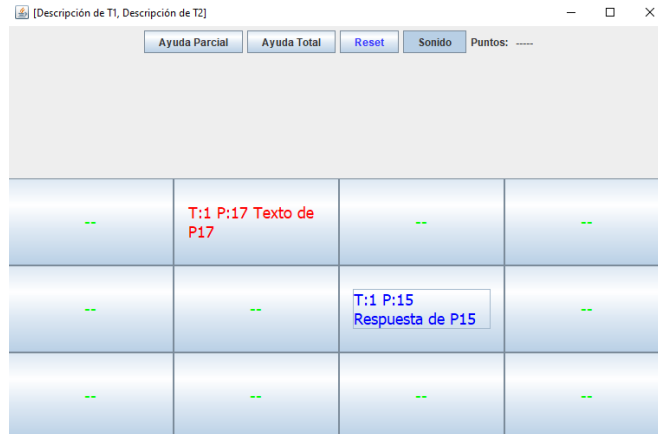


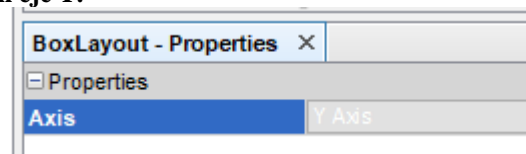
## Programación II: Obligatorio II

### Ventana con 2 paneles que se ajustan según el tamaño de la ventana

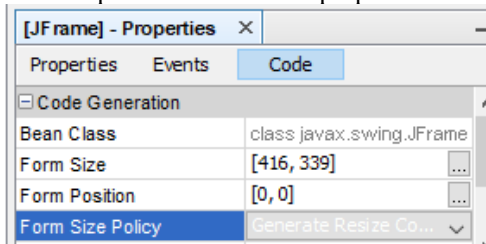
Ejemplo de Juego de Memoria empezado del Tema 1 y Tema 2 con preguntas autogeneradas:



La ventana es un JFrame con **BoxLayout**, con eje **Y**.



Indicarle al Frame que "Generate Resize Code" en la pestaña Code de sus propiedades:



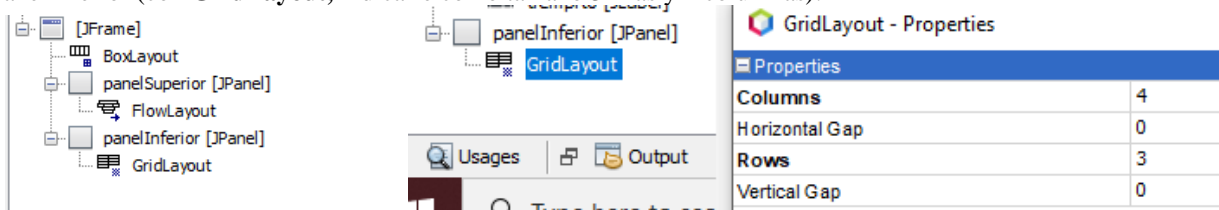
Agregar: `import java.awt.event.*`, `import java.awt.*` e `import javax.swing.*`.

(Nota: En el título se incluyen los temas del juego.)

Esa ventana contendrá dos paneles JPanel:

panelSuperior (con **FlowLayout**)

panelInferior (con **GridLayout**, indicarle como tamaño 3 filas y 4 columnas):



En el panelSuperior, agregar los botones (ej. para ayuda, reset, etc.) e información relevante.

En el panelInferior agregar por programa los botones y el procesamiento del click, como se describe:

Para agregar los botones:

- Definir en la clase una matriz de botones:  

```
private JButton[][] botones;
```
- En el constructor, luego del  `initComponents()`, crear los botones y agregarlos al panel de la matriz.
- `cantFilas` y `cantColumnas` representan respectivamente la cantidad de filas y columnas deseadas, deben ser definidas y cargadas (son 3 filas y 4 columnas).

```
botones = new JButton[cantFilas][cantColumnas];
for (int i = 0; i < cantFilas; i++) {
    for (int j = 0; j < cantColumnas; j++) {
        JButton jButton = new JButton();
        jButton.addActionListener(new ListenerBoton(i, j));
        panelInferior.add(jButton);
        botones[i][j] = jButton;
    }
}
```

Notas:

- para cambiar el ancho del espacio para texto en un botón utilizar:  
`botones[i][j].setMargin(new Insets(-5, -5, -5, -5));`
- para cambiar el color de fondo de un botón por ej. a color Rojo, utilizar:  
`botones[i][j].setBackground(Color.RED);` (Color está en java.awt.Color)

Agregar el texto correspondiente en cada botón. Para visualizar mejor el texto en los botones se puede utilizar, siendo frase el texto a mostrar:

```
botones[i][j].setText("<html><p>" + frase + "</p></html>");
```

- Definir al final de la clase de la Ventana el manejo de listener. Incluir:

```
private class ListenerBoton implements ActionListener {  
    private int x;  
    private int y;  
    public ListenerBoton(int i, int j) {  
        // en el constructor se almacena la fila y columna que se presionó  
        x = i;  
        y = j;  
    }  
    public void actionPerformed(ActionEvent e) {  
        // cuando se presiona un botón, se ejecutará este método  
        clickBoton(x, y);  
    }  
}  
private void clickBoton(int fila, int columna) {  
    // Método a completar!  
    // En fila y columna se reciben las coordenadas donde presionó el usuario, relativas al comienzo de la grilla  
    // fila 0 y columna 0 corresponden a la posición de arriba a la izquierda.  
    // Debe indicarse cómo responder al click de ese botón.  
}
```