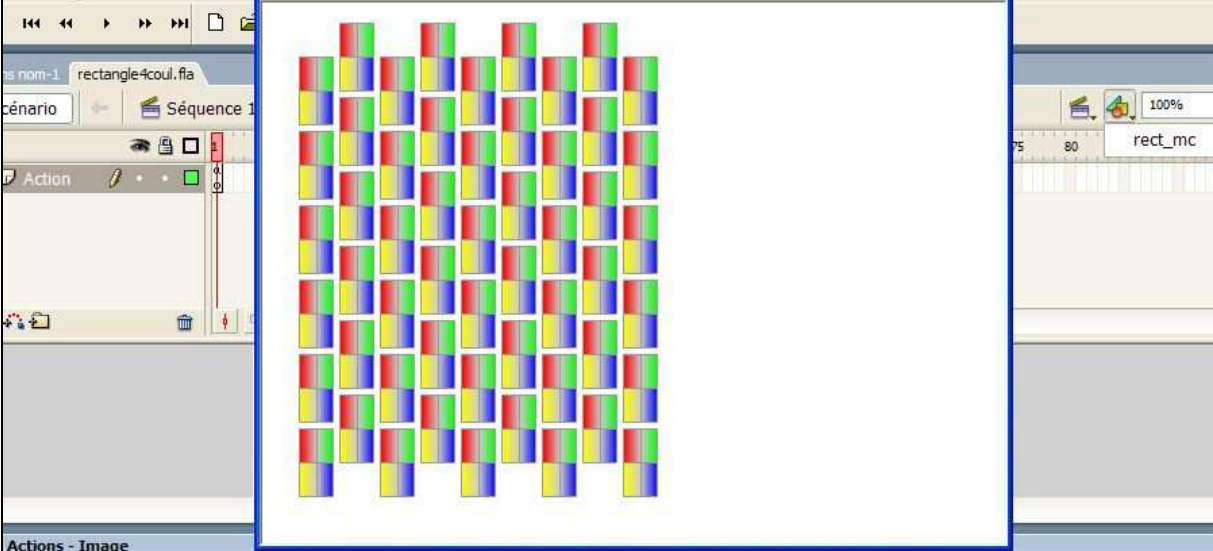


DIVERS PAVAGES

1 Motif de Jos

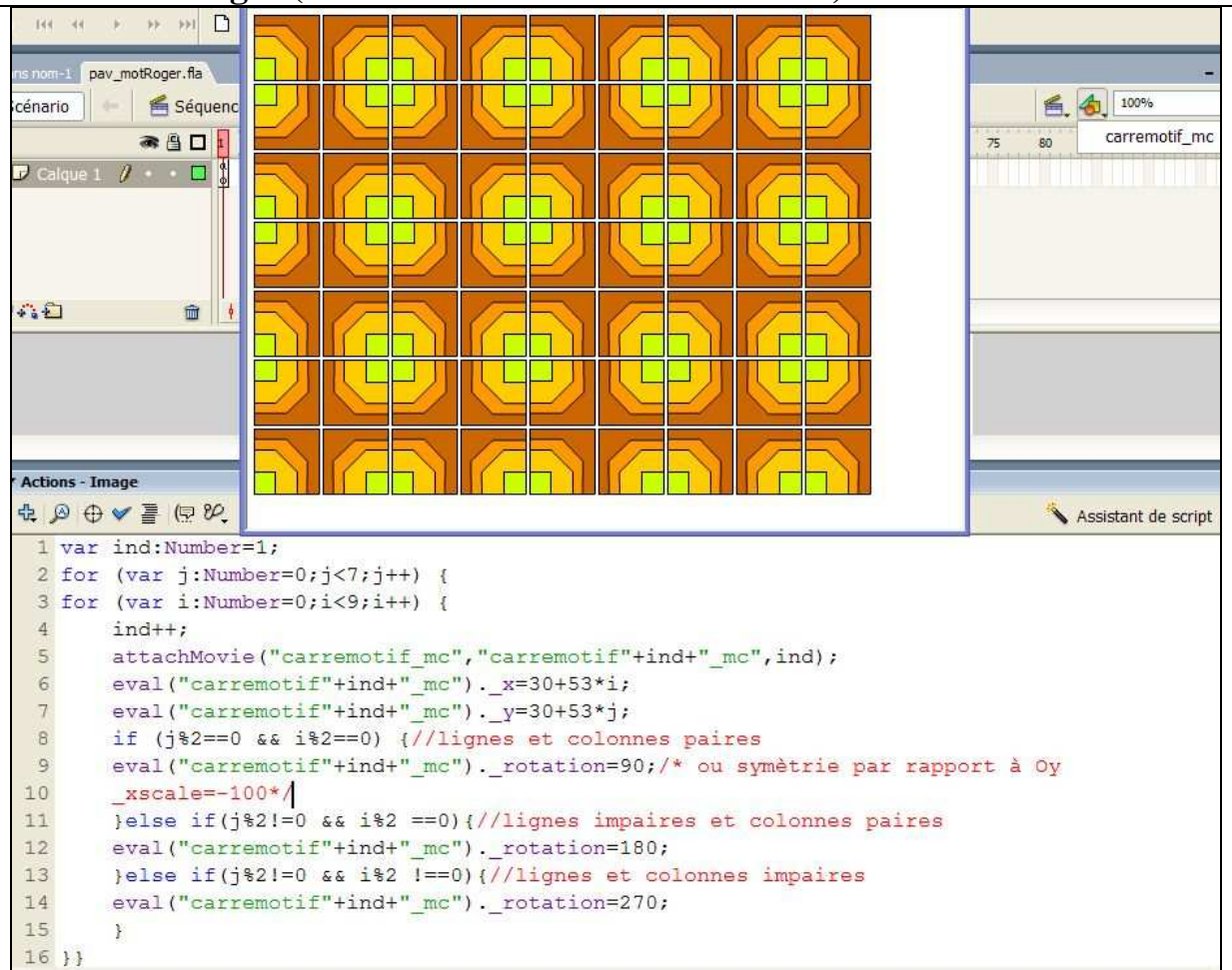


The screenshot shows a software interface with a central scene view displaying a 6x9 grid of colored rectangles. The rectangles are arranged in a staggered pattern, with each row offset by half a column from the previous one. The colors used are red, green, blue, and yellow. The interface includes a timeline at the top, a scene view on the right, and an actions panel at the bottom.

```

1 /*aligner sur 6 lignes j et 9 colonnes i (2 boucles for imbriquées)
2 position initiale à 40 et 40 du bord puis +25 en x et +50 en y et joint 5pixels*/
3 ind=1;
4 for(var j:Number=0; j<6; j++){
5     for(var i:Number=0; i<9; i++){
6         ind++;
7         attachMovie("rect_mc", "rect"+ind+"_mc", ind);
8         if (i%2==0) {
9             eval("rect"+ind+"_mc")._x=40+30*i;
10            eval("rect"+ind+"_mc")._y=40+25+55*j;
11        }else{
12            eval("rect"+ind+"_mc")._x=40+30*i;
13            eval("rect"+ind+"_mc")._y=40+55*j;
14        }}}
    
```

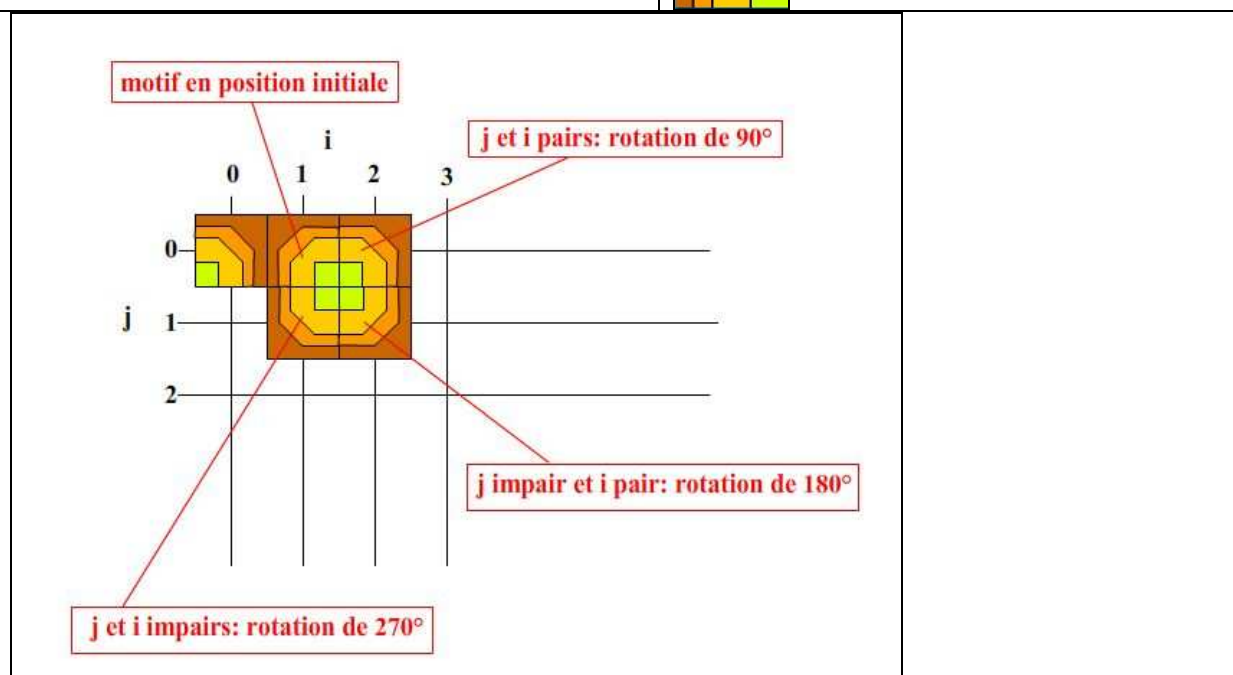
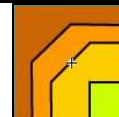
2 Motif de Roger (conditions combinées et rotations)



```

1 var ind:Number=1;
2 for (var j:Number=0;j<7;j++) {
3   for (var i:Number=0;i<9;i++) {
4     ind++;
5     attachMovie("carremotif_mc","carremotif"+ind+"_mc",ind);
6     eval("carremotif"+ind+"_mc")._x=30+53*i;
7     eval("carremotif"+ind+"_mc")._y=30+53*j;
8     if (j%2==0 && i%2==0) { //lignes et colonnes paires
9       eval("carremotif"+ind+"_mc")._rotation=90; /* ou symétrie par rapport à Oy
10      _xscale=-100*/
11     } else if (j%2!=0 && i%2 ==0) { //lignes impaires et colonnes paires
12       eval("carremotif"+ind+"_mc")._rotation=180;
13     } else if (j%2!=0 && i%2 !=0) { //lignes et colonnes impaires
14       eval("carremotif"+ind+"_mc")._rotation=270;
15     }
16  }}
    
```

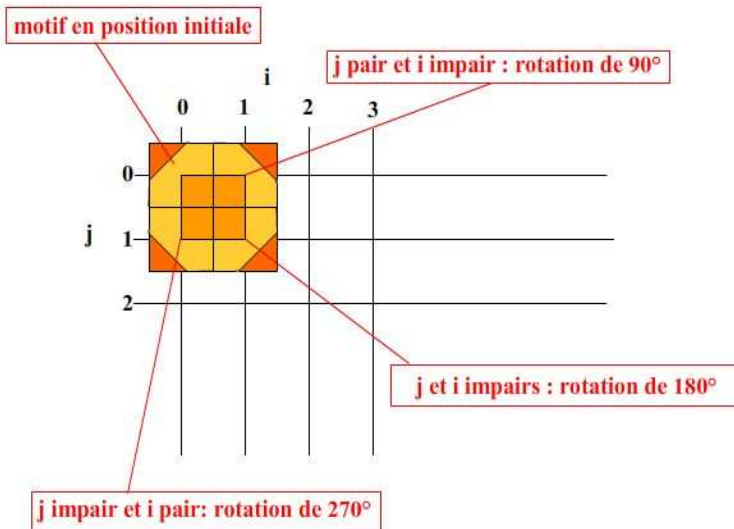
Clip carremotif_mc de 50x50



2 bis 3 rotations

```

1 var ind:Number=1;
2 for (var j:Number=0;j<8;j++) {
3   for (var i:Number=0;i<10;i++) {
4     ind++;
5     attachMovie("carremotif_mc", "carremotif"+ind+"_mc", ind);
6     eval("carremotif"+ind+"_mc")._x=30+53*i;
7     eval("carremotif"+ind+"_mc")._y=30+53*j;
8     if (j%2==0 && i%2!=0) { //lignes paires et colonnes impaires
9       eval("carremotif"+ind+"_mc")._rotation=90; /* ou symétrie par rapport à Oy
10      _xscale=-100*/
11     } else if (j%2!=0 && i%2 !=0) { //lignes et colonnes impaires
12       eval("carremotif"+ind+"_mc")._rotation=180;
13     } else if (j%2!=0 && i%2 ==0) { //lignes impaires et colonnes paires
14       eval("carremotif"+ind+"_mc")._rotation=270;
15     }
16   }
  }
  
```



3 Pavage avec 2 clips alternés

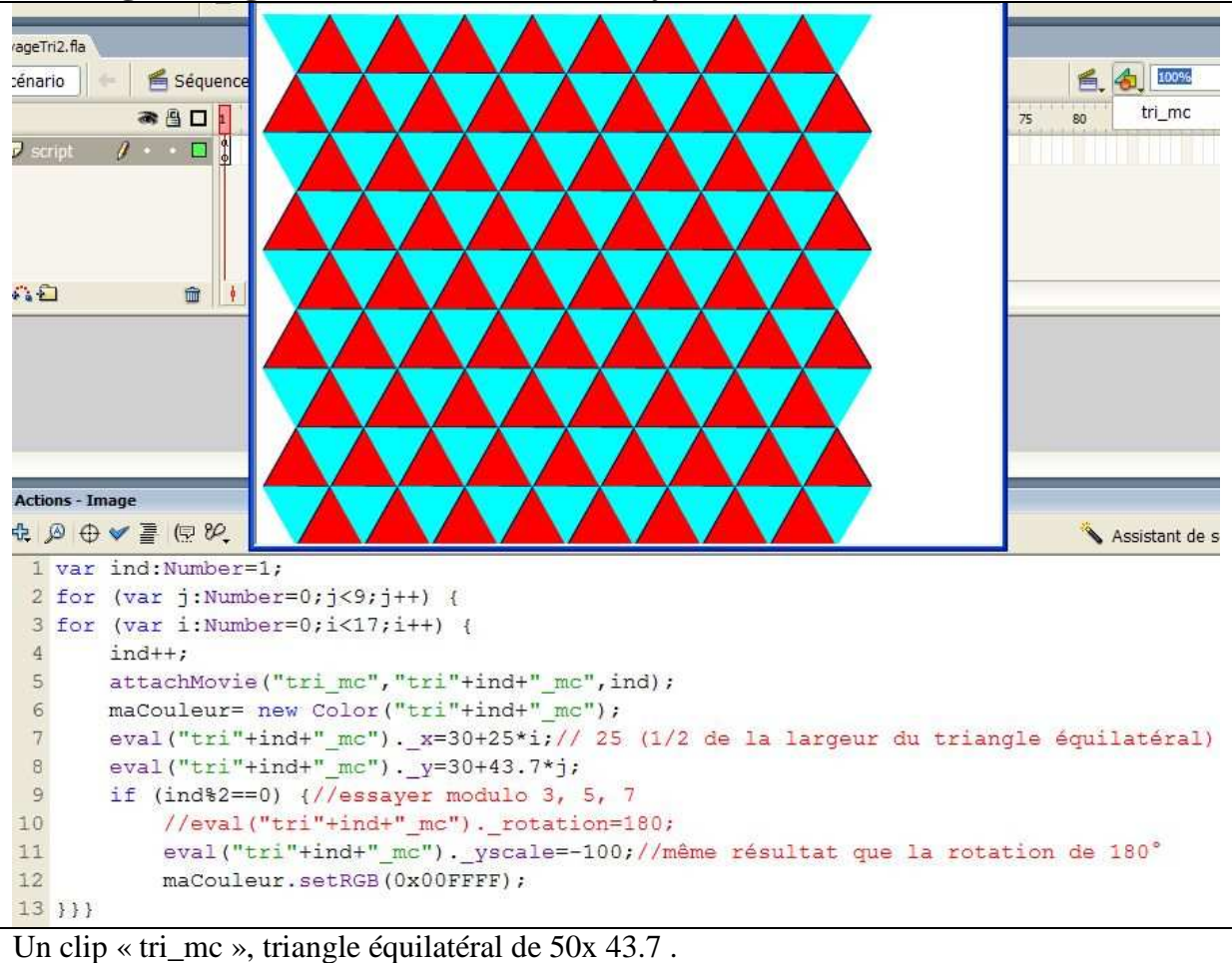
The screenshot displays the animation software interface for a project named 'pav_2clipsAlternes.fla'. The main workspace shows a 6x9 grid of alternating blue squares and circles. The interface includes a timeline at the top right, a sequence editor on the left, and an actions panel at the bottom containing the following ActionScript code:

```

1 //CARRELER mon sol(rien sur la scène; un script et 2 clips appelés avec attachMovie)
2 ind=1;//indice pour les occurrences de clips et leur profondeur.
3 //2 boucles imbriquées
4 for (var j:Number=0;j<6;j++) {/*lignes de 0 à 5, donc 6 lignes.Pour chaque
5 valeur de j la boucle imbriquée des colonnes va s'effectuer, donc les lignes
6 vont se remplir*/
7     for (var i:Number=0; i<9; i++) {//colonnes de 0 à 8 donc 9 colonnes
8         ind++;//incrémementation de l'indice
9         if(ind%2==0 ){
10            _root.attachMovie("carre_mc","carre"+ind+"_mc",ind);
11            eval("carre"+ind+"_mc")._x=i*55+40;//abscisse
12            eval("carre"+ind+"_mc")._y=j*55+40;//ordonnée
13        }else{
14            _root.attachMovie("cercle_mc","cercle"+ind+"_mc",ind);
15            eval("cercle"+ind+"_mc")._x=i*55+40;//abscisse
16            eval("cercle"+ind+"_mc")._y=j*55+40;//ordonnée
17        }}}

```

4 Pavage triangles, couleurs alternées, symétrie (ou rotation).



The screenshot shows a software interface for creating an animation. The central area displays a grid of triangles in a repeating pattern of red and cyan. The interface includes a timeline at the top right, a script editor at the bottom, and a properties panel on the left. The script editor contains the following code:

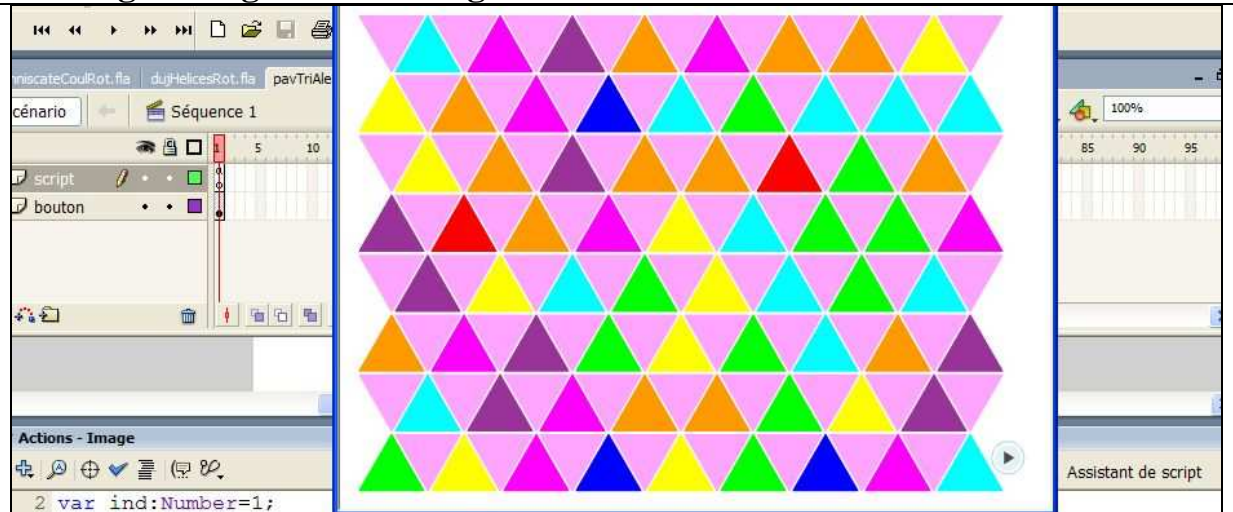
```

1 var ind:Number=1;
2 for (var j:Number=0;j<9;j++) {
3   for (var i:Number=0;i<17;i++) {
4     ind++;
5     attachMovie("tri_mc","tri"+ind+"_mc",ind);
6     maCouleur= new Color("tri"+ind+"_mc");
7     eval("tri"+ind+"_mc")._x=30+25*i;// 25 (1/2 de la largeur du triangle équilatéral)
8     eval("tri"+ind+"_mc")._y=30+43.7*j;
9     if (ind%2==0) { //essayer modulo 3, 5, 7
10      //eval("tri"+ind+"_mc")._rotation=180;
11      eval("tri"+ind+"_mc")._yscale=-100;//même résultat que la rotation de 180°
12      maCouleur.setRGB(0x00FFFF);
13    }}

```

Un clip « tri_mc », triangle équilatéral de 50x 43.7 .

5 Pavage triangles avec changement aléatoire de couleur



The screenshot shows a software interface for creating an animation. The main area displays a grid of 17 columns and 8 rows of equilateral triangles. Each triangle is a different color, including shades of purple, blue, green, yellow, orange, and red. The interface includes a timeline at the top with a play button, a script editor on the left, and a preview window on the right. The script editor contains the following code:

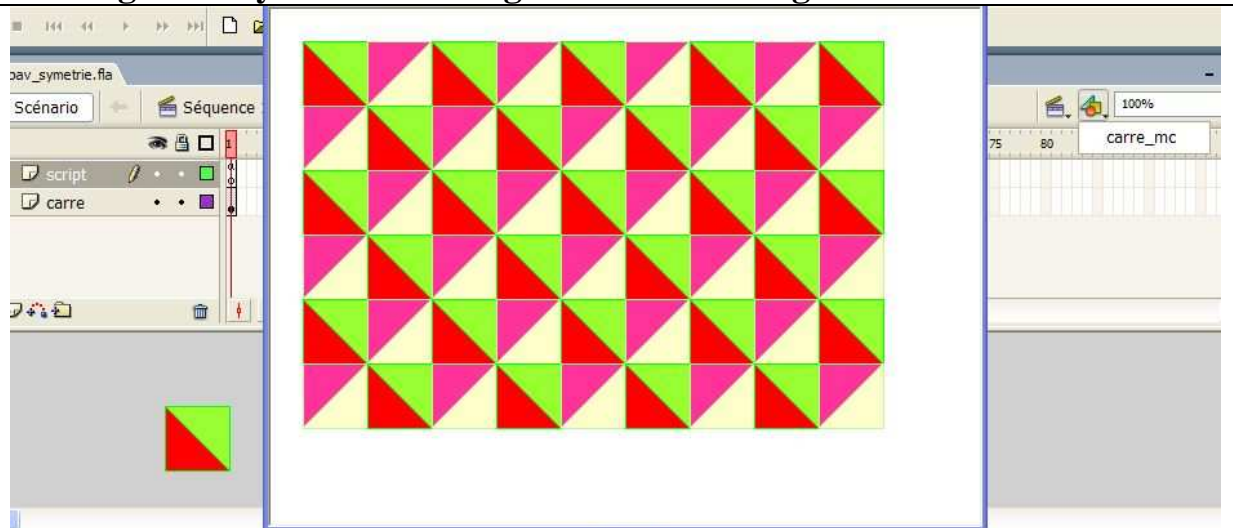
```

2 var ind:Number=1;
3 //déclaration d'un tableau de couleurs
4 var valcouleurs=["FF0000","FF00FF","00FF00","00FFFF","FF9900","993399","FFFF00","0000FF"];
5 for(var j:Number=0;j<8;j++){// 8 lignes
6 for(var i:Number=0;i<17;i++){// 17 colonnes
7 ind++;
8 macouleur=new Color("triangle"+ind+"_mc");/*choisir une nouvelle couleur et
9 l'appliquer aux occurrences de triangle_mc*/
10 attachMovie("triangle_mc","triangle"+ind+"_mc",ind);
11 if(ind%2==0){//test ind pair
12 eval("triangle"+ind+"_mc")._rotation=180;
13 macouleur.setRGB(0xFFA5FF);//mauve (R 255,G 165, B 255)
14 }else{
15 k=Math.round(Math.random()*7);//choix de 0 à 7
16 appelCoul=valcouleurs[k];//appel à la couleur de rang k dans le tableau
17 macouleur.setRGB("0x"+appelCoul);//aléatoire parmi 8 couleurs
18 }
19 eval("triangle"+ind+"_mc")._x=40+28*i;//28= 25(1/2 base triangle)+3(espace ou joint)
20 eval("triangle"+ind+"_mc")._y=40+46.3*j;//46.3= 43.3 (hauteur triangle) +3
21 }}
22 recommencer_btn.onRelease=function(){ //recommencer l'animation
23 _root.loadMovie("pavTriAleaCoul.swf")
24 }

```

Un clip « triangle_mc », équilatéral appelé par liaison. Un bouton choisi dans la bibliothèque commune et de nom d'occurrence « recommencer_btn ».

6 Pavage avec symétrie et changement des décalages des couleurs RVB



The screenshot shows a software interface for creating an animation. The main window displays a 6x6 grid of colored squares. The squares are colored in a repeating pattern of red, green, and yellow. The interface includes a timeline, a script editor, and a preview window.

```

1 ind=0;
2 for(var j:Number=0;j<6;j++){
3   for(var i:Number=0;i<9;i++){
4     ind++;
5     mcouleur=new Color("carre"+ind+"_mc");/*utilisation du constructeur new Color avec
6     affectation au clip dont on veut manipuler les couleurs*/
7     carre_mc.duplicateMovieClip("carre"+ind+"_mc",ind);
8     eval("carre"+ind+"_mc")._x=50+i*50;
9     eval("carre"+ind+"_mc")._y=50+j*50;
10    if (ind%2==0) {
11      eval("carre"+ind+"_mc")._yscale=-100;//symétrie suivant Ox
12      chtC={rb:183,gb:51, bb:153};/*ensemble des paramètres pour changement de couleur;
13      on peut intervenir sur 8 paramètres;ra,ga,ba et aa sont les pourcentages du rouge,
14      du vert, du bleu et de l'alpha (transparence) de -100 à 100;rb,gb,bb et ab sont les
15      décalages des 3 couleurs R,V,B et de l'alpha de -255 à 255*/
16      mcouleur.setTransform(chtC); //méthode de changement de couleur
17    }
18  }}
  
```

Un clip `carre_mc` glissé à côté de la scène (même nom d'occurrence).

Variante de code pour la ligne 12 :

```
chtC={rb:55-3*ind,gb:4*ind, bb:255-3*ind,aa:50};
```