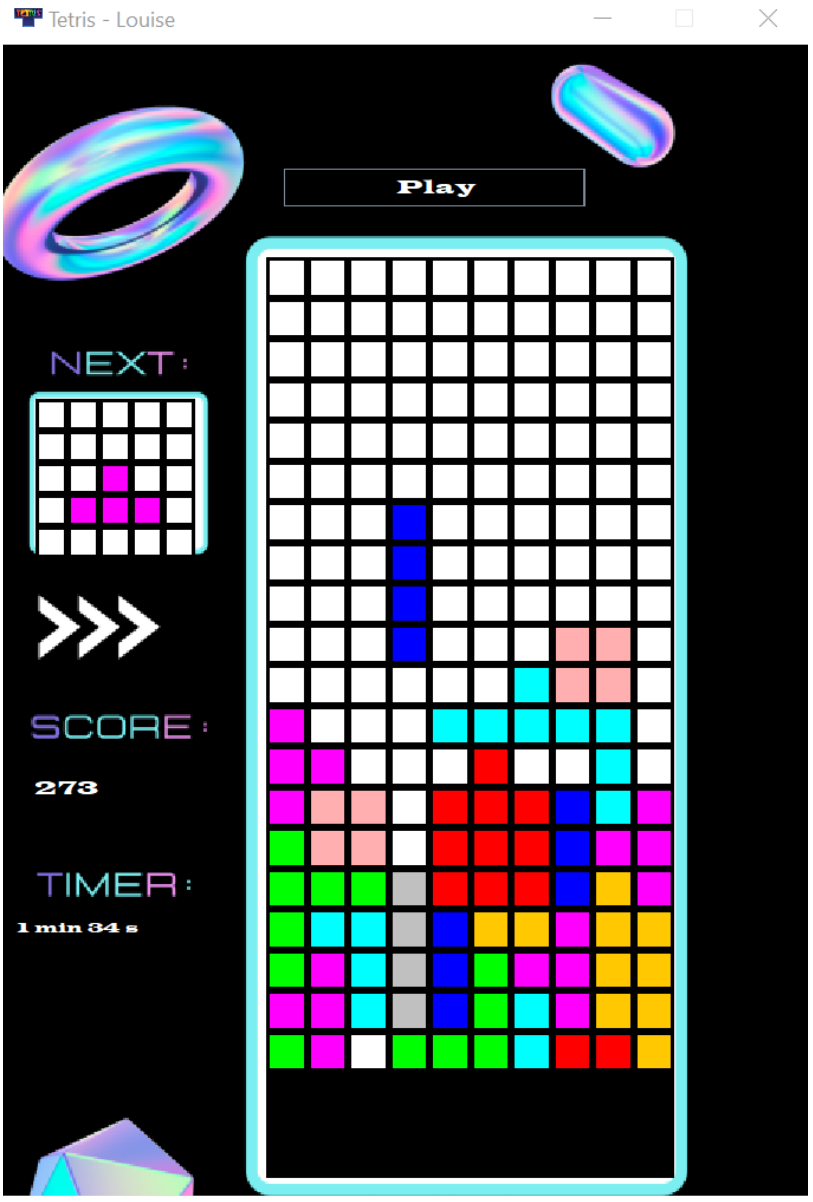
Rapport projet Algorithmique et Programmation Objet

TETRIS



Membres du groupe :
Amine EL GAOUZI
Louise RIBOLLET
Lucas TEXIER

IMT Mines Alès
École Mines-Télécom

Table des matières

Synthèse

Le projet d'Algorithmique et Programmation Objet consiste à analyser et programmer une problématique mettant en jeu la totalité du cours abordé. Dès la formation du groupe, nous voulions programmer un jeu en ligne. Nous avons donc décidé de programmer le célèbre jeu “Tetris” qui possède une interface attractive ainsi qu’un principe de jeu facile à comprendre : compléter des lignes afin de gagner des points.

Notre objectif de base était la création d’un jeu qui fonctionne avec un plateau de jeu, sept pièces toutes avec des formes et des couleurs différentes. Nous voulions que, lorsqu’une ligne est complète, cette dernière se supprime et tout le plateau au-dessus descende.

Nous avons donc pu réaliser un jeu abouti puisque le principe de base du jeu est fonctionnel. Nous avons pu le perfectionner en ajoutant plusieurs interfaces graphiques animées (lancement du jeu, jeu en lui-même et lorsqu’on a perdu), le fonctionnement du jeu est fluide, il y a plusieurs niveaux de difficulté, un score, un classement en fonction du meilleur score de chaque joueur se sauvegardent automatiquement à la fin de chaque partie , un chronomètre. Il y a même de la musique en fond et qui change selon les événements.

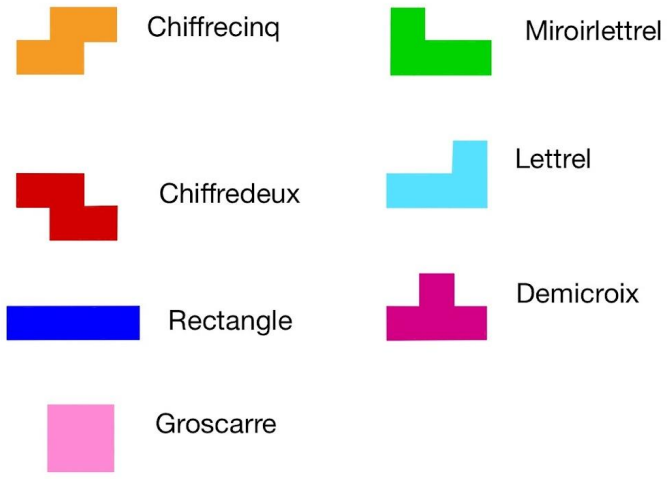
Analyse

Le but du projet a donc été de coder un Tetris fonctionnel. Nous avons codé sur Java et les interfaces graphiques ont été réalisées à l'aide de la bibliothèque Java.swing.

Pour commencer, le plateau de jeu, situé dans la classe **Plateau**, a été fait par le biais d'une matrice comportant des Objets : **la classe Case**. C'est cette classe Case qui nous a permis de mettre des couleurs sur le plateau grâce à ses méthodes .

Le premier problème auquel nous avons été confrontés a été de faire apparaître les différentes formes sur la grille. Pour cela nous avons fait une **classe mère Forme** qui se rappelle des coordonnées sur la matrice de son centre et des coordonnées de ses points par rapport à son centre. Ainsi, en créant **sept classes filles** correspondant à chacune des pièces, nous étions capables de dessiner toutes les pièces sur la grille. La classe Forme contient aussi toutes les méthodes qui permettent de dessiner, savoir si le mouvement (tourner/descendre) est possible ainsi que l'affichage de la position de l'endroit où la pièce va tomber. Cela permet donc d'utiliser toutes ces méthodes pour chacune des pièces.

Correspondance des classes avec les pièces:

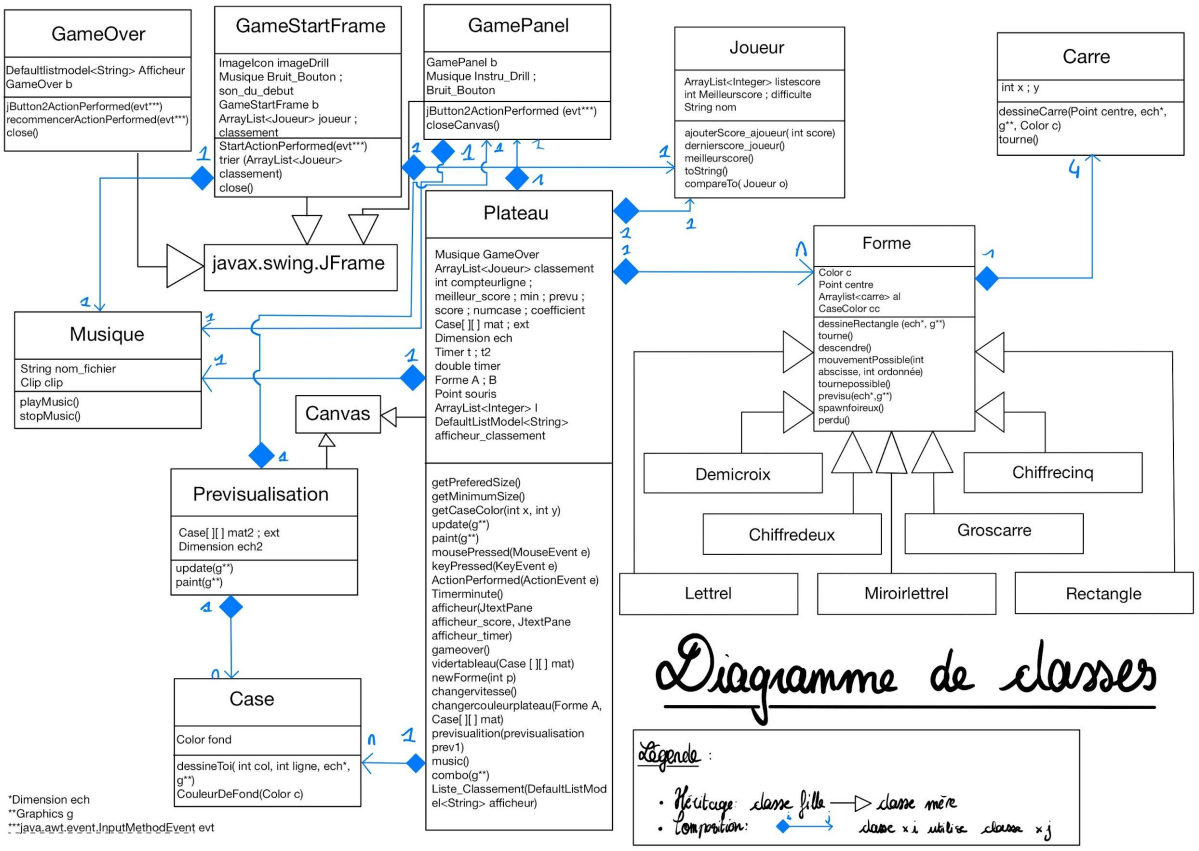


Ensuite, pour réussir à faire descendre les pièces, nous avons mis sur notre plateau de jeu un **timer** qui fait descendre les pièces en augmentant l'ordonnée du centre de la pièce. Mais ensuite, il a fallu trouver la condition pour poser la pièce lorsqu'elle ne peut plus descendre. Pour cela, nous avons créé une fonction dans Forme qui permet de savoir s'il est possible de se déplacer de dx et dy. Ainsi dans notre cas lorsque la pièce ne peut plus se déplacer de dx = 0 et dy = 1, on dessine sur le plateau la pièce. Par la même occasion, on a mis la possibilité de déplacer la pièce droite/gauche/bas avec les flèches directionnels. Pour savoir si la pièce peut se déplacer, nous avons pris la **couleur** comme élément indicateur de si oui ou non une case est libre : ***une case est libre si et seulement si elle est blanche***.

Ensuite, la pièce se posait bien mais le problème a été de faire apparaître les pièces suivantes **aléatoirement**. On tire un nombre au hasard entre 1 et 7, chacun étant attribué à une pièce spécifique, puis selon sa valeur, on fait apparaître cette dernière juste au-dessus de la grille. Les cases correspondantes aux pièces ne se dessinent qu'à partir du moment où elles sont sur le plateau (y >= 0).

Le jeu en lui-même était alors quasi fonctionnel mais il manquait la suppression des lignes. Nous voulions donner un aspect dynamique à cette suppression, pour cela, nous avons mis un **deuxième timer** plus rapide, mis en marche dès qu'au moins une ligne est à supprimer. D'abord il y a un balayage noir puis les lignes supprimées se voient attribuer la ligne d'au dessus.

Sur l'interface graphique, nous avons aussi mis des indicateurs : le score, le temps, le nom d'utilisateur et la prochaine pièce. Pour la pièce qui va arriver ensuite, on a fait la **classe Previsualisation** qui hérite de Canvas affichant la prochaine pièce. Elle est constituée d'une matrice (comme le Plateau) et on affiche dans cette dernière des pièces.



Algorithmes les plus importants

Un moment important pour la programmation du tétris a été l'affichage de la prévisualisation de la chute de la pièce. C'est-à-dire afficher à chaque étape montrer où tombe la pièce si l'on ne touche plus à celle-ci. D'autant plus que le rendu est plutôt agréable et utile sur le jeu.

```
66      void previsu(Dimension ech, Graphics g) {
67          int a = 0;
68          while (mouvementPossible(0, a)) { // on cherche le nombre a de fois qu'il est encore pos
69              a++;
70          }
71          a--;
72          Point cp = new Point(centre.x, centre.y + a); // on construit le centre correspondant à
73          for (int i = 0; i < al.size(); i++) {
74              al.get(i).dessineCarré(cp, ech, g, Color.lightGray); // on dessine à partir de ce no
75          }
76      }
```

On définit donc dans la classe Forme la **fonction previsu** qui permet de dessiner la partie grisée. Cette fonction est juste rappelée dans la fonction qui permet de dessiner la pièce sur la grille afin de dessiner les deux en même temps.
Le fonctionnement de l'algorithme est simple, on initialise un entier puis à l'aide d'une boucle while, on détermine combien de mouvement possible la pièce peut faire vers le bas (donc pour y croissant). On utilise pour cette occasion la fonction mouvementPossible(dx,dy) qui détermine ***si un mouvement de dx et dy est réalisable.***
Une fois que l'on a cette translation maximale, on crée un point correspondant au centre avec +a en 'ordonnée.
Ensuite, on dessine à partir de ce nouveau centre toute la pièce grisée à l'image de celle pour une pièce normale.
A noter que lorsque l'on appelle cette fonction dans la fonction dessinant la pièce, on dessine d'abord la partie grisée puis la pièce de manière à s'il y a une superposition, la pièce prime sur sa prévisualisation.

Interface graphique des différents panels :

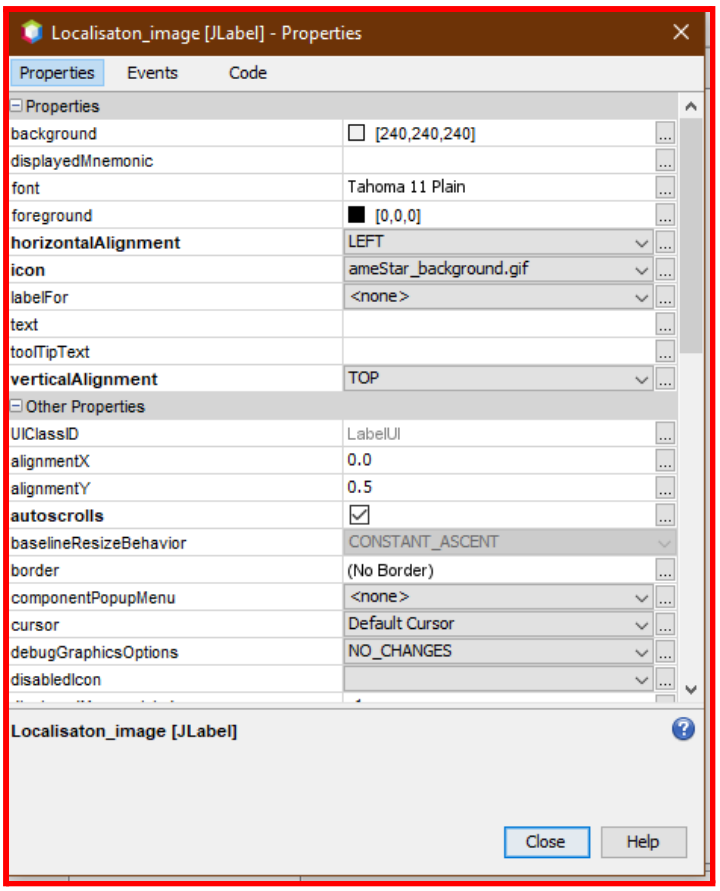
Nous voulions avoir une interface graphique plaisante se rapprochant au maximum d'un vrai jeu vidéo pouvant être commercialisé sur le play store. Afin de faciliter la programmation de cette interface , nous avons décidé de découper notre jeu en 3 Jframe représentant 3 phases du jeu :
Etape 1 : Lancement du jeu (GameStarFrame)
Etape 2 : Début du jeu (GamePanel)
Etape 3: Fin de partie (GameOver)

Le fonctionnement des différentes Jframe étant similaire, je détaillerais dans cette partie le fonctionnement de la Gamme StarFrame.

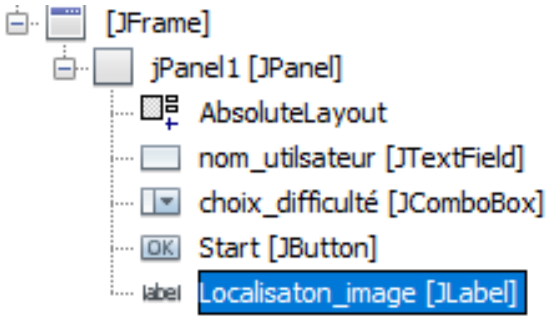
Lorsque que nous lançons le jeu, la GameStarFrame s'affiche. Elle permet grâce à des champs de texte et une liste boxe de choisir le nom et le niveau de difficulté de

Graphique design

Dans un premier temps nous avons dû dessiner grâce aux logiciels Canva le design de notre frame, nous avons ajouté des animations dans le but de dynamiser notre jeu. Grâce à Canva nous avons pu obtenir un Gif Compatible avec JAVA.
Le gif représentant le Game Start panel a été implémenté grâce aux fonctionnalités de Java Swing.



Nous avons transformé un Jlabel “ Localisation_image” en icon le “GameStar_background.gif” en modifiant les propriétés du JLabel 6 ligne de l 'longlet properties.



La listbox, le bouton et le texte field ont été mis au premier plan en faisant click droit move up, afin d’afficher ces éléments au dessus du gif. Le bouton start est sur le gif mais nous avons mis le bouton en opacity en mode false pour le rendre transparent.

Action du bouton start

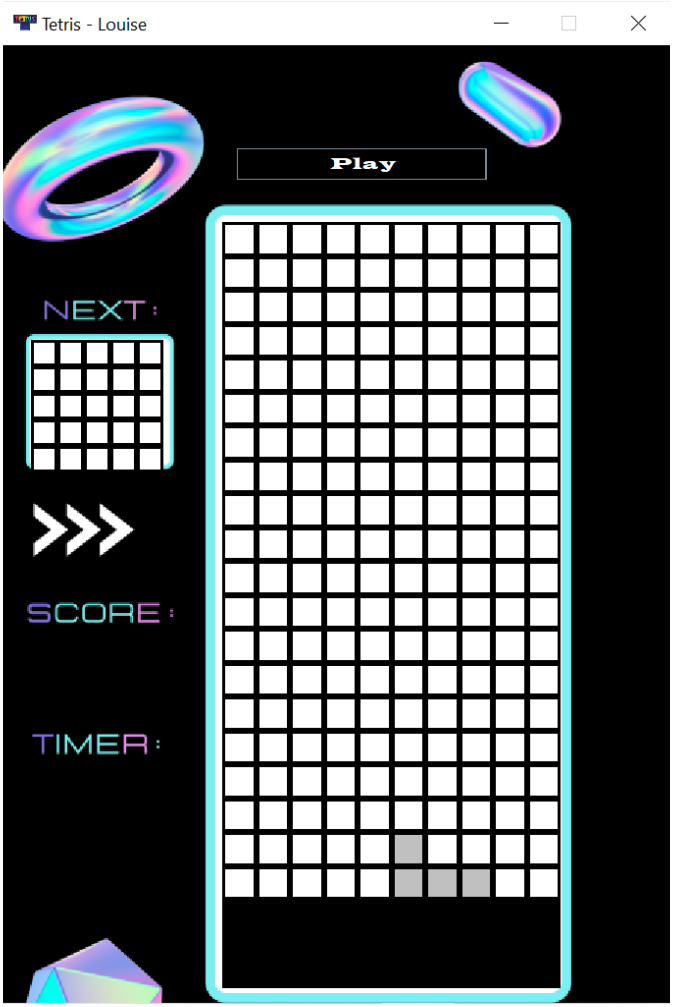
```
private void StartActionPerformed(java.awt.event.ActionEvent evt) {
    // Ajoute l'utilisateur à la liste des joueur
    joueur.add(new Joueur(nom_utilsateur.getText(), choix_difficulté.getSelectedIndex()));

    // créer une nouvelle fenetre
    GamePanel.b = new GamePanel();
    GamePanel.b.setVisible(true);// rend la fenetre visible
    GamePanel.b.setDefaultCloseOperation(DISPOSE_ON_CLOSE);// parametrage qui permet de fermer la fenetre
    GamePanel.b.setTitle("Tetris - " + joueur.get(joueur.size() - 1).nom);
    GamePanel.b.plateau2.t.setDelay(300);
    GamePanel.b.plateau2.coefficientscore = ((joueur.get(joueur.size() - 1).difficulté) + 1);
    GamePanel.b.setLocationRelativeTo(this);

    try {

        son_du_début.clip.close();
        Bruit_bouton = new Musique("Bruit_Boutton.wav");
        Bruit_bouton.playMusic();
    } catch (UnsupportedAudioFileException ex) {
        Logger.getLogger(GameStartFrame.class.getName()).log(Level.SEVERE, null, ex);
    } catch (IOException ex) {
        Logger.getLogger(GameStartFrame.class.getName()).log(Level.SEVERE, null, ex);
    } catch (LineUnavailableException ex) {
        Logger.getLogger(GameStartFrame.class.getName()).log(Level.SEVERE, null, ex);
    }
    // ferme l'ancienne fenetre
    close();
}
```

Lorsque l'on appuie sur le bouton start, on ajoute à une liste de joueur un nouveau joueur ayant les caractéristiques données : le nom et le le niveau de difficulté. Ensuite on crée une nouvelle fenêtre GamePanel qui s'ouvre sur l'ancienne fenêtre grâce à la fonction setRelativeTo, et en titre le nom du joueur grâce à la fonction setTitle. Nous initialisons aussi les paramètres permettant de jouer au tetris : le delay, le coefficient de_difficulté augmentant le score voir mode d'emploi. On lance la musique, puis on ferme l'ancienne fenêtre grâce à la fonction close.

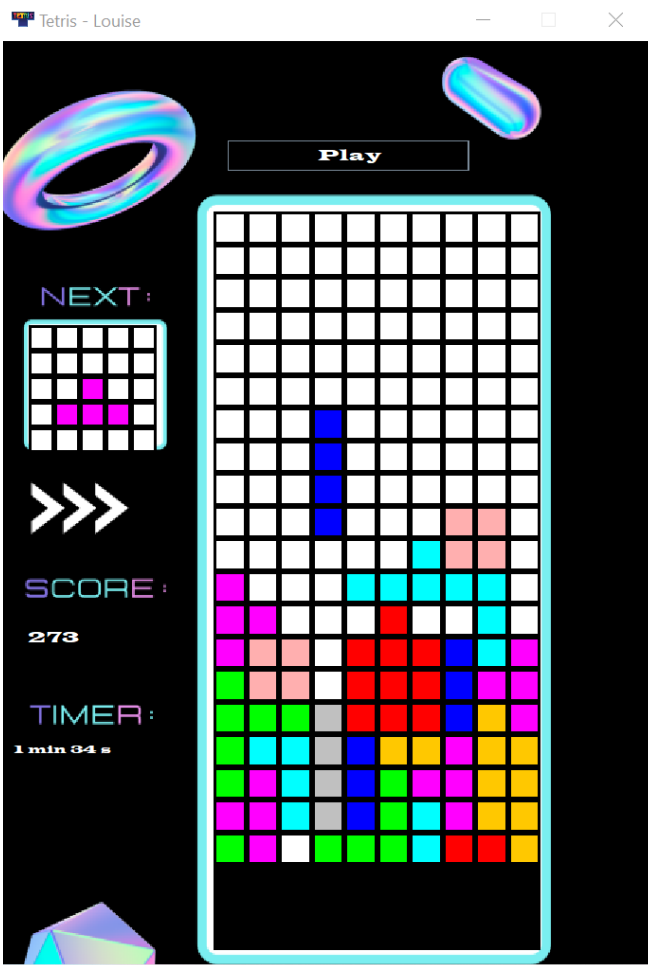
```
// fonction qui permet de fermer une JframeSWING
    private void close() {

        WindowEvent closeWindow = new WindowEvent(this, WindowEvent.WINDOW_CLOSING);
        Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(closeWindow);

    }
```

Jeux de test

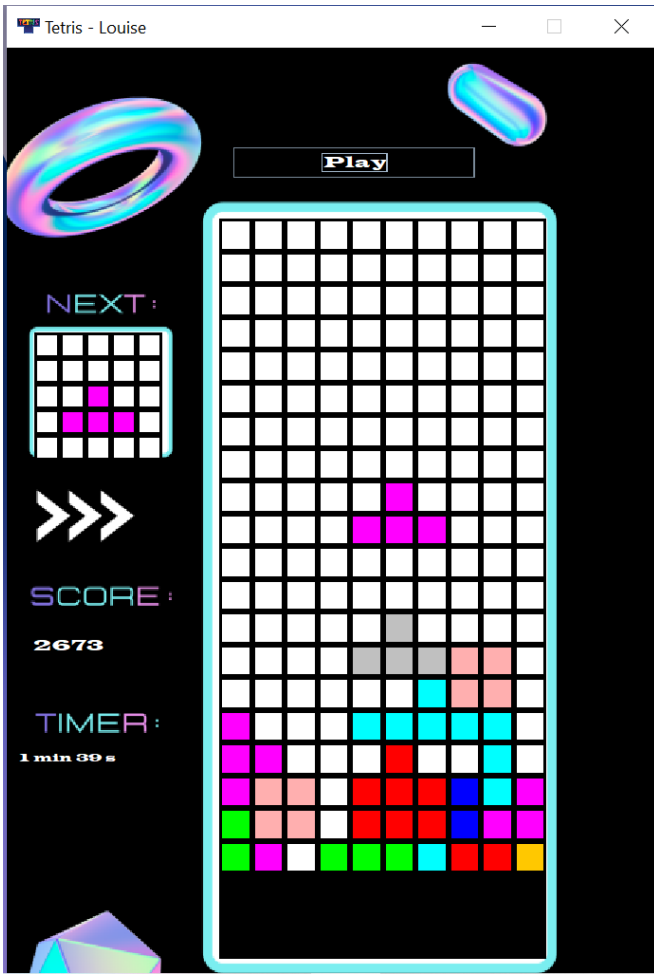
Etape 1 : Lancement du jeu



Etape 2 : début du jeu



Etape 3 : descente des pièces



Etape 4 : suppression des lignes

Etape 5 : Fin de partie

Mode d’emploi

Démarrage :

Le lancement du jeu s’effectue en appuyant sur le bouton démarrer de Netbeans. Il affiche alors la page d’accueil du jeu. À ce stade, le joueur inscrit son nom et décide de la difficulté avec laquelle il veut jouer avant de commencer la partie.Elle correspond à la vitesse d’accélération du jeu. En effet, lorsqu’une ligne est effectuée, le rythme de descente des pièces augmente. Lors du niveau « Facile », le rythme augmente progressivement tandis dis que lors du niveau « Jean-Paul Veuillez », dès qu’une ligne est effectuée, la pièce arrive au bas de l’écran de manière quasi-instantanée. Pour commencer la partie, il faut cliquer sur les flèches en bas à droite.

Principe de jeu :

L’objectif est de compléter des lignes pour avoir un maximum de points.
Pour jouer, on utilise les flèches du clavier. Les **flèches de droite** et **de gauche** afin de déplacer la pièce latéralement. La **flèche du bas** permet d’accélérer la descente de la pièce. La **barre d’espace** ou la **flèche du haut** permet de faire pivoter la pièce sur elle-même.

Le **comptage des points** s’effectue de la manière suivante :
+1 à chaque fois qu’on appuie sur la flèche du bas
+(40*coefficient_de_difficulté) lorsqu’une ligne est complète
+(100*coefficient_de_difficulté) lorsque deux lignes sont complètes simultanément
+(300*coefficient_de_difficulté) lorsque trois lignes sont complètes simultanément
+(1200*coefficient_de_difficulté) lorsque quatre lignes sont complètes simultanément

Fin du jeu :

La partie s’arrête lorsque les pièces ne peuvent plus descendre car la colonne où les pièces apparaissent est pleine. Une nouvelle fenêtre apparaît alors. Sur celle-ci s’affiche le score qui vient d’être réalisé ainsi que le classement des joueurs en fonction de leur score ayant déjà joué. Il n’y a pas de répétition dans les noms des joueurs. Deux possibilités pour la suite : ou bien le joueur peut recommencer ou quelqu’un d’autre souhaite jouer et il suffit de cliquer sur la case : « changer d’utilisateur ».