действия выполняются в конструкторе. Под переменными класса добавим конструктор класса и первым делом загрузим изображения в игру:

```
// Конструктор класса
public pole()
{
  // Попытка загрузки всех изображений для игры
  try
  {
      fon = ImageIO.read(new File("c:\\fon.png"));
      paluba = ImageIO.read(new File("c:\\paluba.png"));
      ranen = ImageIO.read(new File("c:\\ranen.png"));
      ubit = ImageIO.read(new File("c:\\ubit.png"));
      end1 = ImageIO.read(new File("c:\\end1.png"));
          end2 = ImageIO.read(new File("c:\\end2.png"));
          bomba = ImageIO.read(new File("c:\\bomba.png"));
  }
  catch (Exception ex) {}
}
```

В процессе разработки все изображения находятся в корне диска **C:**. При успешной загрузке изображений – они станут доступны в игре. После загрузки изображений создаем, настраиваем и запускаем таймер отрисовки игрового поля. Программный код для таймера будет также внутри конструктора, под загрузкой изображений:

```
// Создаем, настраиваем и запускаем таймер
// для отрисовки игрового поля
tmDraw = new Timer(50,new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent arg0) {
                        // Вызываем перерисовку - paintComponent()
                        repaint();
                    }
                });
tmDraw.start();
```

Отображение фона, сеток игровых полей из линий, выведение надписей мы будем выполнять в методе **paintComponent()**. Две кнопки **Новая игра** и **Выход** добавим в конструкторе класса под кодом для таймера. Полный код конструктора класса **pole** будет выглядеть так:

```
// Конструктор класса
    public pole()
    {
// Попытка загрузки всех изображений для игры
try
```

```
{
    fon = ImageIO.read(new File("c:\\fon.png"));
    paluba = ImageIO.read(new File("c:\\paluba.png"));
    ranen = ImageIO.read(new File("c:\\ranen.png"));
    ubit = ImageIO.read(new File("c:\\ubit.png"));
    end1 = ImageIO.read(new File("c:\\end1.png"));
        end2 = ImageIO.read(new File("c:\\end2.png"));
        bomba = ImageIO.read(new File("c:\\bomba.png"));
}
catch (Exception ex) {}

// Создаем, настраиваем и запускаем таймер
// для отрисовки игрового поля
tmDraw = new Timer(50,new ActionListener() {
                    @Override
                    public void actionPerformed(ActionEvent arg0) {
                            // Вызываем перерисовку - paintComponent()
                            repaint();
                    }
                });
 tmDraw.start();

 // Включаем возможность произвольного размещения
 // элементов интерфейса на панели
 setLayout(null);

 // Создаем кнопку Новая игра
 btn1 = new JButton();
 btn1.setText("Новая игра");
 btn1.setForeground(Color.BLUE);
 btn1.setFont(new Font("serif",0,30));
 btn1.setBounds(130, 450, 200, 80);
 btn1.addActionListener(new ActionListener() {
        // Обработчик события при нажатии на кнопку Новая игра
                    public void actionPerformed(ActionEvent arg0) {

                    }
                });
  add(btn1);

  // Создаем кнопку Выход
```

```
btn2 = new JButton();
btn2.setText("Выход");
btn2.setForeground(Color.RED);
btn2.setFont(new Font("serif",0,30));
btn2.setBounds(530, 450, 200, 80);
btn2.addActionListener(new ActionListener() {
    // Обработчик события при нажатии на кнопку Новая игра
                public void actionPerformed(ActionEvent arg0) {
                // Выход их игры - завершение работы приложения
                        System.exit(0);
                }
            });
add(btn2);
    }
```

Если выполнить запуск, то мы увидим следующее **(см. рис. 26)**.

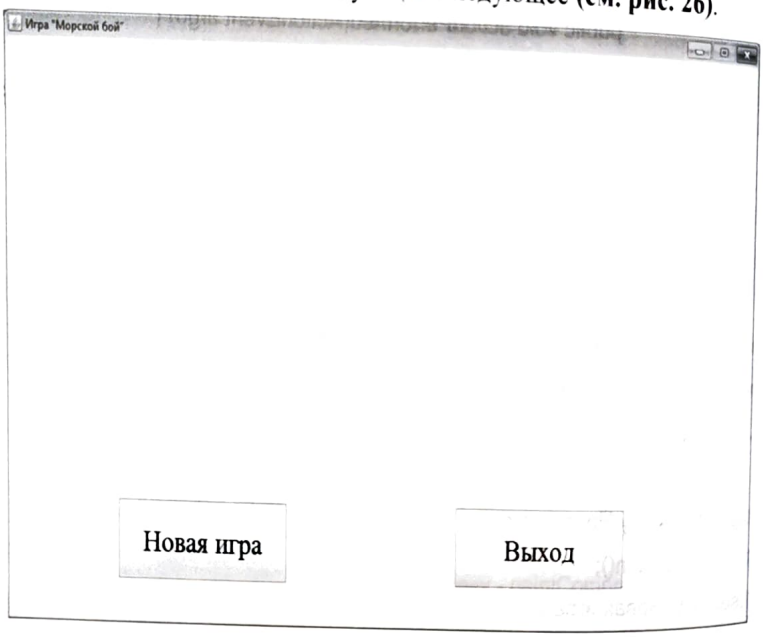

Рис. 26

Добавление кнопок происходит привычным способом. Точно также были добавлены кнопки в игре "Змейка". Обработчик события для кнопки **Новая игра** на первом уровне сложности ничего не выполняет.

Остальные действия для рисования игрового поля мы выполним, как и обычно, в методе **paintComponent()**. Добавим его под конструктором класса.

```
// Метод отрисовки
    public void paintComponent(Graphics gr)
    {
        // Очищение игрового поля
        super.paintComponent(gr);
}
```