

Karel Project #21

ACCESS DENIED

You can **stop** players from using specific **commands**.

Go to the **Creative Suite**, open a new Karel project, and save it in the folder **course-karel-projects/** with the name, **access-denied**.

Goal: Forbid Keywords in Solutions

To create a maze that **forbids keywords**, you will:

1. Use the **Goals menu** to **forbid** keywords.
2. Design a maze with **forbidden keywords**.
3. **Test** your maze.

A Simple Maze

The following maze would be fairly **easy to solve**. What commands would you use to solve this maze?



*This solution to this maze requires **go**, **right**, and **left**. Or does it?*

The maze above could be solved with the following program:

```
1 right_turns=0
2 while not home
3   go
4   if wall
5     if right_turns >= 2
6       left
7     if right_turns < 2
8       right
9       inc(right_turns)
10
```

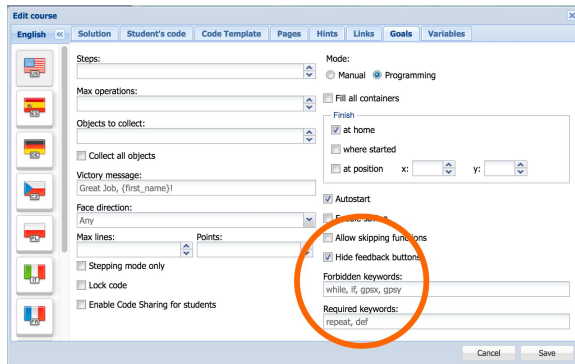
This is one of many possible solutions.

This solution uses the commands: **while**, **go**, **if**, **left**, **right**, and **inc()**.

What would happen if the player was **forbidden** from using one of these commands?

Step 1: Use the **Goals Menu** to Forbid Keywords

In a new maze, open the **Goals** panel:



Any commands you type in this field will be forbidden in the solution for the maze. So, what are some commands that you should **forbid**?

What if you **forbid** the keyword **left**? Then the player would be forced to use 3 **right commands** instead of **left**. Tricky!

In this maze:



The **left** command in the solution would no longer **be allowed**:

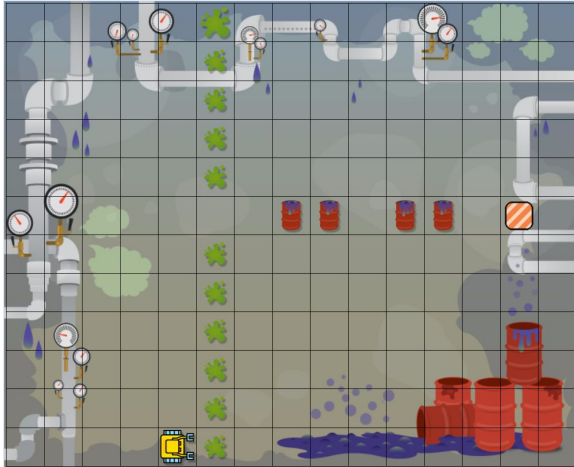
```
1 right_turns=0
2 while not home
3   go
4   if wall
5     if right_turns >= 2
6       left
7     if right_turns < 2
8       right
9     inc(right_turns)
10
```

The player will need to **think harder**, and will feel more **challenged**!

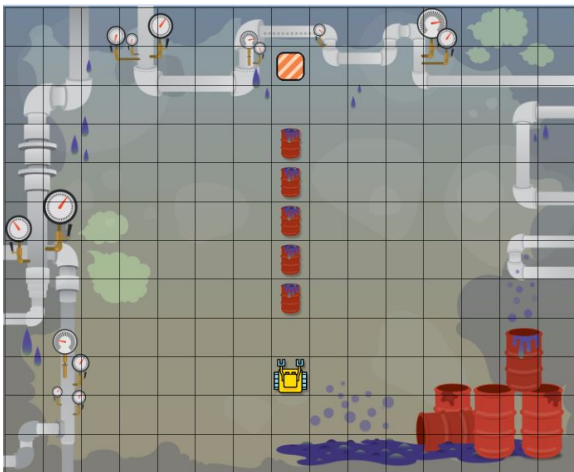
Step 2: Design a Maze With **Forbidden Keywords**

Use a **maze template** or start an **original design** to create a maze. Then use the **Goals** menu to **forbid keywords** in the solution to your maze.

Here are a few examples from the **NCLab Karel Course**:



In this maze, the player is forbidden from using the **repeat** keyword and they must complete the level with less than or exactly **15 lines of code**. Players will use the **while** keyword as an **ideal strategy**. A **strategy** is a method for solving a problem.



This maze is quite similar to the last one. It forbids the player from using the **repeat** keyword. The **while** keyword allows the user to move Karel while still using very few lines of code.

In both of these examples, the player is persuaded to use **while** in their **strategy** for solving the maze. There are many other strategies for solving Karel mazes. Experiment and play in order to **find new strategies!**

Step 3: Test Your Maze

Have fun experimenting with different **forbidden keywords**. Remember, as you **make your maze harder** more people will find it **challenging**, and some may find it **too hard**.



Project Checklist

Your project will be finished when:

1. You created a **simple maze**.
2. You have **forbidden** players from using one of the **commands** you used to solve your maze.
3. You have **solved** your maze.