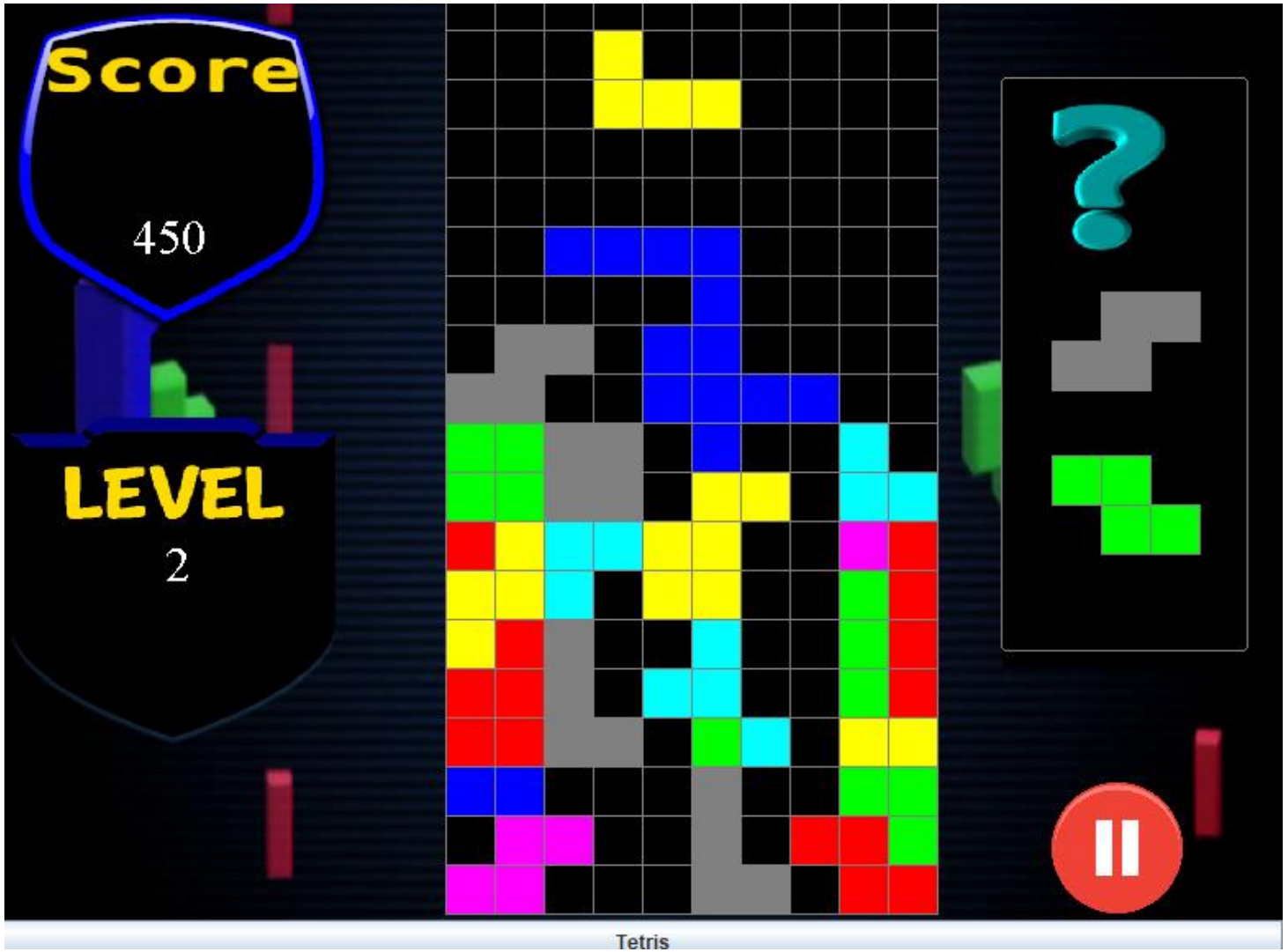


Rapport projet Tetris

LAHCENE Naël
VANG Amanda

Liste des fonctionnalités

Menu :

Menu avec possibilité de lancer une partie ou de quitter en utilisant la souris

Rotation:

Possibilité de faire une rotation aux pièces avec la flèche du haut

Pièces suivantes :

Affiche les 3 prochaines pièces à droite de la grille

Score :

Affichage du score en direct à gauche qui est incrémenté lorsqu'une ligne est faite

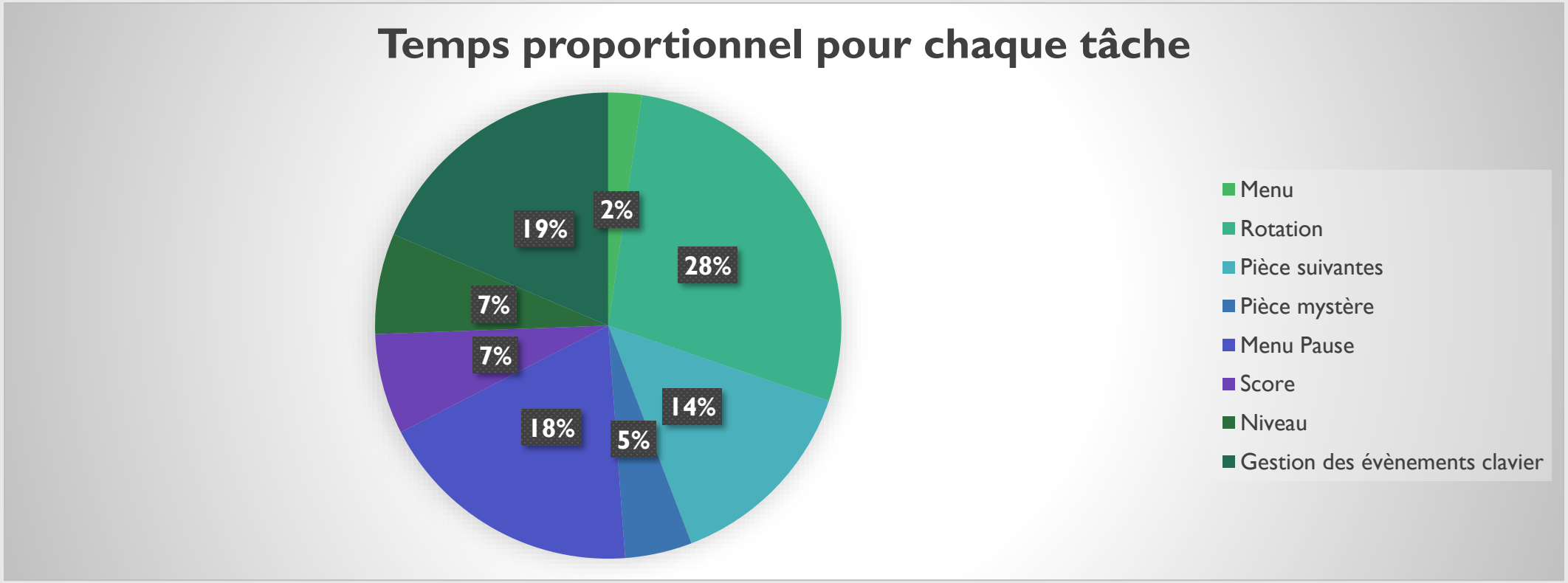
Niveau :

Affichage du niveau et qui en fonction du score est incrémenté. Augmente la vitesse de descente des pièces, ce qui rend le jeu plus dur en fonction du niveau.

Pièce mystère :

Pièce qui est affichée avec un point d'interrogation, empêchant de savoir à l'avance quelle forme elle aura jusqu'à ce qu'elle devienne la pièce courante.

Diagramme des fonctionnalités



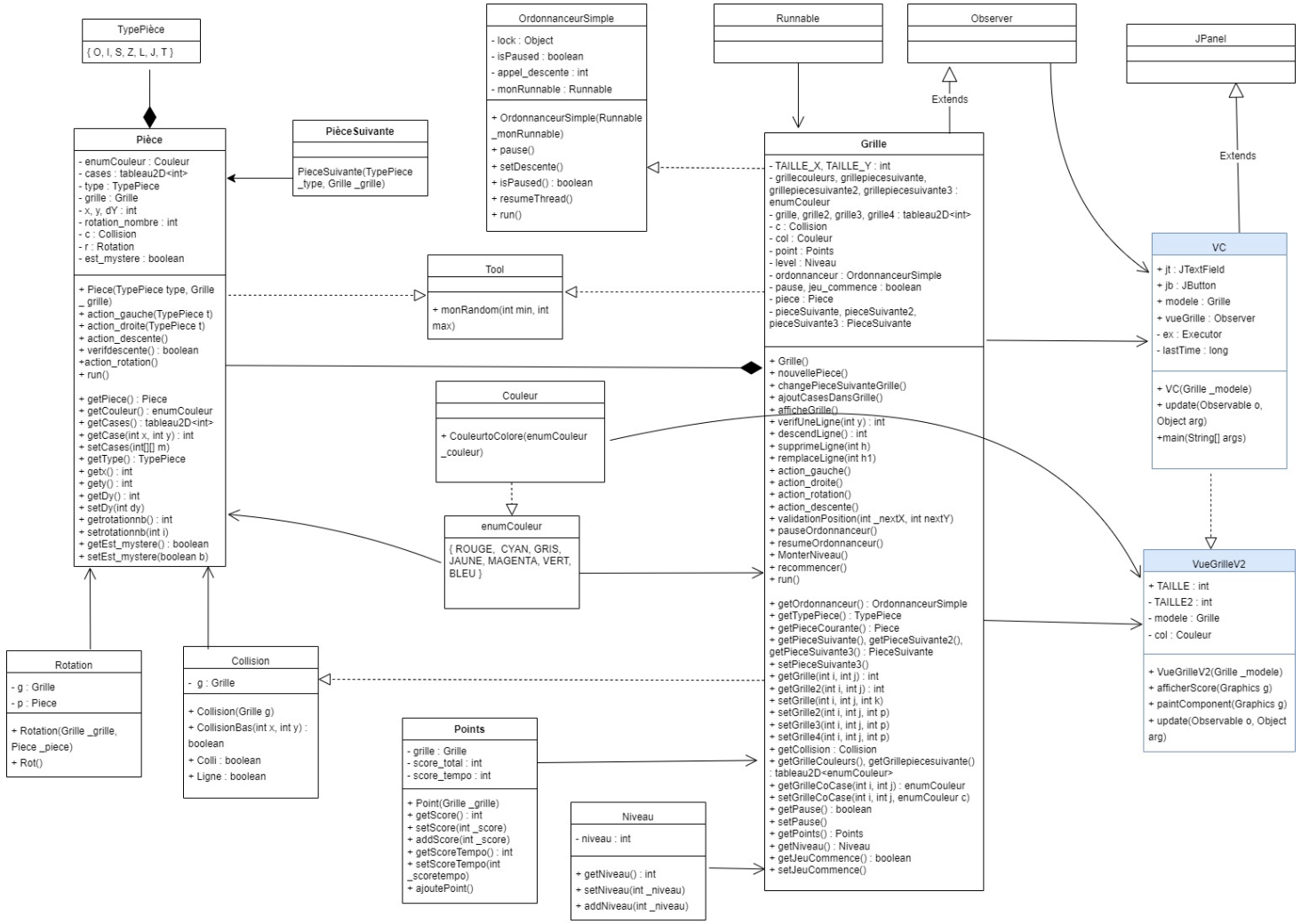


DIAGRAMME UML DES CLASSES DU TÉTRIS

Principales difficultés

Conception des Pièces :

Au début, il était difficile de savoir comment modéliser les pièces pour les utiliser efficacement dans la grille. Nous avons finalement décidé d'utiliser une énumération pour le type de pièce (L, J, O, etc.) avec une matrice représentée par un tableau d'entiers à 2 dimensions. Les "1" représentent une partie de la pièce et les "0" représentent le vide.

Rotation des Pièces :

La difficulté résidait dans le choix de la méthode. Après des recherches, nous avons opté pour l'utilisation de formules mathématiques. Cependant, en raison des problèmes liés à la nécessité d'un point d'origine, nous avons finalement créé une classe Rotation. Nous avons mis en place un algorithme simple avec une structure switch qui, en fonction du type de pièce et de son indice de rotation, modifie la matrice de la pièce.

Implémentation des Pièces dans la Grille :

La tâche la plus chronophage a été de savoir comment stocker les pièces dans la grille. Nous avons rencontré de nombreuses erreurs liées aux dépassements de tableau, en particulier lors des descentes de pièces et des rotations. Nous avons dû vérifier s'il n'y avait pas de collisions entre les pièces. Cette phase a demandé le plus de temps dans le développement du jeu.

Vue :

La gestion de l'affichage de la grille a été complexe. Nous avons utilisé un tableau de couleurs associé au tableau d'entiers de la grille. L'affichage de tous les éléments, que ce soit les images (arrière-plan, score, niveau, pause, etc.) ou les dessins générés par Swing (grille, pièces suivantes), devait être bien ordonnancé. Nous avons utilisé les types ImageIcon et la fonction `drawImage()` pour le rendu des images, et `drawString()` pour afficher le score et le niveau. Il était essentiel de veiller à ce qu'aucun élément ne soit modifié dans la vue, directement ou par l'appel de fonctions. Nous avons uniquement utilisé des accesseurs pour lire la grille et afficher correctement tous les éléments.

Modèle :

La gestion des événements de souris en fonction de l'affichage a été difficile en raison de l'utilisation intensive d'images. Nous avons utilisé des exécuteurs pour appeler des fonctions, par exemple, afficher la pause en cas de clic sur le bouton pause. Sans cela, le processus de la vue restait en sommeil avant de pouvoir afficher la pause.