

COMPTE RENDU PROJET

Info3B (Synthèse d'Images)

Sommaire

Échiquier.....	p2
Pièces d'échecs.....	p2
-Pion (surfaces de révolution)	p2
-Fou (blobs)	p3
-Autres (C.S.G).....	p3/4/5
Animation.....	p6
-Fonctionnement général	p6
-1 ^{er} mouvement.....	p7
-2 ^{ème} mouvement.....	p7
-3 ^{ème} mouvement.....	p8

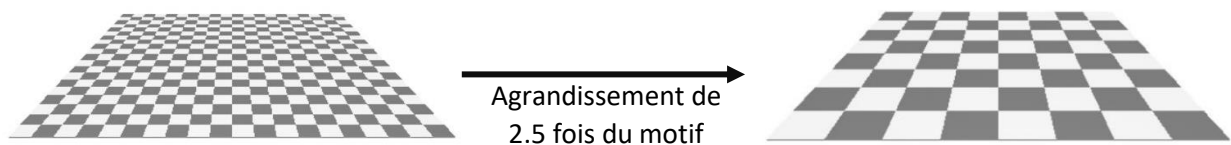
L'ÉCHIQUIER

- **Motif :**

Ici l'objectif était de représenter les 64 cases de l'échiquier avec le moins de code possible, pour ceci nous avons opté pour le motif «checker» permettant de faire directement un quadrillage sur un solide, avec un petit calcul nous pouvons lui appliqué un «scale» permettant d'obtenir 64 cases.

Calcul :

- Le «checker» applique des carrées de longueur et largeur 1
- Nous définissons notre échiquier pour que ses deux points opposés soient de coordonnées $[x=-10 ; z=-10]$ et $[x=10 ; z=10]$, donc nous obtenons de base un damier de $20*20$ cases
- Or nous voulons un damier de $8*8$ cases :
 $20/8=2.5$
- Il faut donc appliquer au motif «checker» une remise à l'échelle de 2.5, le motif est ainsi 2.5 fois plus grand et nous obtenons le nombre de cases voulues



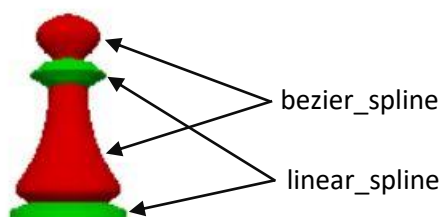
- **Couleurs :**

En ce qui concerne la couleur du plateau, nous avons opté pour un simple noir et gris, mais en rajoutant un petit effet de réflexion des objets environnent et d'éclat de la lumière, ce qui donne un effet entre métal et céramique.

LES PIÈCES D'ÉCHECS

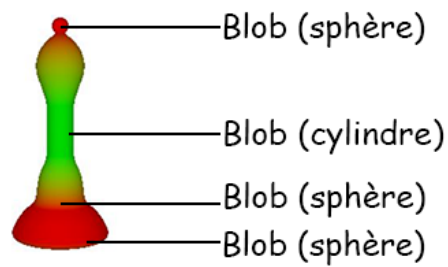
- **Pion :**

Pour le pion nous avons utilisé 4 surfaces de révolutions, dont deux «lathe» qui sont des «bezier_spline» et deux autres qui sont des «linear_spline», ainsi nous n'aurons pas des courbes parfaites sur tout le pion ce qui est plus cohérent avec le reste des pièces. Toutes ces surfaces de révolutions sont raccordées entre elles, ce qui donne le pion suivant :

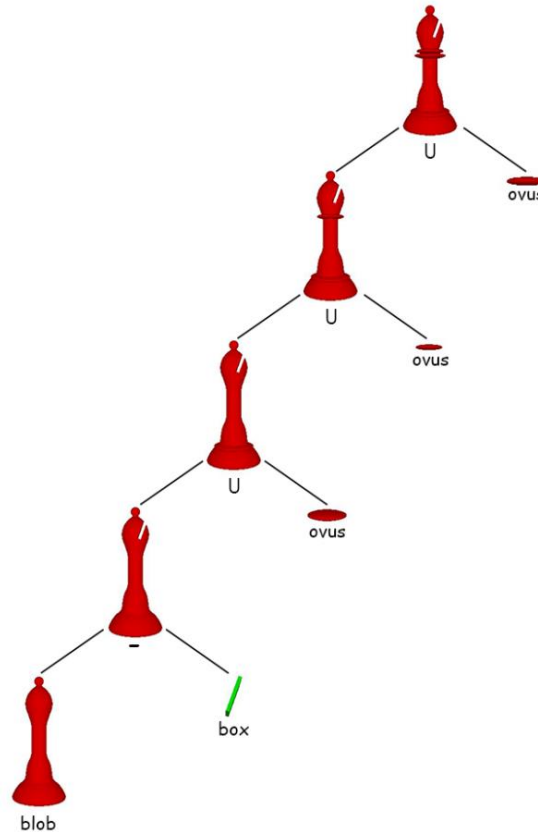


- **Fou :**

Pour le fou nous avons utilisé 4 blobs pour faire la base du pion :

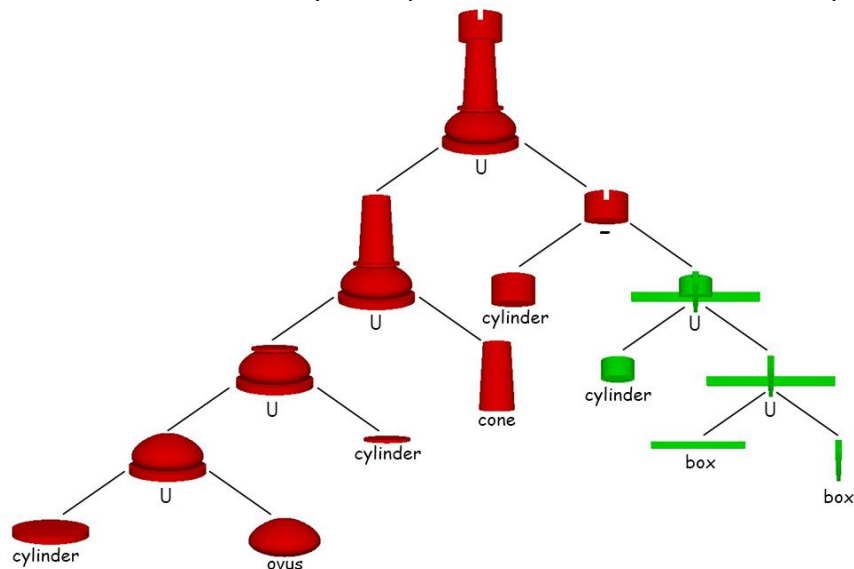


Ensuite, nous avons utilisé des unions ou des différences, pour donner plus de détails à notre pièce, voilà l'arbre C.S.G du fou :



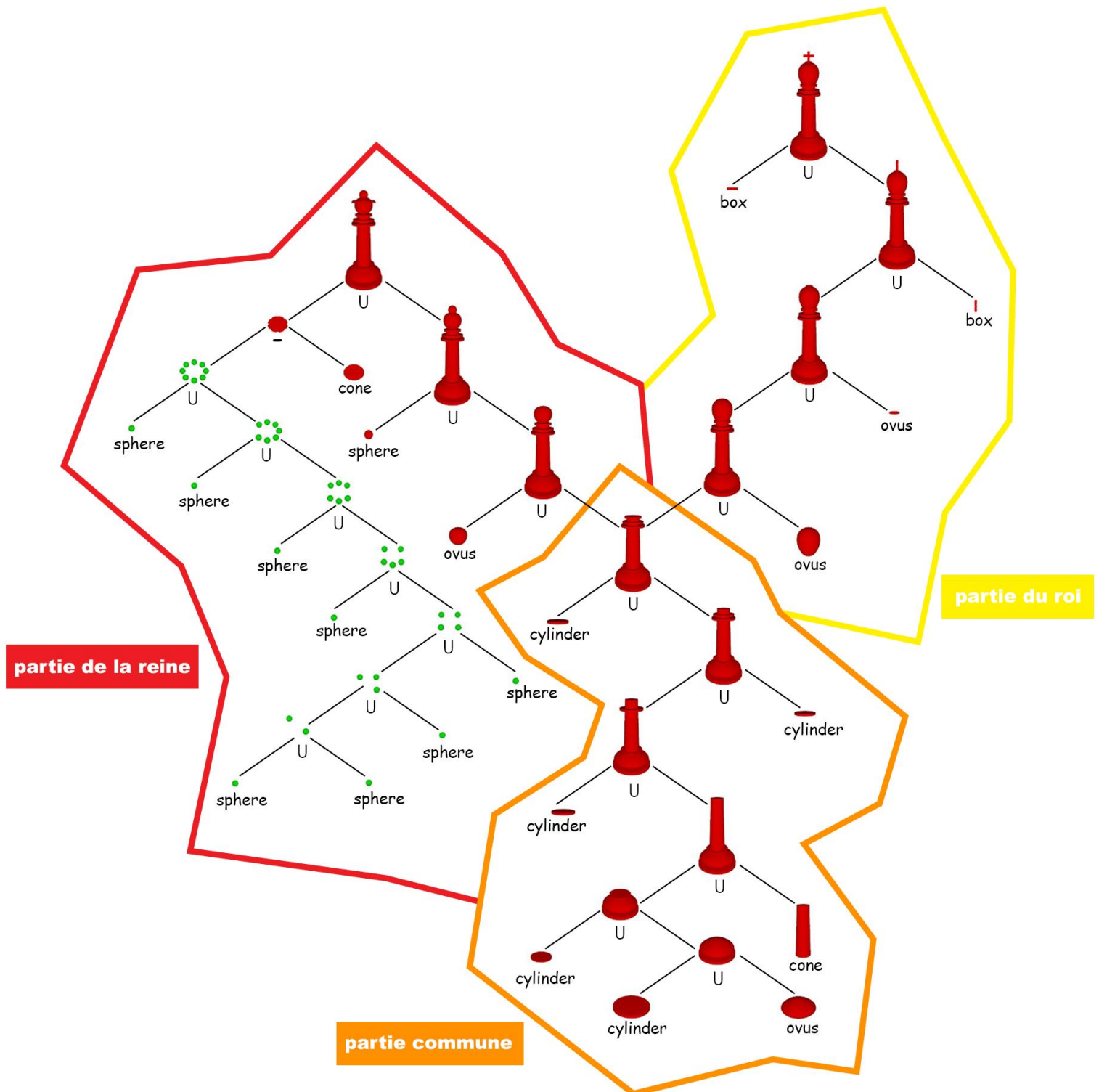
- **Tour :**

Pour les autres pions nous avons utilisé uniquement des formes de bases, et les avons associées grâce à des différences et des unions. La tour étant symétrique sur deux axes a été assez simple à réaliser, voilà son arbre C.S.G :



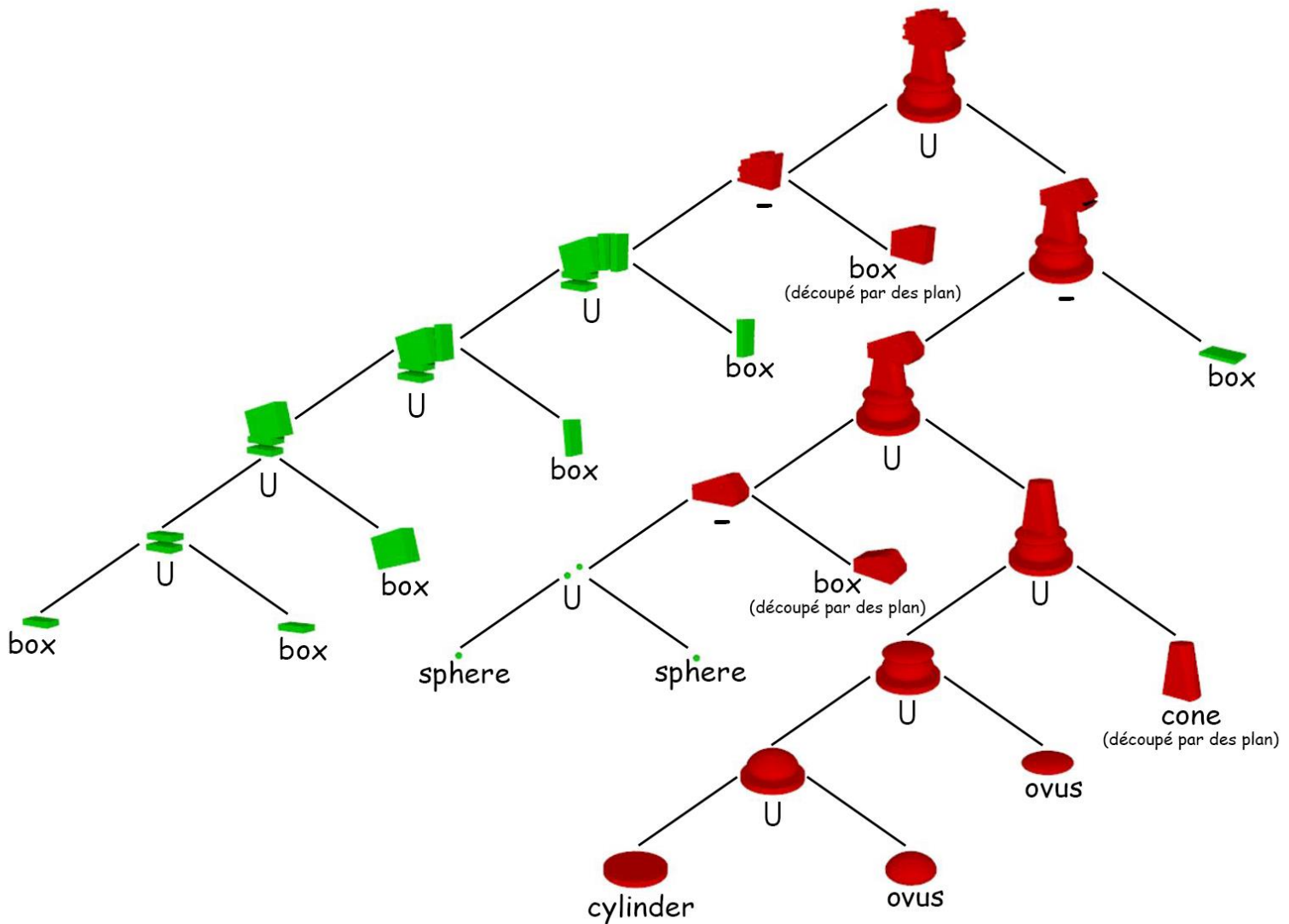
• Roi et Reine :

En ce qui concerne le roi et la reine, là aussi nous avons utilisé des formes simples que nous avons additionnées ou soustraites. Mais ici, nous avons décidé d'utiliser la même base pour ces deux pièces, ainsi seulement le haut de la pièce change. Voici donc l'arbre C.S.G qui regroupe la construction du roi et de la reine :



- **Cavalier :**

Le cavalier a été la pièce la plus compliqué à faire, cette pièce n'est pas centrée par rapport à son pied, et désigner un cheval n'est pas un exercice facile, nous avons réfléchi à des solutions pour représenter autrement le cavalier (style plus abstrait, autre signe caractéristique, ...), or ceci nous menait à un cavalier non cohérent avec le reste des pièces. Donc nous avons quand même représenté un cheval, ceci uniquement avec des formes simples, voilà l'arbre C.S.G de cette pièce (à noter que représenter des plans dans l'arbre n'était pas pratique, les pièces qui ont subi une ou des différences avec des plans sont directement représentées découpées) :



- **Couleurs :**

Pour les couleurs des pièces, nous avons, contrairement au plateau, appliqué une texture de marbre, et enlever la réflexion de couleur sur la moitié de celles-ci pour qu'elles apparaissent noir, les 2 textures des pièces donnent ainsi l'effet de «marbre blanc» et «marbre noir». Ainsi les couleurs du plateau et des pièces vont bien ensembles.

L'ANIMATION

• Fonctionnement général :

Pour l'animation, nous utilisons un fichier «.ini», qui va nous aider à générer une suite d'images que nous pourrons par la suite associer en GIF. Dans ce dossier il y a le nombre d'image que nous voulons, et une «clock», que nous avons définie pour aller de 0 à 1. En utilisant la «clock» dans les coordonnées de nos pièces, nous pouvons ainsi les faire évoluer puisque la «clock» elle-même évolue dans le temps.

Nous avons vu grossièrement le fonctionnement d'une animation en POV-Ray, je vais maintenant vous présenter nos choix personnels pour celle-ci. Nous voulions utiliser un seul fichier «.ini» et une seule «clock» pour faire tous nos mouvements, il a donc fallu gérer les coordonnées de nos pions avec la «clock» mais aussi sans la «clock» puisque nos pions ne bougent pas tous en même temps et tout le long de l'animation, pour faire ceci nous avons donc utilisé des conditions dépendantes de la «clock». Par exemple : “si la clock est à un stade inférieur à 0.10, ma pièce reste fixée, en revanche une fois que la clock passe à 0.10 ma pièce commence à se déplacer”.

Pour nos mouvements, nous n'avons pas bougé nos pièces au hasard, nous avons fait commencer une partie agressive où le fou va mettre en échec le roi adverse :

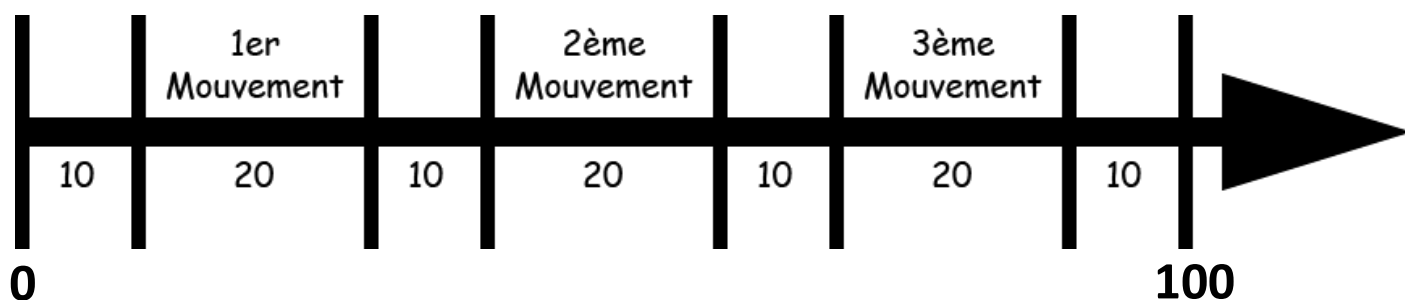
T	P					P	T
C	P					P	C
F	P					P	F
Ro	P				P	P	Ro
Re	P		P			P	Re
F	P					P	F
C	P		F			P	C
T	P					P	T

1^{er} Mouvement

2^{ème} Mouvement

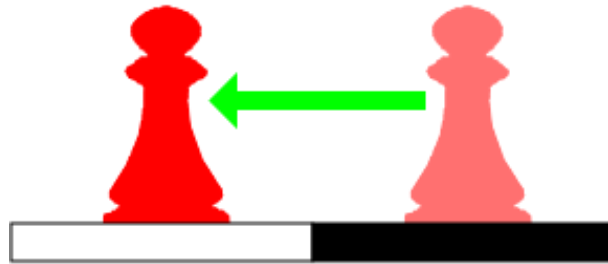
3^{ème} Mouvement

Nous avons également fait le choix de répartir nos mouvements sans trop les enchaîner, le rythme de l'animation est représenté ci-dessous, où les valeurs représentent le pourcentage du statut de la «clock», 0 étant la première image, et 100 étant la dernière image :



- **1^{er} Mouvement :**

Le premier mouvement est le mouvement rectiligne, nous l'avons réalisé en déplaçant un pion de 1 case, le pion se déplace donc de manière linéaire sans que sa vitesse varie :



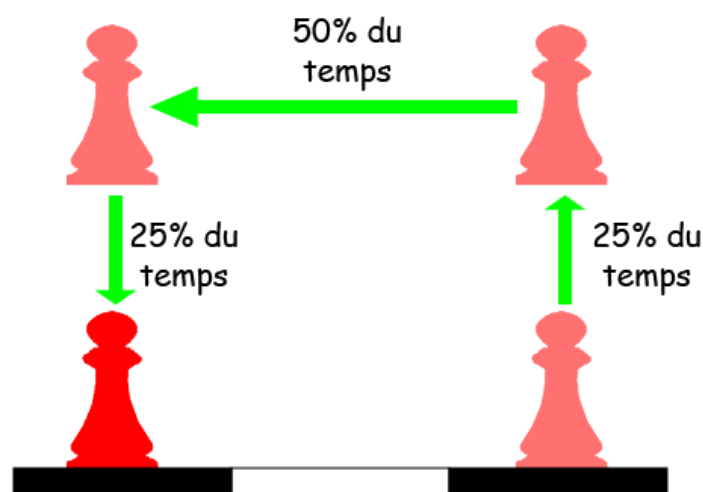
- **2^{ème} Mouvement :**

Le deuxième mouvement est le mouvement vertical, pour celui-ci nous avons fait déplacer le pion adverse de deux places, mouvement possible lorsque le pion est encore à sa place initiale.

Ce mouvement peut être lui-même divisé en trois mouvements, soit la montée, l'avancement, et la descente.

Pour faire ceci, nous avons encore utilisé des conditions, de la même façon que l'on gère s'il y a mouvement ou non, ainsi il y a 3 conditions juste pour ce mouvement (de X1 temps à X2 temps il y a le premier mouvement vertical, ensuite de X3 temps à X4 temps il y a le mouvement horizontal, de X5 temps à X6 temps il y a le dernier mouvement vertical, puis enfin le pion se stabilise à sa place finale).

Nous avons choisi que la montée et la descente utilise chacune 25% du temps total du mouvement de ce pion, et que le mouvement horizontal en utilise 50%, cela permet de garder une vitesse assez homogène sachant que la distance horizontale est plus grande que celles verticales :



- **3^{ème} Mouvement :**

Le troisième mouvement est le mouvement en forme d'arc de parabole. Ici nous avons décidé de déplacer le fou qui va mettre en échec le roi adverse (mouvement en diagonal non représenté sur le schéma ci-dessous). Pour faire un arc de parabole, nous avons utilisé une courbe de Bézier, ceci demandant une formule assez longue, nous avons décidé de faire des «macro» au tout début du code, que nous avons ensuite utilisé plus facilement dans la translation de la pièce.

