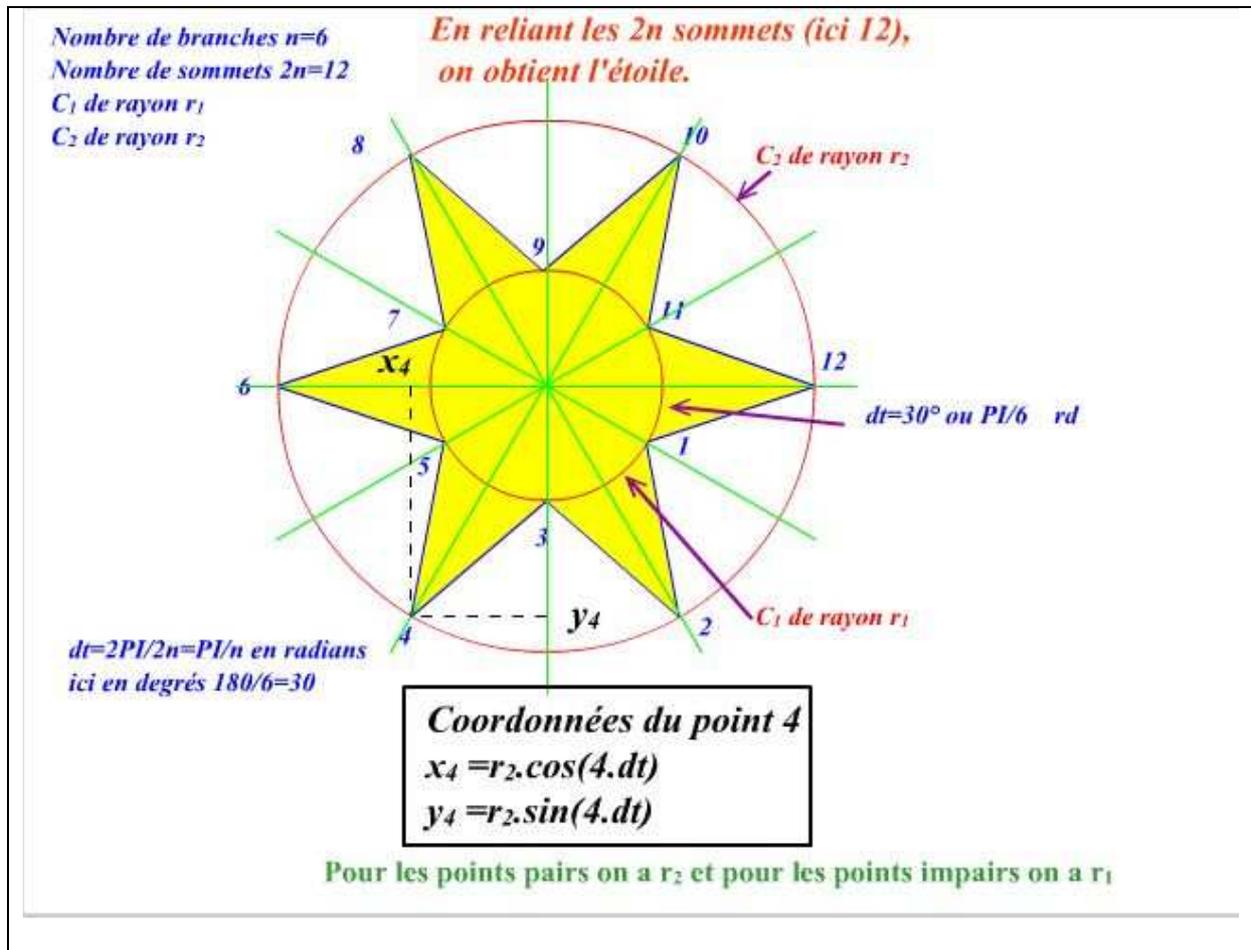


## CREATION DYNAMIQUE d'une ETOILE

### 1\_Observation de ce dessin.



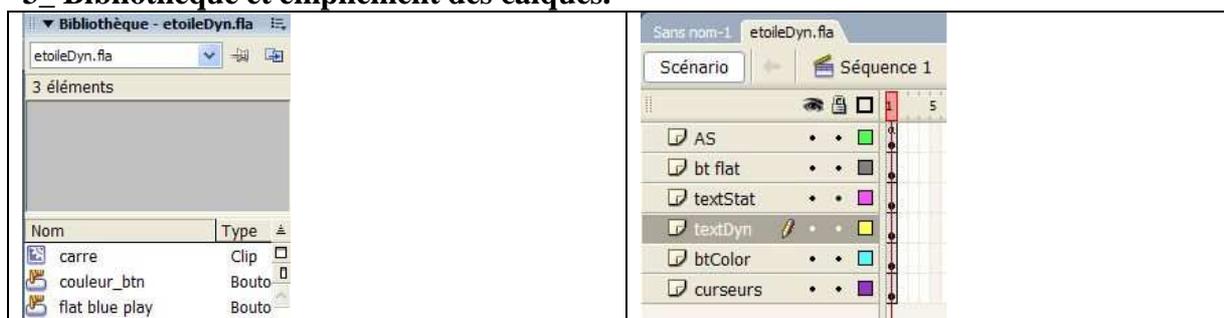
**2\_ Nom du fichier : etoileDyn.fla. Document :** *dimensions* : 800 px x 600 px  
 : *arrière plan* : blanc  
 : *cadence* : 12 im/s

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 r1 et r2. 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 r1 et r2.

Enfin un nuancier permettra de choisir des couleurs pour le remplissage de l'étoile.

### 3\_Bibliothèque et empilement des calques.



#### 4\_ Aspect final de la scène.

**3 occurrences de carré :**  
branches\_mc,  
rayon1\_mc,  
rayon2\_mc

**18 occurrences de couleur\_btn**  
r1\_btn à r6\_btn  
v1\_btn à v6\_btn  
b1\_btn à b6\_btn

**Bouton**  
choixColor\_btn

Nombre de branches  
 Rayon1  
 Rayon2  
  

00 00 00

Choisir les nuances

●	●	●	00
●	●	●	33
●	●	●	66
●	●	●	99
●	●	●	CC
●	●	●	FF

R      V      B

Hexa.      Déci.

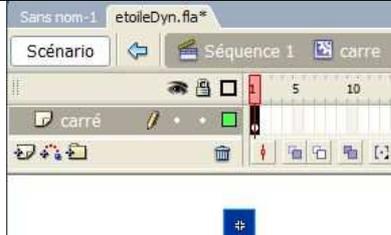
Valider après chaque choix, pour les codes (216 couleurs; 6<sup>3</sup>)

**3 textes dyna.**  
branches\_txt  
rayon1\_txt  
rayon2\_txt

**3 textes dyna**  
r\_txt ; v\_txt ;  
b\_txt

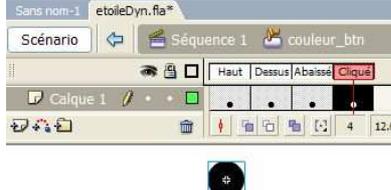
**2 textes dyna.**  
couleur\_txt  
codeD\_txt

#### 5\_ Le clip «carre».



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

#### 6\_ Le bouton couleur\_btn.



Avec Ctrl+F8, **créer** le bouton « couleur\_btn » (32 x 32).  
Même disque noir pour les 4 états.

#### 7\_ Le bouton flat blue play

**Aller** à Fenêtre/Bibliothèques communes/Boutons : clic droit sur le dossier playback.flat pour le développer et **choisir** flat blue play pour le **glisser** dans la bibliothèque.



#### 8\_ Le calque « curseurs ».

**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.

#### 9\_ Le calque « btColor ».

**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<sup>8</sup>, de 0 à 255, soit 256 nuances. Total des couleurs 256<sup>3</sup>, 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

Étudions les six nuances utilisées dans cet exercice pour 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  $16 \times 15$  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  $9 + 16 \times 9 = 153$

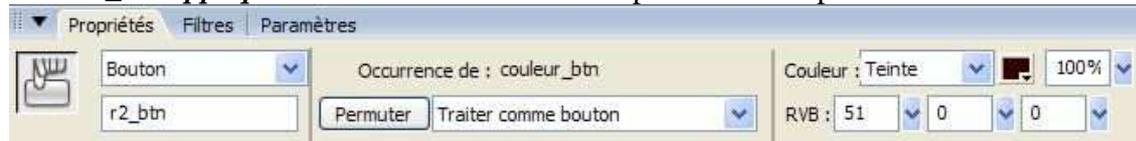
La nuance 66 correspond à  $6 + 16 \times 6 = 102$

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

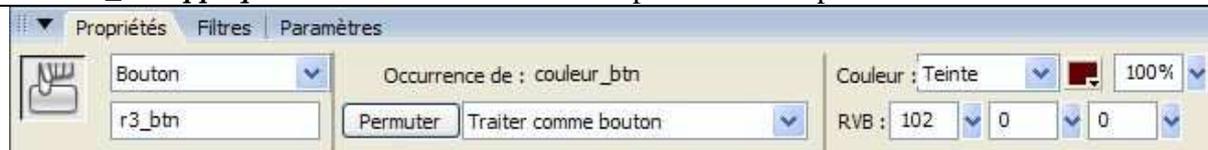
La nuance 00 correspond à 0.

**Insérer** un nouveau calque à renommer « btColor ». De la bibliothèque *glisser déposer* le bouton « couleur\_btn ». **Dupliquer** le 5 fois et **aligner** verticalement les six occurrences. De haut en bas les noms d'occurrences sont r1\_btn à r6\_btn ; r1\_btn reste noir.

Pour r2\_btn **appliquer** la teinte à choisir dans le panneau « Propriétés ».



Pour r3\_btn **appliquer** la teinte à choisir dans le panneau « Propriétés ».

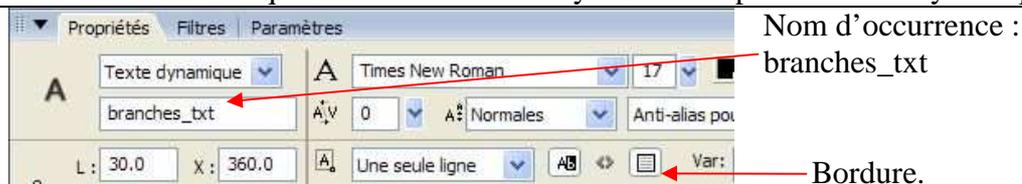


**Terminer** pour r4\_btn, r5\_btn et r6\_btn.

De la même façon **placer** les 6 occurrences v1\_btn à v6\_btn puis b1\_btn à b6\_btn et **appliquer** les teintes.

## 10\_ Le calque « textDyn ».

**Insérer** un nouveau calque à renommer « textDyn. **Créer** le premier texte dynamique.



**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.

## 11\_ Le calque « textStat ».

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

## 12\_ Le calque « bt flat ».

**Insérer** un nouveau calque à renommer « bt flat ». De la bibliothèque *glisser déposer* le bouton flat blue play. **Donner** lui le nom d'occurrence choixColor\_btn et **placer** le convenablement.

### 13\_ Le calque « AS ».

```

1 //etoileDyn 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 position y ordonnée fixe du curseur*/
9 function deplace(curseur,positiony){
10     //déclenchement movt 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 //moteur pour modifications Nombre de branches, Rayon1, Rayon2
26 _root.onEnterFrame=function(){
27 //Le nbre de branches varie de 0 à 100 et s'inscrit ds le texte dyn
28 branches_txt.text=n=Math.round((branches_mc._x-100)/2); //de 0 à 100
29 //Le rayon1 varie de 0 à 260 et s'inscrit ds le texte dyn
30 rayon1_txt.text=r1=Math.round((rayon1_mc._x-100)*1.3);
31 //r2 varie de 0 à 260 et s'inscrit ds le texte dyn
32 rayon2_txt.text=r2=Math.round((rayon2_mc._x-100)*1.3);
33 //*****
34 /*Création dynamique d'une étoile. Voir avant le dessin explicatif
35 montrant les 2 cercles de rayons r1 et r2, le nombre n de
36 branches et l'angle dt; c1 couleur trait, c2 couleur remplissage;
37 la propriété prototype permet d'affecter des méthodes à l'objet MovieClip*,
38 MovieClip.prototype.tracerEtoile=function(r1,r2,n,c1,c2){
39     var dt:Number=Math.PI/n;
40     this.lineStyle(1,c1,100); //épaisseur, couleur, alpha
41     this.moveTo(r2,0); //départ au point (r2,0)
42     this.beginFill(c2,100); //remplissage, couleur, alpha
43     for(var i:Number=1;i<2*n+1;i++){
44         var t:Number=dt*i;
45         var r:Number=(i%2==0) ? r2:r1; //i pair r=r2, i impair r=r1
46         var x:Number=r*Math.cos(t); //abscisse
47         var y:Number=r*Math.sin(t); //ordonnée
48         this.lineTo(x,y); //jonction
49     }
50     this.endFill(); //fin remplissage
51 }

```

```

52 //*****
53 var etoile:MovieClip=this.createEmptyMovieClip("etoile",0);
54 etoile._x=300;//étoile centrée à (300,350)
55 etoile._y=350;
56 etoile.tracerEtoile(r1,r2,n,"0x0000FF","0x"+(r+v+b).toUpperCase());
57 //écriture des codes hexa. pour r, v, b
58 r_txt.text=r; v_txt.text=v; b_txt.text=b;
59 }
60 //*****pour changement couleur remplissage
61 //déclaration et initialisation variables
62 var r:String="00";
63 var v:String="00";
64 var b:String="00";
65 //le rouge
66 r1_btn.onPress=function() {r="00";}
67 r2_btn.onPress=function() {r="33";}
68 r3_btn.onPress=function() {r="66";}
69 r4_btn.onPress=function() {r="99";}
70 r5_btn.onPress=function() {r="CC";}
71 r6_btn.onPress=function() {r="FF";}
72 //le vert
73 v1_btn.onPress=function() {v="00";}
74 v2_btn.onPress=function() {v="33";}
75 v3_btn.onPress=function() {v="66";}
76 v4_btn.onPress=function() {v="99";}
77 v5_btn.onPress=function() {v="CC";}
78 v6_btn.onPress=function() {v="FF";}
79 //le bleu
80 b1_btn.onPress=function() {b="00";}
81 b2_btn.onPress=function() {b="33";}
82 b3_btn.onPress=function() {b="66";}
83 b4_btn.onPress=function() {b="99";}
84 b5_btn.onPress=function() {b="CC";}
85 b6_btn.onPress=function() {b="FF";}
86 //afficher code couleur et le convertir
87 choixColor_btn.onRelease=function(){
88     //toUpperCase( ) fait passer en majuscules
89     couleur_txt.text=(r+v+b).toUpperCase();
90 var a:Number;
91 //parseInt( ) permet de définir la base (ici 16)
92 a=parseInt(r+v+b,16);
93 codeD_txt.text=a;
94 }

```