nclab
Learning for Industry 4.0

Karel Project #16

More Rooms

Complexity can become out of control. You need to plan ahead to make a complex maze.

Go to the Creative Suite, open a new Karel project, and save it in the folder course-karel-projects/ as five-rooms.

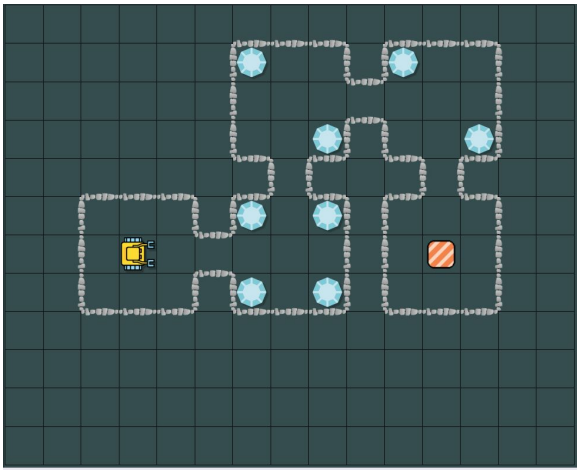
Goal: Create a Multi-Room Puzzle

To create a maze with more rooms, you will:

1. Create a maze from the 3-rooms template.
2. Add an extra room to your maze.
3. Test your maze.

An Ideal Solution

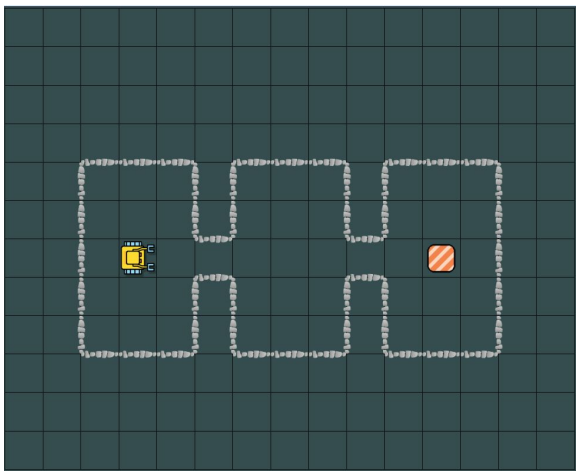
A good design can help lead the player to an ideal solution. Consider how you might solve the following maze. Can you see a pattern?

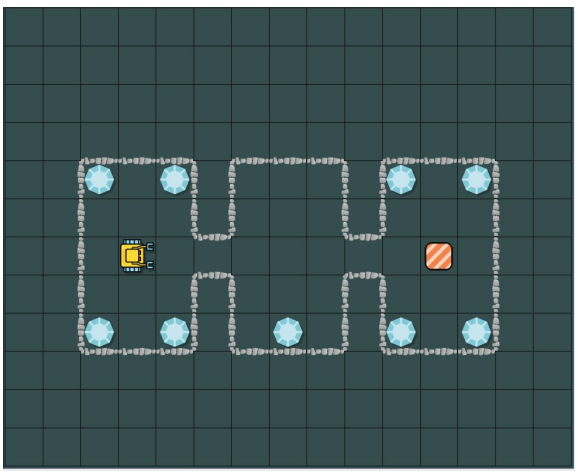


In this maze, the player needs to turn a different direction if there are more than 2 gems in the current room. This maze could be solved with the > operator. Navigate to the last page of this Project for a detailed solution for this maze.

Step 1: Create a Maze From the 3-Rooms Template

In this Project you will add complexity to the design you practiced in Project 16. Start this Project with the 3-rooms template. Create 3 rooms in your maze and place a few Objects in each room.

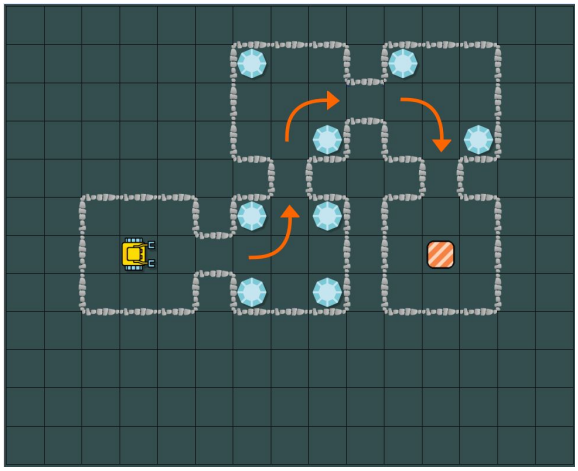




Step 2: Add More Rooms to Your Maze

In this step you need to add at least one more room to your maze. Use Objects in each of your rooms. As you place Objects, consider the ideal solution for your maze.

How can you use Objects to help the player solve your maze?



In this maze, 4 gems leads to a left turn and 2 gems leads to a right turn.

Step 3: Test Your Maze

Always test your maze with others. If they find your maze too difficult, consider adding some clues in your Summary or try using different Goals.

Use more rooms to add more complexity.

Project Checklist

Your project will be finished when:

1. You saved the project in the folder course-karel-projects/.
2. You created a simple maze from the 3-rooms template.
3. You added more rooms to the maze.
4. You tested your maze and solved it.

Solution For the Five-Rooms Maze

```
def evaluate_room
  gems = 0
  repeat 3
    go
  left
  repeat 4
    go
    if gem
      inc(gems)
    right
    go
  right

  # Change direction based on quantity of gems
  if gems > 2
    go
    left
  else
    go
    right

# Main Program
repeat 3
  evaluate_room
```