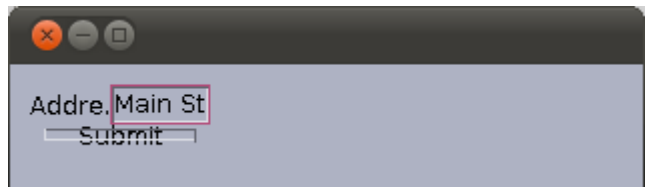
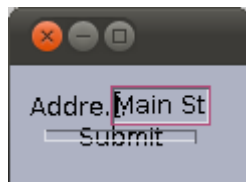
- achizitionarea informatiilor de dimensionare necesare pentru layout manager si componentele administrate;
- anuntarea unui layout manager despre adaugarea/eliminarea componentelor grafice in container;
- dimensionarea si pozitionarea componentelor administrate.

Utilitatea unui LayoutManager

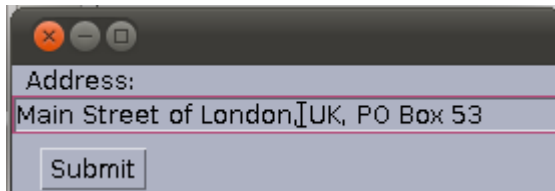
Administrarea pozitionarii, dimensionarii si redimensionarii componentelor grafice implica cateva probleme comune tuturor aplicatiilor grafice. Un layout manager rezolva aceste probleme comune precum si o serie de probleme specifice aplicatiei dezvoltate.

Probleme comune:

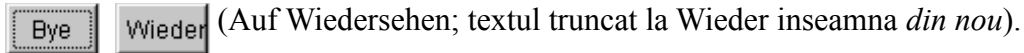
- redimensionare realizata de catre utilizator;
- utilizarea unui anumit look-and-feel;
 - prin utilizarea unui null layout manager se observa ca la redimensionarea realizata de utilizator campul unde a fost introdusa adresa nu este actualizat astfel ca nu pot fi vizualizate usor datele acelei componente; de asemenea fereastra foloseste look-and-feel Motif ceea ce face ca textul afisat pe comonenta JButton sa apara gresit datorita dimensiunilor asociate butonului;



- prin utilizarea unui layout manager (e.g. FlowLayout) componentele raspund bine atat la redimensionarile utilizatorilor cat si la diversele look-and-feel existente pe sistemele unde v-a rula aplicatia:



- la utilizarea unei limbi specifice unei anumite zone (e.g. o aplicatie care isi schimba textul in functie de optiunile utilizatorului) pot aparea probleme de trunchiere:



Exercitiu: Implementati urmatoarul cod pentru a testa problemele de mai sus;

```
package demo.layout.manager.sins;
import java.awt.Dimension;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JFrame;

import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;

public class LayoutManagerDemoJPanel extends JPanel {

    public LayoutManagerDemoJPanel() {
        super();
    }

    public void addWidgets() {
        /* create some widgets */
        // add a label
        JLabel label = new JLabel("Address: ");
        this.add(label);

        // add input data text field
        JTextField textField = new JTextField();
        this.add(textField);
        textField.setPreferredSize(new Dimension(60,20));

        // add a button for submission
        JButton submitButton = new JButton("Submit");
        this.add(submitButton);

        if(this.getLayout() == null) {
            System.out.println("Layout is null!");
            label.setSize(50,20);
            label.setLocation(10,10);

            textField.setSize(new Dimension(50,20));
            textField.setLocation(50,10);
        }
    }
}
```

```

        submitButton.setSize(new Dimension(90,10));
        submitButton.setLocation(10,30);
    } else {
        System.out.println("Layout is: " + this.getLayout());
    }
}

public static void main(String[] args) throws ClassNotFoundException,
InstantiationException, IllegalAccessException, UnsupportedLookAndFeelException {

    UIManager.setLookAndFeel("com.sun.java.swing.plaf.motif.MotifLookAndFeel");

    SwingUtilities.invokeLater(new Runnable() {

        @Override
        public void run() {
            // TODO Auto-generated method stub

            JFrame frame = new JFrame();
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            LayoutManagerDemoJPanel panel = new
LayoutManagerDemoJPanel();
            frame.setContentPane(panel);

            // set LayoutManager: default is BorderLayout
            // comment/uncomment the following line and test the
results

            panel.setLayout(null);
            //panel.setLayout(new BorderLayout(panel, BorderLayout.Y_AXIS));
            frame.setSize(140,100);
            frame.setLocation(500,500);

            panel.addWidgets();
            frame.setVisible(true);
        }
    });
}
}

```

Componentele de tip `LayoutManager` in `Swing`

Componente `LayoutManager` existente in `Swing`

`Swing` API ofera urmatoarele clase ca si layout manager:

- `BorderLayout`
- `BoxLayout`
- `CardLayout`
- `FlowLayout`
- `GridBagLayout`
- `GridLayout`
- `GroupLayout`
- `SpringLayout`.

O descriere vizuala a acestora poate fi gasita la [aceasta adresa](#). Pentru a observa diferitele caracteristici de utilizare ale unui layout manager se vor consulta sectiunile *how-to* ale [acestei documentatii](#).

Utilizarea componentelor `LayoutManager`

Un layout manager este utilizat in contextul unui container de componente grafice Swing de tip:

- `JPanel`
- `Content Pane`

Un container `JPanel` este initializat sa utilizeze implicit un layout manager de tip `FlowLayout` in timp ce un content pane utilizeaza `BorderLayout` ca si layout manager implicit.

Specificarea explicita a componentei layout manager folosite intr-un container

Pentru a specifica explicit tipul de layout manager pe care un container il foloseste se pot folosi urmatoarele mecanisme:

- folosind un constructor la initializarea referintei containerului:

```
JPanel panel = new JPanel(new CardLayout());
```

- utilizand metoda `setLayout` a containerului deja creat:

```
Container contentPane = frame.getContentPane();
contentPane.setLayout(new CardLayout());
```

In cazul in care se doreste realizarea pozitionarilor si dimensionarilor fara a utiliza un layout manager atunci se poate specifica proprietatea de layout manager a containerului ca fiind `null`.

Adaugarea componentelor intr-un container

Prin utilizarea unui layout manager un container prezinta o serie de metode de `add(...)` ce primesc ca parametri valori care indica felul in care o componenta este pozitionata relativ la restul componentelor si caracteristicilor componentei layout manager utilizate. Containerele Swing care nu sunt content pane sau `JPanel` prezinta un API dedicat pentru adaugarea componentelor grafice.

Pentru a observa diferitele caracteristici se vor consulta sectiunile *how-to* ale [acestei documentatii](#) si/sau se v-a consulta API-ul Swing corespunzator containerelor si componentelor layout manager corespunzatoare.

Indicii de dimensionare si aliniere

Un layout manager decide modul de aliniere si dimensionare a componentelor grafice adaugate unui container. Pentru a indica preferinte de aliniere si dimensionare altele decat cele ce ar fi utilizate implicit de un layout manager se folosesc metodele:

- `setAlignmentX`
- `setAlignmentY`
- `setMinimumSize`
- `setPreferredSize`
- `setMaximumSize`

Multe componente layout manager pot ignora insa aceste indicii. Pentru a afla daca un layout manager aplica sau nu aceste indicii se consulta documentatia API specifica aceluia layout manager.

Exercitiu: Aflati - consultand documentatia Swing API sau realizand un test practic – ce indicii ignora sau nu BorderLayout.

Distantierea componentelor

Spatiul ce separa componentele grafice ale unei interfete este influentat de urmatoorii factori:

- layout manager
 - unele componente de tip layout manager – dar nu toate – adauga implicit spatiu intre componentele grafice pe care le afiseaza;
- componente grafice invizibile
 - se pot utiliza componente grafice care nu realizeaza nici o desenare pe suprafata containerelor care le contin pentru a forta distantierea intre componente vizibile;
- margini goale
 - spatiul dintre componente poate fi afectat (atat in prezenta unui layout manager dar si in absenta lui) prin adaugarea de margini goale componentelor care sunt adaugate interfeței grafice. (A se consulta sectiunea: [How to Use Borders](#)).

Alegerea unui layout manager

Fiecare componenta layout manager are avatanjele si dezavatajele sale. Complexitatea unei aplicatii grafice poate determina necesitatea unei pozitionari absolute sau a crearii unui layout manager dedicat. De asemenea Swing ofera trei layout manager suficient de flexibile pentru a indeplini multe necesitati de amplasare a componentelor, si anume: GridBagLayout si SpringLayout.

- O componenta trebuie sa ocupe cat mai mult spatiu in containerul in care se afla:
 - in cazul in care este singura componenta atunci se poate utiliza GridLayout sau BorderLayout
 - in cazul in care exista mai multe componente atunci se pot utiliza cu succes BorderLayout sau GridBagLayout
- Mai multe componente trebuie amplasate intr-un rand si la dimensiunea lor naturala:
 - FlowLayout, BorderLayout sau SpringLayout
- Afisarea mai multor componente de aceasi dimensiune pe randuri sau coloane:
 - GridLayout
- Afisarea componentelor pe randuri sau coloane cu diverse cerinte de spatiu intre acestea, de aliniere si dimensionare:
 - BorderLayout
- Formarea unei interfete de tip formular (coloane de etichete si campuri de text):
 - SpringLayout
- Amplasare complexa cu o multitudine de componente:
 - SpringLayout si GridBagLayout.
 - Gruparea componentelor in mai multe containere JPanel fiecare cu un layout manager corespunzator.

Exercitiu: Folosind GroupLayout se va implementa urmatoarea interfata grafica:

TV Acquisition Form

Producer:

Diagonal: < 80" 80" - 100" > 100"

Technology: LCD LED CRT YES NO What is 1D

Aspect Ratio: 16:9

Smart TV: NO YES

created with Balsamiq Mockups - www.balsamiq.com

Dezvoltarea unui LayoutManager

Dezvoltarea unui layout manager este o operatie ce trebuie efectuata in conditiile in care nici un alt layout manager nu corespunde necesitatilor aplicatiei sau pentru a crea produse competitive relativ la piata carei se adreseaza aplicatia. De regula insa componente layout manager flexibile precum GridBagLayout, SpringLayout si BoxLayout acopera cerintelor multor aplicatii iar in combinatie cu organizarea in mai multe containere de tip JPanel se pot crea interfete grafice oricât de complexe.

Pentru a crea un layout manager trebuie implementata o clasa care extinde interfața LayoutManager si care implementeaza urmatoarele metode:

- void addLayoutComponent(String, Component) – apelata in metoda add a clasei containerului asociat;
- void removeLayoutComponent(Component) – apelata de metodele remove si removeAll ale containerului asociat;
- Dimension preferredLayoutSize(Container) – apelata de metoda getPreferredSize a containerului primit ca argument;
- Dimension minimumLayoutSize(Container)
- void layoutContainer(Container) – apelata pentru a pozitiona si dimensiona componentele conainerului primit ca parametru. Aceasta metoda trebuie sa tina cont de:
 - marginile interioare ale containerului dupa cum sunt returnate de metoda getInsets()
 - orientarea containerului: getComponentOrientation()
 - containerul nu este obligat sa epeleze preferredLayoutSize sau minimumLayoutSize inainte de apelarea layoutContainer.

Exercitiu: Sa se implementeze un layout manager care adauga componentele grafice ale unui container pe marginea acestuia formand o spirala dreptunghiulara:

Spiral Layout Test

Button1 Button2 Button3

Button12 Button13 Button4

Button11 Button5

Button10 Button6

Button9 Button8 Button7

created with Balsamiq Mockups - www.balsamiq.com

Amplasarea absoluta a componentelor grafice intr-un GUI Swing

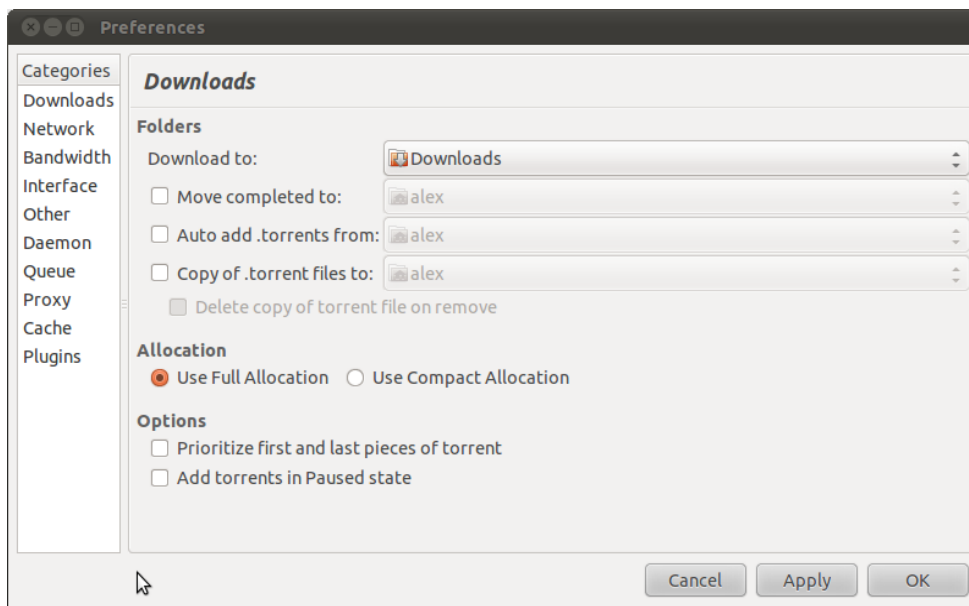
Amplasarea absoluta a componentelor grafice implica absentia unui layout manager. Creatorii API-ului Swing precum si dezvoltatorii oricarui API pentru GUI incurajeaza de regula utilizarea unui layout manager insa pot aparea situatii in care utilizarea acestuia poate deveni dezavantajoasa (fie pentru aplicatii extrem de simple: e.g. un singur label si un buton; fie in situatii de aplicatii care folosesc componente grafice si containere non-standard).

Pentru a utiliza amplasarea absoluta implica crearea unui container fara layout manager:

- odata creat containerul se specifica proprietatea de layout manager ca fiind null:
 - `container.setLayout(null)`
- se apeleaza metoda `setBounds` pentru componentele existente in container
- se apeleaza metoda `repaint` a componentelor existente in container.

```
pane.setLayout(null);
JButton b1 = new JButton("one");
JButton b2 = new JButton("two");
pane.add(b1);
pane.add(b2);
Insets insets = pane.getInsets();
Dimension size = b1.getPreferredSize();
b1.setBounds(25 + insets.left, 5 + insets.top,
             size.width, size.height);
size = b2.getPreferredSize();
b2.setBounds(55 + insets.left, 40 + insets.top,
             size.width, size.height);
...//In the main method:
Insets insets = frame.getInsets();
frame.setSize(300 + insets.left + insets.right,
              125 + insets.top + insets.bottom);
```



“X and 0” GUI



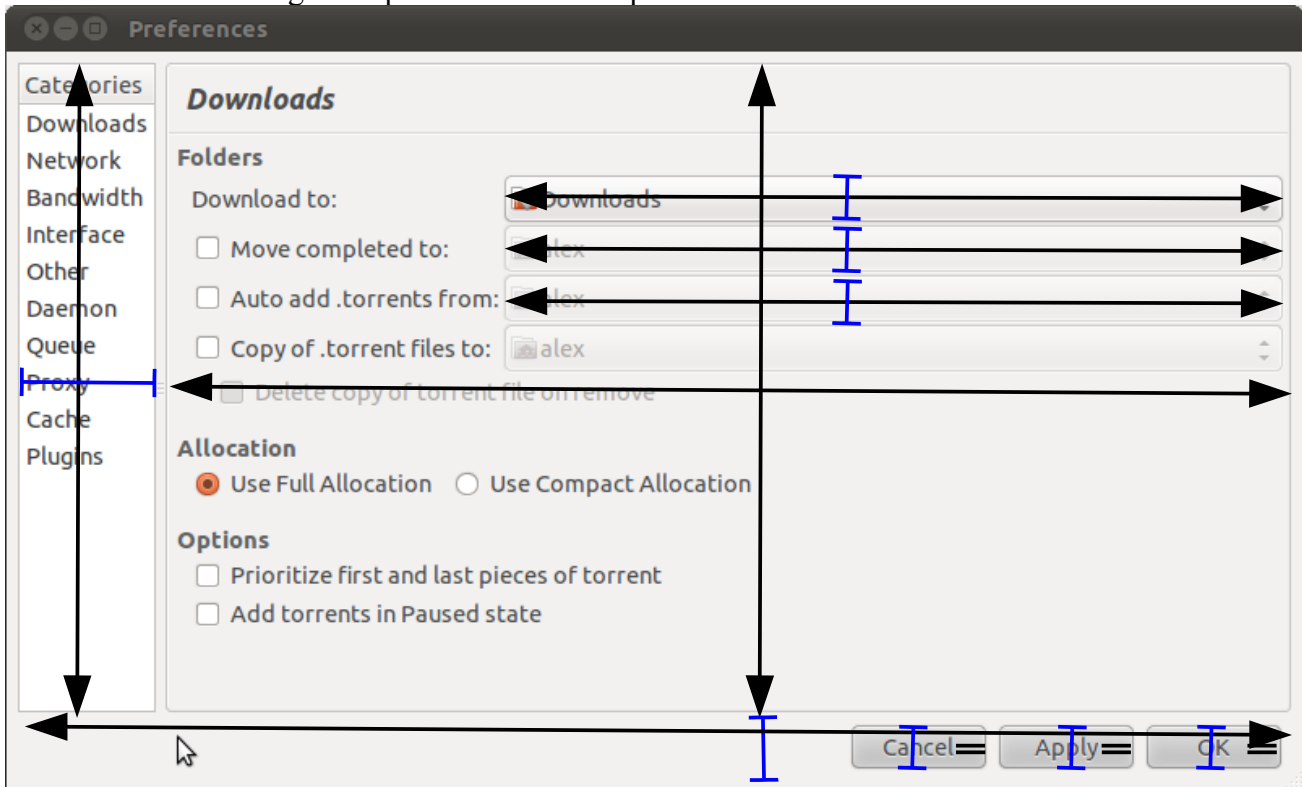
Pentru a implementa o interfata grafica (e.g. vezi figura de mai sus) trebuie evaluati urmatoorii pasi:

1. Desenarea interfetei

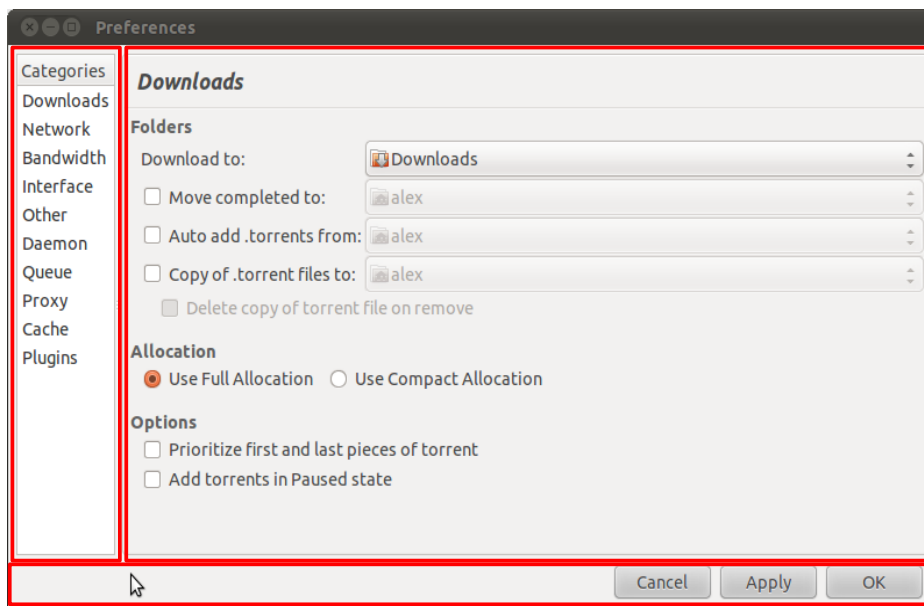
2. Descrierea comportamentului interfeței la redimensionare.

-  - pentru a sublinia zonele care nu suferă redimensionare
-  - pentru a sublinia zonele și componentele care se redimensionează odată cu containerul
- semnul egal este utilizat pentru a indica componente de aceeași dimensiune poziționate adiacent.

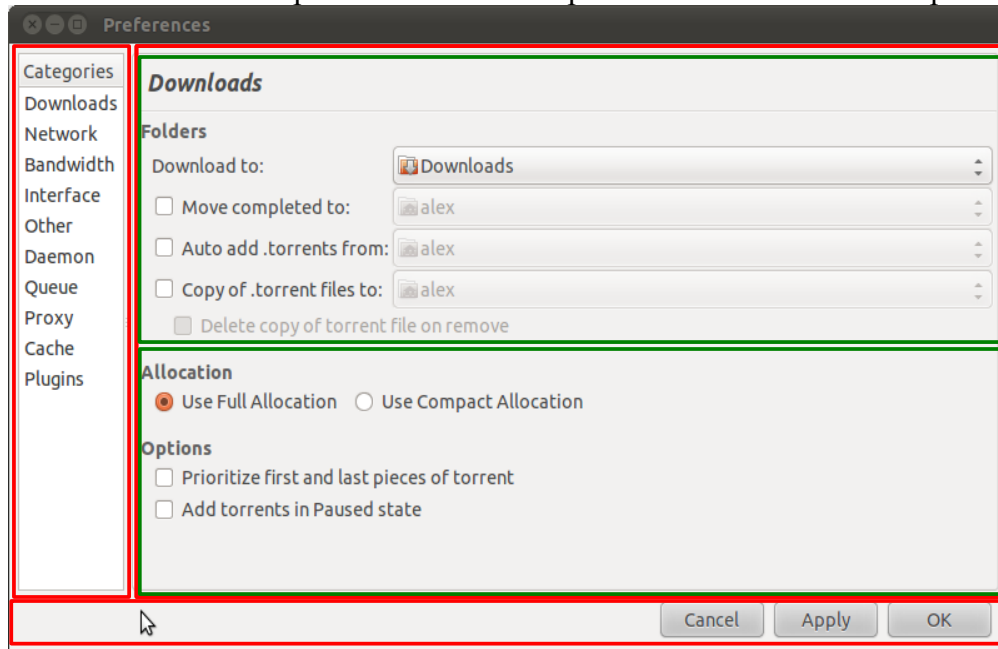
Pentru interfața grafică prezentată ca exemplu mai sus:



Analizând în continuare desenul interfeței se pot observa trei containere diferite:



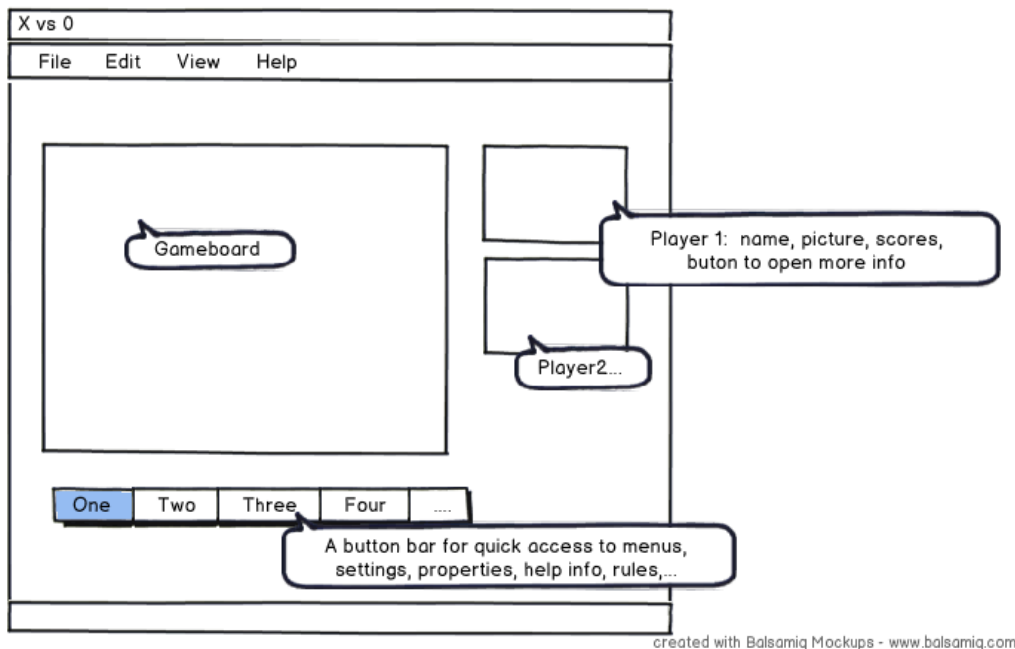
Containerul central se poate de asemenea separa in doua containere independente:



Dupa aceasta separare se pot reface elementele de redimensionare si pozitionare.

Pentru primul container se poate utiliza FlowLayout, pentru panoul cuprinzand butoanele se poate utiliza BorderLayout sau FlowLayout. Pentru cele doua containere centrale se poate utiliza GroupLayout sau SpringLayout.

Exercitiu: Realizati amplasarea componentelor grafice pentru interfata jocului X si 0 folosind componente de tip layout manager si/sau descomponerea in containere.



NOTA: O descriere detaliata a tuturor conceptelor precedente se poate consulta la [aceasta adresa](#).