

Комитет по образованию г. Санкт-Петербург

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

**ПРЕЗИДЕНТСКИЙ ФИЗИКО-МАТЕМАТИЧЕСКИЙ
ЛИЦЕЙ №239**

**Отчет о практике
«Создание графической игры на Java»**

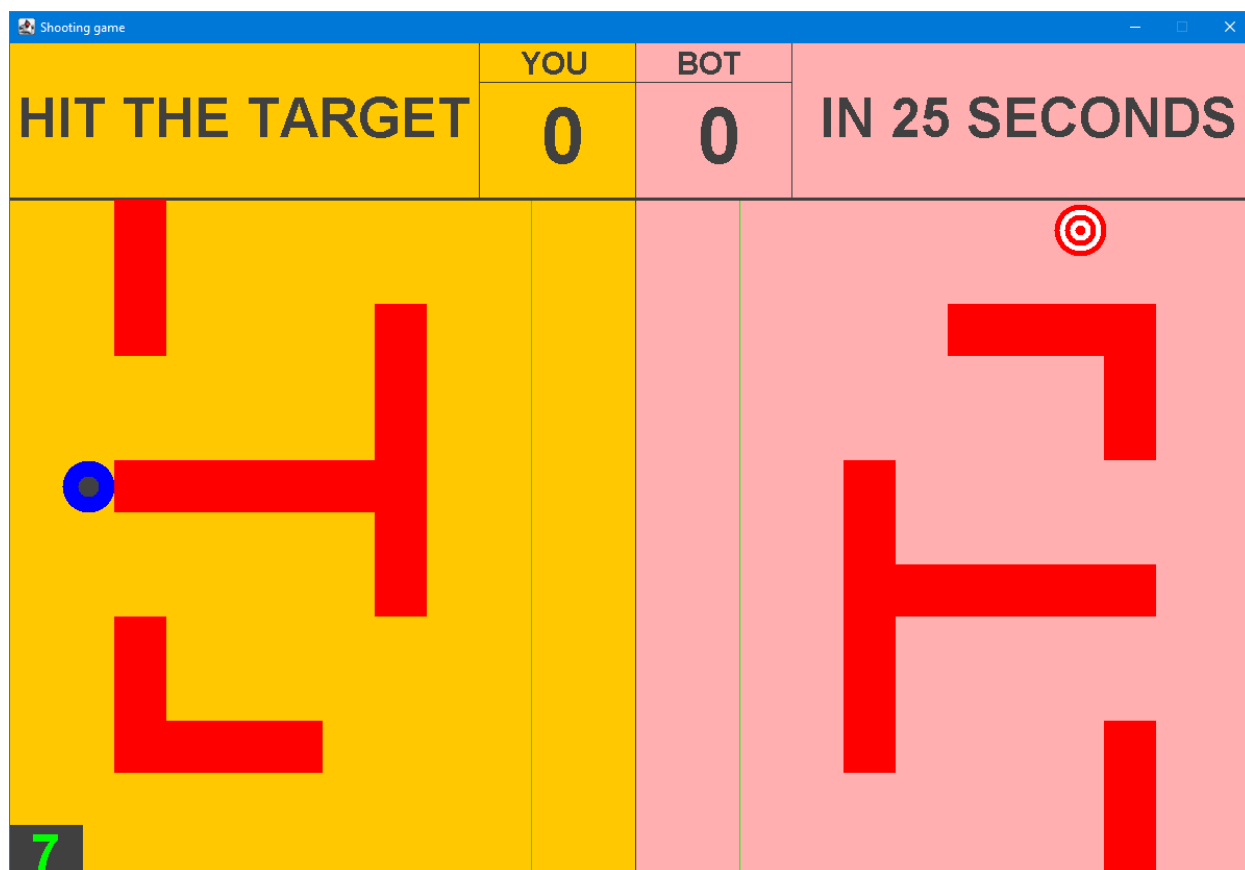
Учащийся 10-3 класса
Борисов П.Г.

Преподаватель:
Клюнин А.О.

Санкт-Петербург – 2022 год

1. Постановка задачи

Требуется создать окно, нарисовать карту, игрока и мишень, расписать физику полета пули и вывести окна-«помощники», подсказывающие, что делать, чтобы продолжить игру или начать ее, создать счет и таймер



2. Элементы управления

В рамках данной задачи необходимо было реализовать следующие элементы управления:

На кнопки w-вверх

A-влево

S-вниз

D-вправо

Осуществляется управление

На кнопки e,r,f осуществляется стрельба вверх вправо, вправо, вниз вправо соответственно

На enter начинается новый раунд



3. Рисование

Чтобы нарисовать карту, состоящую из прямоугольных блоков и прямоугольного заднего фона, использовалась команда **g.fillRect()**;

Чтобы нарисовать игрока, мишень и пулю, использовалась команда **g.fillOval()**;

Чтобы нарисовать счет и таймер, использовалась команда **g.drawString()**;

Чтобы нарисовать барьеры стрельбы и ограничители карты сверху, использовалась команда **g.drawLine()**;

```
@Override
public void paint(Graphics g) {
    //fields
    g.setColor(Color.ORANGE);
    g.fillRect(0, 0, 600, 801);
    g.setColor(Color.PINK);
    g.fillRect(601, 0, 601, 801);
    //line between fields
    g.setColor(Color.DARK_GRAY);
    g.drawLine(600, 0, 600, 801);
    //score bottom line
    g.setColor(Color.DARK_GRAY);
    g.drawLine(450, 150, 750, 150);
    //score left n right borders
    g.setColor(Color.DARK_GRAY);
    //left
    g.drawLine(450, 0, 450, 150);
    //right
    g.drawLine(750, 0, 750, 150);
    //your score writing
    g.setColor(Color.DARK_GRAY);
    Font f1 = new Font("Arial", Font.BOLD, 30);
    g.setFont(f1);
    g.drawString("YOU", 490, 30);
    //bot's score writing
    g.setFont(f1);
    g.drawString("BOT", 640, 30);
    //writings underline
    g.setColor(Color.DARK_GRAY);
    g.drawLine(450, 37, 750, 37);
    //bot and players score
    Font f = new Font("Arial", Font.BOLD, 75);
    //you
    g.setFont(f);
    g.drawString(String.valueOf(playerscore), 510, 115);
    //bot
    g.setFont(f);
    g.drawString(String.valueOf(botscore), 660, 115);
    //timer on screen
    Font f7 = new Font("Arial", Font.BOLD, 50);
    g.setColor(Color.darkGray);
    g.fillRect(0, 750, 70, 50);
    if(timeonscreen<=19){
        g.setColor(Color.green);
    }
    if(timeonscreen>=20 && timeonscreen<=25){
        g.setColor(Color.red);
    }

    g.setFont(f7);
    if(timeonscreen<10) {
        g.drawString(String.valueOf(timeonscreen), 20, 79)
    }
    if(timeonscreen>=10){
        g.drawString(String.valueOf(timeonscreen), 5, 795)
    }
    //shotborders
    g.setColor(Color.GREEN);
    g.drawLine(500, 150, 500, 800);
    g.drawLine(700, 150, 700, 800);
    g.setColor(Color.DARK_GRAY);
    g.fillRect(0, 148, 1200, 3);
    Font f2 = new Font("Arial", Font.BOLD, 60);
    Font f3 = new Font("Arial", Font.BOLD, 53);
    g.setFont(f3);
    g.drawString("HIT THE TARGET", 7, 90);
    g.setFont(f3);
    g.drawString("IN 25 SECONDS", 777, 90);
    //map
    g.setColor(Color.RED);
    g.fillRect(100, 150, 50, 150);
    g.fillRect(100, 400, 300, 50);
    g.fillRect(350, 250, 50, 300);
    g.fillRect(100, 550, 50, 150);
    g.fillRect(100, 650, 200, 50);

    g.fillRect(1050, 650, 50, 150);
    g.fillRect(800, 500, 300, 50);
    g.fillRect(800, 400, 50, 300);
    g.fillRect(900, 250, 200, 50);
    g.fillRect(1050, 250, 50, 150);
    //player
    if(playerspawn==false) {
        g.setColor(Color.BLUE);
        playerx = playerxold;
        playery = playeryold;
        g.fillOval(playerx, playery, 50, 50);
        playerspawn=true;
    }
    if(playerspawn==true){
        g.setColor(Color.BLUE);
        g.fillOval(playerx, playery, 50, 50);
    }
}
```

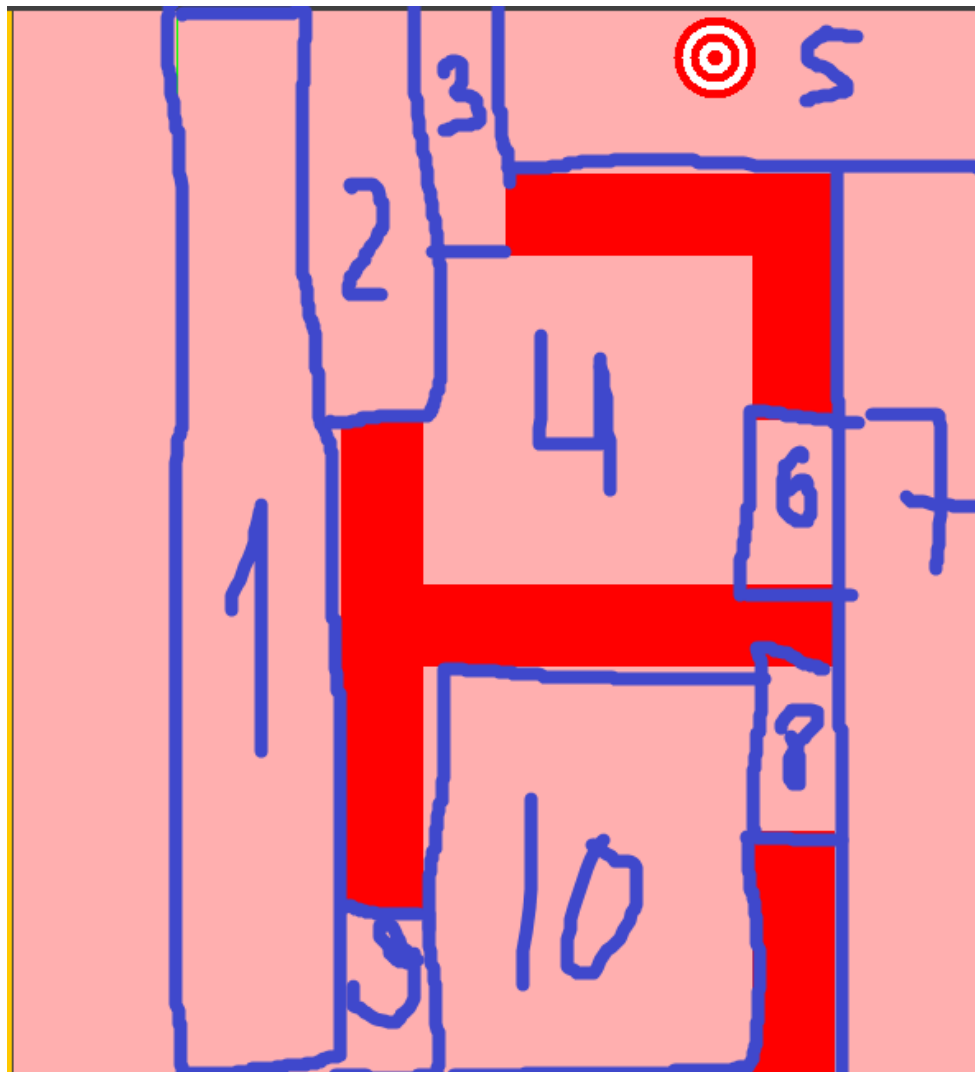
```

//player's bullet
g.setColor(Color.RED);
g.fillOval(botx, boty, 50, 50);
g.setColor(Color.white);
g.fillOval(botx+5, boty+5, 40, 40);
g.setColor(Color.RED);
g.fillOval(botx+10, boty+10, 30, 30);
g.setColor(Color.white);
g.fillOval(botx+15, boty+15, 20, 20);
g.setColor(Color.RED);
g.fillOval(botx+20, boty+20, 10, 10);
//player's bullet
g.setColor(Color.DARK_GRAY);
if(pshoot==false) {
    pbulletx = playerx + 15;
    pbullety = playery + 15;
}
g.fillOval(pbulletx,pbullety,20,20);
//bot's bullet
/*
if(bshoot==false) {
    bbulletx = botx + 15;
    bbulety = boty + 15;
}
g.fillOval(bbulletx,bbulety,20,20);
*/
//dialogue window
if(botscore==5){
    menuOpacityf = 255;
    g.setColor(new Color(139, 0, 255, menuOpacityf));
    g.fillRect(300, 400, 600, 150);
    g.setColor(new Color(255, 140, 0, menuOpacityf));
    Font f4 = new Font("Arial", Font.BOLD, 53);
    g.setFont(f4);
    g.drawString("you are loseer", 333, 485);
}
if(playerscore==5){
    menuOpacityf=255;
    g.setColor(new Color(139, 0, 255, menuOpacityf));
    g.fillRect(300, 400, 600, 150);
    g.setColor(new Color(255, 140, 0, menuOpacityf));
    Font f4 = new Font("Arial", Font.BOLD, 53);
    g.setFont(f4);
    g.drawString("gg", 515, 485);
}

```

4. Решение задачи

Для генерации мишени карта разбита на 10 квадрантов, и на каждом из них равновероятно появляется мишень



Для отскоков пули от стенок использовано изменение нужной скорости пули на отрицательную в зависимости от места столкновения со стенкой

```
@Override
public void actionPerformed(ActionEvent e) {
    repaint();
    if(pshoot==true){
        if (playerx<400){
            pshoot=false;
        }
    }
    else {
        pbulletx += bulletspeedx;
        pbulleyy += bulletspeedy;
        //screenborders bounceoff
        if (pbulletx > 1180 || pbulletx < 0) {
            bulletspeedx *= -1;
            pbounceoff += 1;
        }
        if (pbulleyy < 150 || pbulleyy > 780) {
            bulletspeedy *= -1;
            pbounceoff += 1;
        }
        //vertical bounceoffs
        //mapborders bounceoff botside forward
        if ((pbulleyy > 380 && pbulleyy < 700 && pbulletx > 780 && pbulletx < 790) || (pbulleyy > 230 && pbulleyy < 300 && pbulletx > 880 && pbulletx < 890) || (pbulletx > 695 && pbulletx < 705)) {
            bulletspeedx *= -1;
            pbounceoff += 1;
        }
        //mapborders bounceoff botside backward
        if ((pbulleyy > 380 && pbulleyy < 700 && pbulletx > 840 && pbulletx < 850) || (pbulleyy > 480 && pbulleyy < 550 && pbulletx > 1090 && pbulletx < 1100) || (pbulletx > 540 && pbulletx < 550 && pbulletx > 830 && pbulletx < 1100)) {
            bulletspeedx *= -1;
            pbounceoff += 1;
        }
        //horizontal bounceoffs
        //mapborders bounceoff botside upward
        if ((pbulletx > 230 && pbulletx < 240 && pbulletx > 880 && pbulletx < 1100) || (pbulletx > 380 && pbulletx < 390 && pbulletx > 780 && pbulletx < 850) || (pbulletx > 695 && pbulletx < 705)) {
            bulletspeedy *= -1;
            pbounceoff += 1;
        }
        //mapborders bounceoff botside downward
        if ((pbulletx > 690 && pbulletx < 700 && pbulletx > 780 && pbulletx < 850) || (pbulletx > 540 && pbulletx < 550 && pbulletx > 830 && pbulletx < 1100) || (pbulletx > 695 && pbulletx < 705)) {
            bulletspeedy *= -1;
            pbounceoff += 1;
        }
        if (pbulletx > 695 && pbulletx < 705) {
            pbulletcrossborder += 1;
        }
        if (pbulletcrossborder == 2) {
            pbounceoff = 14;
            pbulletcrossborder = 0;
        }
    }
}
```

```
//player's bullet
g.setColor(Color.DARK_GRAY);
if(pshoot==false) {
    pbulletx = playerx + 15;
    pbullety = playery + 15;
}

if (new Rectangle(pbulletx, pbullety, 20, 20).intersects(new Rectangle(botx, boty, 50, 50))) {
    playerscore += 1;
    time.stop();
    botspawn=false;
    playerspawn=false;
    procTime.restart();
    timetablet.restart();
    timeonscreen=0;
    pshoot = false;
}
```

5. Проверка

Для проверки правильности решённой задачи игра запускалась много раз с разных устройств. Проверка показала, что игра работает без сбоев.

6. Заключение

В рамках выполнения поставленной задачи была создана игра с полностью выполненным требуемым функционалом. Правильность работы проверена многочисленными запусками и проверкой всех действий.