

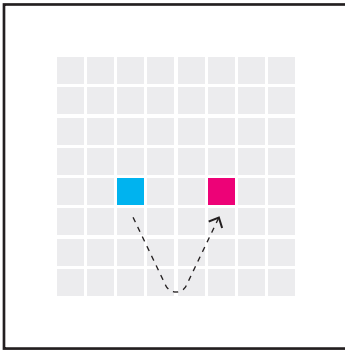
# WORKSHEET W012

## ANGLE

Again we work with animations.  
The goal is to write a program with  
a pixel flying around.

### EXERCISE

Program a pixel that bounces off  
the walls according to the princi-  
ple: «angle of incidence equals angle  
of precipitation».

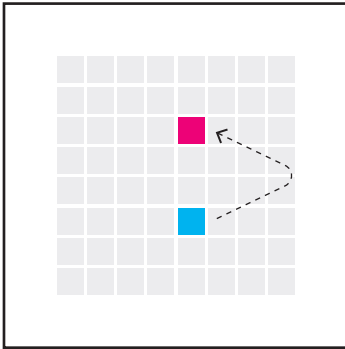


**Blue pixel:**

xDirection = 1;  
yDirection = 1;

**Red pixel:**

xDirection = 1;  
yDirection = -1;



**Blue pixel:**

xDirection = 1;  
yDirection = -1;

**Red pixel:**

xDirection = -1;  
yDirection = -1;

### HINT

Define position and direction, build  
the «if/then» blocks and put them  
in the loop. The rest is appended  
below in logical order.

#### LOGIC

Do forever (1x)

If (2x)  
then

or (2x)

yPosition = 0 (1x)

xPosition = 0 (1x)

yPosition = 7 (1x)

xPosition > 7 (1x)

#### VARIABLES

Set xPosition to 4 (1x)

Set yPosition to 6 (1x)

Set xDirection to xDirection \* -1 (1x)

Set yDirection to yDirection \* -1 (1x)

Set xPosition to xPosition + xDirection (1x)

Set yPosition to yPosition + yDirection (1x)

Set xDirection to 1 (1x)

Set yDirection to 1 (1x)

#### MATRIX

Set random pixel color (1x)

Delete all pixels (1x)

Draw pixel xPosition yPosition (1x)

#### TIME

Wait 50 milliseconds (1x)

#### PARTS LIST LIST OF BLOCKS TO BE USED



LEVEL EXPERT

# SOLUTION

PROPOSED SOLUTION

The code consists of the following blocks:

- Set `xPosition` to `4`
- Set `yPosition` to `6`
- Set `xDirection` to `1`
- Set `yDirection` to `1`
- Do forever loop:
  - Set `xPosition` to `xPosition + xDirection`
  - Set `yPosition` to `yPosition + yDirection`
  - If `yPosition = 7` or `yPosition = 0`, then Set `yDirection` to `yDirection * -1`
  - If `xPosition = 7` or `xPosition = 0`, then Set `xDirection` to `xDirection * -1`
  - Set random pixel color
  - Draw pixel `xPosition` `xPosition`
  - Wait `50` milliseconds
  - Delete all pixels

# WORKSHEET

## W012

### ANGLE

#### Learning objective:

Capacity to know the «and/or» block and to be able to build a more complex program.

## WHAT TO DO

1.  
First the variables xPosition, yPosition, xDirection and yDirection are defined.
2.  
Next up is the «Do forever» block.
3.  
Then you insert the two additions into the repetition.
4.  
Now position and assemble the «if/then» blocks. Place the «or» blocks at the back according to the «puzzle principle». In the «then» column the direction changes are set.
5.  
Finally, the pixel should light up in a random color for 50 milliseconds and be deleted again.

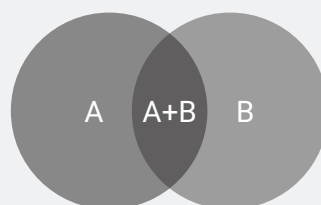
## NEW COMMANDS



With «and» or «or» we can logically extend conditions.

## ADDITIONAL INFORMATION: AND/OR

With the module «or» or «and» you have many new possibilities. With «or» you can program two different conditions which can apply individually. With «and» two different conditions must apply at the same time so that an overall condition applies.



The overall condition «A and B» is only true if both conditions are true.



The overall condition «A or B» is true as soon as one of the two conditions is true.