

# Projet Tetris

*Modélisation Objet, Modèle Vue Contrôleur, Swing*

*Critères d'évaluation :*

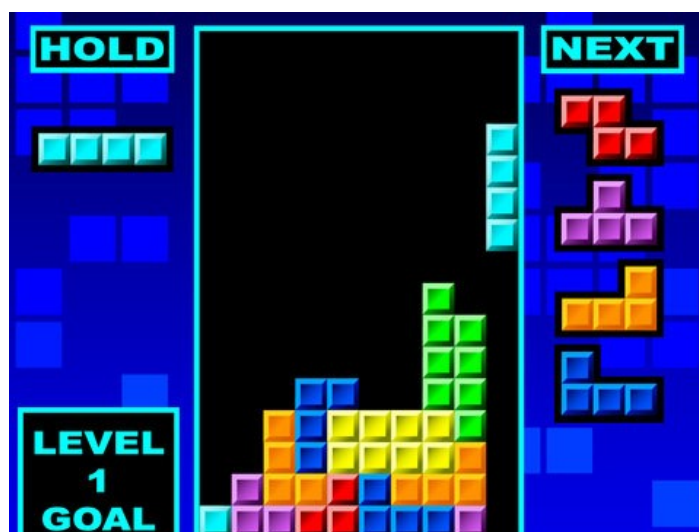
- *Qualité de l'analyse et du code associé*
- *Respect du Modèle Vue Contrôleur Strict*
- *Modularité*
- *Extensions proposées*

## 1 Le Tetris

Ce projet consiste à développer le jeu du Tetris (<https://fr.wikipedia.org/wiki/Tetris>).

### 1.1 Travail en binôme

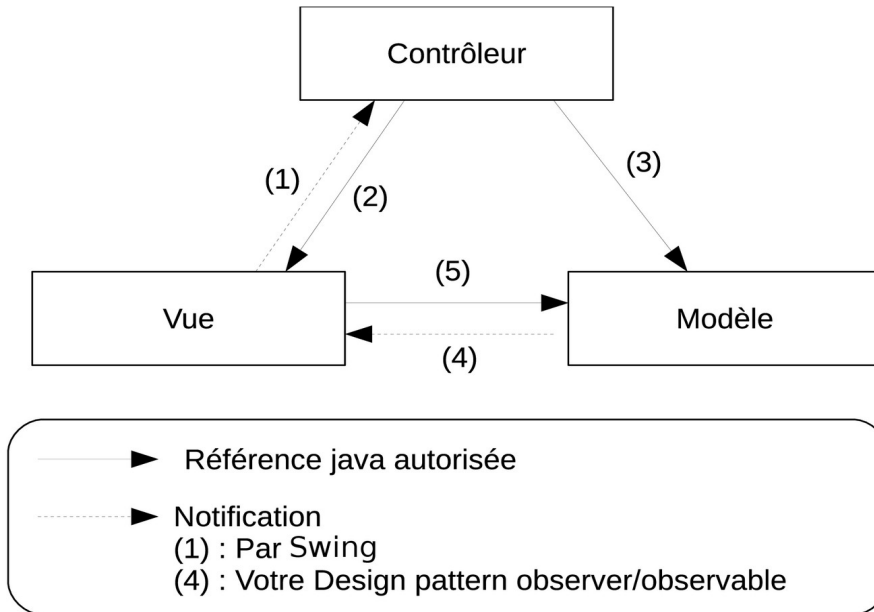
- Travail personnel entre les séances ;
- Évaluations individuelles ;
- Rapport par binôme : 6 pages au maximum, listes de fonctionnalités et extensions (indiquer la proportion de temps associée à chacune d'elles), copies d'écran, 1 diagramme UML de votre choix.
- Démonstration lors de la dernière séance encadrée (présence des deux binômes obligatoire).



### 1.2 Travail à réaliser

Développer une application graphique Java la plus aboutie possible de l'application, en respectant le modèle MVC Strict. Vous êtes responsables et libres de votre analyse, et pouvez proposer des fonctionnalités supplémentaires. Veillez à proposer ces fonctionnalités incrémentalement, afin d'avoir une démonstration opérationnelle le jour de la soutenance. **Le code doit être le plus objet possible. Privilégiez donc une programmation objet plutôt qu'un algorithme central long et complexe.**

## 2 Rappel : MVC Strict



### MVC Strict :

- (1) Récupération de l'événement Swing par le controleur
- (2) Répercutions locale directe sur la vue sans exploitation du modèle
- (3) Déclenchement d'un traitement pour le modèle
- (4) Notification du modèle pour déclencher une mise à jour graphique
- (5) Consultation de la vue pour réaliser la mise à jour Graphique

### Application Calculatrice :

- (1) récupération clic sur bouton de calculatrice
- (2) construction de l'expression dans la vue (1,2...9, (, ))
- (3) déclenchement calcul (=)
- (4) Calcul terminé, notification de la vue
- (5) La vue consulte le résultat et l'affiche

Remarque : le code associé au contrôleur et à la vue peut être réalisé dans le même fichier .java tel que vu en cours (utilisation de classes anonymes ou de classes internes).

### 2.1 Modèle

#### Données :

Grille, Pièces, Temps, Points, Pièces suivantes, etc.

#### Fonctionnalités :

- Paramétrage de la partie (démarrer, quitter, pause, etc.)
- Initialiser : créer une grille vide
- Déroulement du jeu :
  - Un thread fait descendre la pièce courante, et réalise les traitements pour figer la pièce le moment venu
  - Des évènements (clavier ou souris) permettent de déplacer la pièce courante (translations, rotations)
  - etc.

### **3 Extensions, suivant votre avancement**

**Sauvegarde des meilleurs scores**

**Mode deux joueurs**

**IA**

**Niveaux de difficulté**

**Tetris 3D**

**Rendu graphique et sonore**

**Un extension que vous souhaitez proposer (à valider avec votre encadrant)**