

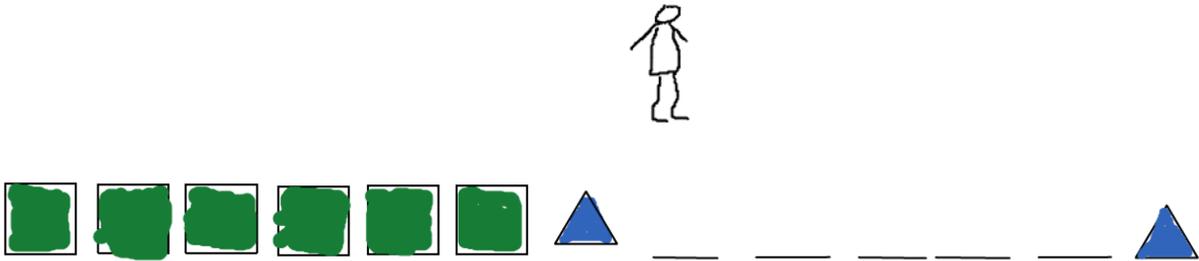
**General template to build a detailed  
sheet for each puzzle and problems  
treated by cMinds**

## The problem

Title of the problem: **Freezes**

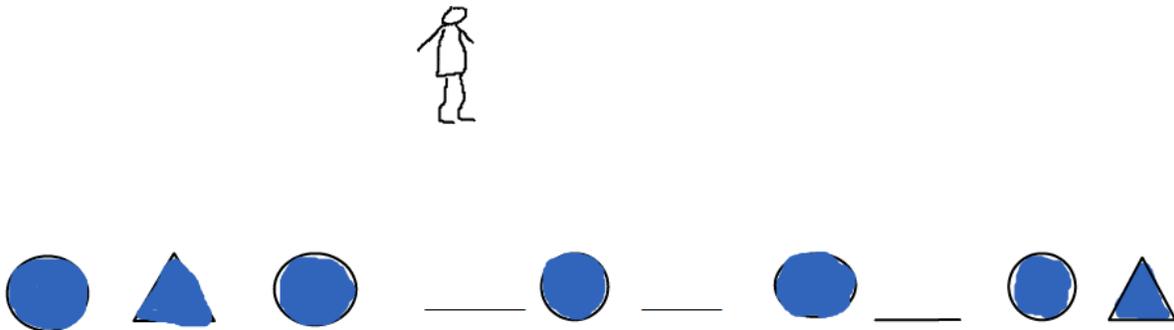
//Long description of the problem. Be as detailed as possible.

### LEVEL 1



Help the robot fill the empty spaces with the correct figures so that a pattern to be produced. The robot is on the position indicated on the screen (on the first blank space). Can you recognize the pattern? Can you instruct the robot to produce this pattern?

### LEVEL 2



Help the robot fill the empty spaces with the correct figures so that a pattern to be produced. The robot is on the position indicated on the screen (on the first blank space). Can you recognize the pattern? Can you instruct the robot to produce this pattern?

**LEVEL 3**



Help the robot fill the empty spaces with the correct figures so that a pattern to be produced. The robot is on the position indicated on the screen (on the first blank space). Can you recognize the pattern? Can you instruct the robot to produce this pattern?

**LEVEL 4**



Help the robot fill the empty spaces with the correct figures so that a pattern to be produced. The robot is on the position indicated on the screen (on the first blank space). Can you recognize the pattern? Can you instruct the robot to produce this pattern?

**LEVEL 5**



Help the robot fill the empty spaces with the correct figures so that a pattern to be produced. The robot is on the position indicated on the screen (at the beginning of the pattern). Can you recognize the pattern? Can you instruct the robot to produce this pattern?

## Data

//All the numerical parameters of the problem.

### LEVEL 1

**2 types of figures: squares, triangles.**

**13 figures in total: 11 green squares, 2 blue triangles from which 6 figures are missing: 6 green squares**

**1 robot**

**6 blank spaces**

### LEVEL 2

**2 types of figures: triangles, circles.**

**10 figures in total: 5 blue circles, 5 blue triangles from which 4 figures are missing: 2 blue triangles and 2 pink triangles.**

**1 robot**

**4 blank spaces**

### LEVEL 3

**2 types of figures: triangles, circles.**

**9 figures in total: 5 blue circles, 4 yellow triangles from which 3 figures are missing: 2 blue circles, 1 yellow triangle**

**1 robot**

**3 blank spaces**

#### **LEVEL 4**

**2 types of figures: triangles, circles.**

**12 figures in total: 3 pink circles, 3 pink triangles, 3 blue circles, 3 pink triangles from which 3 figures are missing: 2 blue circles, 1 yellow triangle**

**1 robot**

**3 blank spaces**

#### **LEVEL 5**

**3 types of figures: triangles, circles, squares.**

**9 figures in total: 3 blue circles, 3 blue triangles, 3 blue squares from which 3 figures are missing: 1 circle, 1 triangle, 1 square**

**1 robot**

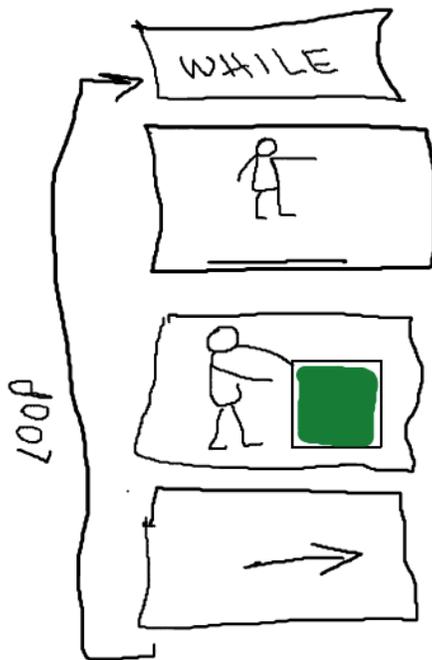
**3 blank spaces**

Solution

We need to describe here the most ideal solution for the problem, both with a simple algorithm and with a graphical solution

LEVEL 1

### 1. Graphical



### 2. Algorithmic - Pseudocode

```
While { on a blank space}{
```

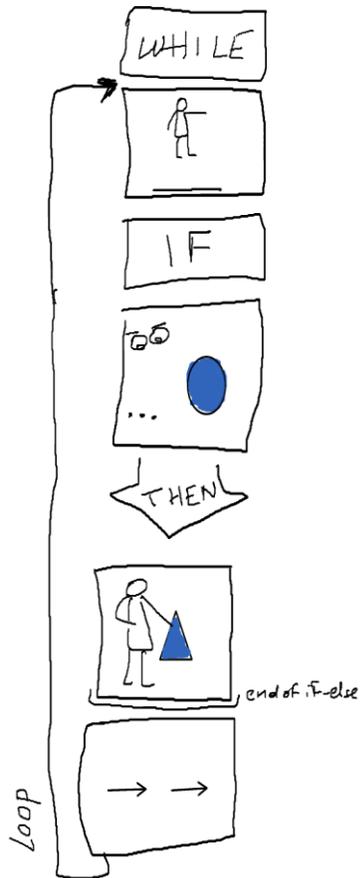
```
Place a green square
```

```
Move one step
```

```
}
```

## LEVEL 2

### 3. Graphical



### 4. Algorithmic - Pseudocode

**While** (on a blank space){

**IF** (the last figure is a blue circle) **then**

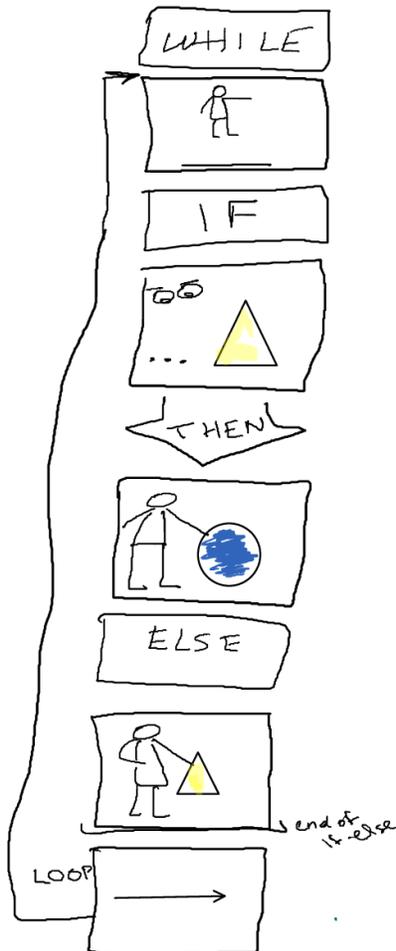
{place a blue circle}

Move 2 steps

}

LEVEL 3

## 5. Graphical



## 6. Algorithmic - Pseudocode

**While** (on a blank space){

**IF** (the last figure is a yellow triangle) **then**

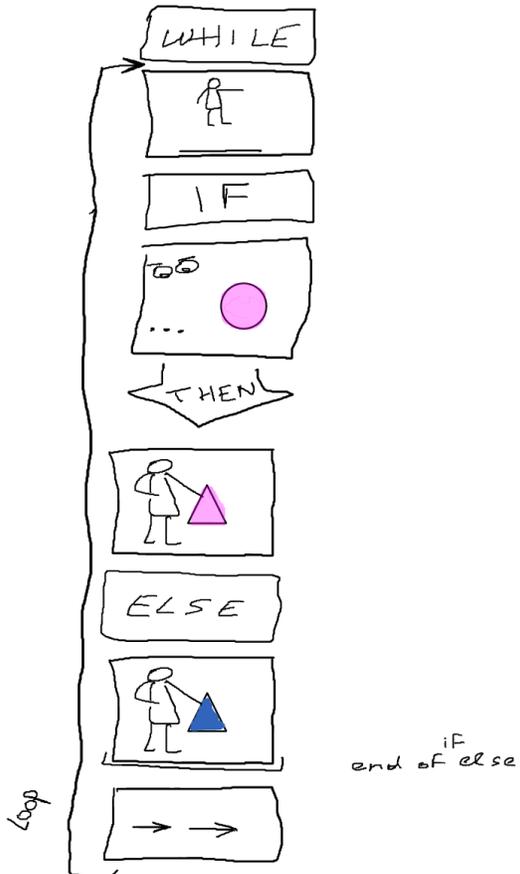
{place a blue circle}

**Else** {place a yellow triangle}

Move 1 step }

## LEVEL 4

### 7. Graphical

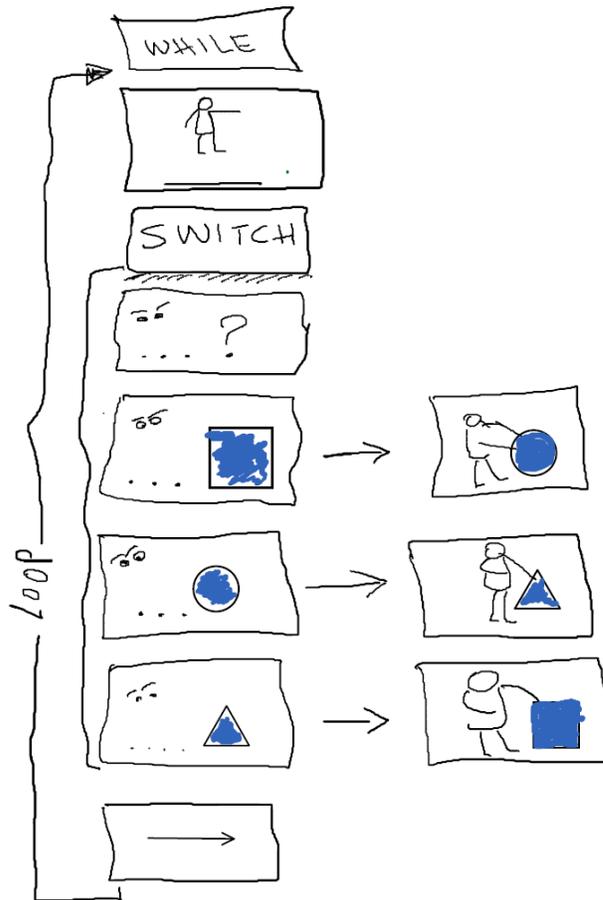


### 8. Algorithmic

```
While { on a blank space } {  
  IF {last figure is a pink circle} then  
  { place a pink triangle}  
  Else {place a blue triangle}  
  Move two steps  
}
```

LEVEL 5

## 9. Graphical



## 10. Algorithmic

```
While (on blank space){
```

```
Switch (last figure) {
```

```
Case 1 (the last figure is triangle): place a blue square
```

```
Case 2 (the last figure is a square): place a blue circle
```

```
Case 3 (the last figure is a circle): place a blue triangle
```

```
}
```

```
Move one step
```

```
}
```

## List of needed assets (aka talking about the assets)

*//Now that we have a solution, we can describe different scenarios (of course not all of them) requiring different assets. This will give us all the assets we need. We need to keep it as simple as possible. It is a list of all the assets used in the result zone and all their possible variations.*

*For example, if we have a result box for eggs with dots, we need to have it in different states:*

- empty
- one egg
- two eggs
- full

*Important: all the assets described here MIGHT NOT be new. For example, we already have assets for the robot walking or picking stuff. But they still need to be described here. The brand new assets will be singled out in the list below. Also, the assets are just described not pictured here. Again, only the new assets will be pictured below. We could use the same numbering system as below for the assets to keep some level of coherence. In order to have a list as clear as possible, we divide the assets into four categories: action, tests, code, result zone//*

### A. Action

- A1.** Robot moves 1 step
- A2.** Robot moves 2 steps
- A3.** Robot places a green square on the blank space
- A4.** Robot places a blue circle on the blank space
- A5.** Robot places a yellow triangle on the blank space
- A6.** Robot places a pink triangle on the blank space
- A7.** Robot places a blue triangle on the blank space
- A8.** Robot places a blue square on the blank space

### B. Tests

- B1.** The robot is on a blank space
- B2.** Check the last figure
- B3.** the last figure is a blue triangle
- B4.** the last figure is a blue square
- B5.** the last figure is a blue circle
- B6.** The last figure is a pink circle
- B7.** the last figure is a yellow triangle

### C. Code

C1. WHILE

C2. IF- THEN

C3. IF- THEN ELSE

C4. SWITCH

### D. Result zone general purpose assets

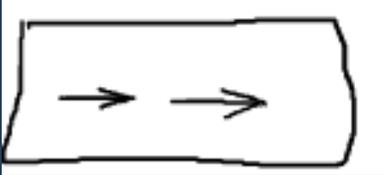
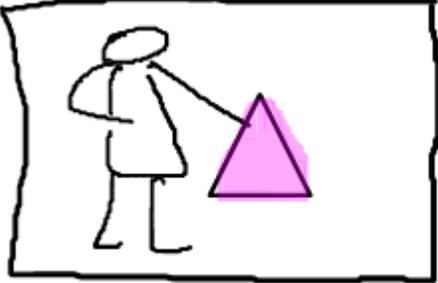
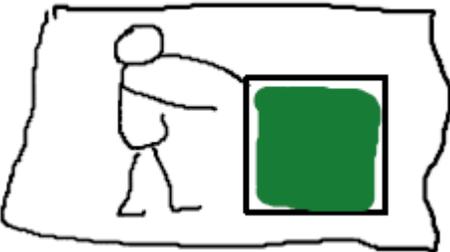
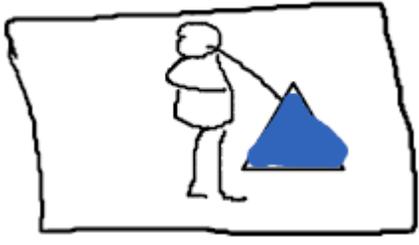
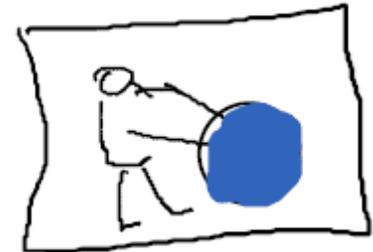
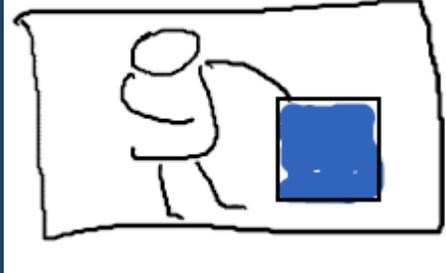
<p>D1. A place with a green square D2. A place with a blue square</p> <p>D3. A place with a blue triangle D4. A place with a yellow triangle D5. A place with a pink triangle</p> <p>D6. A place with a blue circle D7. A place with a pink circle</p>	<p>D8. A blank place</p>
<p>D9. Robot is still D10. Robot is moving D11. Robot is checking the type and the colour of the figure D12. Robot is walking holding a blue circle D13. Robot is walking holding a blue square D14. Robot is walking holding a blue triangle D15. Robot is walking holding a yellow triangle D16. Robot is walking holding a pink triangle D17. Robot is walking holding a green square</p>	<p><b>Assets needed in the effect zone – already implemented above (not new assets)</b></p> <p><b>A3.</b> Robot places a green square on the blank space <b>A4.</b> Robot places a blue circle on the blank space <b>A5.</b> Robot places a yellow triangle on the blank space <b>A6.</b> Robot places a pink triangle on the blank space <b>A7.</b> Robot places a blue triangle on the blank space <b>A8.</b> Robot places a blue square on the blank space</p>

## Visualization (aka drawing the assets)

Nothing more than a drawing of an asset and a number which will be used throughout this document.

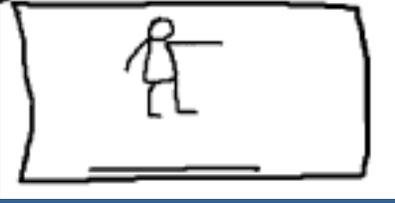
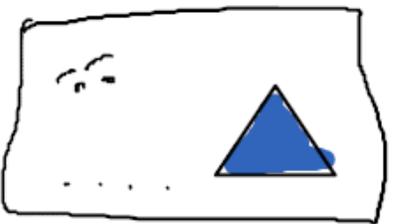
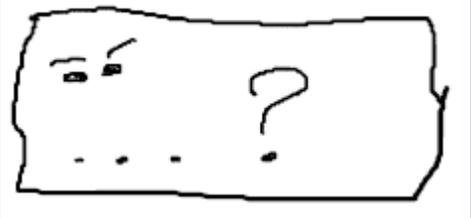
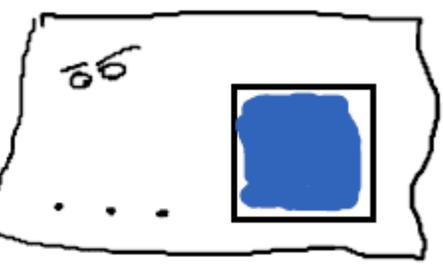
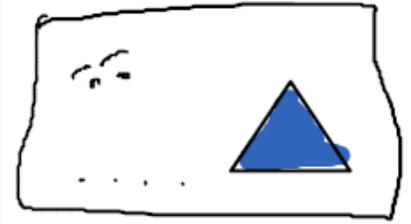
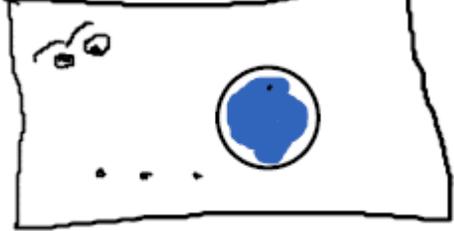
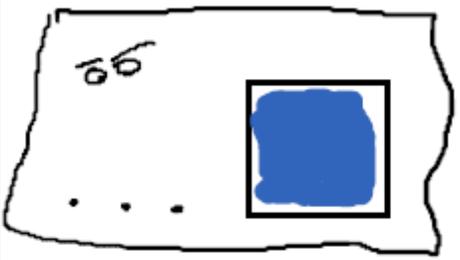
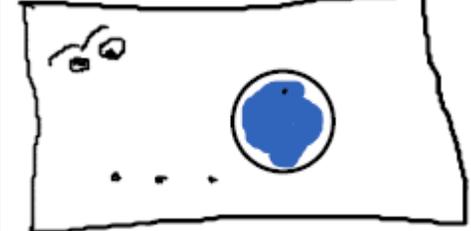
### 1. Action

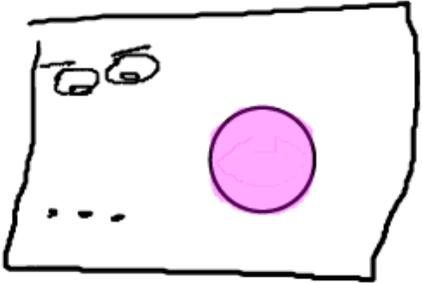
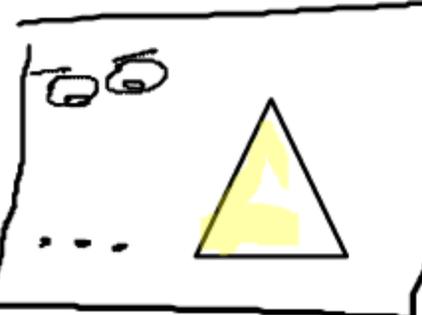
//All the assets which will be available in the action toolbox.

	A1		A5
	A2		A6
	A3		A7
	A4		A8

## 2. Test

All the assets which will be available in the tests toolbox. Warning: as we are using switch, we actually need to sets of tests : one very general which is used when we pick what the switch is about, one detailed which describes all the possible test in one of the selected test category

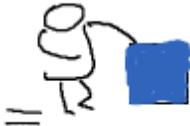
TESTS		CORRESPONDING SUB-CATEGORIES OF TESTS FOR LEVEL 5	
	B1		B3
	B2		B4
	B3		B5
	B4		
	B5		

	B6		
	B7		

### 3. General purpose

//Here we have all the new assets which will be used in the result zone.

	D1		D9
	D2		D10
	D3		D11

	<p>D4</p>		<p>D12</p>
	<p>D5</p>		<p>D13</p>
	<p>D6</p>		<p>D14</p>
	<p>D7</p>		<p>D15</p>
	<p>D8</p>		<p>D16</p>
			<p>D17</p>