# CODEWARDS

## Hour of Code

Introduction to programming

# Hour of Code

Introduction to Programming

| | |
|---|---|
| **Lesson aims:** | To introduce students to the basic concepts of programming. |
| **Computer activity:** | Master commands and basic programming concepts<br><br>Commands: `move, rotate, right, left, load, put`<br><br>Object: `cat`<br><br>Cycyle operator: `loop ... end.` |
| **Materials required:** | 1. Individual computers with access to the internet and simulator for each student. |

# Part 1. Organizational conditions

- To organize a lesson using the simulator, the teacher must provide all students with a workplace, including a computer with Internet access and access to the promotion website https://codewards.org/hoc-cats.

- All modern browsers and operating systems can be used, for working with the simulator!
- It is advisable to organize the opportunity for students to work individually with a stimulator, drawing their attention to the importance of successive passing levels, according to the chosen trajectory (determined by the teacher)

- Upon completion of the work with the simulator, in the class, students will get to the completion screen, where they will see a message thanking them for their participation in the "Hour of Code". They will also have the option to print or upload their certificate of participation.

# Часть 2. The first part of the job

Before getting to work, draw the students' attention to the following:

1. Writing a program is done using visual objects (buttons, icons) at the bottom of the screen. A text editor is also available for students, and that will allow them to write a program themselves, on their keyboards, and feel like a real programmer.

2. The simulator has been enriched with a system of tips and learning tasks, with which schoolchildren can independently get acquainted with the mechanics of generating and writing codes.

Before starting to work with the simulator, give the class some rules to follow in case someone has a problem that they can't solve themselves, while working on the simulator.

A. "Read the condition and try again."
B. “Ask three people, and then me,” i.e. first you need to ask three classmates, and if they can not help, then ask the teacher.
C. "The strong help the weak" (help, but do not suggest).

# Part 3. Computer practice

- Let's explain that the simplest program is a system of commands, that is, WHO + WHAT (must do) + HOW (how many steps, which way, etc.).

## *Concept No.1:*

---

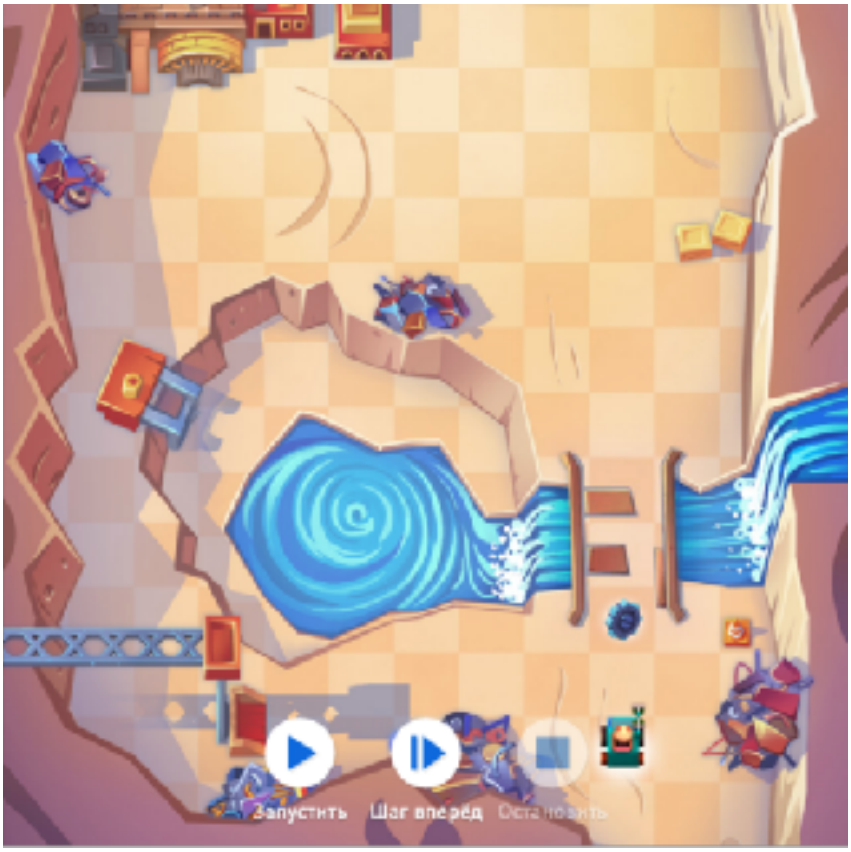
## WHO + WHAT + HOW

- Students walk up to the black board, one by one (holding a tablet with a robot). The groups program it. (the commands should be written on the blackboard in English).

- Go to the computers. Give out instructions.

There are comics embeded in the online platform, and they will immerse the children in a story with a plot.

## Task No.1



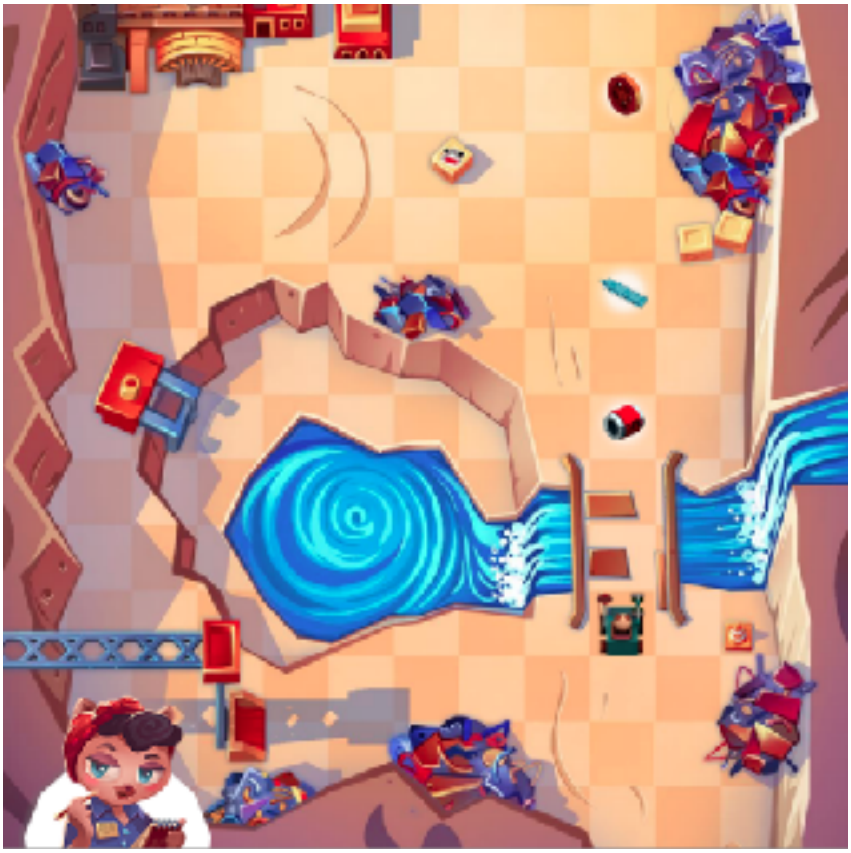

**Task**:
Let's learn how to maneuver a robot. Try to collect the first detail. Use the move command (move forward).

**Hints:**
1. If you need help, ask me.

2. Go to the cell where the detail is. The robot will get it himself.

**Correct code:**
```
cat.move 2
```

## Task No.2



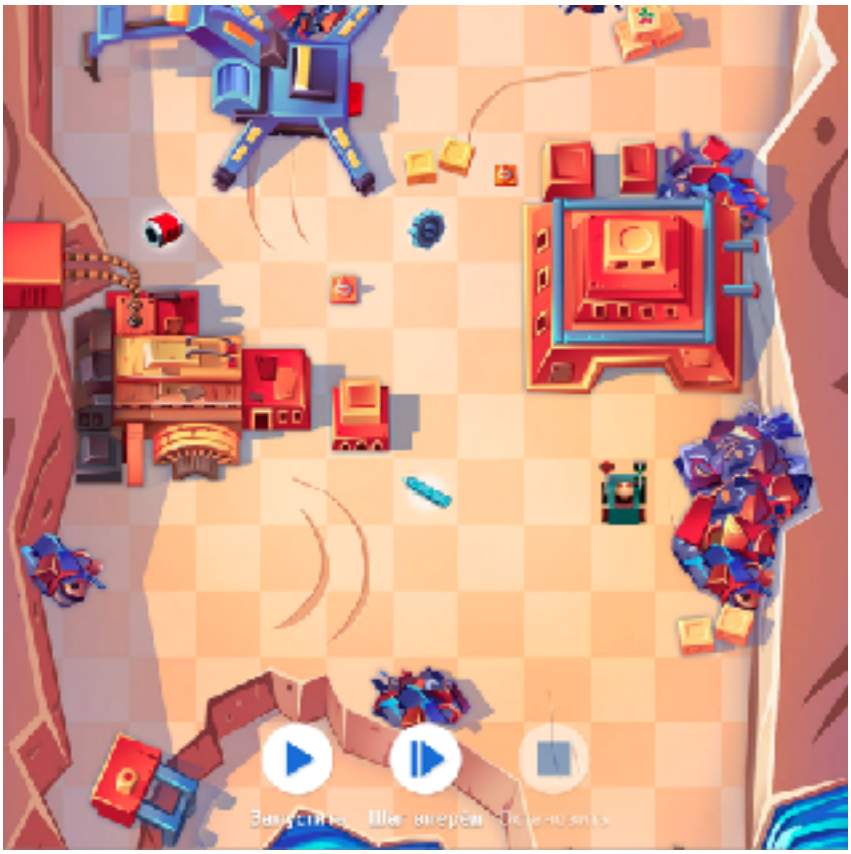
**Task:**
We need more details.  I see 3 more over there. Let's get them all!

**Hints:**
1. Don't forget to stop in every cell that has a detail.

2. All 3 details need to be collected.

**Correct code:**
```
1.    cat.move 3
2.    cat.move 2
3.    cat.move 3
```

## Task No.3



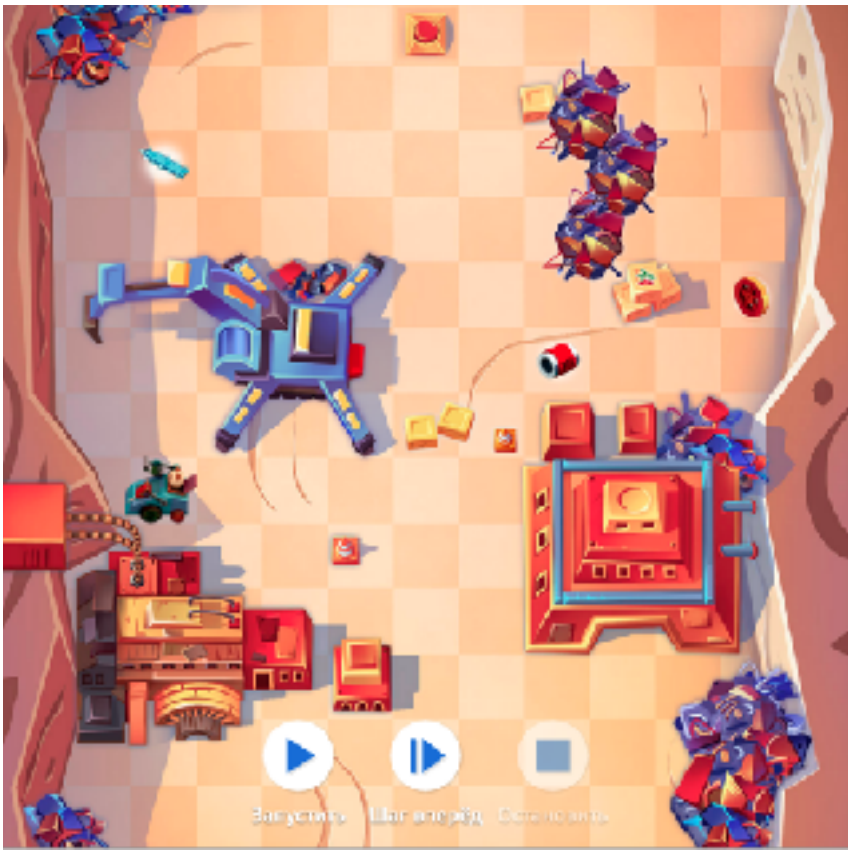
**Task:**
We've got the straight forward movement all figured out.  Now you need to master the turns, with the "rotate" (turn around) command.  Get the next 3 details.

**Hints:**
1. For a left turn, use the argument left. For a right turn - right.

2. All 3 details need to be collected.

**Correct code:**
```
1.    cat.rotate left
2.    cat.move 3
3.    cat.rotate right
4.    cat.move 4
5.    cat.rotate left
6.    cat.move 4
```

## Task No.4



**Task:**
3 more to go, and we're going to have a complete set of parts to assemble! Program the path for the collection of the remaining details.

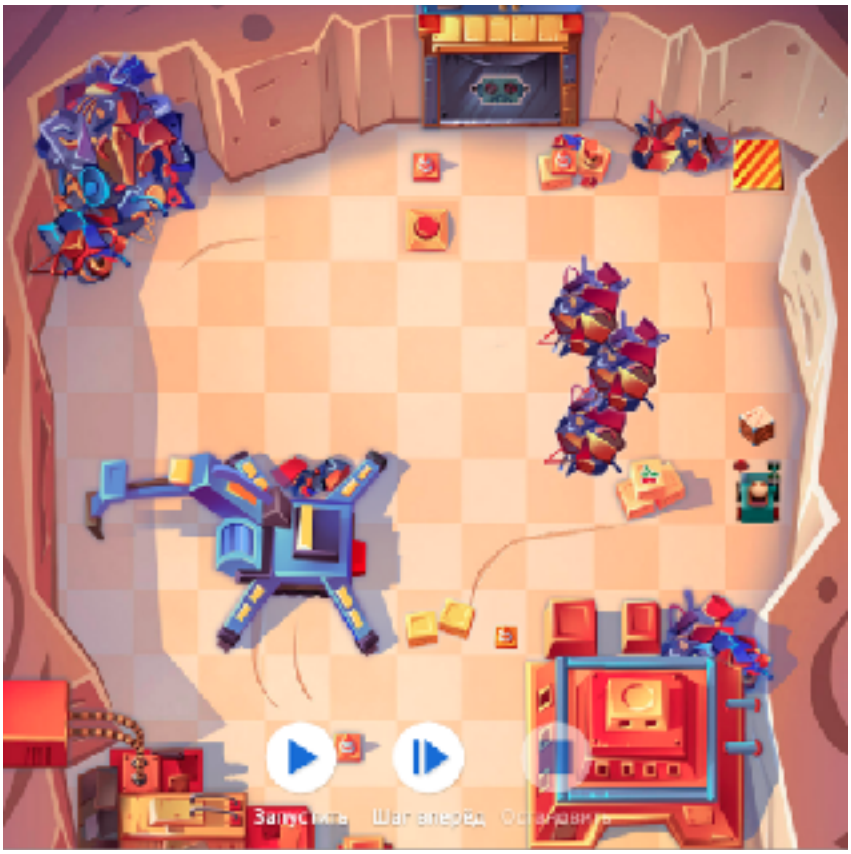
**Hints:**
1. Hey, try not to crash into anything! We need the robot to stay intact.

2. You need to turn, using the rotate command.

3. In order to turn left, use the argument left. To turn right - right.

**Correct code:**
```
1.    cat.rotate right
2.    cat.move 5
3.    cat.rotate right
4.    cat.move 5
5.    cat.rotate right
6.    cat.move 3
7.    cat.rotate left
8.    cat.move 1
9.    cat.move 3
10.   cat.rotate left
11.   cat.move 2
```

## Task No.5



**Task:**
We can't move forward anymore.  Let's get the boxes out of our way to the checkpoint.  To do this, use the commands load and put.

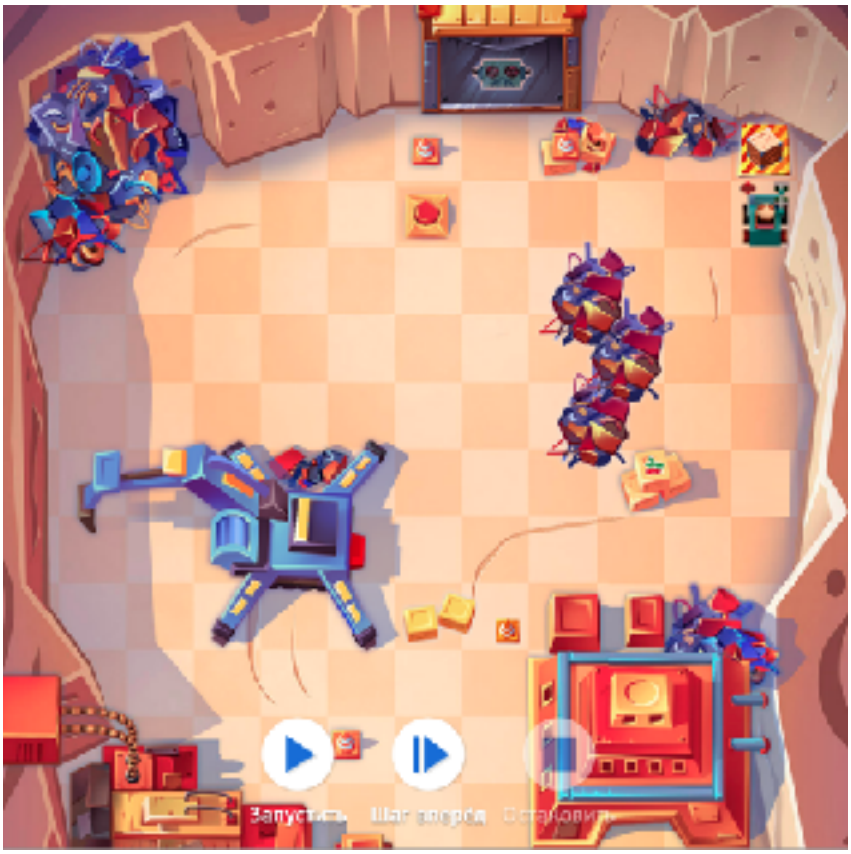
**Hints:**
1. To load the boxes, use the load command. To put it down - put.

2. To take a box or put it down, the robot has to stop in front of it.

3. Put the boxes in the marked cell.

**Correct code:**
```
1.    cat.load
2.    cat.move 4
3.    cat.put
```

## Task No.6



**Task:**
Great! We've gathered enough details. Now it's time to go to a factory, where we can put those details together to create something useful, and install it on a robot. Open the door and drive inside the building.

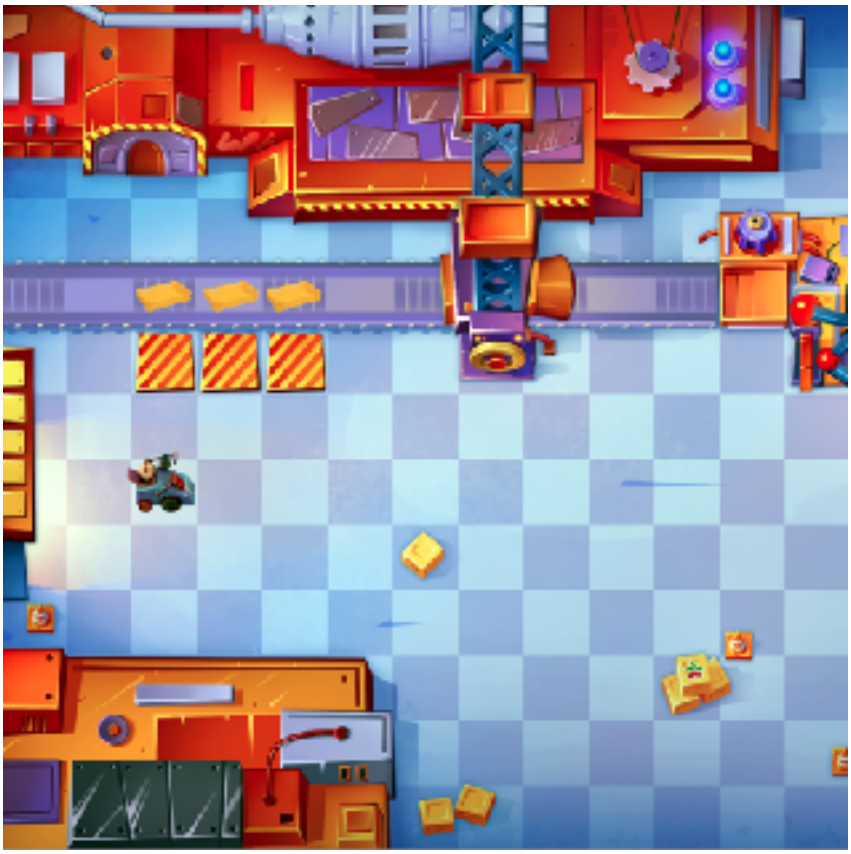
**Hints:**
1. I have a feeling that we need to put something heavy on the button.

2. Try putting a box on the button. Use the commands load and put (put it down).

3. To take and put a box on the button, you need to stop the robot in front of it.

**Correct code:**
```
1.    cat.load
2.    cat.rotate left
3.    cat.move 4
4.    cat.put
5.    cat.rotate right
6.    cat.move 2
```

## Task No.7



**Task:**
It's time to assemble the device for the robot.  Lay out 3 details in 3 boxes, on the conveyor belt.

**Hints:**
1. Place the collected details into the boxes, on the conveyor belt.  One detail goes in one box. Quite simple!

2. In order to put down the detail, use the command put.

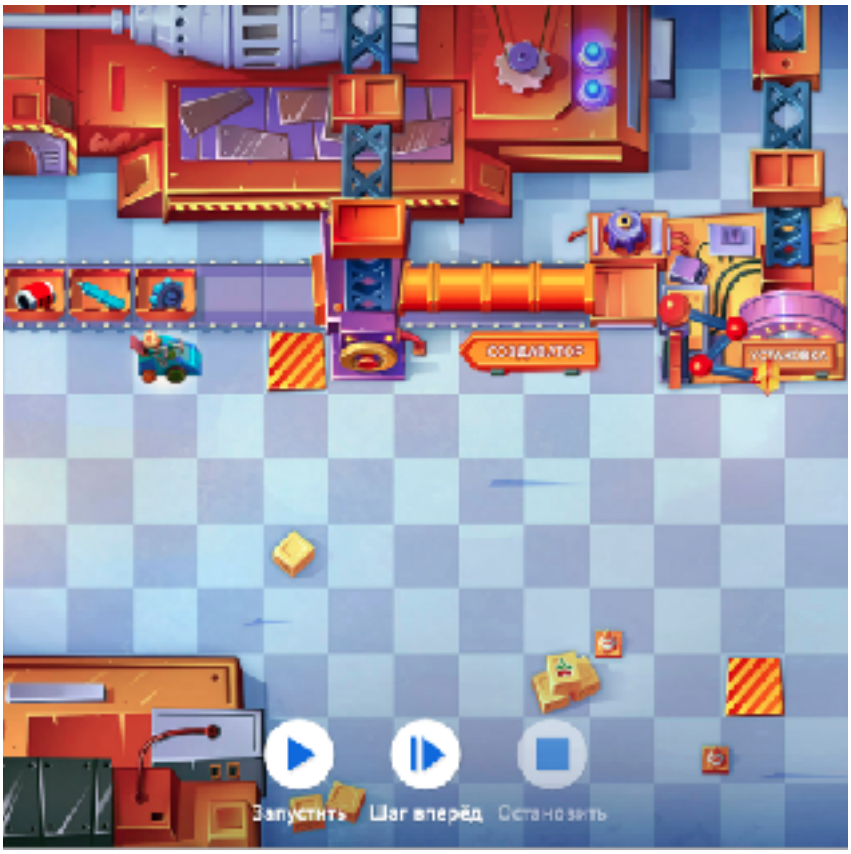
3. You can put down a detail, only if you go to an empty box.

**Correct code:**

```
1. cat.rotate left
2. cat.move 2
3. cat.put
4. cat.rotate right
5. cat.move 1
6. cat.rotate left
7. cat.put
8. cat.rotate right
9. cat.move 1
10.cat.rotate left
```

```
11.cat.put
```

## Task No.8




**Task:**
Now we can assemble the device for our robot. Spin the "Creator" wheel, using the spin command, so that all of the parts get loaded onto it. After the device has been created, go on to the installation.

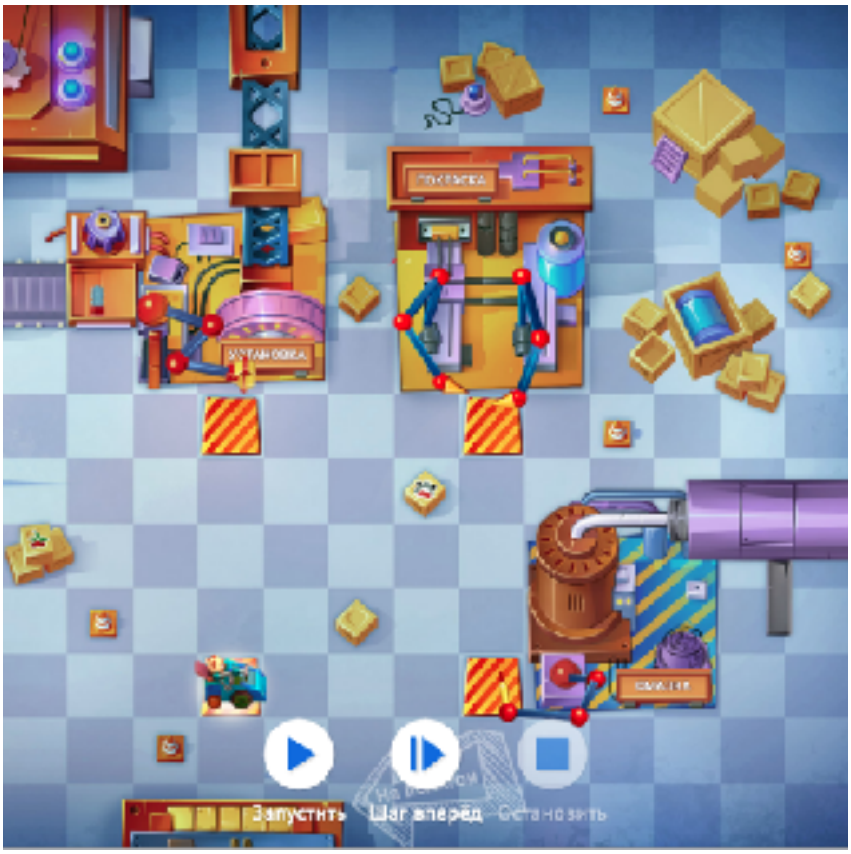
**Hints:**
1. Something new and useful will turn out, only when all three details have been loaded in the "Creator."

2. One turn of the wheel moves the conveyor belt one division.

3. You can spin the wheel only from the checkpoint, using the spin command.

**Correct code:**
```
1. cat.move 2
2. cat.spin 5
3. cat.rotate right
4. cat.move 2
5. cat.rotate left
6. cat.move 7
7. cat.rotate right
8. cat.move 3
```

## Task No.9




**Task:**
Let's install the device at last! Every detail needs to be installed, painted, and greased. Three times over. Use the loop (cycle) function, so that the set of commands will be automatically repeated. After installing, go to the exit to the testing range.

**Hints:**
1. 3 kinds of devices need to be installed on to the robot. For that, you need to drive up to each machine 3 times. For repeating commands, use a loop (cycle).

2. Try to understand which commands should be repeated, and in what sequence. Then it'll become clear how you're supposed to use a loop. By the way, a loop always starts at the same point.

3. After installing the device, drive to the exit.

4. The loop should start with the loop command and end with the end command.

**Correct code – version 1:**

```
1.    cat.rotate left
2.    loop 3
3.         cat.move 4
4.         cat.rotate right
```

```
5.          cat.move 4
6.          cat.rotate right
7.          cat.move 4
8.          cat.rotate right
9.          cat.move 4
10.         cat.rotate right
11.   end
```

**Correct code– version 2:**

```
1.    cat.rotate left
2.    loop 11
3.          cat.move 4
4.          cat.rotate right
5.    end
6.    cat.move 4
7.    cat.rotate left
8.    cat.move 1
```

## Task No.10

**Task:**
We are at a testing range! Let's test the saw that we've put on our robot. In order to do that, cut the wooden robot figurines. Use the saw command.

**Hints:**
1. Notice how the figurines are alligned.

2. It seems like it's worth using the loop (cycle) function, so that the code will be optimal.

**Correct code:**
```
loop 4
    cat.rotate left
    cat.move 1
    cat.saw
    cat.move 1
    cat.rotate right
    cat.move 1
    cat.saw
    cat.move 1
end
```

# Part 4. Reflection

The live action game **"I will program you!"**

Before the game, towards the end of each lesson, you need to hand out cards of 3 colors (types):

1. object,
2. commands (actions),
3. arguments (numbers).

The team is given a little task: They have to program the teacher so that he/she will carry out 2 steps. The teacher is able to do that if all three teams create the needed command, out of the cards.

They select an object (cat) and make up a program that reflects the material covered, out of the cards. (cards can be stuck onto the blackboard with magnets / masking tape, and each student takes one card - three students make up the program).