

## Compte-rendu projet

### Sommaire :

I / Échiquier

II / Pion

III / Fou

IV / Tour

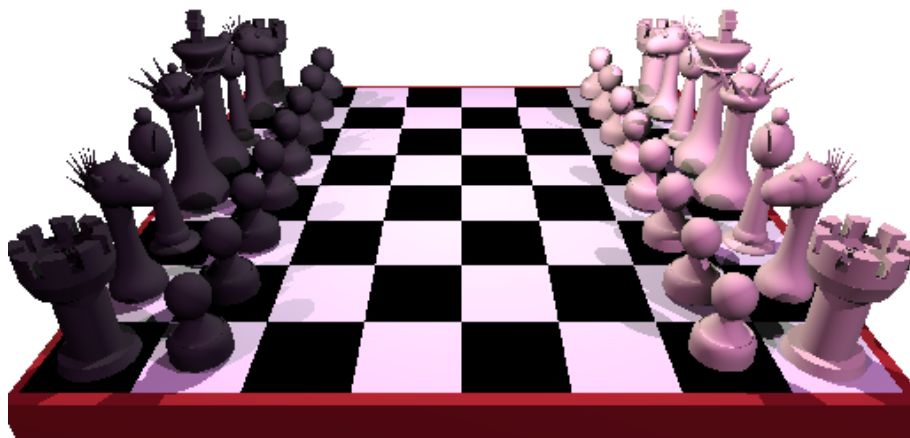
V / Roi

VI / Reine

VII / Cavalier

VIII / Placement des pièces

IX / Animation



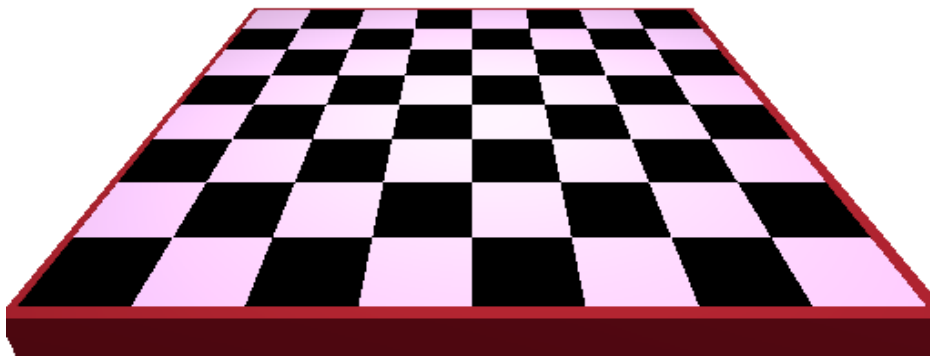
## I / Échiquier

L'objectif premier de l'échiquier est d'avoir 64 cases avec le moins de code possible.

Dans un premier temps j'ai créé une box qui fait du 8x8 avec comme coordonnées point le premier point  $x = y = 4$  et pour le second point  $x = y = -4$  avec une hauteur allant de 0 à 0.5, je lui ajoute ensuite deux couleur (blanc et noir) et j'utilise la texture « checker » pour que les couleurs alternent toute les 1x1 cases.

Ainsi la base de l'échiquier est créé, afin de cacher les faces sur le coté je décide de créer une autre box de couleur marron, et pour ce faire je lui donne une taille un petit plus grande que la première box,  $p1 : x = y = 4.15$  et  $p2 : x = y = -4.15$  avec une hauteur allant de 0 à 0.49 pour éviter les bugs de textures.

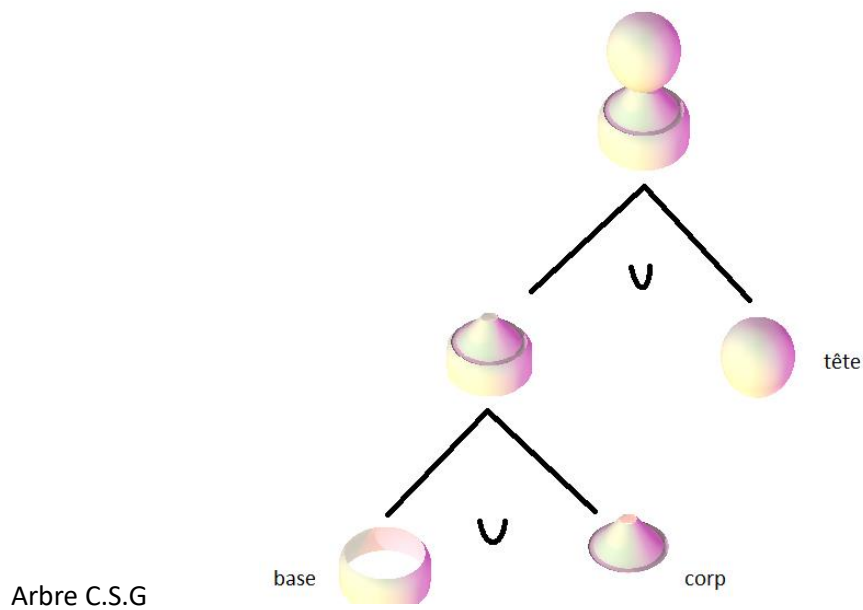
Mon échiquier est donc désormais terminé. II



## II / Pion

Pour faire mon pion j'utilise une macro que je nomme pion avec comme paramètres les coordonnées et la couleur du pion. Afin de lui donner forme, j'utilise 3 courbes de Bézier que je vais unifier.

Je commence donc par la base, ensuite le corp puis finalement la tête.



### III / Fou

Comme imposé, le fou est composé de blob, 4 au total composés de cylindres et sphères le tout unifié

Blob 1 : dessus de tête (cylindre + sphère)

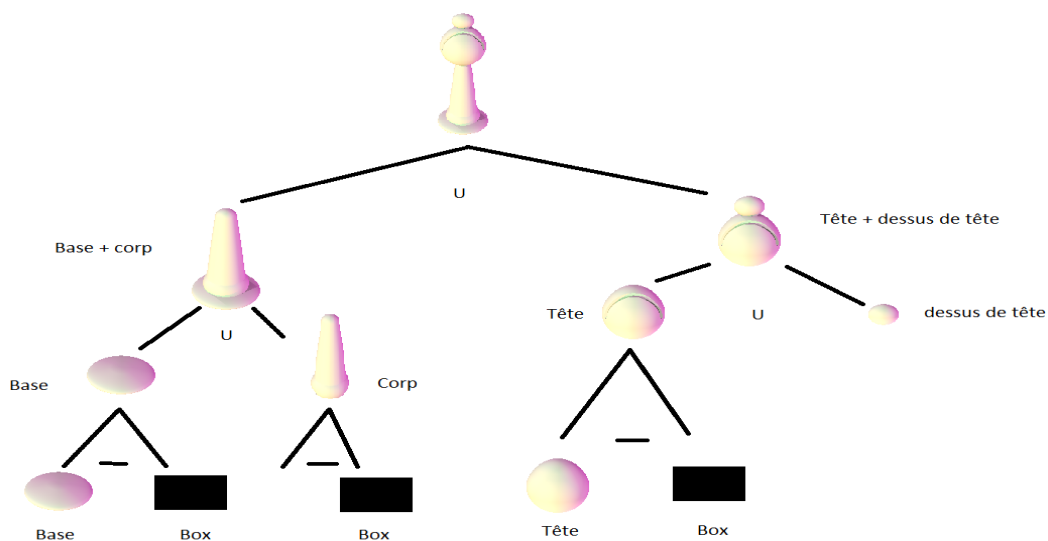
Blob 2 : tête (cylindre + sphère et soustraction d'une box pour créer une « entaille »)

Blob 3 : corp (cylindre + sphère et soustraction d'une box pour retirer l'excédent qui dépasse sous l'échiquier)

Blob 4 : base (cylindre + sphère et soustraction d'une box pour retirer l'excédent qui dépasse sous l'échiquier)

La seule difficulté à été ici la rotation pour le fou noir, par rapport à l'entaille.

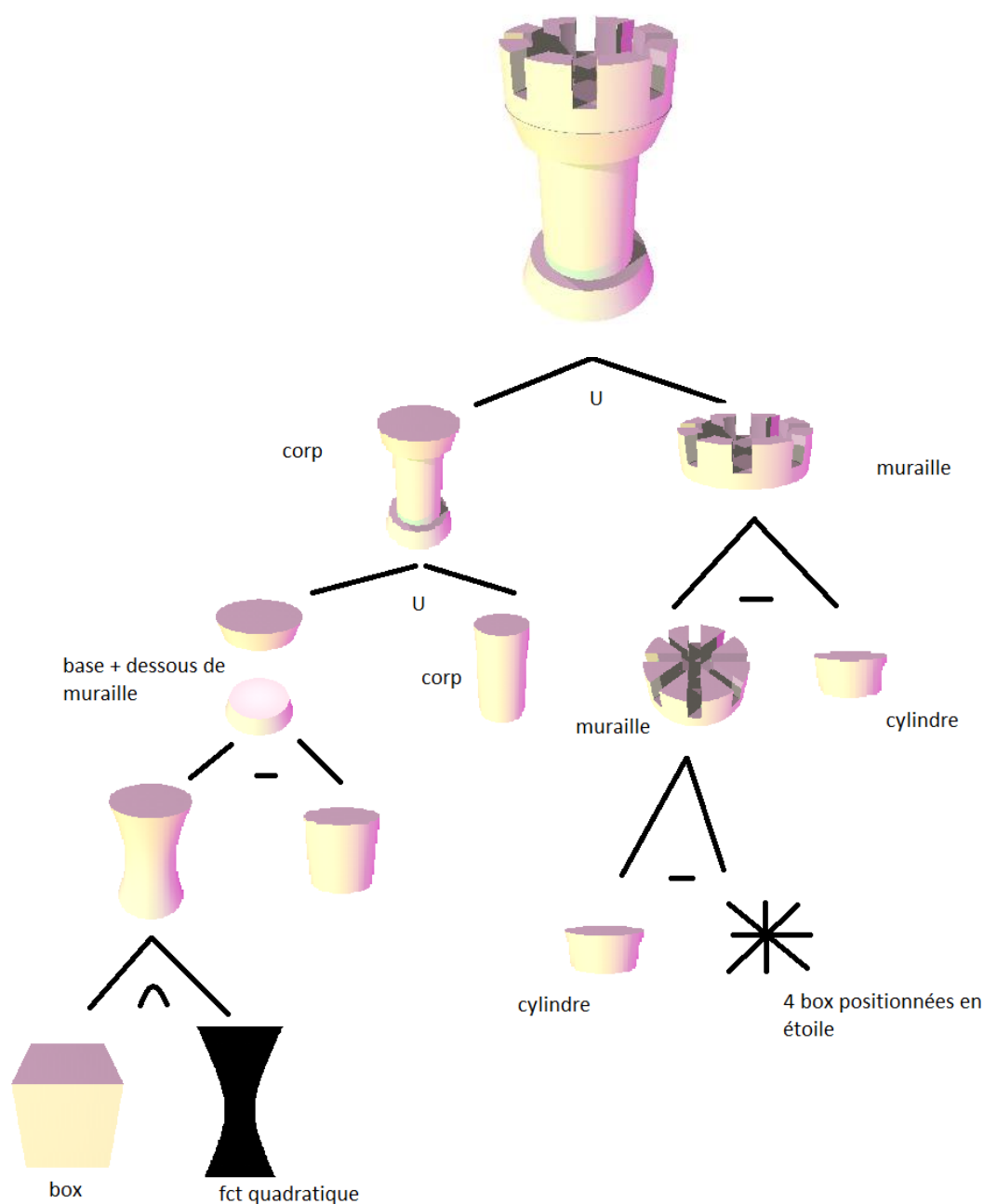
Arbre C.S.G



## IV / Tour

Pour la base de la tour j'ai utilisé une box à laquelle j'ai ajouté une fonction quadratique pour amincir les cotés ( comme dans le cour ) et je lui ai soustrait un cylindre au milieu de la box pour ne garder que le haut et le bas de la box, j'ai donc rajouté un cylindre pour faire le corp du pion et pour la muraille j'ai créé une macro permettant de générer des box avec lesquelles je forme une étoile et je la soustrait à un cylindre, je soustrait encore un plus petit cylindre pour créer l'espace en cercle.

Arbre C.S.G



## V / Roi

Mon roi a été créé à partir de 3 courbes de Bézier le tout unifié

1<sup>ère</sup> courbe : le corp

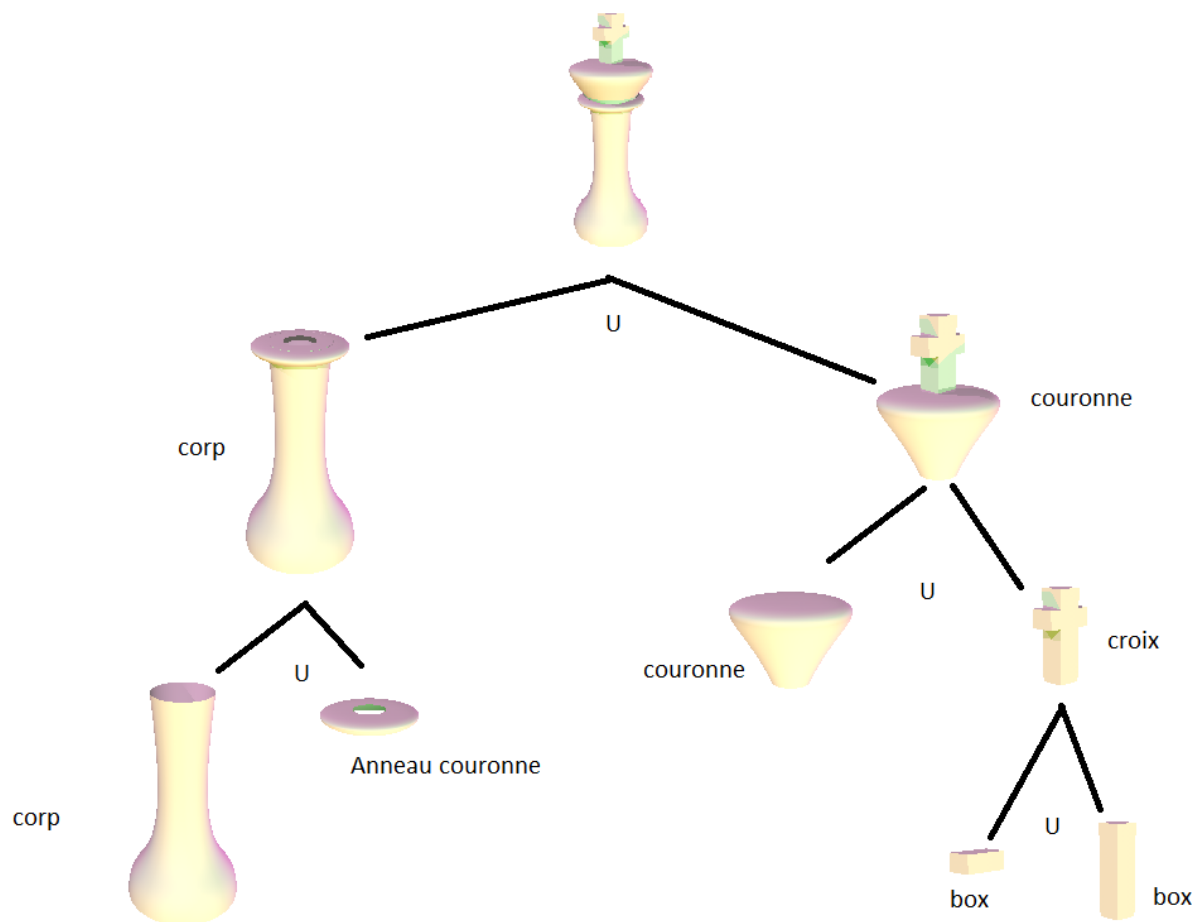
2<sup>ème</sup> courbe : l'anneau de la couronne

3<sup>ème</sup> courbe : forme de la couronne

J'utilise ensuite deux box pour faire la croix sur la couronne.

La seule difficulté a été de tourner le roi pour que la croix soit dans le bon sens.

Arbre C.S.G



## VI / Reine

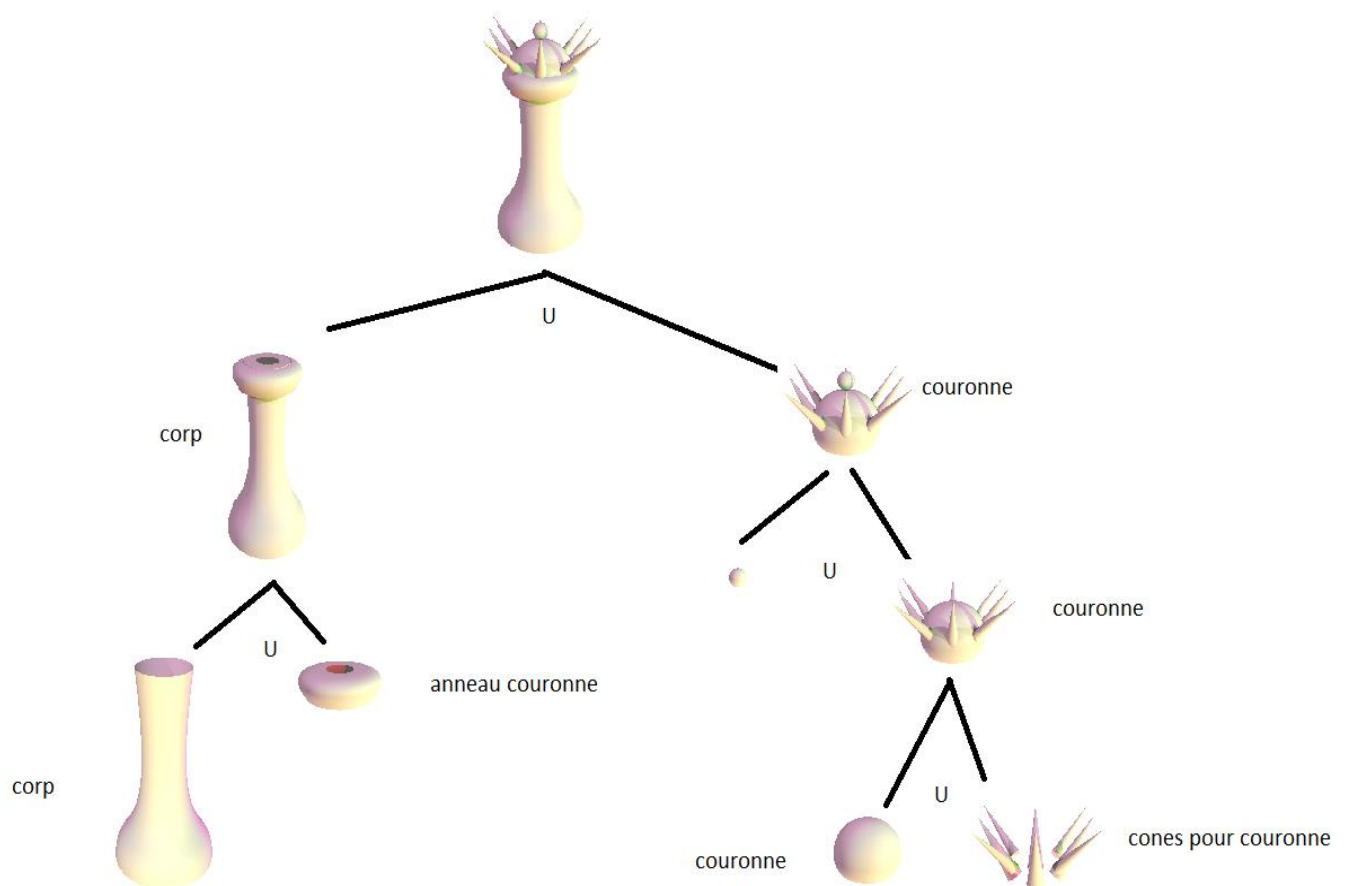
Ma reine a été créée à partir de 3 courbes de Bézier tout comme le roi le tout unifié.

1<sup>ère</sup> courbe : la base étant la même

2<sup>ème</sup> courbe : l'anneau de couronne est plus gros

3<sup>ème</sup> courbe : la couronne est arrondie cette fois-ci.

Afin de donner forme à la couronne j'ai ajouté tout autour d'elle des cones pour qu'elle soit reconnaissable, j'ai ajouté une sphère par-dessus la couronne.



## VII / Cavalier

Pour cette pièce on reprend la même corp que le roi et la reine la pièce se compose de :

Corp : courbe de Bézier

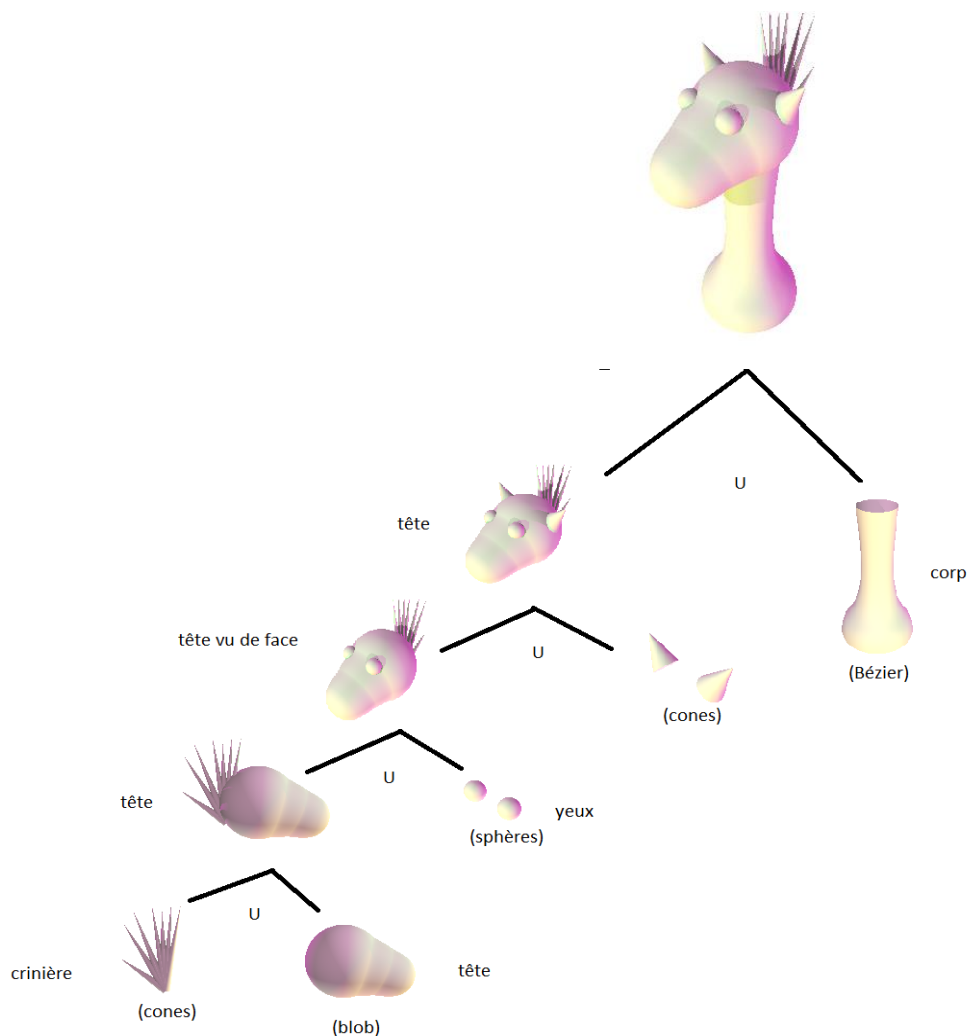
Tête : Blob à 3 sphères, les trois sphères sont cotes à cotes, avec une sphère moins grande que la précédente afin de donner une forme de « cône » afin de faire le museau du cavalier

Oreilles : 2 cônes sur les côtés de la tête

Yeux : 2 sphères positionnées à l'avant de la tête

Crinière : 7 cônes à l'arrière de la tête qui pointent vers l'arrière à l'avant

La difficulté ici à été d'imaginer une crinière pouvant être faite avec des box mais qui nécessitait une rotation précise ainsi que des coordonnées propres à elles même, j'ai donc changé en optant pour des cônes, il faut aussi changer la rotation du cavalier pour les pions noir et blanc pour pointer dans la bonne direction.



## VIII / Placement des pièces

Etant donné que chaque pièces sont faites à partir de macro, nous allons donner les coordonnées que l'on veut à telle pièce à partir des paramètres.

Pour ce faire j'ai créé une variable pour chaque emplacement de pièce en suivant les coordonnées de base d'un échiquier

C'est-à-dire A1,B1...



et étant donné que le centre de mon échiquier est aux coordonnées (0,0)

le centre de chaque case donne (x,5 ; y,5) et donc x et y varient de -3 à 3, 0 inclus.

Par exemple A1 = (3,5 ; 3,5), une fois toute les variables remplies , elles sont appelées dans les paramètres lors de la création de la pièce et ainsi la pièce sera générée au bon endroit

Cependant il peut être plus compliqué de placer des pièces à sens unique comme le Fou ou le Cavalier , une rotation suffira à tourner la pièce en question, tout ça intégré en paramètres.



## IX / Animation

Pour débiter l'animation il nous faut un fichier pov ray en .ini auquel on vas associer notre modélisation de l'échiquier.

Afin de faire nos animations, je l'ai initialisé à 50 images et l'horloge à 50 tours, l'horloge vas être la base de notre animation car elle va permettre de se repérer dans le « temps » et d'enchaîner les tours sans qu'ils s'animent tous en même temps, la valeur de l'horloge vas aussi permettre aux pions d'avancer puisque la valeur augmente à chaque image

Dans un premier temps on prend une valeur A et B qui définissent le début et la fin de notre animation, dans ce même laps de temps on doit faire en sorte que la pièce se lève, se déplace et redescend.

Pour se faire j'utilise des conditions **#if** qui permettent de donner un ordre durant un laps de temps, étant donné que l'animation d'une pièce se fait en trois temps, on a donc trois **#if**.

On va donc donner des conditions comme ceci,

Si clock > 1 , lever la pièce (change les coordonnées de la variable de la pièce)

Si clock >=2 et si clock <= 4, déplacer la pièce ( la pièce ne s'arrêteras pas avant que la clock ai dépassée 4)

Si clock > 4 , poser la pièce ( j'ai indiqué la position moi-même pour que la pièce soit bien centré sur la case)

Pour le déplacement, on additionne ou on soustrait à x ou y la clock à laquelle on soustrait ou additionne la valeur du début de la condition

Ex : Si clock >=2 et si clock <= 4,

$A1 = x + (\text{clock} - 2), y, z$

Ou

$A1 = x - (\text{clock} + 2), y, z$

Si on veut un déplacement en diagonale, on utilise la clock sur x et y en même temps ( pour les pions par ex).

Si on veut un déplacement à l'horizontal, on utilise la clock sur x ou y mais un à la fois ( pour le fou )

Vien ensuite le déplacement à l'aide d'une courbe

J'ai décidé de faire ce déplacement à l'un de mes cavalier blanc, pour ce faire je veux dans un premier temps faire « 2 » courbe pour faire un déplacement en deux temps et créer un rebond.

**déplacement**



**direction**



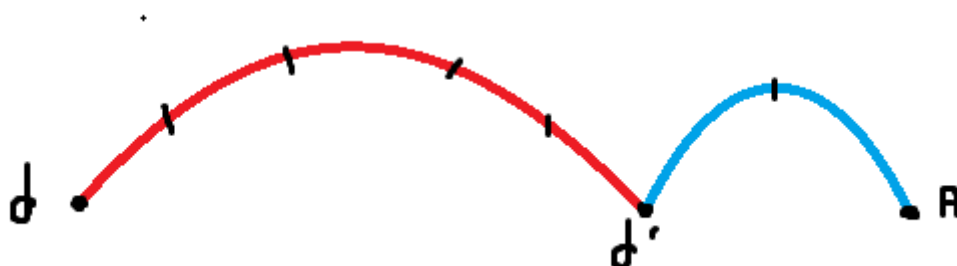
comme ceci

Pour ce faire un pose déjà plusieurs point,

Le premier point qui est la position de base de notre cavalier

Le dernier qui sera la position à la fin de notre animation

Ma courbe se présente ainsi



Ces deux courbes sont bien dans une seule et même courbe, les bâtons représentent mes points et les points noir représentent le départ, le premier arrêt et l'arriver.

J'utilise comme avant, une condition **#if** afin de démarrer l'animation et vu que à une valeur de la clock et fait en sorte que la clock se déplacer sur la courbe et que à chaque image les coordonnées soit transmises à mon cavalier.