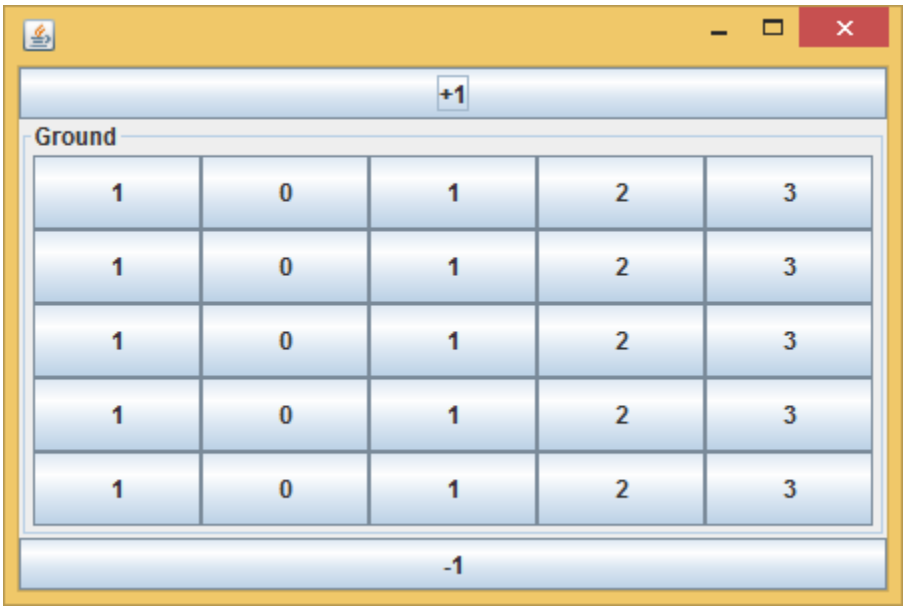
BÀI THỰC HÀNH
MA TRẬN SỐ MÔ HÌNH HÓA GIAO DIỆN

Bài 1

Yêu cầu:

Viết chương trình xây dựng giao diện như hình vẽ.



Khi ấn vào nút +1, tất cả các giá trị trong gound sẽ tăng lên 1.

Khi ấn vào nút -1, tất cả các giá trị trong gound sẽ giảm đi 1.

Khi click vào một nút bất kì trong gound, giá trị trong nút đó sẽ tăng lên 1.

Hướng dẫn:

1. Sử dụng một mảng hai chiều lưu giá trị các số trong các nút của gound.

```
private int[][] matrix = { { 1, 0, 1, 2, 3 }, { 1, 0, 1, 2, 3 },
                  { 1, 0, 1, 2, 3 }, { 1, 0, 1, 2, 3 }, { 1, 0, 1, 2, 3 } };
```

2. Dựa theo mảng hai chiều này, hiển thị các nút trong ground theo dữ liệu này.

i. Khởi tạo:

```
private void initButton() {
	buttons = new JButton[matrix.length][matrix[0].length];

	ActionListener al = new MyButtonEvent();

	for (int i = 0; i < matrix.length; i++) {
		for (int j = 0; j < matrix[i].length; j++) {
			buttons[i][j] = new JButton(String.valueOf(matrix[i][j]));

			buttons[i][j].addActionListener(al);
			panel.add(buttons[i][j]);
		}
	}
}
```

ii. Cập nhật (mỗi khi dữ liệu thay đổi, phần hiển thị sẽ thay đổi theo):

```
private void updateAllButtons() {
	for (int i = 0; i < matrix.length; i++) {
		for (int j = 0; j < matrix[i].length; j++) {
			buttons[i][j].setText(String.valueOf(matrix[i][j]));
		}
	}
}
```

3. Khi xử lý sự kiên, có hai bước cần thực hiện:

i. Cập nhật ma trận số mô tả dữ liệu:

```
//Cập nhật một ô
private void updateCell(int i, int j){
	matrix[i][j]++;
}

//Cập nhật cả ma trận tăng 1
private void updateMatrixByAdd(){
	for (int i = 0; i < matrix.length; i++) {
		for (int j = 0; j < matrix[i].length; j++) {
			matrix[i][j]++;
		}
	}
}

//Cập nhật cả ma trận giảm 1
private void updateMatrixBySub(){
	for (int i = 0; i < matrix.length; i++) {
		for (int j = 0; j < matrix[i].length; j++) {
			matrix[i][j]--;
		}
	}
}
```

ii. Cập nhật hiển thị theo ma trận số bằng cách gọi đến phương thức Cập nhật (mỗi khi dữ liệu thay đổi, phần hiển thị sẽ thay đổi theo): `private void updateAllButtons().`

4. Lưu ý: Cách xác định vị trí của button vừa click.
   i. Xác định nút vừa được bấm:

```
JButton button = (JButton) e.getSource();
```

   ii. Xác định chỉ số của nút bấm (tương ứng với giá trị trong ma trận số)

```
int order = panel.getComponentZOrder(button);
int row = (order) / matrix[0].length;
int col = (order) % matrix[0].length;
```

5. Tham khảo chương trình: NumberMatrix

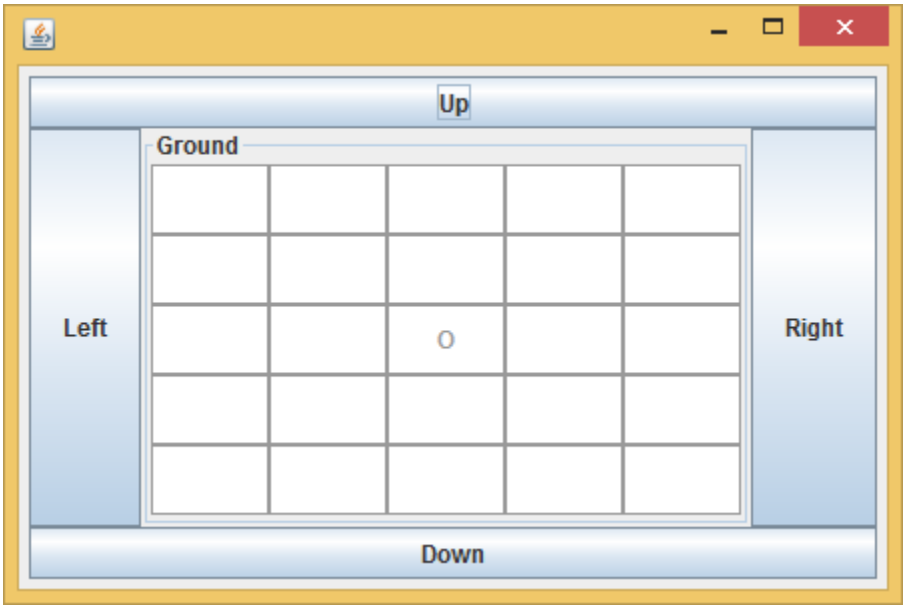
Nâng cao

1. Viết riêng các thông tin mô tả trong một lớp riêng có nhiệm vụ chứa tất cả các thông tin về game.
2. Xây dựng trò chơi Dò mìn, Kim cương,...

Bài 2

Yêu cầu

Xây dựng chương trình có giao diện như trên



Trên ground có một viên bi.

Khi chọn các nút Up, Down, Left, Right viên bi lần lượt chuyển sang các ô bên trên, dưới, trái, phải tương ứng. Nếu đã đến sát cạnh, không thể di chuyển, viên bi sẽ đứng yên.

Hướng dẫn:

1. Lưu thông tin về sân

```
private int maxCol = 4;
private int maxRow = 4;
```

2. Các ô trên sân là một mảng các nút

```
private JButton[][] buttons;
```

3. Vị trí viên bi

```
private int row = 2;
private int col = 2;
```

4. Mỗi khi có một lệnh, vị trí của viên bi sẽ được cập nhật tương ứng:

```
protected void moveRight() {
	// TODO Auto-generated method stub
	removeBall();
	if (col < maxCol)
		col++;
	addBall();
}

protected void moveLeft() {
	// TODO Auto-generated method stub
	removeBall();
	if (col > 0)
		col--;
	addBall();
}

protected void moveDown() {
	// TODO Auto-generated method stub
	removeBall();
	if (row < maxRow)
		row++;
	addBall();
}

protected void moveUp() {
	// TODO Auto-generated method stub
	removeBall();
	if (row > 0)
		row--;
	addBall();
}
```

5. Việc cập nhật hiển thị của viên bi được thay đổi thông qua hai hàm

```
//Thêm ball vào ô cần hiển thị. Gọi sau khi thay đổi vị trí ball.

private void addBall() {
	buttons[row][col].setText("O");
}

//Xóa ball khỏi ô đang hiển thị. Gọi trước khi thay đổi vị trí ball.
private void removeBall() {
	buttons[row][col].setText("");
}
```

6. Tham khảo: BallMoving.

Nâng cao

1. Thêm vào thông tin về chướng ngại vật trên sân (ball không thể di chuyển đến)
2. Game Tank.