Projet de conception et programmation orientées objet
Polytech Clermont-Ferrand :
Tetris

Lucas DUCHESNE, Valentin GIBOULOT
Encadré par M. BOUET et C. DEVAULX

Année universitaire 2012/2013

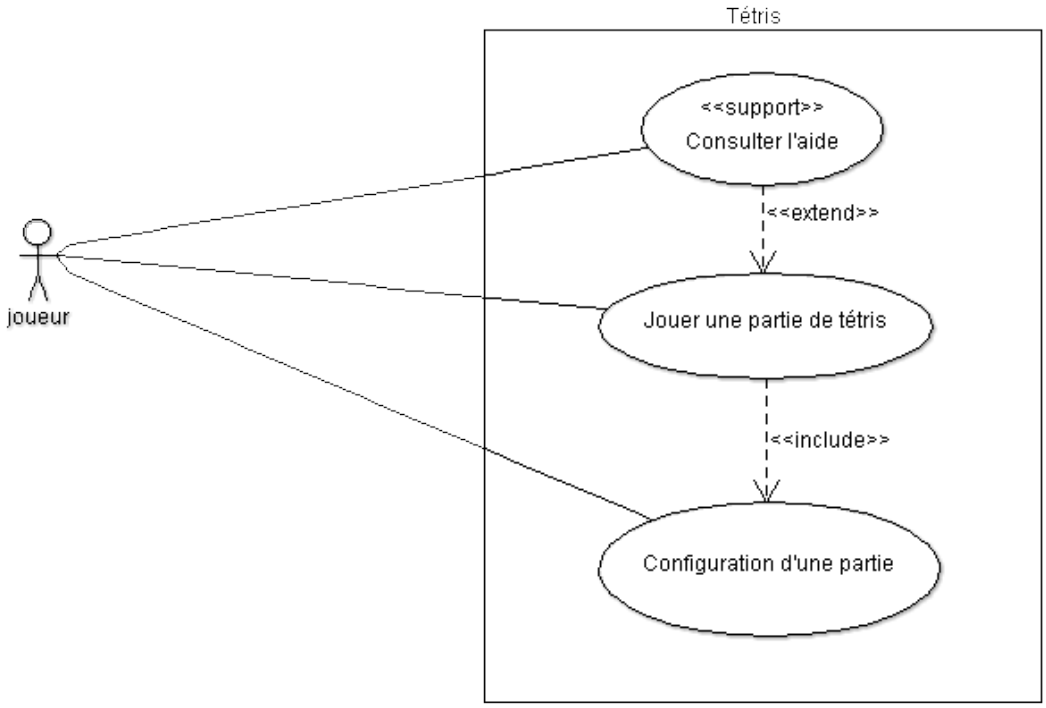
I – Introduction

-Principe du jeu :
Des pièces nommées des Tétrominos descendent du haut de l'écran les unes après les autres. Le joueur ne peut pas stopper la chute d'une pièce. En revanche, il peut soit la déplacer latéralement au moyen des flèches ← et →, soit la tourner de 90° au moyen de la flèche ↑.
Le joueur a pour objectif de compléter un maximum de lignes horizontales. Lorsqu'une ligne est complétée (c'est-à-dire sans aucune case vide), elle disparaît et les blocs supérieurs tombent à leur tour. Si le joueur ne fait pas disparaître assez rapidement les lignes, l'écran se remplit jusqu'en haut, le joueur ne peut plus jouer et il perd. Le jeu se termine donc toujours sur une défaite du joueur.
En vue d'atteindre un bon score, le joueur a tout intérêt à compléter un maximum de lignes horizontales en même temps si possible. En effet, compléter une seule ligne rapporte 40 points, alors qu'en compléter 2 en rapporte 100, 3 lignes en rapporte 300 et 4 (le maximum atteignable) en rapporte 1200. De plus, le nombre de points augmente à chaque niveau selon l'équation $F(p,n) = p(n+1)$ où p représente le nombre de points au niveau 0 et n le niveau.

Pour créer nos diagrammes, nous avons décortiqué le problème, et, nous avons réfléchi au déroulement d'une partie, tout particulièrement au cycle entre la chute de deux Tétrominos .

II – Diagrammes UML

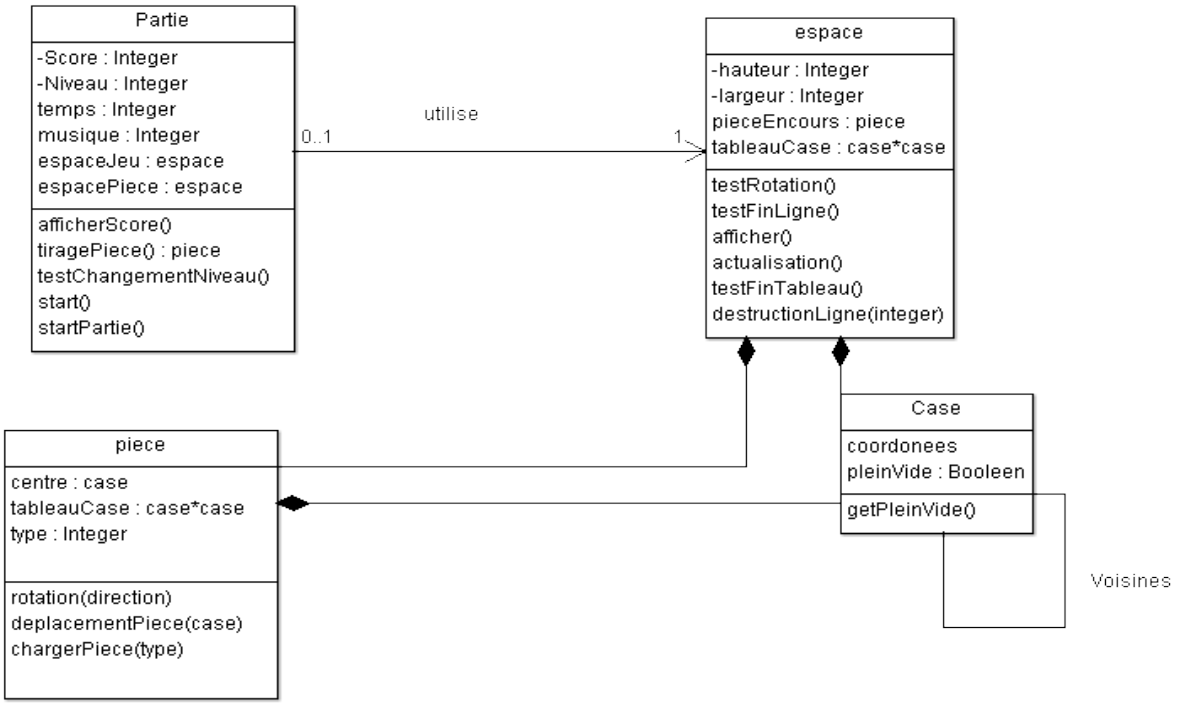
1) Les cas d’utilisation :



Explications :

-Pour lancer une partie, il est nécessaire de passer par le menu des options. On peut régler la musique et la difficulté initiale. Il est également possible de lancer directement et de laisser les réglages par défaut.

2) Diagrammes de classes



Explications :

-La classe « Espace »

Un espace est un ensemble de case. Il est caractérisé par un hauteur et une largeur (10*22 pour l'espace de jeu). On utilise cette classe pour l'espace de jeu, mais aussi pour le petit espace affichant la pièce suivante qui est dans l'attribut *pieceEncours*. Cet attribut permet de savoir quelle est la pièce qui tombe dans l'espace de jeu. La méthode *testRotation* permet de savoir savoir si il est possible de tourner une pièce ou si elle est gênée par un mur ou une autre pièce. *testFinLigne* vérifie si il y a des lignes entières. *testFinTableau* regarde si il y a une pièce trop haute ce qui entrainerait la défaite.

-La classe « Case»

Une classe est caractérisé par ses attributs coordonnées et si elle est occupé par une case ou non.

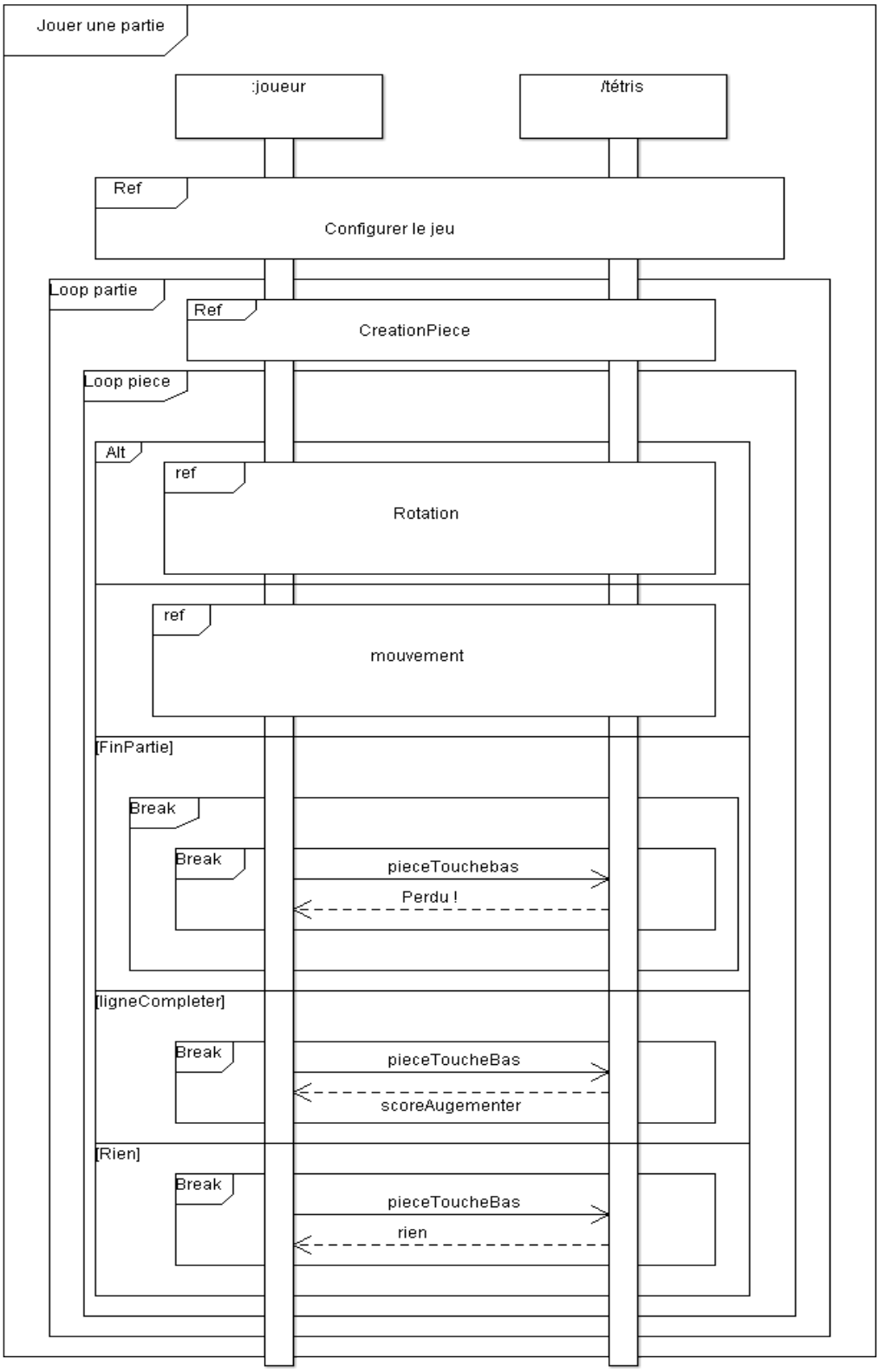
-La classe « Pièce »

Une pièce est caractérisé par un son *centre* et l'ensemble de ses cases contenues dans *tableauCase*. Le centre permet d'effectuer les rotations. On stock le *type* de la pièce(carré, L, barre ...). On pourrait déduire le type de la pièce a partir de son ensemble de case et de son centre, mais notamment pour savoir de quelle couleur afficher les cases, il est mieux d'avoir un attribut supplémentaire. chargerPiece permet d 'aller chercher une pièce en ayant son type. Les pièces et leur types seront défini dans un fichier.

-La classe Partie « Partie »

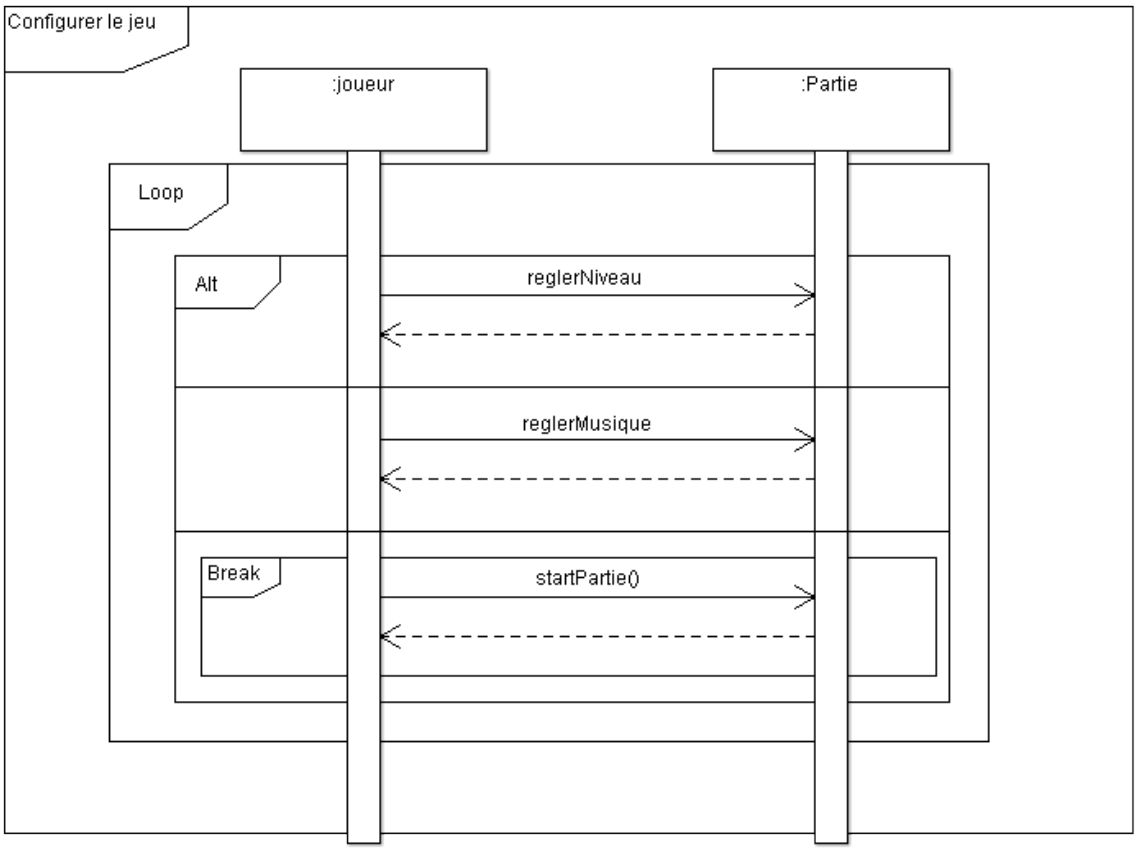
Une partie contient l'ensemble des élément nécessaire pour jouer.

3) Les diagrammes de séquence :


Jouer une partie
:joueur
/tétris
Ref
Configurer le jeu
Loop partie
Ref
CreationPiece
Loop piece
Alt
ref
Rotation
ref
mouvement
[FinPartie]
Break
Break
pieceTouchebas
Perdu !
[ligneCompleter]
Break
pieceToucheBas
scoreAugementer
[Rien]
Break
pieceToucheBas
rien

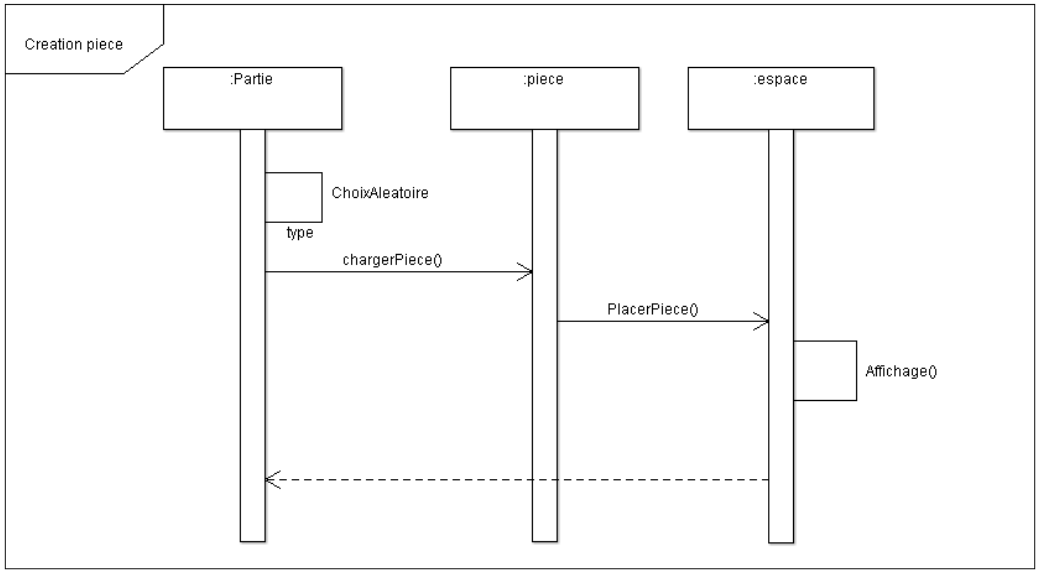
Explications :

Ce diagramme décrit le déroulement général d'une partie. On commence par configurer la partie. Ensuite on va rentrer dans deux boucles imbriquer. La première boucle partie, qui consiste a créer une pièce. A l'intérieur de cette boucle, on entre dans la boucle pièce qui permet de gérer la pièce dans l'espace de jeu pendant sa chute. On vérifie à chaque instant si la pièce quand la pièce touche le fond. On a alors trois possibilités soit on complète une ou plusieurs lignes et il faut les détruires, soit on est au dessus de la limite supérieure et c'est la défaite, soit rien.



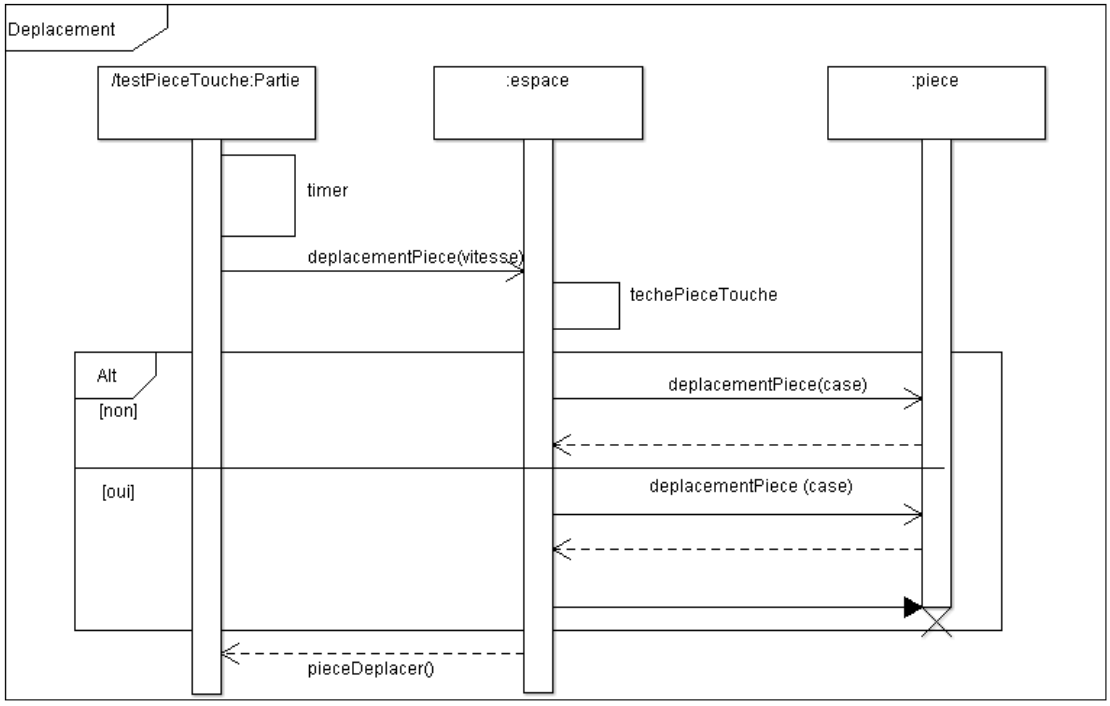
Explications :

Ce diagramme explique le fonctionne du menu pour régler les options sur la musique et le niveau de difficulté initial avant la partie.



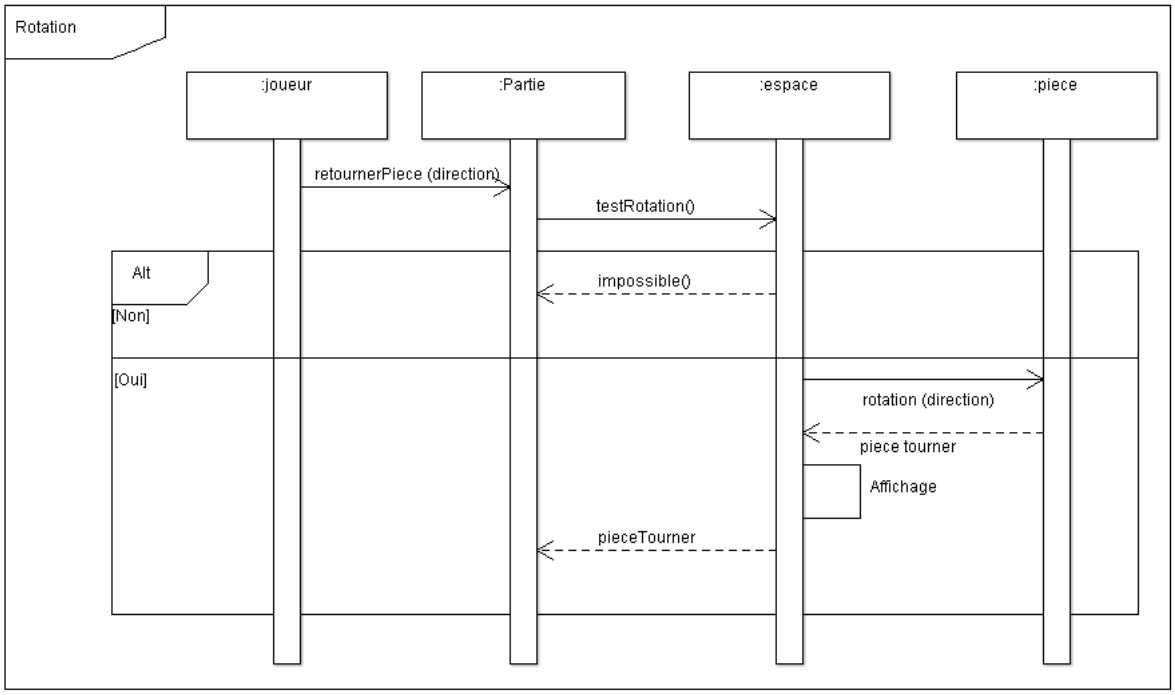
Explications :

Ce diagramme décrit le processus de création d'une pièce. On commence par choisir le type de pièce au hasard. On crée puis on place et on affiche cette pièce.



Explications :

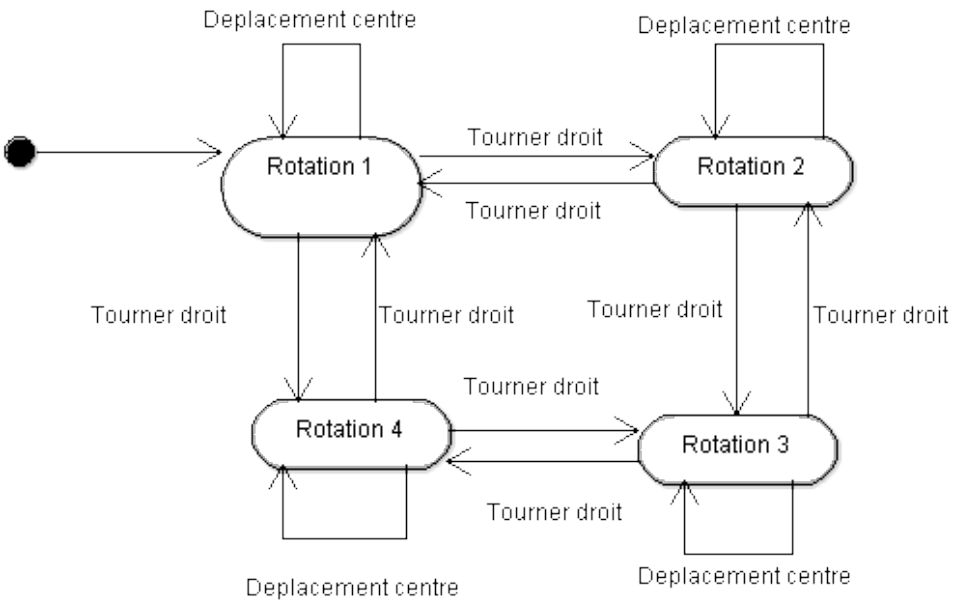
Ce diagramme décrit le processus de déplacement d'une pièce. On vérifie a chaque déplacement que le déplacement est possible, et si on touche le sol. Si le déplacement est effectivement possible on le fait. Si la pièce touche le sol, on détruit l'objet pièce, mais en gardant les cases ou elle était pleines. On peut alors en créer une nouvelle.



Explications :

Ce diagramme décrit le processus de rotation. On vérifie que la rotation est possible, et on l'effectue si elle l'est.

4) Diagramme d'état



Explications :

Ce diagramme décrit le passage d'une position de rotation à une autre.