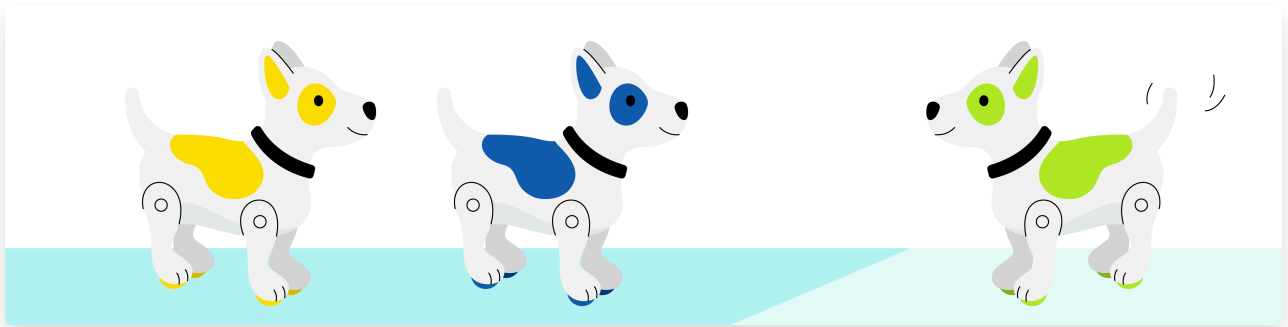


Events

Textbook

Events



How Events Work in Programming

A program is a set of instructions a computer follows. In programming, an **event** is an action that a program responds to. Events are what make a program work when you use it. For example, when you click a button or type on a keyboard, that is an event. The program notices the event and does something.

Here are some common events:

- **Click events:** When you click a button, a link, or a picture.
- **Hover events:** When you move your mouse pointer over an object.
- **Key events:** When you press a key on your keyboard.
- **Submit events:** When you send information from a form.

Programmers use events to make programs react to users. Without events, you could not tell a program what to do.

Events and Computational Steps

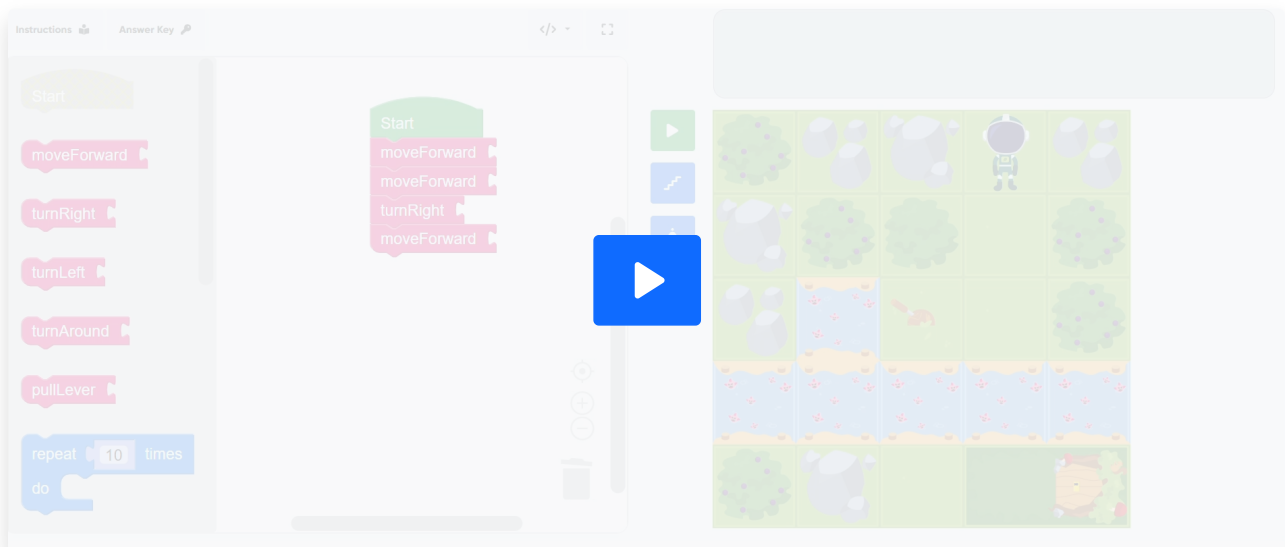
When an event happens, a computer follows a series of steps. This is a process of **computational steps**.

For example, imagine you click a "Submit" button on a form. The computer follows these steps:

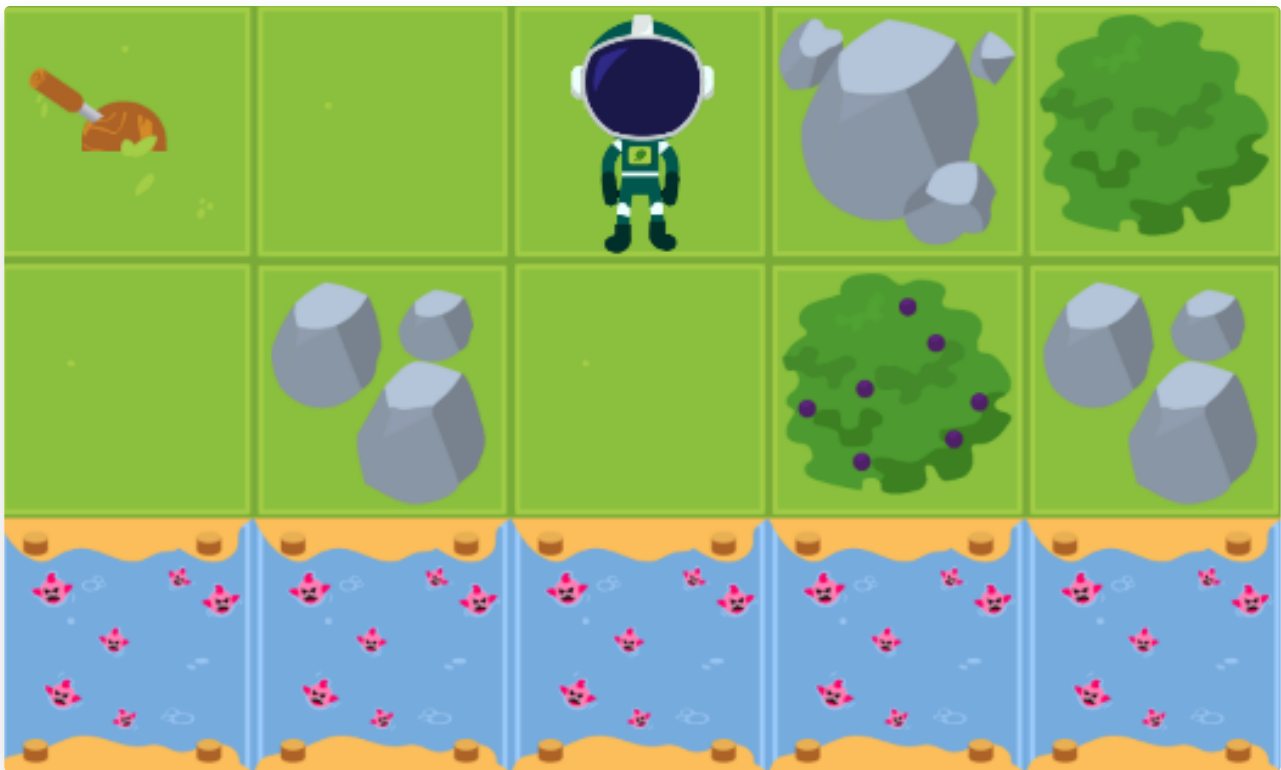
1. **Detect the click:** The program first records that the "Submit" button was pressed.
2. **Gather data:** It then collects all the information you typed into the form.
3. **Process data:** The program checks the information and sends it to a server.
4. **Display feedback:** Finally, the program shows you a message, such as "Form sent!"

Breaking down events into these steps helps programmers understand exactly how a program works. This helps them fix problems and build better software. To understand how an event works, ask yourself: "What are the exact steps the computer must follow?"

Events Video




Practice Using Events



The code `pullLever();` is an event. Remember that an event is something the user does that causes things to happen in the program. See what happens when your character pulls the lever!

Here is both the block and syntax code that will allow the character above to pull the lever.

Blocks	Syntax
	<pre> 1 turnRight(); 2 moveForward(); 3 moveForward(); 4 pullLever(); 5 </pre>

Critical Thinking Questions

1. If you were making an app for a school project, what events would you need to include?
2. How would websites and apps work differently without events?

Questions (5)

1. What is an event in programming?

MULTIPLE CHOICE

Choose the correct answer:

- A. When an algorithm ends
- B. Something that happens on a computer when the user does something
- C. When the program finishes
- D. When a variable gets used

2. Select the 2 examples of events on the computer.

SELECT MULTIPLE

Select all that apply:

- A. Pressing a button
- B. Hovering over a link
- C. Waiting for an image or video to load
- D. Planning out an algorithm

3. True or False: Selecting a button on a touch screen is an example of an event

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

4. What helps programs become useful for users by allowing interaction?

MULTIPLE CHOICE

Choose the correct answer:

- A. Buttons and icons
- B. Pictures and videos
- C. Background colors
- D. Font styles

5. Imagine you hover your mouse over a picture on a website and a description pops up. What type of event is this?

MULTIPLE CHOICE

Choose the correct answer:

- A. Click event
- B. Hover event
- C. Submit event
- D. Play event

Games (2)

1. Events and Computer Systems Typing Game


Full Screen

Audio

Instructions

Restart

Pause



0s 100%

An event is something th

2. Events and Computer Systems Category Game

Full Screen

Audio

Instructions

Answer Key

Pause

Clear All

Check Order

Attempts: 0

The first day of school

A family reunion

Pressing a button on the keyboard

A birthday party

Clicking "Submit" on a form

Clicking a button on a website

A Computer Event

Not a Computer Event



Puzzles (15)

1. Events #1

Instructions

Answer Key

</>

Start

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

2. Events #2

Instructions

Answer Key

</>

Start

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do


and


not

isClearAhead


isClearLeft

3. Events #3

Instructions 

Answer Key 

</>



Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do


and


not


isClearAhead





isClearLeft

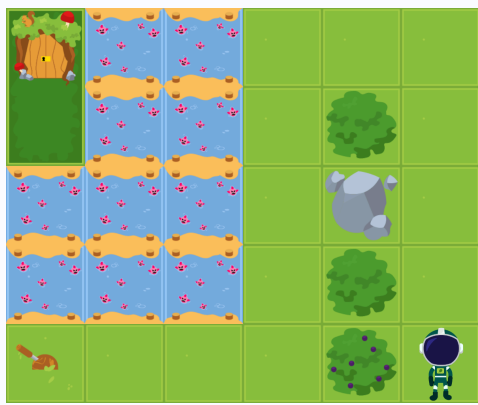
Start














4. Events #4

Instructions 

Answer Key 

</>



Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do


and


not


isClearAhead





isClearLeft


Start











5. Events #5

Instructions

Answer Key

</>

↺

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

Start



▶

↗

↑


⏮

6. Events #6

Instructions 
Answer Key 

</>
▼

[]

Start


moveForward

turnRight

turnLeft


turnAround

pullLever

repeat


10


times




repeat


while






if








and





not


isClearAhead



isClearLeft



Start










































7. Events #7

Instructions 
Answer Key 
</> 

Start
Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if


do


and


not


isClearAhead


isClearLeft


























Instructions 
Answer Key 
</> 

Start
Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if


do


and


not


isClearAhead


isClearLeft
























Instructions 

8. Events #8

Instructions
Answer Key
</>

Start

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times
do

repeat while
do

if
do

and

not

isClearAhead

isClearLeft

Start

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times
do

repeat while
do

if
do

and

not

isClearAhead

isClearLeft

▶

⬆

⬆

⬆

⬆

9. Events #9

Instructions

Answer Key

</>

↺

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

Start

🔍

+

-

▶

📈

⬆

🏠

10. Events #10

Instructions
Answer Key

Start

```

moveForward
turnRight
turnLeft
turnAround
pullLever

repeat 10 times
do

repeat while
do

if
do

and

not

isClearAhead

isClearLeft
        
```


▶


⬆

⬇


⬅

11. Events #11

Instructions 

Answer Key 

</>



Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do


and


not


isClearAhead





isClearLeft


Start














12. Events #12

Instructions 

Answer Key 

</>



Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do


and


not


isClearAhead





isClearLeft


Start











13. Events #13

Instructions
Answer Key

Start

- moveForward
- turnRight
- turnLeft
- turnAround
- pullLever

repeat 10 times
do

repeat while
do

if
do

and

not

isClearAhead

isClearLeft

Start

Play

Step

Up

Down

14. Events #14

Instructions

Answer Key

</>

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

Start


▶


↗


↑

⬇

15. Events #15

Instructions 

Answer Key 

`</>` 

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do




and





not


























isClearAhead

isClearLeft

Start

Answer Keys & Solutions

Questions

1. What is an event in programming?

MULTIPLE CHOICE

Correct Answer:

- A. When an algorithm ends ✗ Incorrect
- B. Something that happens on a computer when the user does something ✓ Correct
- C. When the program finishes ✗ Incorrect
- D. When a variable gets used ✗ Incorrect

Explanation:

Remember, an event can trigger the computer to do something

2. Select the 2 examples of events on the computer.

SELECT MULTIPLE

Correct Answers:

- A. Pressing a button ✓ Correct
- B. Hovering over a link ✓ Correct
- C. Waiting for an image or video to load ✗ Incorrect
- D. Planning out an algorithm ✗ Incorrect

Explanation:

An event is when you do something and the computer responds.

3. True or False: Selecting a button on a touch screen is an example of an event

MULTIPLE CHOICE

Correct Answer:

A. True

✓ Correct

B. False

✗ Incorrect

Explanation:

An event is when you do something and the computer responds

4. What helps programs become useful for users by allowing interaction?

MULTIPLE CHOICE

Correct Answer:

A. Buttons and icons

✓ Correct

B. Pictures and videos

✗ Incorrect

C. Background colors

✗ Incorrect

D. Font styles

✗ Incorrect

Explanation:

Consider what elements on a website or software allow you to do things like submit forms or make purchases.

5. Imagine you hover your mouse over a picture on a website and a description pops up. What type of event is this?

MULTIPLE CHOICE

Correct Answer:

A. Click event

✗ Incorrect

B. Hover event

✓ Correct

C. Submit event

✗ Incorrect

D. Play event

✗ Incorrect

Explanation:

Pay attention to the action described (hovering over something) and how it triggers a response.

Games

1. Events and Computer Systems Typing Game

Typing game - no answer key needed. Students practice typing the provided content.

2. Events and Computer Systems Category Game

Category Solutions:

Category 1: A Computer Event

- Clicking "Submit" on a form
- Clicking a button on a website
- Pressing a button on the keyboard

Category 2: Not a Computer Event

- A birthday party
- A family reunion
- The first day of school

Scoring:

- Gold: 1 attempts or fewer
- Silver: 2 attempts or fewer
- Bronze: 3 attempts or fewer

Students must sort items into their correct categories.

Puzzles

1. Events #1

Solution:

```
1 moveForward();
2 moveForward();
3 turnRight();
4 moveForward();
5 pullLever();
6 turnLeft();
7 moveForward();
8 moveForward();
9 turnLeft();
10 moveForward();
```

2. Events #2

Solution:

```
1 moveForward();
2 moveForward();
```



```
3 moveForward();
4 turnRight();
5 moveForward();
6 moveForward();
7 moveForward();
8 moveForward();
9 moveForward();
10 turnRight();
11 moveForward();
12 moveForward();
13 moveForward();
14 turnRight();
15 moveForward();
16 pullLever()
17 moveForward();
18 turnRight();
19 moveForward();
```

3. Events #3

Solution:

```
1 turnAround();
2 moveForward();
3 moveForward();
4 moveForward();
5 moveForward();
6 turnLeft();
7 moveForward();
8 moveForward();
9 turnLeft();
10 moveForward();
11 moveForward();
12 moveForward();
13 moveForward();
14 turnRight();
15 moveForward();
16 moveForward();
17 moveForward();
18 pullLever()
19 turnRight();
20 moveForward();
21 moveForward();
22 moveForward();
```

4. Events #4

Solution:

```
1 turnRight();
2 moveForward();
3 turnRight();
4 moveForward();
5 turnLeft();
6 for (var count = 0; count < 2; count++) {
```

```

7   moveForward();
8 }
9 turnRight();
10 for (var count2 = 0; count2 < 2; count2++) {
11   moveForward();
12 }
13 turnLeft();
14 moveForward();
15 turnRight();
16 moveForward();
17 pullLever()
18 turnAround();
19 moveForward();
20 turnLeft();
21 for (var count3 = 0; count3 < 4; count3++) {
22   moveForward();
23 }
24 turnRight();
25 moveForward();

```

5. Events #5

Solution:

```

1 turnLeft();
2 moveForward();
3 moveForward();
4 turnRight();
5 moveForward();
6 moveForward();
7 turnRight();
8 moveForward();
9 moveForward();
10 moveForward();
11 moveForward();
12 turnRight();
13 moveForward();
14 moveForward();
15 pullLever()
16 turnAround();
17 moveForward();
18 moveForward();
19 turnLeft();
20 moveForward();
21 moveForward();
22 moveForward();
23 moveForward();
24 turnRight();
25 moveForward();
26 moveForward();
27 turnRight();
28 moveForward();
29 moveForward();

```

```
30 moveForward();
```

6. Events #6

Solution:

```
1 moveForward();
2 moveForward();
3 pullLever()
4 moveForward();
5 moveForward();
6 moveForward();
7 pullLever()
8 moveForward();
9 moveForward();
10 moveForward();
```

7. Events #7

Solution:

```
1 turnAround();
2 moveForward();
3 moveForward();
4 moveForward();
5 turnRight();
6 moveForward();
7 pullLever()
8 turnAround();
9 moveForward();
10 turnLeft();
11 moveForward();
12 moveForward();
13 moveForward();
14 turnLeft();
15 moveForward();
16 moveForward();
17 turnLeft();
18 moveForward();
19 moveForward();
20 turnRight();
21 moveForward();
22 turnLeft();
23 moveForward();
24 pullLever()
25 turnAround();
26 moveForward();
27 turnRight();
28 moveForward();
29 turnLeft();
30 moveForward();
31 moveForward();
32 turnLeft();
33 moveForward();
34 moveForward();
35 moveForward();
36 turnLeft();
37 moveForward();
```

```
38 moveForward();
39 moveForward();
```

8. Events #8

Solution:

```
1  turnLeft();
2  moveForward();
3  moveForward();
4  moveForward();
5  moveForward();
6  moveForward();
7  moveForward();
8  pullLever()
9  turnAround();
10 moveForward();
11 moveForward();
12 moveForward();
13 moveForward();
14 moveForward();
15 turnLeft();
16 moveForward();
17 moveForward();
18 turnRight();
19 moveForward();
20 turnLeft();
21 moveForward();
22 moveForward();
23 moveForward();
24 moveForward();
25 turnLeft();
26 pullLever();
27 moveForward();
28 moveForward();
29 moveForward();
30 moveForward();
31 moveForward();
32 moveForward();
33 turnLeft();
34 moveForward();
35 moveForward();
36 moveForward();
37 moveForward();
38 turnLeft();
39 moveForward();
40 pullLever();
41 turnAround();
42 moveForward();
43 turnRight();
44 moveForward();
45 moveForward();
46 turnRight();
47 moveForward();
48 moveForward();
49 moveForward();
50 turnRight();
51 moveForward();
```

9. Events #9

Solution:

```
1 turnLeft();
2 moveForward();
3 turnLeft();
4 moveForward();
5 turnRight();
6 moveForward();
7 turnLeft();
8 moveForward();
9 moveForward();
10 turnLeft();
11 moveForward();
12 turnRight();
13 moveForward();
14 moveForward();
15 turnLeft();
16 moveForward();
17 pullLever()
18 turnAround();
19 moveForward();
20 turnRight();
21 moveForward();
22 moveForward();
23 turnLeft();
24 moveForward();
25 moveForward();
26 moveForward();
27 pullLever()
28 turnLeft();
29 moveForward();
30 turnLeft();
31 moveForward();
32 turnRight();
33 moveForward();
```

10. Events #10

Solution:

```
1 turnLeft();
2 moveForward();
3 turnLeft();
4 moveForward();
5 turnRight();
```

```
6 moveForward();
7 turnLeft();
8 moveForward();
9 moveForward();
10 turnLeft();
11 moveForward();
12 moveForward();
13 turnLeft();
14 moveForward();
15 pullLever()
16 turnAround();
17 moveForward();
18 moveForward();
19 moveForward();
20 turnRight();
21 moveForward();
22 moveForward();
23 moveForward();
24 turnRight();
25 moveForward();
26 turnLeft();
27 moveForward();
28 moveForward();
29 turnLeft();
30 moveForward();
31 turnAround();
32 pullLever()
33 moveForward();
34 turnRight();
35 moveForward();
36 turnLeft();
37 moveForward();
38 moveForward();
39 moveForward();
40 turnLeft();
41 moveForward();
42 turnRight();
43 moveForward();
44 pullLever();
45 turnRight();
46 moveForward();
47 moveForward();
```

11. Events #11

Solution:

```
1 turnRight();
2 moveForward();
3 turnRight();
4 moveForward();
5 moveForward();
6 turnLeft();
7 moveForward();
8 turnRight();
9 moveForward();
10 moveForward();
```

```
11 moveForward();
12 pullLever()
13 turnRight();
14 moveForward();
15 moveForward();
16 moveForward();
17 turnRight();
18 moveForward();
19 moveForward();
20 turnRight();
21 moveForward();
22 pullLever();
23 turnAround();
24 moveForward();
25 turnLeft();
26 moveForward();
27 moveForward();
28 turnRight();
29 moveForward();
30 moveForward();
31 turnRight();
32 moveForward();
33 moveForward();
34 moveForward();
35 moveForward();
36 moveForward();
37 turnRight();
38 moveForward();
```

12. Events #12

Solution:

```
1 turnLeft();
2 moveForward();
3 pullLever()
4 moveForward();
5 moveForward();
6 pullLever()
7 moveForward();
8 moveForward();
9 pullLever()
10 moveForward();
11 moveForward();
12 pullLever()
13 moveForward();
14 moveForward();
```

13. Events #13

Solution:

```
1 turnRight();
2 moveForward();
3 moveForward();
4 turnLeft();
5 moveForward();
6 turnRight();
7 moveForward();
8 turnLeft();
9 moveForward();
10 moveForward();
11 turnLeft();
12 moveForward();
13 moveForward();
14 turnRight();
15 moveForward();
16 moveForward();
17 turnLeft();
18 moveForward();
19 pullLever()
20 turnAround();
21 moveForward();
22 turnRight();
23 moveForward();
24 moveForward();
25 turnLeft();
26 moveForward();
27 moveForward();
28 turnLeft();
29 moveForward();
30 moveForward();
31 moveForward();
32 turnLeft();
33 moveForward();
34 pullLever()
35 turnRight();
36 moveForward();
37 turnLeft();
38 moveForward();
39 moveForward();
```

14. Events #14

Solution:

```
1 moveForward();
2 turnLeft();
3 moveForward();
4 moveForward();
5 turnLeft();
6 moveForward();
7 moveForward();
8 pullLever()
9 turnAround();
10 moveForward();
11 moveForward();
12 turnRight();
13 moveForward();
14 moveForward();
```



```
15 moveForward();
16 turnLeft();
17 moveForward();
18 turnRight();
19 moveForward();
20 turnLeft();
21 moveForward();
22 moveForward();
23 turnLeft();
24 moveForward();
25 turnRight();
26 moveForward();
```

15. Events #15

Solution:

```
1  turnLeft();
2  moveForward();
3  turnLeft();
4  moveForward();
5  turnRight();
6  moveForward();
7  turnLeft();
8  moveForward();
9  turnRight();
10 moveForward();
11 turnLeft();
12 moveForward();
13 turnRight();
14 moveForward();
15 turnLeft();
16 moveForward();
17 turnRight();
18 moveForward();
19 turnLeft();
20 moveForward();
21 moveForward();
22 turnLeft();
23 moveForward();
24 moveForward();
25 moveForward();
26 moveForward();
27 moveForward();
28 turnLeft();
29 moveForward();
30 moveForward();
31 moveForward();
32 moveForward();
33 pullLever()
34 turnAround();
35 moveForward();
36 moveForward();
37 moveForward();
38 moveForward();
39 turnRight();
40 moveForward();
41 moveForward();
42 moveForward();
43 moveForward();
```

```
44 moveForward();
45 moveForward();
46 turnRight();
47 moveForward();
48 moveForward();
49 moveForward();
50 moveForward();
51 moveForward();
52 moveForward();
53 turnRight();
54 moveForward();
55 moveForward();
56 moveForward();
```