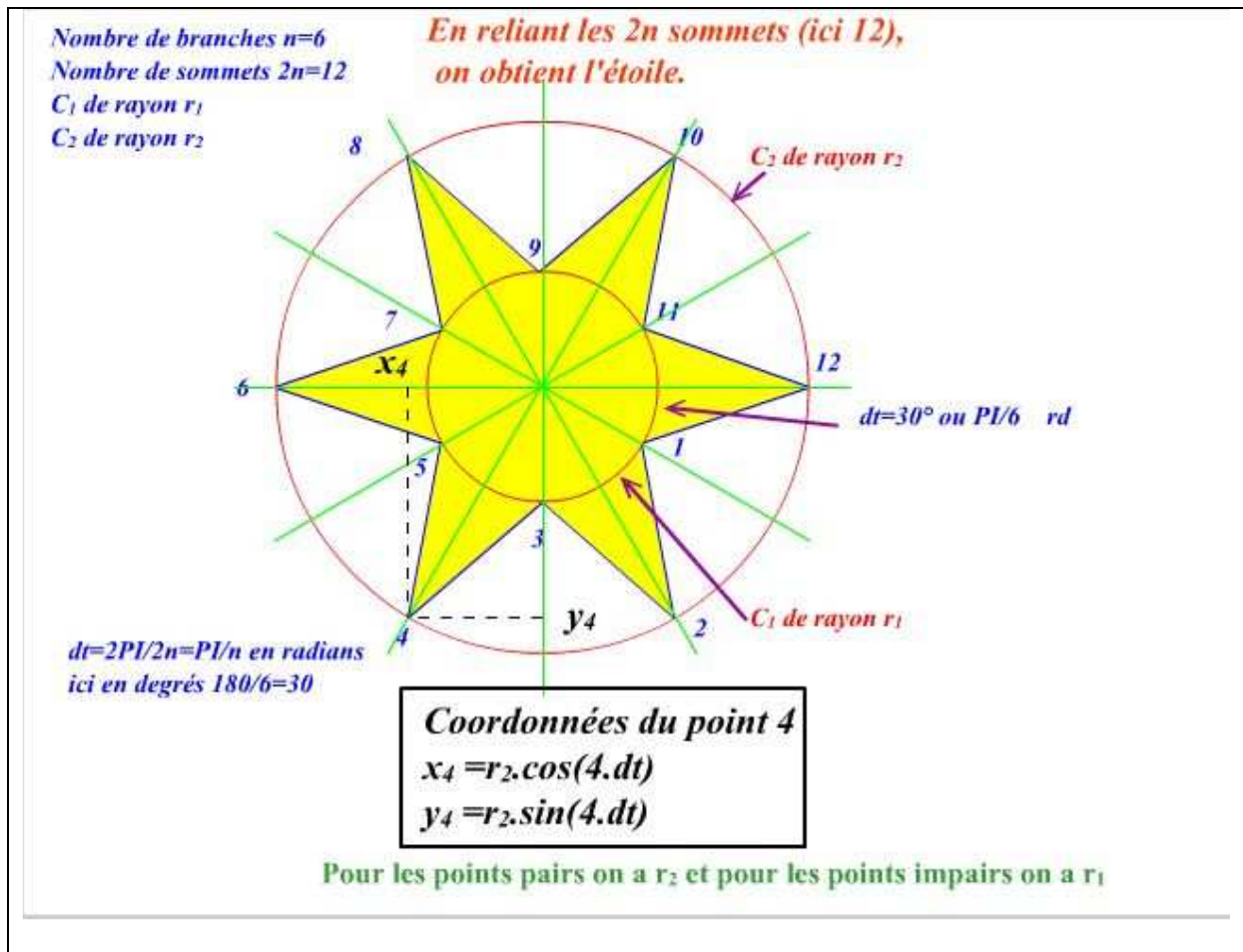


CREATION DYNAMIQUE d'une ETOILE

1_Observation de ce dessin.



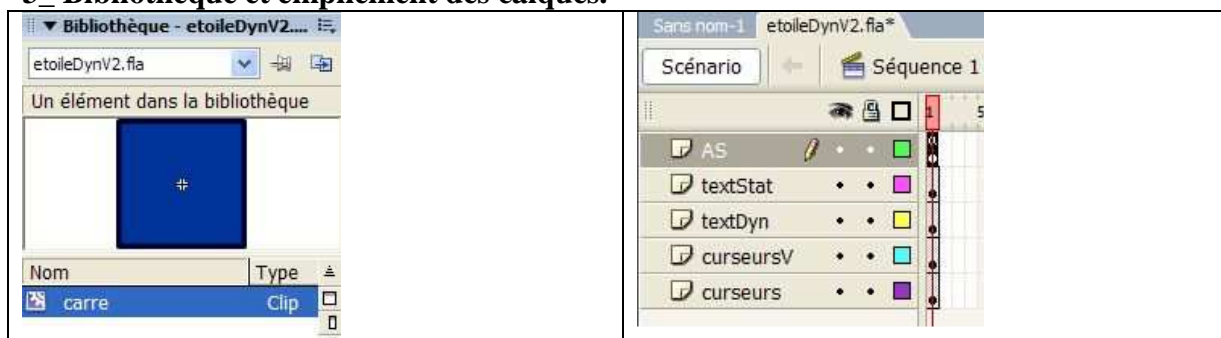
2_ Nom du fichier : etoileDynV2.fla. Document : *dimensions* : 800 px x 600 px
 : *arrière plan* : blanc
 : *cadence* : 12 im/s

L'étoile sera créée à l'aide d'un code, faisant intervenir le nombre de branches, donc de sommets répartis sur les cercles de rayons r_1 et r_2 . Les coordonnées des différents sommets seront calculées (voir sommet 4). Ces sommets seront reliés successivement.

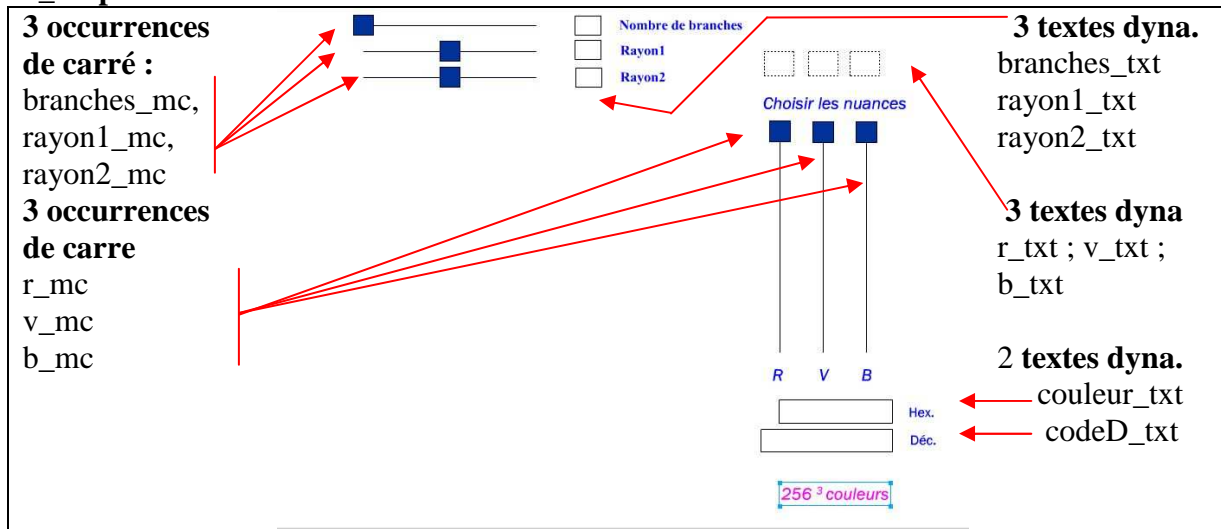
Des curseurs feront varier le nombre de branches, les rayons r_1 et r_2 .

Enfin des curseurs permettront de choisir les composantes r , v , b des couleurs de 0 à 255, pour le remplissage de l'étoile.

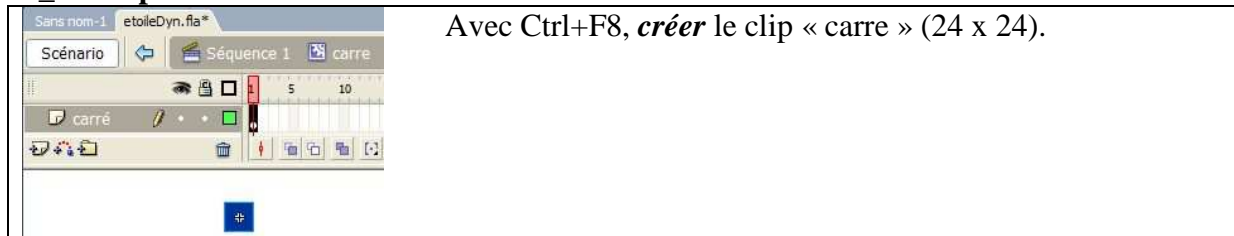
3_ Bibliothèque et empilement des calques.



4_ Aspect final de la scène.



5_ Le clip «carre ».



Avec Ctrl+F8, **créer** le clip « carre » (24 x 24).

6_ Le calque « curseurs » (curseurs horizontaux).

Retour dans la séquence :

- **Renommer** le calque 1 « curseurs ». **Tracer** un premier segment, L = 200 et (x, y) = (200, 20). **Dupliquer** ce segment et **placer** le à (x, y) = (200, 50). **Dupliquer** à nouveau et (x, y) = (200, 80) .Sur ce calque **glisser déposer** le clip « carre » et l'**ajuster** à (x, y) = (100, 20). **Donner** le nom d'occurrence branches_mc. **Dupliquer** ce clip, le **placer** à (x, y) = (200, 50) et **changer** le nom d'occurrence en rayon1_mc. **Dupliquer** à nouveau, (x, y) = (200, 80) et nom d'occurrence rayon2_mc.

7_ Le calque « curseursV » (curseurs verticaux).

Rappel sur les couleurs, système R V B (ou R G B).

Les nuances de chaque composante R V B sont définies sur un octet 2^8 , de 0 à 255, soit 256 nuances. Total des couleurs 256^3 , soit 16 777 216

Pour exprimer les nuances de 0 à 255 on utilise aussi la base hexadécimale (base 16).

0 1 2 3 4 5 6 7 8 9 A B C D E F
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Exemples pour 6 des 256 nuances de chaque composante.

La nuance FF : le F de droite est au rang des unités (donc 15), le F de gauche est au rang des « seizaines », donc 16×15 et

FF correspond donc à $15 + 16 \times 15 = 255$

La nuance CC correspond à $12 + 16 \times 12 = 204$

La nuance 99 correspond à $9 + 16 \times 9 = 153$

La nuance 66 correspond à $6 + 16 \times 6 = 102$

La nuance 33 correspond à $3 + 16 \times 3 = 51$

La nuance 00 correspond à 0.

Donc dans la séquence, sur le calque « curseursV », **tracer** un segment vertical de hauteur H=255 px et (x, y) = (581, 270.5). **Dupliquer** 2 fois le segment. Le 2^{ème} segment est placé à (x, y) = (631, 270.5). Le 3^{ème} segment est placé à (x, y) = (681, 270.5). **Glisser** sur la scène et sur le calque « curseursV », le clip « carre ». **Dupliquer** le 2 fois. Les coordonnées des curseurs sont respectivement (581, 143), (631, 143), (681, 143). Les noms d'occurrences sont r_mc, v_mc, b_mc.

7_Le calque « textDyn ».

Insérer un nouveau calque à renommer « textDyn. **Créer** le premier texte dynamique branches_txt. **Créer** les autres textes dynamiques de noms d'occurrences rayon1_txt, rayon2_txt, r_txt, v_txt, b_txt, couleur_txt et codeD_txt. [Voir : 3_Aspect final, pour l'emplacement de chacun.](#)

8_Le calque « textStat ».

Insérer un nouveau calque à renommer « textStat ». **Ecrire** tous les textes statiques figurant sur [3_Aspect final.](#)

9_Le calque « AS ».

```

1 //etoileDynV2 fla
2 //MODIFICATION propriétés du clip par curseurs.
3 //initialisation des positions des 3 curseurs en abscisse
4 branches_mc._x=200;
5 rayon1_mc._x=200;
6 rayon2_mc._x=250;
7 /*Création de la fonction deplace avec ses 2 arguments
8 curseur et positiony ordonnée fixe du curseur*/
9 fonction deplace(curseur,positiony) {
10     //déclenchement mouvt du curseur pointé(gestionnaire onPress)
11     _root[curseur].onPress=function() {
12         //false pour verrouillage sur position où on a cliqué sur le clip
13         this.startDrag(false,100,positiony,300,positiony);
14     }
15 //pour quitter le drag avec le relâché sur et en dehors du curseur
16 _root[curseur].onRelease=_root[curseur].onReleaseOutside=function() {
17     this.stopDrag();
18 }
19 }
20 //utilisation de la fonction deplace (déplacement des curseurs)
21 deplace("branches_mc",20); //déplacement sur y=20
22 deplace("rayon1_mc",50);
23 deplace("rayon2_mc",80);
24 //*****

```



```

25 function deplace2(curseur,positionx){
26     //déclenchement mouvt du curseur pointé(gestionnaire onPress)
27     _root[curseur].onPress=function(){
28         //false pour verrouillage sur position où on a cliqué sur le clip
29         this.startDrag(false,positionx,143,positionx,398);
30     }
31 //pour quitter le drag avec le relâché sur et en dehors du curseur
32 _root[curseur].onRelease=_root[curseur].onReleaseOutside=function(){
33     this.stopDrag();
34 }
35 }
36 //utilisation de la fonction deplace2 (déplacement des curseurs V)
37 deplace2("r_mc",581);//déplacement sur x=581
38 deplace2("v_mc",631);
39 deplace2("b_mc",681);
40 //*****
41 //moteur pour modifications Nombre de branches,Rayon1,Rayon2
42 _root.onEnterFrame=function(){
43 //Le nbre de branches varie de 0 à 100 et s'inscrit ds le texte dyn
44 branches_txt.text=n=Math.round((branches_mc._x-100)/2);//de 0 à 100
45 //Le rayon1 varie de 0 à 260 et s'inscrit ds le texte dyn
46 rayon1_txt.text=r1=Math.round((rayon1_mc._x-100)*1.3);
47 //r2 varie de 0 à 260 et s'inscrit ds le texte dyn
48 rayon2_txt.text=r2=Math.round((rayon2_mc._x-100)*1.3);
49 //*****
50 //Variation des valeurs rc,vc,bc des couleurs
51 var r:String;
52 var v:String;
53 var b:String;
54 rc=Math.round(r_mc._y-143);
55 r=rc.toString(16);
56 if(r.length==1) r="0"+r;
57 vc=Math.round(v_mc._y-143);
58 v=vc.toString(16);
59 if(v.length==1) v="0"+v;
60 bc=Math.round(b_mc._y-143);
61 b=bc.toString(16);
62 if(b.length==1) b="0"+b;
63 //*****
64 /*Création dynamique d'une étoile. Voir avant le dessin explicatif
65 montrant les 2 cercles de rayons r1 et r2, le nombre n de
66 branches et l'angle dt; c1 couleur trait, c2 couleur remplissage;*/
67 function forme(r1,r2,n,c1,c2){
68     _root.createEmptyMovieClip("etoile",0);
69     var dt:Number=Math.PI/n;
70     with(etoile){
71         lineStyle(1,c1,100);//épaisseur,couleur,alpha
72         moveTo(r2,0);//départ au point (r2,0)
73         beginFill(c2,100);//remplissage,couleur,alpha
74         for(var i:Number=1;i<2*n+1;i++){
75             var t:Number=dt*i;
76             var r:Number=(i%2==0) ? r2:r1;//i pair r=r2,i impair r=r1

```

```
77         var x:Number=r*Math.cos(t);//abscisse
78         var y:Number=r*Math.sin(t);//ordonnée
79         lineTo(x,y);//jonction
80     }
81     endFill();//fin remplissage
82 }}
83 //*****
84 forme(r1,r2,n,"0x0000FF","0x"+(r+v+b).toUpperCase());
85 _root.etoile._x=300;//étoile centrée à (300,350)
86 _root.etoile._y=350;
87 //renseigner les textes dynamiques teintés r v b
88 r_txt.text=r;v_txt.text=v;b_txt.text=b;
89 //*****Pour changement couleur remplissage
90 //déclaration et initialisation variables
91 //afficher code couleur et le convertir
92 //toUpperCase( ) fait passer en majuscules
93 couleur_txt.text=(r+v+b).toUpperCase();
94 var a:Number;
95 /*parseInt( ) permet de définir la base (ici 16)
96 et convertit*/
97 a=parseInt(r+v+b,16);
98 codeD_txt.text=a;
99 }
```