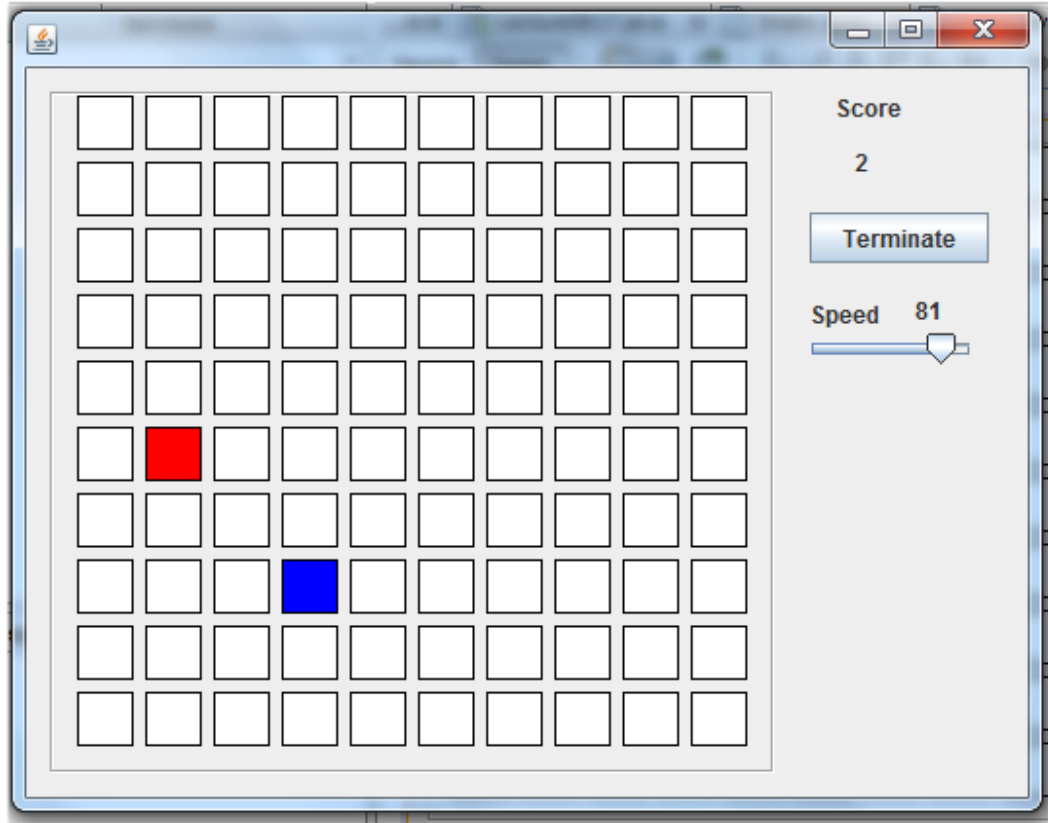


Fundamentals of Programming II Assignment 07 Snake Phase 02



```
int board[][] = {{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {1, 0, 0, 0, 0, 0, 0, 0, 0, 0}};
int targets[][] = {{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0}};
boolean target=false;
String direction = "RIGHT";
int delay = 1000;
Timer t;
```

```
boolean checkWin()
{
    for(int r=0;r<10;r++)
    {
        for(int c=0;c<10;c++)
        {
            if(targets[r][c]==1)
            {
                if(board[r][c]==1)
                {
                    target=false;
                    targets[r][c]=0;
                    Score.setText(Integer.toString(Integer.parseInt(Score.getText()+1)));
                    return true;
                }
            }
        }
    }
    return false;
}
```

```
void renderTargets()
{
    if(target==false)
    {
        Random rand = new Random();
        int r = rand.nextInt(10);
        int c=rand.nextInt(10);
        targets[r][c]=1;
        target=true;
    }

    {
        if (targets[0][0]==1)_00.setBackground(Color.blue);
        if (targets[0][1]==1)_01.setBackground(Color.blue);

        if (targets[9][8]==1)_98.setBackground(Color.blue);
        if (targets[9][9]==1)_99.setBackground(Color.blue);
    }
}
```

```
void renderBoard() {
    if(board[0][0]==0)_00.setBackground(Color.white); else _00.setBackground(Color.red);
    if(board[0][1]==0)_01.setBackground(Color.white); else _01.setBackground(Color.red);

    if(board[9][8]==0)_98.setBackground(Color.white); else _98.setBackground(Color.red);
    if(board[9][9]==0)_99.setBackground(Color.white); else _99.setBackground(Color.red);
}
```

```
void updateBoard() {
    int r, c;
    if (direction.equals("LEFT")) {
        for (r = 0; r < 10; r++) {
            for (c = 9; c >= 0; c--) {
                if (board[r][c] == 1) {
                    int num = c;
                    board[r][num] = 0;
                }
            }
        }
    }
}
```

```

        num--;
        if (num == -1) {
            num = 9;
        }
        board[r][num] = 1;
        break;
    }
}

if (direction.equals("RIGHT")) {
    for (r = 0; r < 10; r++) {
        for (c = 0; c < 10; c++) {
            if (board[r][c] == 1) {
                board[r][c] = 0;
                board[r][(c + 1) % 10] = 1;
                break;
            }
        }
    }
}

if (direction.equals("UP")) {
    for (c = 0; c < 10; c++) {
        for (r = 9; r >= 0; r--) {
            if (board[r][c] == 1) {
                int num = r;
                board[num][c] = 0;
                num--;
                if (num == -1) {
                    num = 9;
                }
                board[num][c] = 1;
                break;
            }
        }
    }
}

if (direction.equals("DOWN")) {
    for (c = 0; c < 10; c++) {
        for (r = 0; r < 10; r++) {
            if (board[r][c] == 1) {
                board[r][c] = 0;
                board[(r + 1) % 10][c] = 1;
                break;
            }
        }
    }
}

boolean b=checkWin();
renderBoard();
renderTargets();
}

```

```

private void formKeyPressed(java.awt.event.KeyEvent evt) {
    int r,c;
    switch (evt.getKeyCode())
    {
        case KeyEvent.VK_LEFT:
            direction="LEFT";
            break;
    }
}

```



```
case KeyEvent.VK_RIGHT:
    direction="RIGHT";
    break;
case KeyEvent.VK_UP:
    direction="UP";
    break;
case KeyEvent.VK_DOWN:
    direction="DOWN";
    break;
}
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    delay=(101-jSlider1.getValue())*20;
    if(jButton1.getText().equals("Start"))
    {
        t= new Timer(delay,new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                updateBoard();
            }
        });
        t.start();
        jButton1.setText("Terminate");
        jSlider1.disable();
    }
    else if(jButton1.getText().equals("Terminate"))
    {
        t.stop();
        jButton1.setText("Start");
        jSlider1.enable();
    }
}
```

```
private void jSlider1StateChanged(javax.swing.event.ChangeEvent evt) {
    Speed.setText(Integer.toString(jSlider1.getValue()));
}
```