Фрагмент кода

draw(Canvas canvas, FieldBlock[] activeFields) - Отрисовка поля

```
public void draw(Canvas canvas, FieldBlock[] activeFields) {
        Paint paint = new Paint();

        int color = Color.BLACK;
        boolean stActive = false;

        if (this.state == 2) color = Color.GRAY;
        else {
                for (int i = 0; i < activeFields.length; i++) {
                        if (activeFields[i].x == this.x && activeFields[i].y == this.y) {
                                color = Color.RED;
                                stActive = true;
                                break;
                        }
                }
        }
        paint.setColor(color);
        canvas.drawRect (this.xPX + tetrisDraw.leftMargin, this.yPX + tetrisDraw.topMargin,
        this.xPX + tetrisDraw.leftMargin + tetrisDraw.blockWidth, this.yPX +
        tetrisDraw.topMargin + tetrisDraw.blockWidth, paint);
        //разметка поля
        if (this.state == 0 && !stActive) {
                paint.setColor(Color.WHITE);
                canvas.drawCircle(this.xPX+tetrisDraw.leftMargin + tetrisDraw.blockWidth/2,
                this.yPX + tetrisDraw.topMargin + tetrisDraw.blockWidth/2, 2, paint);
        }
}
```

KillLine() – очищает полностью заполнившиеся строки

```
public void KillLine() {
        int i, j;
        boolean f;

        for(i = 0; i < glassY; i++){
                f = true;
                for (j = 0; j < glassX; j++){
                        if (this.fields[j][i].state == 0){
                                f = false;
                                break;
                        }
                }
                if (f){
                        for(j = 0; j < glassX; j++) this.fields[j][i].state = 0;
                        for (int l = i-1; l >= 0; l--){
                                for (j = 0; j < glassX; j++){
                                        if (this.fields[j][l].state == 2) {
                                                this.fields[j][l].state = 0;
                                                this.fields[j][l+1].state = 2;
                                        }
                                }
                        }

                        //добавляем N-очков за сокращение линии00
                        this.score += this.killLineBonus;
                }
        }
}
```

setFields() – создание фигуры и её вариаций

```
public void setFields() {
   //положения фигур
   switch (this.id){
   case 0:
    //_
         if (this.points == null) this.points = new FieldBlock[4];
         switch(this.pos){
              case 0:
                   this.points[0] = new FieldBlock(this.point.x - 1, this.point.y);
                   this.points[1] = new FieldBlock(this.point.x, this.point.y);
                   this.points[2] = new FieldBlock(this.point.x + 1, this.point.y);
                   this.points[3] = new FieldBlock(this.point.x + 2, this.point.y);
                   break;
             case 1:
                   this.points[0] = new FieldBlock(this.point.x, this.point.y - 1);
                   this.points[1] = new FieldBlock(this.point.x, this.point.y);
                   this.points[2] = new FieldBlock(this.point.x, this.point.y + 1);
                   this.points[3] = new FieldBlock(this.point.x, this.point.y + 2);
                   break;
             }
             //Log.e ("T1", "F_");
             break;
    case 1:
    //Z
             if (this.points == null) this.points = new FieldBlock[4];
             switch(this.pos){
                case 0:
                   this.points[0] = new FieldBlock(this.point.x - 1, this.point.y);
                   this.points[1] = new FieldBlock(this.point.x, this.point.y);
                   this.points[2] = new FieldBlock(this.point.x, this.point.y + 1);
                   this.points[3] = new FieldBlock(this.point.x + 1, this.point.y + 1);
```

```
            break;
        case 1:
            this.points[0] = new FieldBlock(this.point.x, this.point.y - 1);
            this.points[1] = new FieldBlock(this.point.x, this.point.y);
            this.points[2] = new FieldBlock(this.point.x - 1, this.point.y);
            this.points[3] = new FieldBlock(this.point.x - 1, this.point.y + 1);
            break;
    }
    // Log.e ("T1", "FZ");
    break;
case 2:
    //Γ
    if (this.points == null) this.points = new FieldBlock[4];
    switch(this.pos){
        case 0:
            this.points[0] = new FieldBlock(this.point.x, this.point.y);
            this.points[1] = new FieldBlock(this.point.x - 1, this.point.y);
            this.points[2] = new FieldBlock(this.point.x + 1, this.point.y);
            this.points[3] = new FieldBlock(this.point.x + 1, this.point.y + 1);
            break;
        case 1:
            this.points[0] = new FieldBlock(this.point.x, this.point.y);
            this.points[1] = new FieldBlock(this.point.x, this.point.y - 1);
            this.points[2] = new FieldBlock(this.point.x, this.point.y + 1);
            this.points[3] = new FieldBlock(this.point.x - 1, this.point.y + 1);
            break;
        case 2:
            this.points[0] = new FieldBlock(this.point.x - 1, this.point.y - 1);
            this.points[1] = new FieldBlock(this.point.x - 1, this.point.y);
            this.points[2] = new FieldBlock(this.point.x, this.point.y);
            this.points[3] = new FieldBlock(this.point.x + 1, this.point.y);
            break;
```

```
            case 3:
                this.points[0] = new FieldBlock(this.point.x, this.point.y - 1);
                this.points[1] = new FieldBlock(this.point.x + 1, this.point.y - 1);
                this.points[2] = new FieldBlock(this.point.x, this.point.y);
                this.points[3] = new FieldBlock(this.point.x, this.point.y + 1);
                break;
        }
        // Log.e ("T1", "FΓ");
        break;
    case 3:
        //[]
        if (this.points == null) this.points = new FieldBlock[4];
        this.points[0] = new FieldBlock(this.point.x - 1, this.point.y);
        this.points[1] = new FieldBlock(this.point.x, this.point.y);
        this.points[2] = new FieldBlock(this.point.x - 1, this.point.y + 1);
        this.points[3] = new FieldBlock(this.point.x, this.point.y + 1);
        // Log.e ("T1", "F[]");
        break;
  }
}
```