



TETRIS



Introduction

- Présentation du projet

1. Architecture logicielle

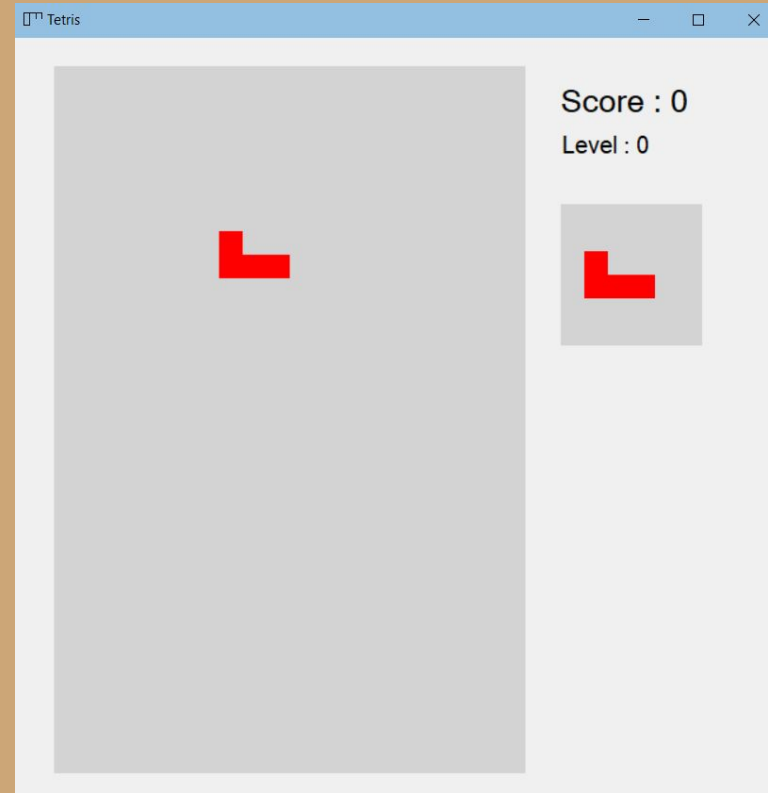
- Technologies utilisées
- Présentation du code

2. Développement

3. Démonstration

4. Organisation

Présentation du projet



ARCHITECTURE

- Technologies utilisées
 - C#
 - Windows Form

ARCHITECTURE

- Présentation du code
 - Organisation des classes

Program.cs : Lancement du jeu

Form1.cs : Gestion du canvas du jeu

Form2.cs : Gestion du menu

Shape.cs : Creation de la forme

Handler.cs : Conteneur de formes

DÉVELOPPEMENT

- Présentation de la classe clearFilledRowsAndUpdateScore

```
1 reference
public void clearFilledRowsAndUpdateScore()
{
    // Verifier la ligne
    for (int i = 0; i < canvasHeight; i++) {
        int j;
        for (j = canvasWidth - 1; j >= 0; j--) {
            if (canvasDotArray[j, i] == 0)
                break;
        }

        if (j == -1) {
            score++;
            label_Score.Text = "Score: " + score;
            label_Level.Text = "Level: " + score / 10;

            // Augmente vitesse
            timer.Interval -= 10;

            // Mettre a jour le tableau de points
            for (j = 0; j < canvasWidth; j++) {
                for (int k = i; k > 0; k--) {
                    canvasDotArray[j, k] = canvasDotArray[j, k - 1];
                }

                canvasDotArray[j, 0] = 0;
            }
        }
    }

    // Redessiner le tableau en fonction du nouveau
    for (int i = 0; i < canvasWidth; i++) {
        for (int j = 0; j < canvasHeight; j++) {
            canvasGraphics = Graphics.FromImage(canvasBitmap);
            canvasGraphics.FillRectangle( canvasDotArray[i, j] == 1 ? Brushes.Red : Brushes.LightGray, i * pointSize, j * pointSize, pointSize, pointSize );
        }
    }

    pictureBox_Game.Image = canvasBitmap;
}
```

DÉVELOPPEMENT

- Présentation du code

```
4 références
public partial class Form1 : Form
{
    /*****|Properties|*****/
    Bitmap canvasBitmap;
    Graphics canvasGraphics;

    int canvasWidth = 20;
    int canvasHeight = 30;
    int[,] canvasDotArray;
    int pointSize = 20;
    int currentX;
    int currentY;

    Shape currentShape;
    Shape nextShape;

    Timer timer = new Timer();

    int score;

    System.Media.SoundPlayer musique = new System.Media.SoundPlayer(@"C:\Users\romal\Desktop\Tetris\Tetris\assets\main_theme.wav");

    /*****|Constructor|*****/
    1 référence
    public Form1()
    {
        InitializeComponent();

        loadCanvas();

        currentShape = getRandomShapeAtCenter();
        nextShape = getNextShape();

        timer.Tick += Timer_Tick;
        timer.Interval = 500;
        timer.Start();

        musique.Play();

        this.KeyDown += Form1_KeyDown;
    }

    /*****|Methode|*****/
    // Canvas
    1 référence
}
```

DÉMONSTRATION

ORGANISATION

- Etablir les règles du jeu
- Etablir les problèmes pouvant être rencontrés
- Mise en place d'un git commun
- Etablir une version logique
- Etablir une version graphique

CONCLUSION
