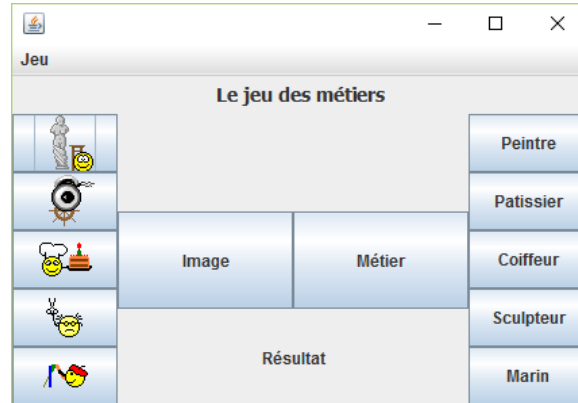


TP2

Exercice 1 : le jeu des métiers

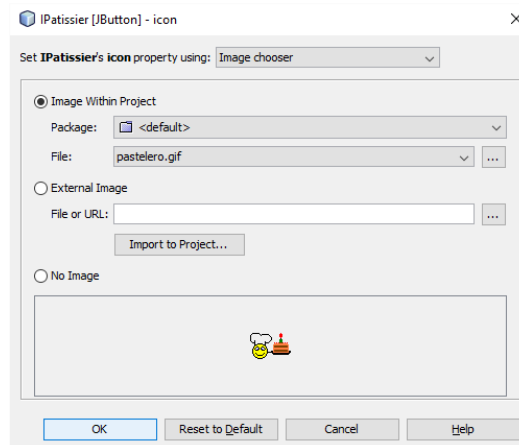
Implémenter le code complet du Jeu des métiers qui a été étudié au TD1 (interface et partie de code) et dont l'interface est rappelée ci-dessous.



Des images de smileys, comme ceux présentés dans l'interface peuvent être téléchargées sur le site <http://www.emoticones.com/profesiones.htm>.

Télécharger 5 images smileys de votre choix et les copier dans le répertoire « src » du projet (en dehors de Netbeans).

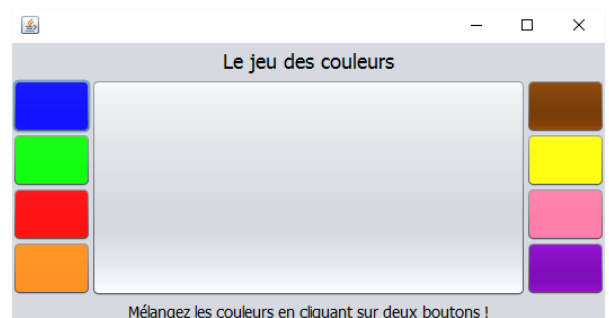
Ensuite pour fixer une image comme valeur de l'icône d'un bouton, cliquer droit sur le bouton, choisir « Propriétés » puis « Icon », cliquer sur le bouton qui comporte trois petits points qui ouvre la fenêtre de sélection d'une image et paramétrer cette fenêtre comme ci-dessous. Si l'image est bien dans le répertoire « src », elle doit apparaître dans la liste déroulante « File ».



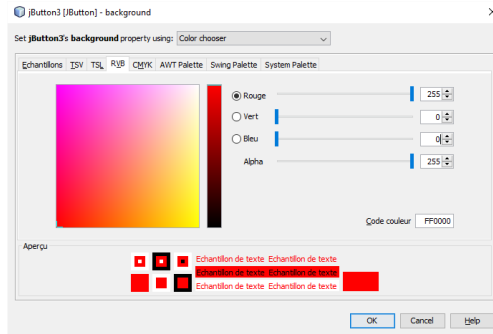
Exercice 2 : Le jeu des couleurs

Le jeu des couleurs est une application qui permet de tester le mélange de deux couleurs. Le joueur clique successivement sur deux boutons de couleur et la couleur de mélange s'affiche comme couleur de fond du panneau au centre de la fenêtre.

Les boutons ont différentes couleurs. Pour choisir la couleur d'un bouton, il faut cliquer droit sur le bouton, sélectionner « Propriétés » puis choisir la propriété « background » et cliquer sur le bouton à droite qui comporte trois petits points, pour ouvrir la fenêtre de choix des couleurs. On choisit de travailler avec des couleurs RGB (Red Green Blue) ou RVB (Rouge Vert Bleu) en français.



L'exemple ci-dessous montre la sélection de la couleur rouge (255,0,0).



1. En suivant les indications données dans le TP1, créer une nouvelle application appelée « JeuCouleurs » et son interface comme décrit ci-dessus.

Pour gérer les couleurs, la classe principale (JFrame) doit disposer de 3 attributs : 2 attributs « couleur1 » et « couleur2 » de type « Color » pour stocker les couleurs choisies et un entier « num » pour connaître le numéro de la couleur à saisir.

```
private Color couleur1 ;
private Color couleur2 ;
private int num ;
```

L'attribut « num » est initialisé à la valeur 1 dans le constructeur.

```
public JeuCouleurs()
{
    initComponents() ;
    num=1 ;
}
```

Chaque fois que l'utilisateur clique sur un bouton de couleur, il faut récupérer la couleur de ce bouton (par ex : Color c= bouton.getBackground()). Si la valeur de num vaut 1, cette couleur est affectée à couleur1, et num passe à 2. Et si la valeur de num vaut 2, la couleur est affectée à couleur2, la valeur de num repasse à 1. Le mélange des couleurs est effectué par la méthode « melanger() » qui colorie également le fond du panneau avec cette couleur.

2. Écrire le code du gestionnaire correspondant au clic sur les boutons de couleur.

Une couleur étant gérée par un triplet de valeurs pour chacune des 3 couleurs (rouge, vert, bleu), pour les mélanger il faut faire la moyenne des valeurs de chacune de ces 3 valeurs. Les méthodes « getRed() », « getGreen() » et « getBlue() » permettent de récupérer respectivement la composante rouge, verte et bleue d'une couleur.

```
int r= (couleur1.getRed() + couleur2.getRed())/2 ;
int v =(couleur1.getGreen() + couleur2.getGreen())/2;
int b =(couleur1.getBlue() + couleur2.getBlue())/2;
Color melange = new Color(r, v, b);
```

3. Écrire le code de la méthode « melanger() ».
4. Tester le jeu

Exercice 3 : Les boutons (JRadioButton, JCheckBox, JToggleButton, JButton)

Implémenter le programme « LesBoutons » dont l'interface est fournie ci-dessous. Cette dernière permet de choisir les propriétés du bouton situé en bas à droite. L'utilisateur peut cliquer sur un des boutons radios (« JRadioButton ») pour choisir la couleur de fond, sur un des composants de type « JToggleButton » pour la couleur d'écriture et sur une des cases à cocher (« JCheckBox ») pour la police d'écriture. Ces propriétés sont directement appliquées au bouton de type « JButton ». Les boutons radios et les cases à cocher sont groupés (mutuellement exclusifs) grâce à un composant non visuel de type « ButtonGroup ».

