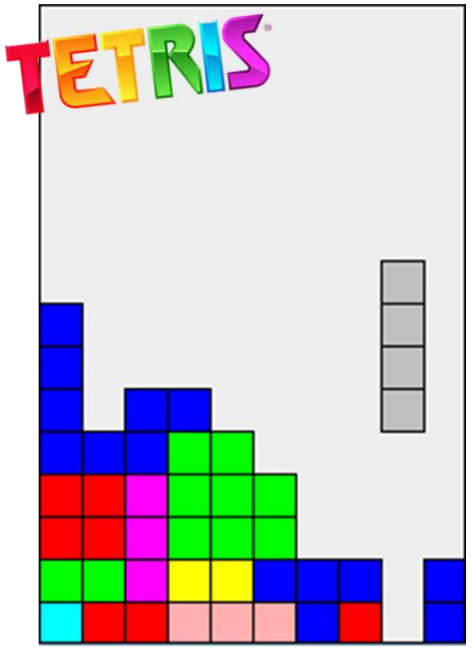
Main Function

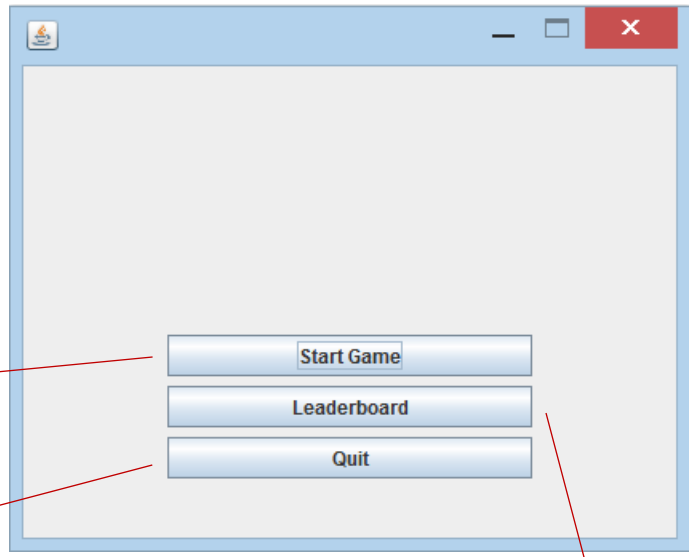
```
public static void main(String[] args){
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            gf = new GameForm();
            sf = new StartupForm();
            lf = new LeaderboardForm();

            sf.setVisible(true);
        }
    });


}
```



- Startup Form
- Game Form
- Shape of block
- Move Block
- Game Over
- DataBase

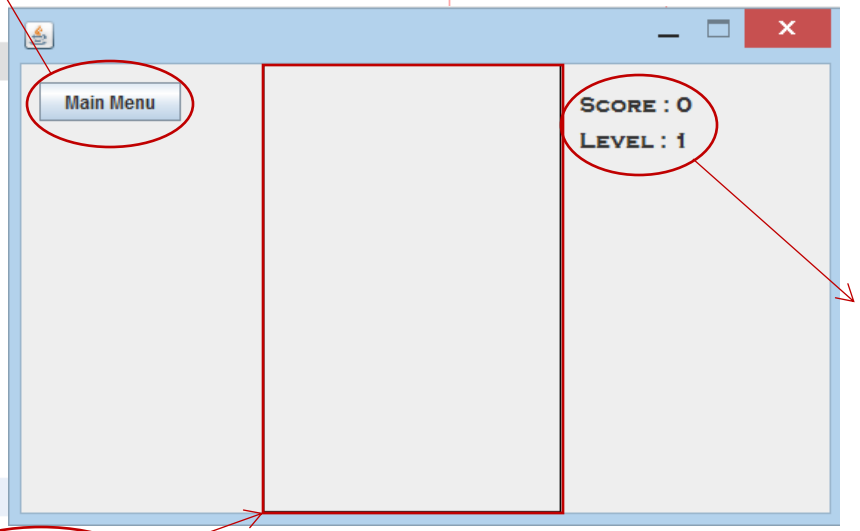
Startup Form

```
package tetris;

public class StartupForm extends javax.swing.JFrame {

    public StartupForm() {
        initComponents();
    }

                    //...

    private void btnStartActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        Tetris.start();
    }

    private void btnQuitActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);
    }

    private void btnLeaderboardActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        Tetris.showLeaderboard();
    }
```



Game Form

```
private void btnMainMenuActionPerformed(java.awt.event.ActionEvent evt) {
    gt.interrupt();
    this.setVisible(false);
    Tetris.showStartup();
}
```




```
public void updateScore(int score) {
    scoreDisplay.setText("Score : " + score);
}

public void updateLevel(int level) {
    levelDisplay.setText("Level : " + level);
}
```

```
public GameForm() {
    initComponents();

    ga = new GameArea(gameAreaPlaceholder, 10);
    this.add(ga);

    initControls();
}
```

Shape of Blocks

```java
package tetrisblocks;

import tetris.TetrisBlock;

public class IShape extends TetrisBlock{
    public IShape(){
        super( new int[][]{ {1, 1, 1, 1} } );
    }
```

1	1	1	1

```java
package tetrisblocks;

import tetris.TetrisBlock;

public class OShape extends TetrisBlock{
    public OShape(){
        super( new int[][]{ {1, 1},
                            {1, 1} });
    }
```

1	1
1	1

```java
package tetrisblocks;

import tetris.TetrisBlock;

public class JShape extends TetrisBlock {
    public JShape(){
        super( new int[][]{ {0, 1},
                            {0, 1},
                            {1, 1} });
    }
```

0	1
0	1
1	1

```java
package tetrisblocks;

import tetris.TetrisBlock;

public class SShape extends TetrisBlock{
    public SShape(){
        super( new int[][]{ {1, 0},
                            {1, 1},
                            {0, 1} });
    }
```

1	0
1	1
0	1

```java
package tetrisblocks;

import tetris.TetrisBlock;

public class LShape extends TetrisBlock {
    public LShape(){
        super( new int[][]{ {1, 0},
                            {1, 0},
                            {1, 1} });
    }
```

1	0
1	0
1	1

```java
package tetrisblocks;

import tetris.TetrisBlock;

public class TShape extends TetrisBlock{
    public TShape(){
        super( new int[][]{ {1, 1, 1},
                            {0, 1, 0} });
    }
```

1	1	1
0	1	0

```java
package tetrisblocks;

import tetris.TetrisBlock;

public class ZShape extends TetrisBlock{
    public ZShape(){
        super( new int[][]{ {1, 1, 0},
                            {0, 1, 1} });
    }
```

1	1	0
0	1	1

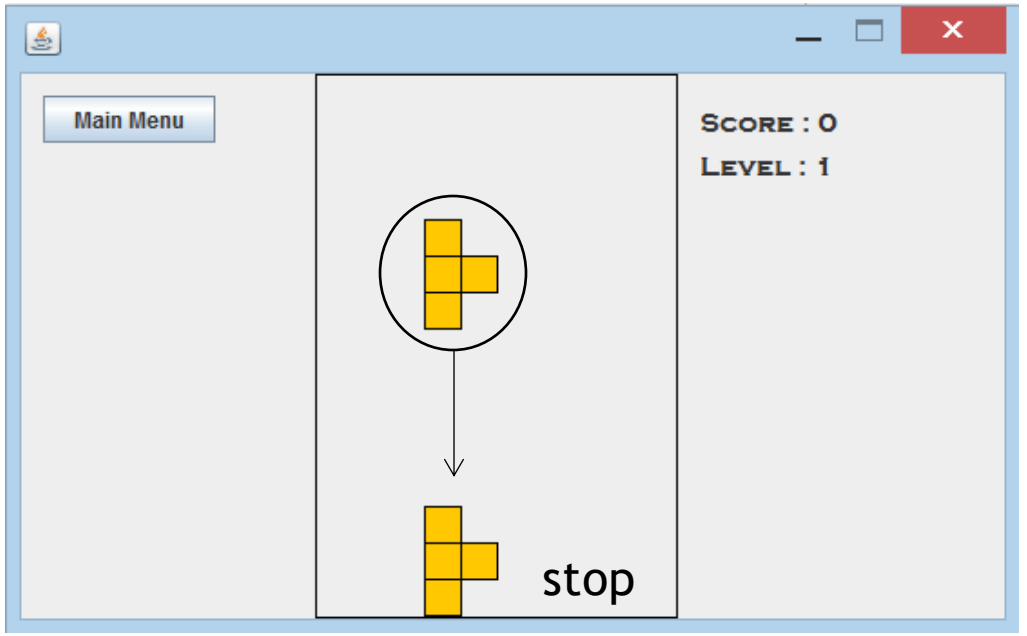
Game Form

//Game Area.java
//let block move down

```
public void dropBlock() {
    if( block == null ) return;
    while( checkBottom() ){
        block.moveDown();
    }
    repaint();
}

private boolean checkBottom(){
    if( block.getBottomEdge() == gridRows){
        return false;
    }
    int[][]shape = block.getShape();
    int w = block.getWidth();
    int h = block.getHeight();

    for(int col = 0; col < w; col++){
        for(int row = h - 1; row >= 0; row--){
            if(shape[row][col] != 0){
                int x = col + block.getX();
                int y = row + block.getY() + 1;
                if(y < 0) break;
                if(background[y][x] != null) return false;
                break;
            }
        }
    }
    return true;
}
```

//Tetris Block.java

```
public void moveDown(){ y++; }
```



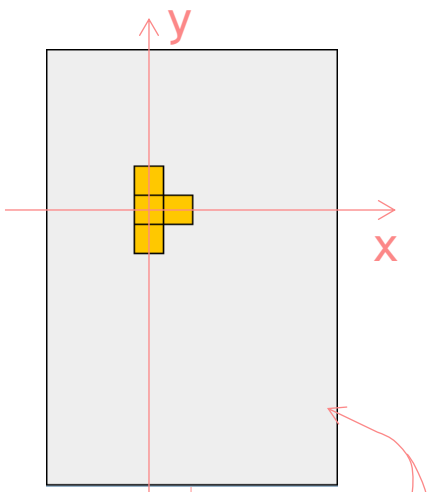
Move block

//Game Form.java

```
private void initControls(){
    InputMap im = this.getRootPane().getInputMap();
    ActionMap am = this.getRootPane().getActionMap();

    im.put(KeyStroke.getKeyStroke("RIGHT"), "right");
    im.put(KeyStroke.getKeyStroke("LEFT"), "left");
    im.put(KeyStroke.getKeyStroke("UP"), "up");
    im.put(KeyStroke.getKeyStroke("DOWN"), "down");

    am.put("right", new AbstractAction() {
        @Override
        public void actionPerformed(ActionEvent ae) {
            ga.moveBlockRight();
        }
    });
    am.put("left", new AbstractAction() {
        @Override
        public void actionPerformed(ActionEvent ae) {
            ga.moveBlockLeft();
        }
    });
    am.put("up", new AbstractAction() {
        @Override
        public void actionPerformed(ActionEvent ae) {
            ga.rotateBlock();
        }
    });
    am.put("down", new AbstractAction() {
        @Override
        public void actionPerformed(ActionEvent ae) {
            ga.dropBlock();
        }
    });
}
```

//Game Area.java

```
public void moveBlockRight() {
    if( block == null ) return;
    if( !checkRight() ) return;

    block.moveRight();
    repaint();
}
public void moveBlockLeft(){
    if( block == null ) return;
    if( !checkLeft() ) return;

    block.moveLeft();
    repaint();
}

public void rotateBlock() {
    if( block == null ) return;
    block.rotate();

    if(block.getLeftEdge() < 0) block.setX(0);
    if(block.getRightEdge() >= gridColumns) block.setX(gridColumns - block.getWidth());
    if(block.getBottomEdge() >= gridRows) block.setY(gridRows - block.getHeight());
    repaint();
}

public void dropBlock() {
    if( block == null ) return;
    while( checkBottom() ){
        block.moveDown();
    }
    repaint();
}
```

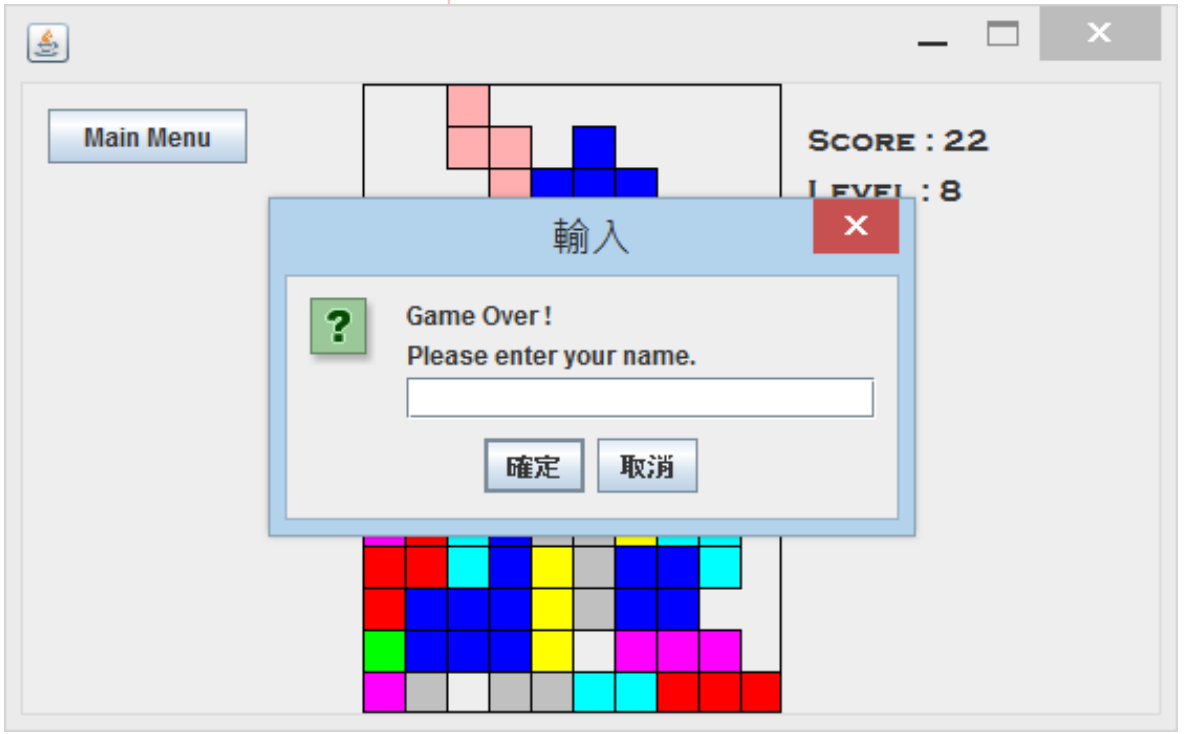



```
public void moveRight(){ x++; }

public void moveLeft(){ x--; }

public void rotate() {
    currentRotation++;
    if(currentRotation > 3) currentRotation = 0;
    shape = shapes[currentRotation];
}
```

Game Over

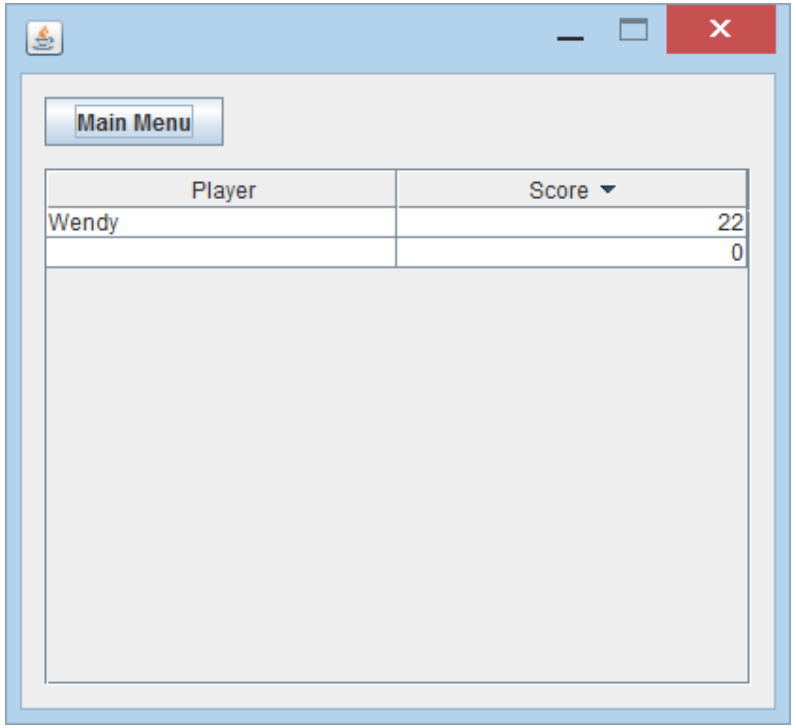
Input and Output your name(data)

```
public static void gameOver(int score){
    playGameover();

     String playerName =  JOptionPane.showInputDialog("Game Over !\nPlease enter your name.");

    gf.setVisible(false);
    lf.addPlayer(playerName, score);
}
```



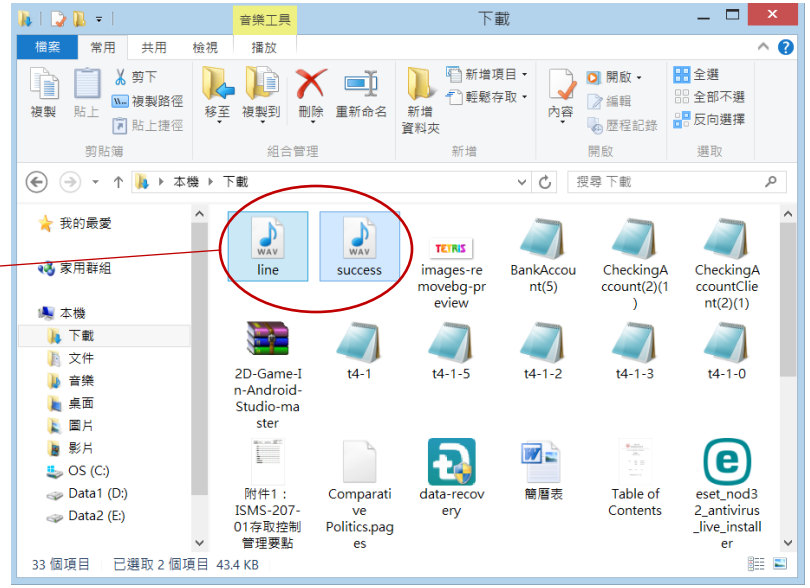
Leaderboard Form

```
public class LeaderboardForm extends javax.swing.JFrame {

    private DefaultTableModel tm;

    private String leaderboardFile = "leaderboard";

    private TableRowSorter<TableModel> sorter;

    public LeaderboardForm() {
        initComponents();
        initTableData();
        initTableSorter();
    }

    private void initTableData(){
        Vector ci = new Vector();
        ci.add("Player");
        ci.add("Score");
        tm = ( DefaultTableModel ) leaderboard.getModel();

        try{
            FileOutputStream fs = new FileOutputStream(leaderboardFile);
            ObjectOutputStream os = new ObjectOutputStream(fs);
            tm.setDataVector( (Vector<Vector>)os.readObject(), ci );

            os.close();
            fs.close();
        }catch(Exception e){}
    }
```



DataBase

Read the Files(Add music)



//Audio Player.java

```java
public class AudioPlayer {
    private String soundsFolder = "tetrissounds" + File.separator;
    private String clearLinePath = soundsFolder + "line.wav";
    private String gameoverPath = soundsFolder + "success.wav";

    private Clip clearLineSound, gameoverSound;

    public AudioPlayer(){
        try {
            clearLineSound = AudioSystem.getClip();
            gameoverSound = AudioSystem.getClip();

            clearLineSound.open(AudioSystem.getAudioInputStream(new File(clearLinePath).getAbsoluteFile()));
            gameoverSound.open(AudioSystem.getAudioInputStream(new File(gameoverPath).getAbsoluteFile()));
        } catch (LineUnavailableException ex) {
            Logger.getLogger(AudioPlayer.class.getName()).log(Level.SEVERE, null, ex);
        } catch (UnsupportedAudioFileException ex) {
            Logger.getLogger(AudioPlayer.class.getName()).log(Level.SEVERE, null, ex);
        } catch (IOException ex) {
            Logger.getLogger(AudioPlayer.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public void playClearLine(){
        clearLineSound.setFramePosition(0);
        clearLineSound.start();
    }

    public void playGameover(){
        gameoverSound.setFramePosition(0);
        gameoverSound.start();
    }
}
```

//Tetris.java

```java
public static void gameOver(int score){
    playGameover();

    String playerName = JOptionPane.showInputDialog("Game Over !\nPlease enter your name.");

    gf.setVisible(false);
    lf.addPlayer(playerName, score);
}
```