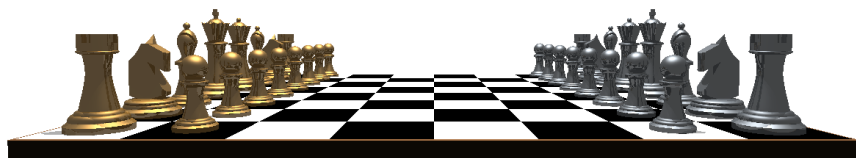


L2 : projet Info3B, année 2020-2021

Agustín Cartaya

jeu d'échecs



Organisation du projet

Le projet est divisé en plusieurs modules chargés de faire des calculs ou de fournir des informations aux scènes, il y a 15 fichiers qui seront expliqués ci-dessous:

main.sh

Si Linux est le système d'exploitation, ce fichier automatise la tâche de création des animations et de convertir les images résultantes en fichiers .gif et .mp4
3 paramètres sont nécessaires (nom de la scène; durée de la scène en (sec); paramètre booléen 0 ou 1)

pieces.inc:

Contient la modélisation de toutes les pièces du jeu

table.inc, tableau.inc:

Contiennent la modélisation de la table et de l'échiquier

macros.inc:

Contient toutes les macros nécessaires pour réaliser les animations et positionner les éléments

initialisation.inc:

Initialise tous les objets de la scène et les stocke dans des constantes

scene01.pov, scene02.pov ... scene08.pov, images.pov:

Contiennent différentes scènes et animations

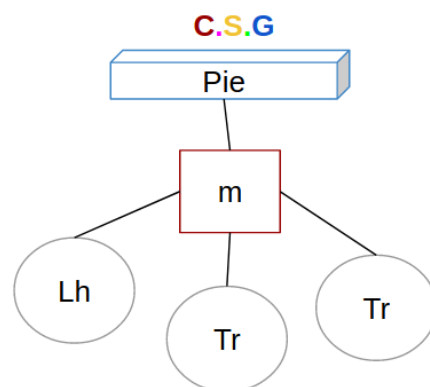
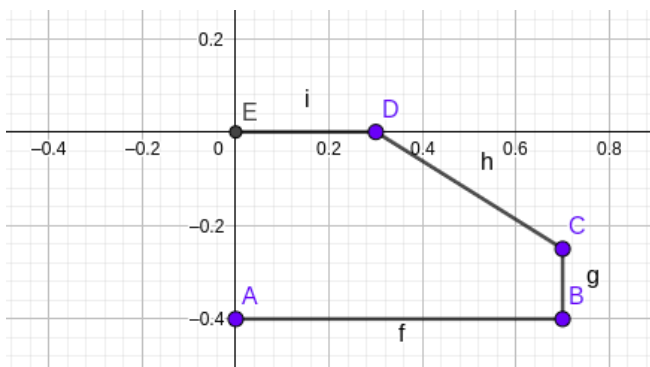
Création de pièces

Les pièces sont principalement constituées de l'union d'un **pie**, d'une **base** et d'une **tête** où cette dernière varie en fonction de la pièce créée et les deux premières peuvent ou non varier. Les diagrammes CSG de chaque pièce seront montrés, quelques abréviations sont utilisées dans les diagrammes (voir glossaire à la fin du rapport)

Pie.

Formé par la fusion (m) de 1 "Lathe" (Lh) et 2 "Torus"(Tr).

- Le "Lathe" utilisé est de type "linear_spline" et ses points ont été choisis pour créer la figure suivante dans le plan XY, (pie.ggb)



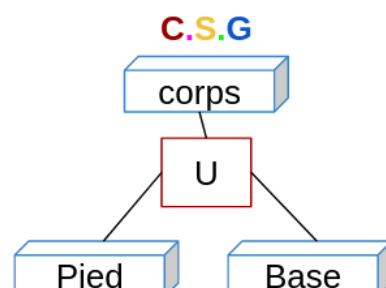
Base.

Formé par 1 "Lathe" (Lh) de type "cubic_spline" avec 7 points de contrôle choisis pour créer le corps suivant.



corps.

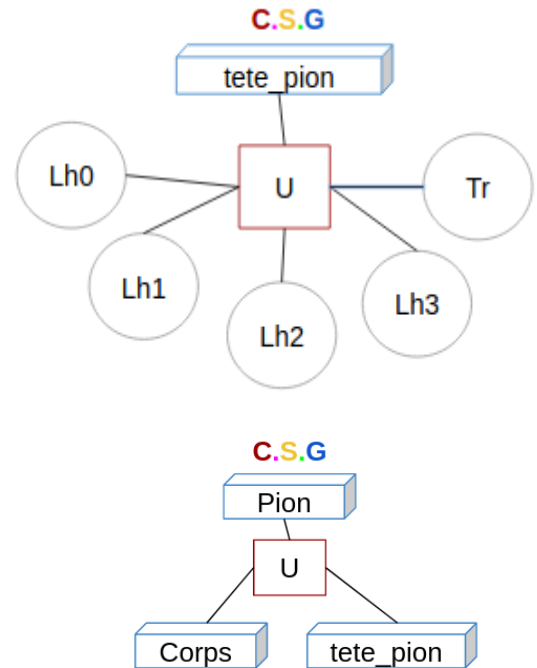
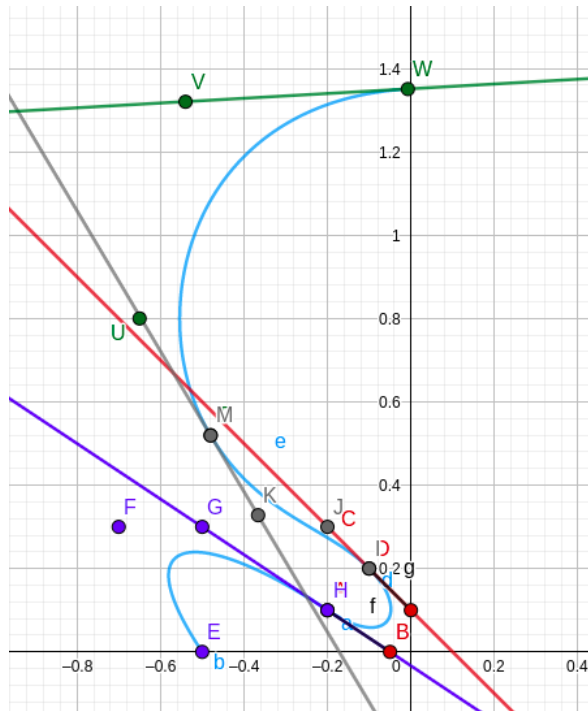
Formé par l'union (u) du pie et la base.



Pion.

- *Tête du pion.*

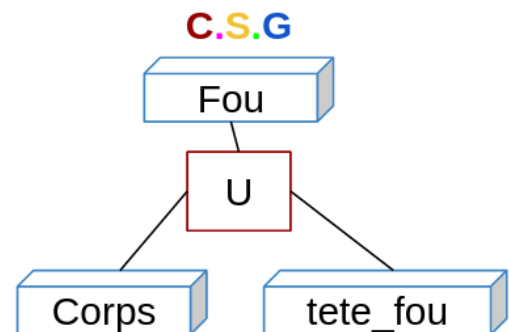
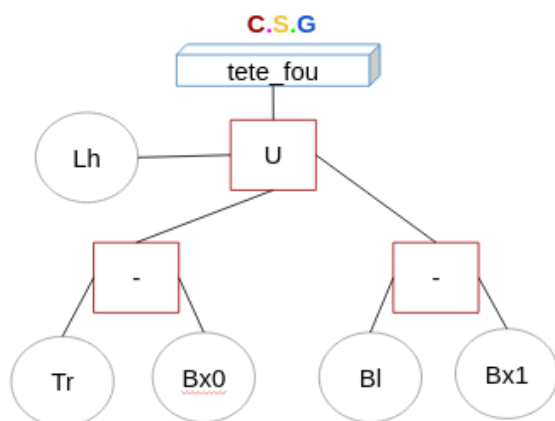
Formé par l'union (U) de 4 "lathe" (Lh) et 1 "torus" (Tr). les "lathes" son de type "bezier_spline" et es points de ceux-ci sont alignés de manière à ce que les 4 aient un raccord de type G1 (tete_pion.ggb)



Fou.

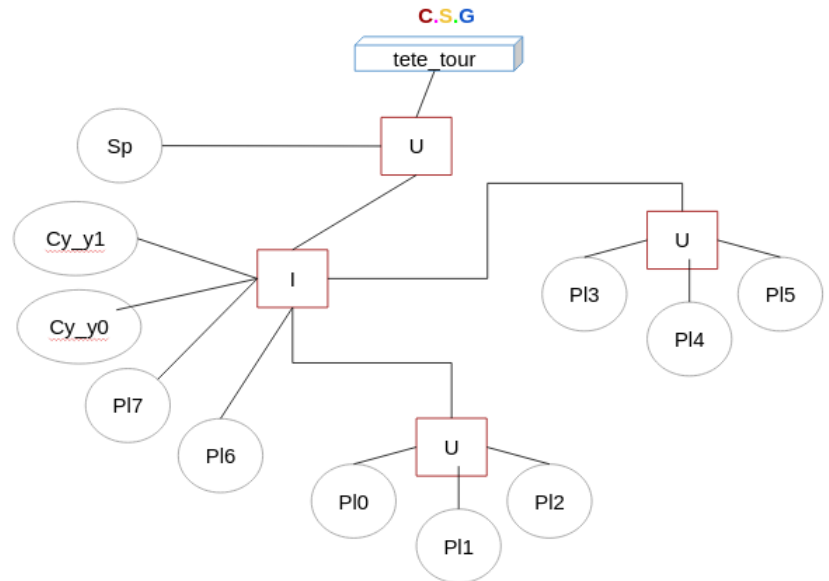
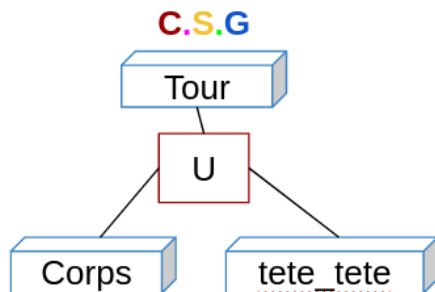
- *Tête du fou.*

Formé par l'union (U) de 1 "lathe" la différence de 1 "torus" y 1 "box" et la différence de 1 "blob" y 1 "box". Le "lathe" est de type "bezier_spline", et le "blob" est formé par 1 "cylinder" et 1 "sphere"



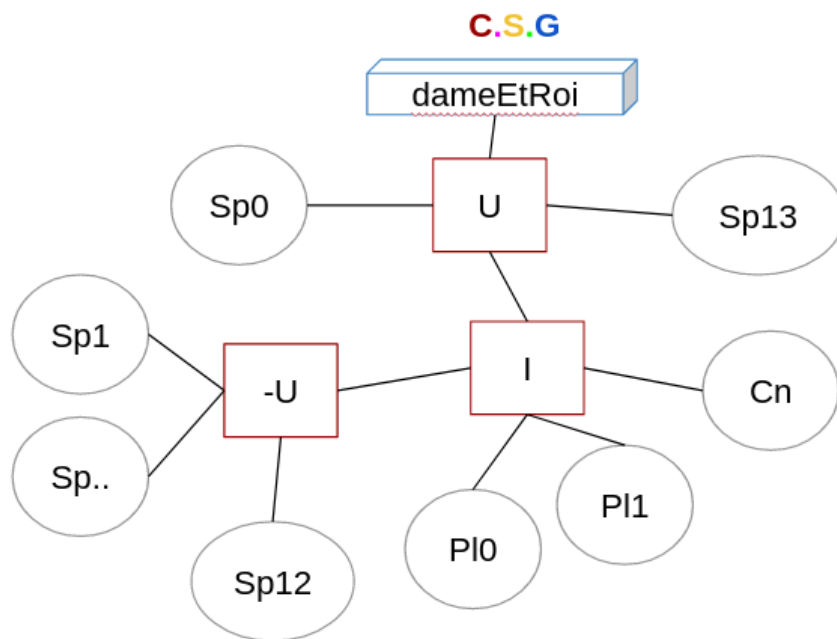
Tour

- tête de la tour.
 - 1 "sphere"
 - 2 "cylinder y"
 - 8 "plane"



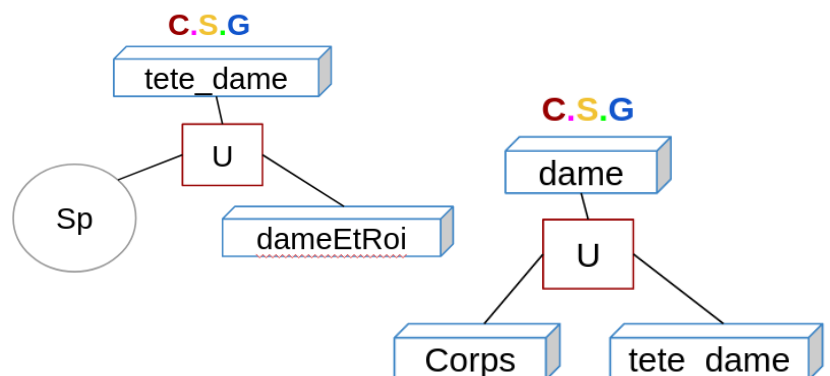
Dame et Roi

- 14 "sphere"
- 2 "plane"
- 1 "cone"



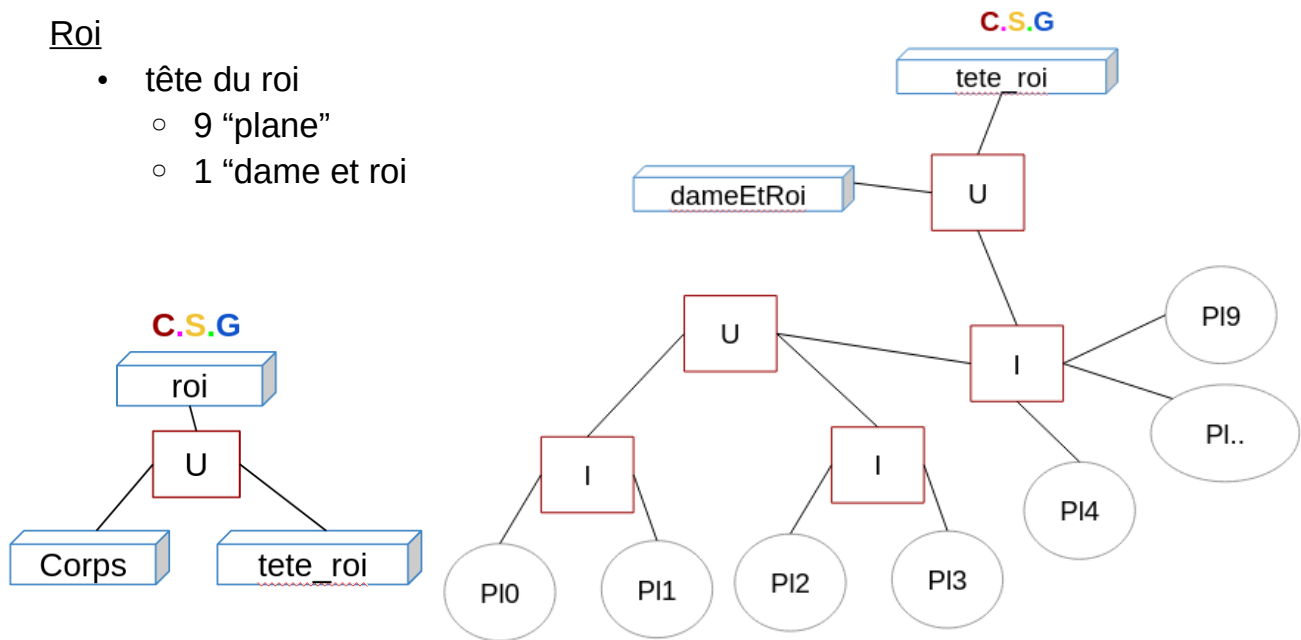
Dame

- tête de la dame
 - 1 "sphere"
 - 1 "dameEtRoi"



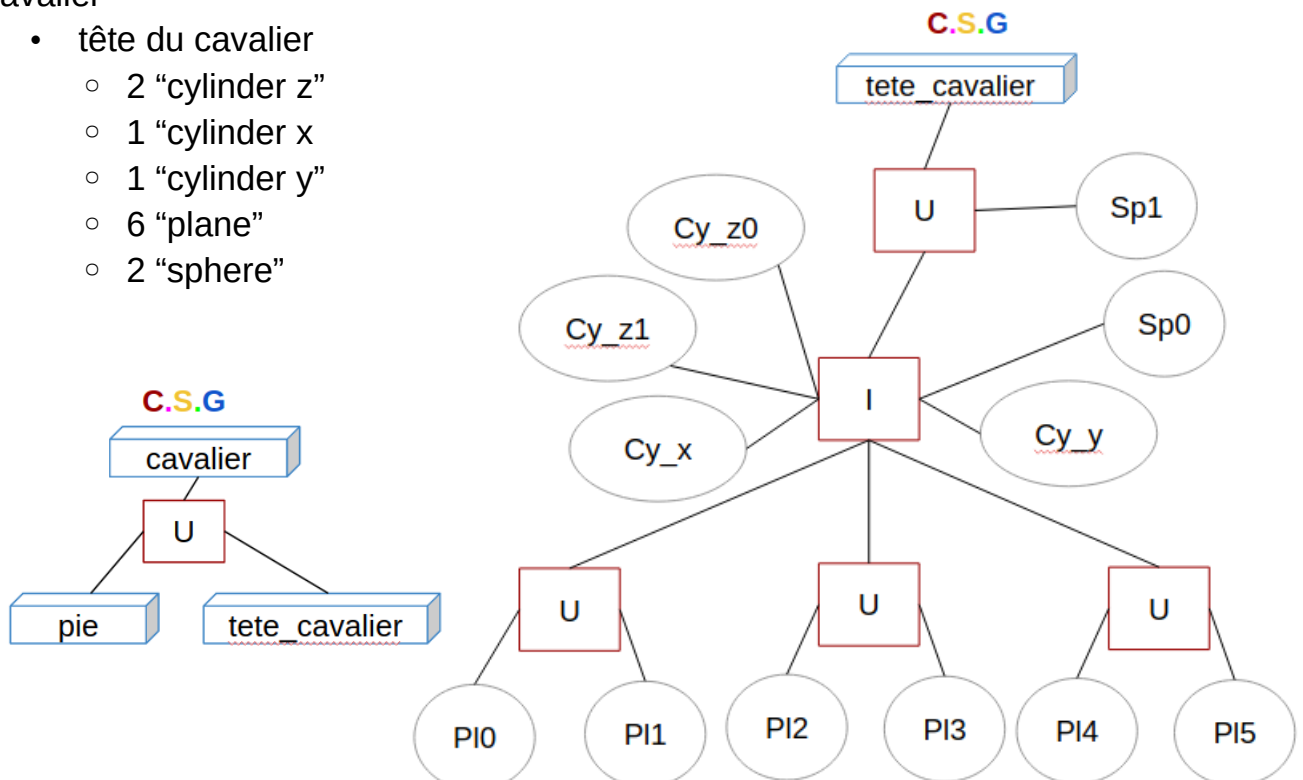
Roi

- tête du roi
 - 9 "plane"
 - 1 "dame et roi"



cavalier

- tête du cavalier
 - 2 "cylinder z"
 - 1 "cylinder x"
 - 1 "cylinder y"
 - 6 "plane"
 - 2 "sphere"



Affichage des pièces

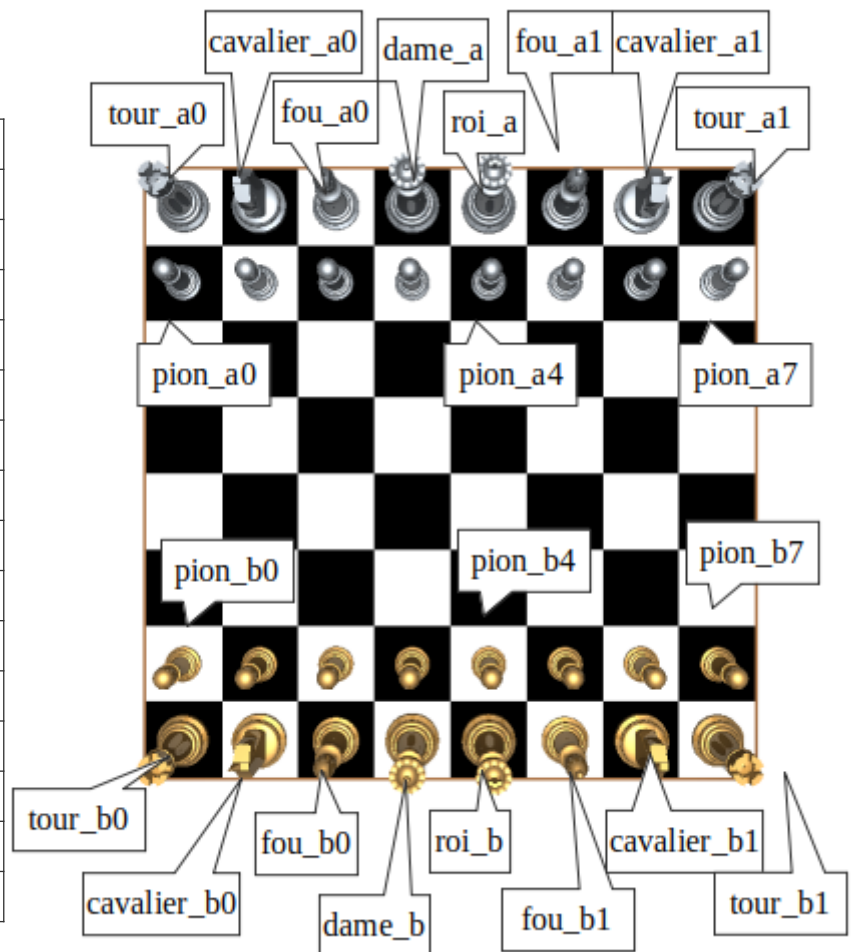


Positionnement des pièces

Pour le positionnement des pièces, un array bidimensionnel avec les positions des pièces est nécessaire, l'array doit être rempli de la manière suivante:

Array[16][2]

tour_a0	tour_b0
cavalier_a0	cavalier_b0
fou_a0	fou_b0
dame_a	dame_b
roi_a	roi_b
fou_a1	fou_b1
cavalier_a1	cavalier_b1
tour_a1	tour_b1
pion_a0	pion_b0
pion_a1	pion_b1
pion_a2	pion_b2
pion_a3	pion_b3
pion_a4	pion_b4
pion_a5	pion_b5
pion_a6	pion_b6
pion_a7	pion_b7



Dans chaque case de l'array il doit y avoir un vecteur avec la position de la pièce, la position peut s'écrire de deux manières "Position relative" ou "Position absolue"

- **Position Relative (RP)**

Ce type de positionnement prend l'échiquier et ses case comme référence, en partant de la case [0; 0] jusqu'à la [7; 7], dans l'exemple précédent la pièce "pion_b0" est située dans la case [1; 0] qui sera écrit en vecteur <1,0,0> (la position z par défaut est 0), il est possible de positionner une pièce à l'extérieur de la carte comme c'est le cas de la position [8,0,0] (ce défaut est utilisé pour retirer plus facilement les pièces de l'échiquier)

- **Position Absolue (AP)**

Ce positionnement est le positionnement par défaut de pov-ray en prenant l'origine comme référence, dans ce cas la pièce peut être placée n'importe où dans l'espace, par exemple si une pièce doit être placée à l'origine DE LA SCÈNE, il faut donner le vecteur <0,0,0> comme position de la pièce

Dans le document « initialisation.inc » les positions par défaut des pièces sur le plateau (**RP**) seront trouvées dans une constante appelée « DEFAULT_POSITIONS »

les fonctions "placement_pieces_tableau" et "placement_pieces_ext" s'occupent de placer les pièces avec les positions RP et AP respectivement, ces fonctions se retrouvent dans le document "macros.inc".

Note.

Il existe une troisième fonction appelée "placement_pieces_table" qui positionne les pièces sur la table

La fonction "calc_positions_tableau" prend comme paramètre un array avec des positions **RP** et calcule les positions **AP** respectives

Animations

Le projet est composé de 8 scènes, dont 3 sont des animations de caméra et 5 sont des animations des pièces.

scene01

Cette scène fait un mouvement de caméra où la position de la caméra suit une trajectoire de type spirale, en utilisant l'équation polaire de la spirale d'Archimède, et le look_at effectue un mouvement linéaire.

Le point de départ de la caméra et du look_at est respectivement "PC_RIGHT_UP" et "PCK_RIGHT_UP" (voir initialisation.inc) et le point d'arrivée est "PC_RIGHT_EXT" et "PCK_RIGHT_EXT"

Note

Les calculs de jeu pour trouver les constantes a et b ont été faits à la main et vérifiés à l'aide du programme Octave, le fichier de vérification est joint au projet et se trouve dans "octave/archimedean_spiral.m"

scene02

Cette scène déplace l'ensemble de pièces de la table au plateau de jeu avec un mouvement parabolique utilisant des courbes de Bézier de degré 2, la position initiale des pièces (point de contrôle P0) est celle obtenue avec la macro "calc_position_table", la position finale des pièces (point de contrôle P2) est obtenue avec la macro "calc_position_tableau" en passant par paramètre les positions par défaut ("DEFAULT_POSITIONS") et la position intermédiaire (point de contrôle P1) est obtenue en calculant la moyenne entre P0 et P2 et en augmentant la composante z d'un facteur $k = -10$

scene03

Effectue un mouvement parabolique de la caméra en utilisant une courbe de Bézier ou les points de contrôle sont les suivants :

P0 = "PC_RIGHT_EXT"

P1 = (<15, -3, 8>)

P2 = "PC_UP"

scene04

Effectue un mouvement linéaire (vers l'avant) de la pièce "pion_b3" de la position <3,1,0> (**RP**) à la position <3,3,0> (**RP**)

scene05

Effectue un mouvement linéaire (vers l'avant*) de la pièce "pion_a7" de la position $\langle 7,6,0 \rangle$ (**RP**) à la position $\langle 7,5,0 \rangle$ (**RP**)

scene06

Effectue 2 mouvements linéaires

- Un mouvement diagonal (vers l'avant) de la pièce «fou_b0» de la position $\langle 2,0,0 \rangle$ (**RP**) à la position $\langle 7,5,0 \rangle$ (**RP**).
- Lorsque la pièce «fou_b0» est suffisamment proche de la pièce «pion_a7», cette dernière effectuera un déplacement de sa case actuelle $\langle 7,5,0 \rangle$ vers la droite $\langle 8,5,0 \rangle$ simulant une sortie de jeu

scene07

Cette scène fait un mouvement de caméra en 2 parties, en utilisant 2 courbes de Bézier,

- pour le premier mouvement, les points de contrôle de la courbe sont:
P0 = "PC_UP"
P1 = $\langle 18, 0, 13 \rangle$
P2 = "PC_RIGHT"
- pour le deuxième mouvement, les points de contrôle de la courbe sont:
P0 = "PC_RIGHT"
P1 = $\langle 15, 24, -2 \rangle$
P2 = "PC_BACK"

scene08

Cette scène simule le mouvement de la pièce "cavalier_a0" qui est composée de 4 mouvements linéaires

1. mouvement linéaire en z vers le haut
La pièce se déplace de sa position initiale $\langle 1,7,0 \rangle$ 3 unités vers le haut $\langle 1,7,3 \rangle$
2. mouvement linéaire en y
La pièce se déplace de sa position actuelle $\langle 1,7,3 \rangle$ 2 unités en avant $\langle 1,5,3 \rangle$
3. mouvement linéaire en x
La pièce se déplace de sa position actuelle $\langle 1,5,3 \rangle$ 1 unité vers la droite $\langle 2,5,3 \rangle$
4. mouvement linéaire en z vers le bas
La pièce se déplace de sa position actuelle $\langle 2,5,3 \rangle$ 3 unités vers le bas $\langle 1,7,0 \rangle$

Problèmes trouvés.

- Les objets Blob ont un comportement un peu compliqué à comprendre, donc construire la tête du fou n'a pas été très facile.
- certaines courbes (cubiq_spline) étaient difficiles à manipuler, quelques points supplémentaires ont dû être ajoutés pour obtenir le résultat souhaité.

Glossaire CSG:

- m: merge
- U: union
- I: intersection
- -: difference
- Lh: lathe
- Tr: torus
- Bx: box
- Sp: sphere
- Cy: cylinder
- Cy_y: cylinder y
- Cy_x: cylinder x
- Cy_z: cylinder z
- Pl: plane
- Cn: cone