More Than Coding

Nevada Ready 21 Webinar 2: Focus on Karel Coding

http://nclab.com

The purpose of today’s webinar

Why start with Karel?

What will my students learn?

- How will they learn it?
- How are they assessed?

How does Karel fit into my curriculum?

What else is there beside the course?

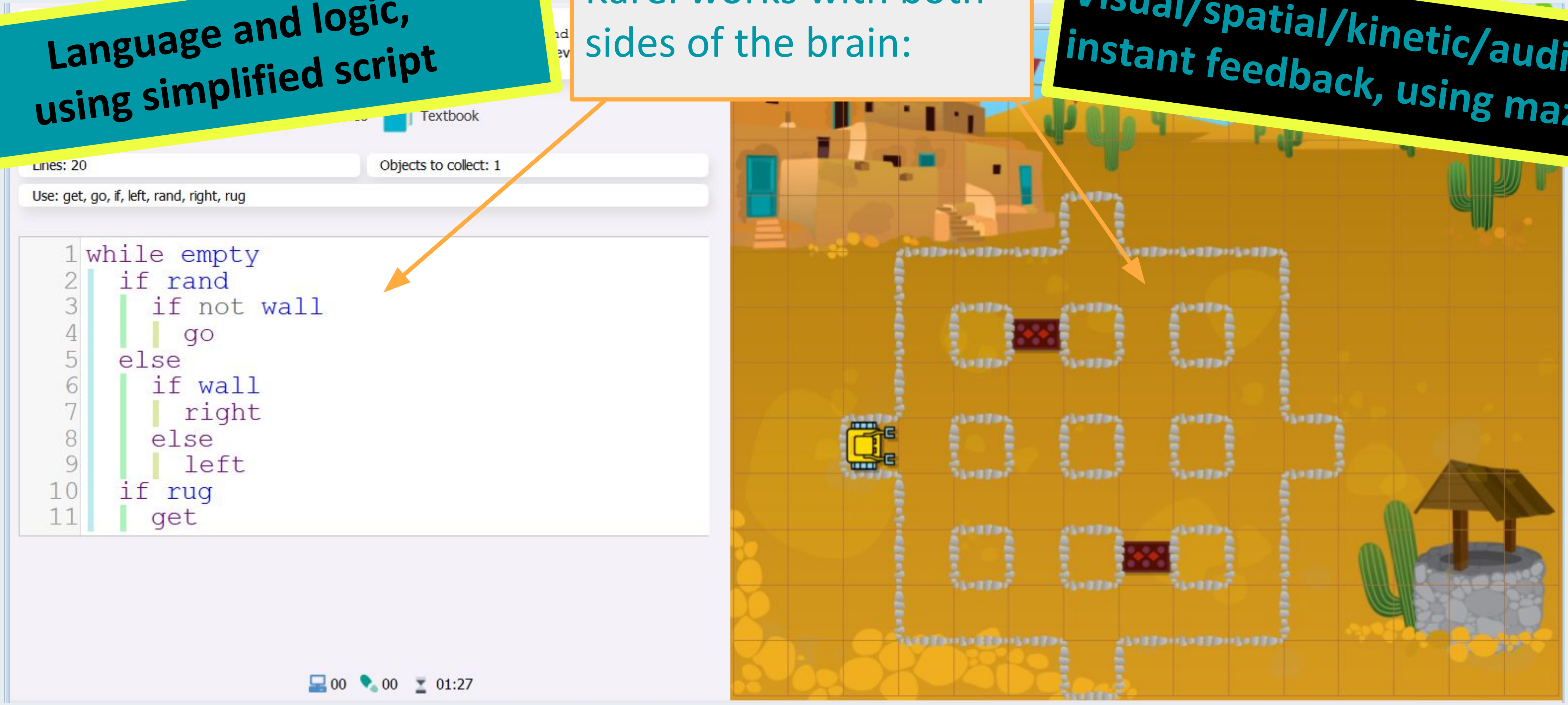
Why start with Karel?

Before tackling complex languages, computations, and tasks, students need to develop **logical reasoning**.

Language and logic, using simplified script

Karel works with both sides of the brain:

Visual/spatial/kinetic/auditory instant feedback, using mazes



Lines: 20

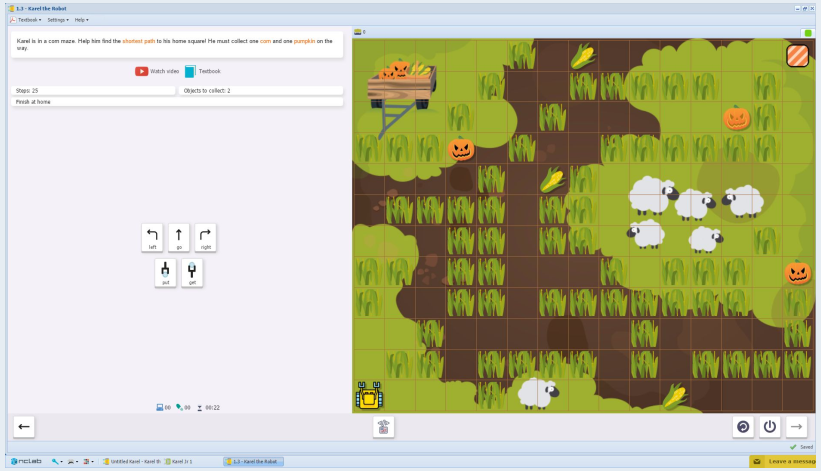
Objects to collect: 1

Use: get, go, if, left, rand, right, rug

```
while empty
  if rand
    if not wall
      go
  else
    if wall
      right
    else
      left
  if rug
    get
```

00 00 01:27

What do my students learn?

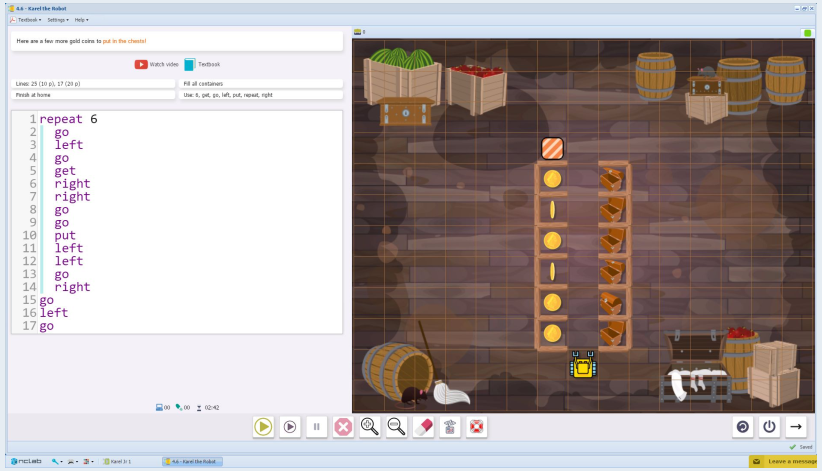


Karel is in a corn maze. Help him find the shortest path to his home square! He must collect one corn and one pumpkin on the way.

Steps: 25

Objects to collect: 2

Finish at home



Here are a few more gold coins to put in the chests!

Lines: 25 (10 p), 17 (20 p)

Fill all containers

Finish at home

Use: 6, get, go, left, put, repeat, right

```
repeat 6
  go
  left
  go
  get
  right
  right
  go
  go
  put
  left
  left
  go
  right
go
left
go
```

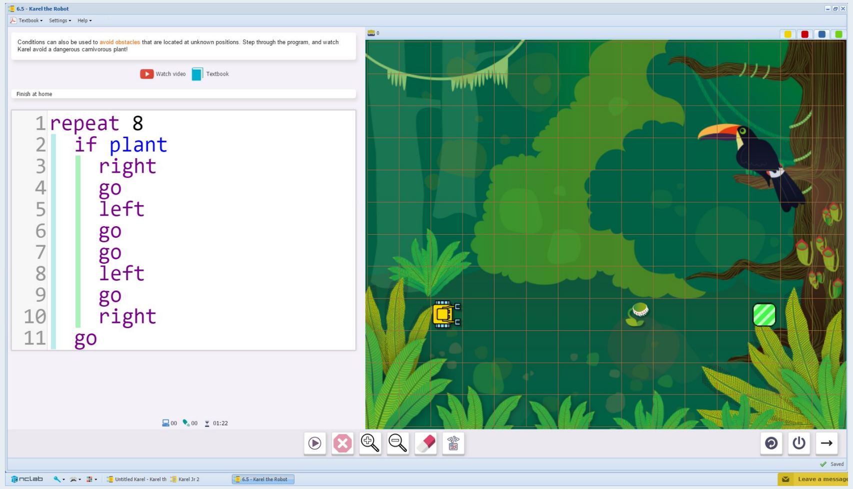
Beginning level skills - quick to learn

UNIT 1: Students learn how to guide the robot, type simple programs, recognize repeated patterns, and use the repeat loop.

At the end of this course, they should be able to create their own mazes with features such as nested loops.

What will my students learn?

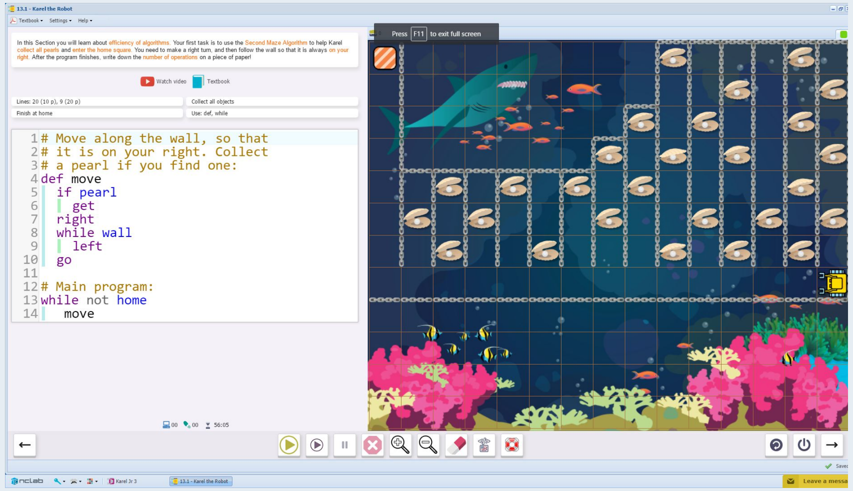
Intermediate level skills



Conditions can also be used to avoid obstacles that are located at unknown positions. Step through the program, and watch Karel avoid a dangerous carnivorous plant!

Finish at home

```
repeat 8
  if plant
    right
    go
    left
    go
    go
    left
    go
    right
  go
```



In this Section you will learn about efficiency of algorithms. Your first task is to use the Second Maze Algorithm to help Karel collect all pearls and enter the home square. You need to make a right turn, and then follow the wall so that it is always on your right. After the program finishes, write down the number of operations on a piece of paper!

Lines: 20 (10 p), 9 (20 p)

Collect all objects

Finish at home

Use: def, while

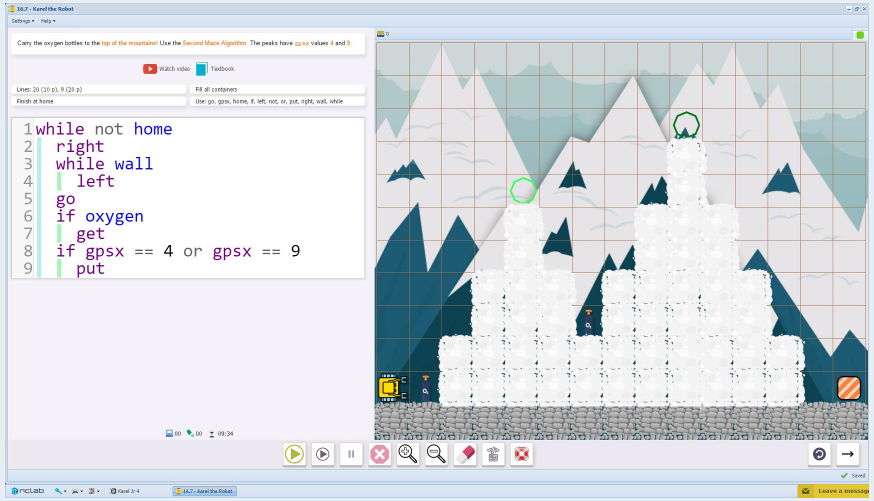
```
# Move along the wall, so that
# it is on your right. Collect
# a pearl if you find one:
def move
  if pearl
    get
  right
  while wall
    left
  go

# Main program:
while not home
  move
```

UNIT 2: Students will learn how to use **if/else conditions**, the **while loop**, and how to combine loops and conditions together.

UNIT 3: Students will learn how to use **custom commands**, local and global **variables**, and **functions** that return values.

What will my students learn?

Advanced level skills



Carry the oxygen bottles to the top of the mountains! Use the Second Maze Algorithm. The peaks have gpsx values 4 and 9.

Lines: 20 (10 p), 9 (20 p)

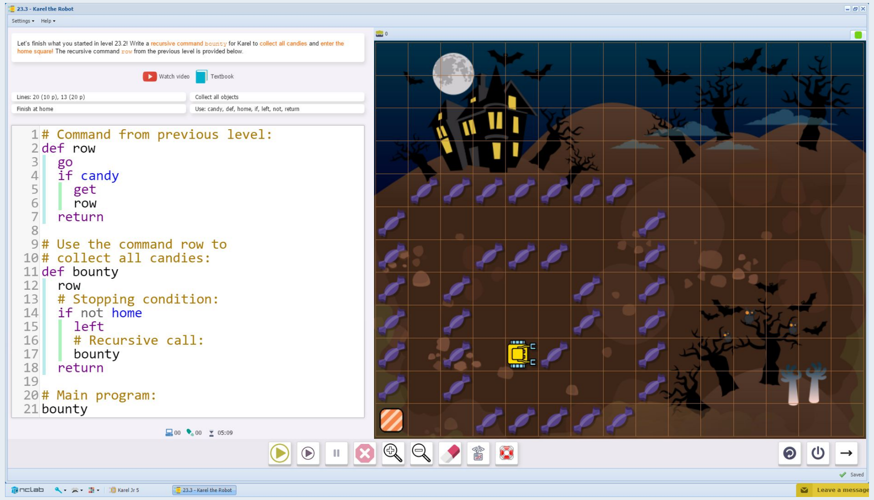
Fill all containers

Finish at home

Use: go, gpsx, home, if, left, not, or, put, right, wall, while

```
while not home
  right
  while wall
    left
  go
  if oxygen
    get
  if gpsx == 4 or gpsx == 9
    put
```

UNIT 4: Students will learn how to use **GPS coordinates**, **comparison symbols**, **Boolean values**, and **random variables**.



Let's finish what you started in level 23.2! Write a recursive command bounty for Karel to collect all candies and enter the home square! The recursive command row from the previous level is provided below.

Lines: 20 (10 p), 13 (20 p)

Collect all objects

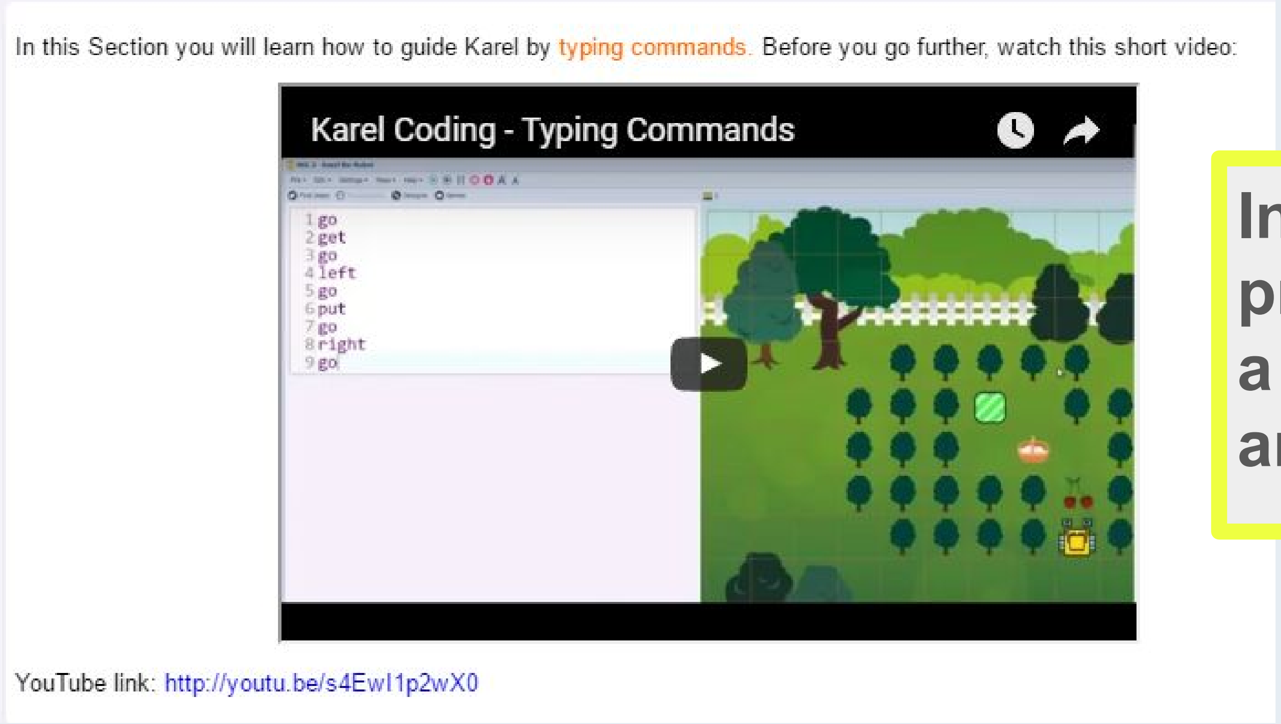
Finish at home

Use: candy, def, home, if, left, not, return

```
# Command from previous level:
def row
  go
  if candy
    get
    row
  return

# Use the command row to
# collect all candies:
def bounty
  row
  # Stopping condition:
  if not home
    left
    # Recursive call:
    bounty
  return

# Main program:
bounty
```

UNIT 5: Students will learn how to make **random decisions**, use **recursion**, and solve **advanced** programming challenges.

How do my students learn?



In this Section you will learn how to guide Karel by typing commands. Before you go further, watch this short video:

Karel Coding - Typing Commands

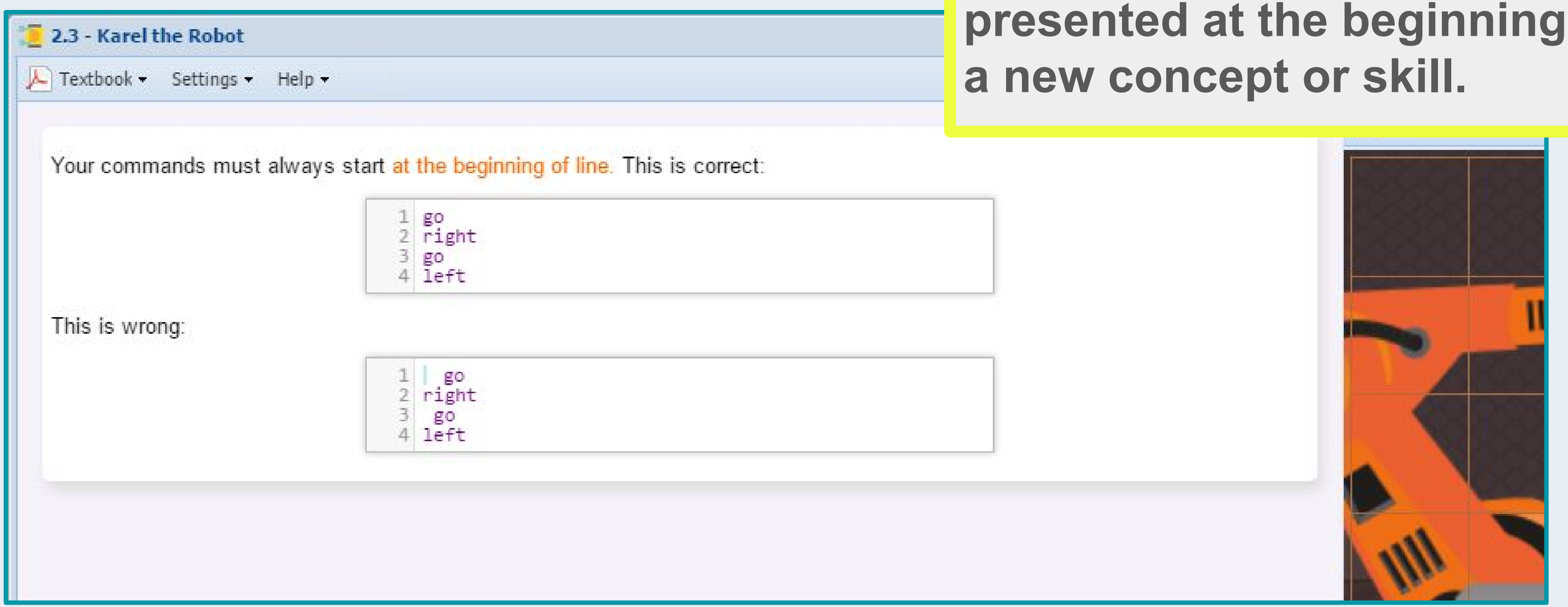
```
go
get
go
left
go
put
go
right
go
```

YouTube link: http://youtu.be/s4Ewl1p2wX0

Instructional Youtube videos, presented at the beginning of a new concept and available any time.

How do my students learn?

Instructional screens, presented at the beginning of a new concept or skill.



Your commands must always start at the beginning of line. This is correct:

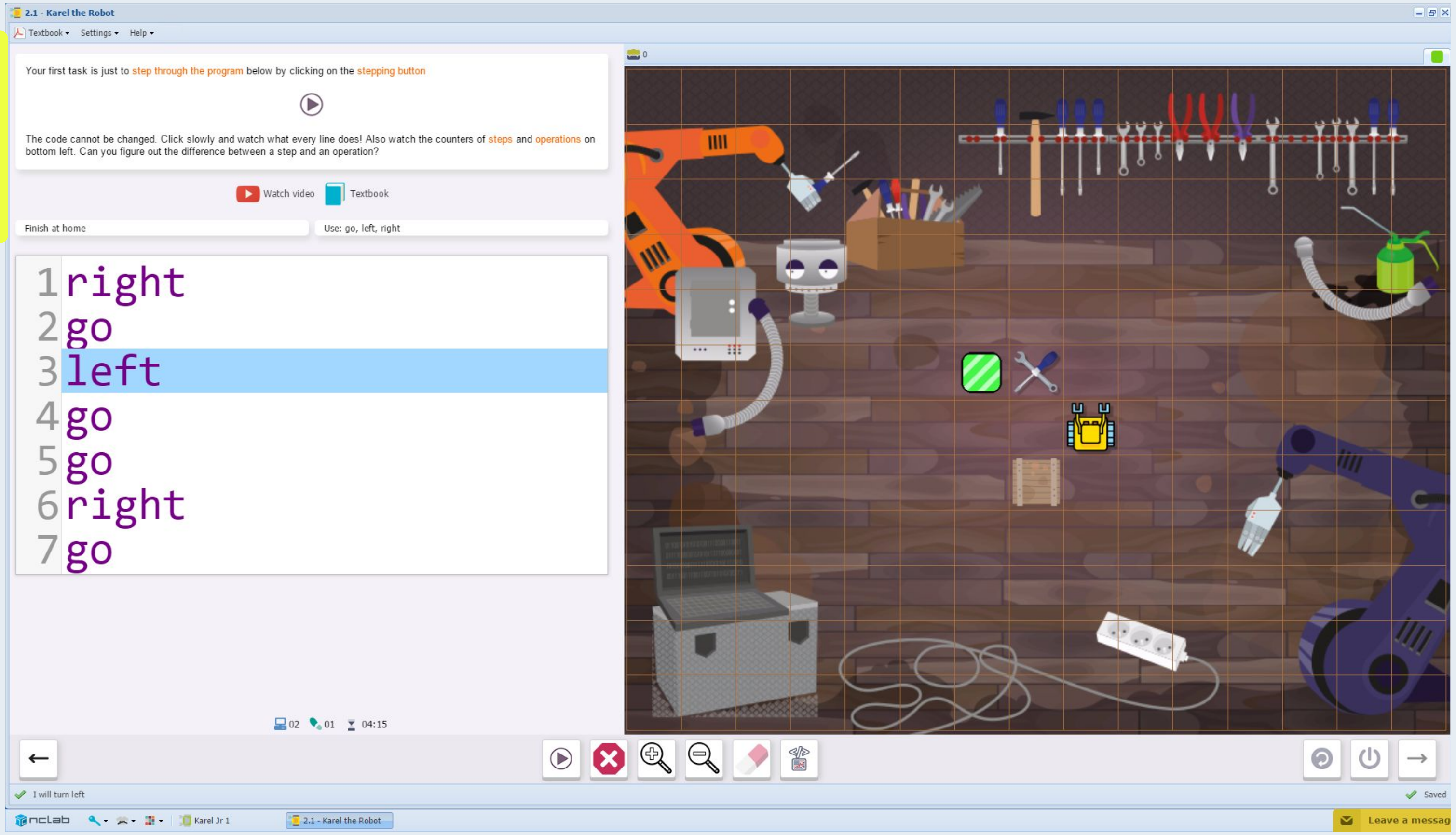
```
go
right
go
left
```

This is wrong:

```
  go
right
 go
left
```

How do my students learn?

Step by step demonstration levels.



Your first task is just to step through the program below by clicking on the stepping button

The code cannot be changed. Click slowly and watch what every line does! Also watch the counters of steps and operations on bottom left. Can you figure out the difference between a step and an operation?

Finish at home

Use: go, left, right

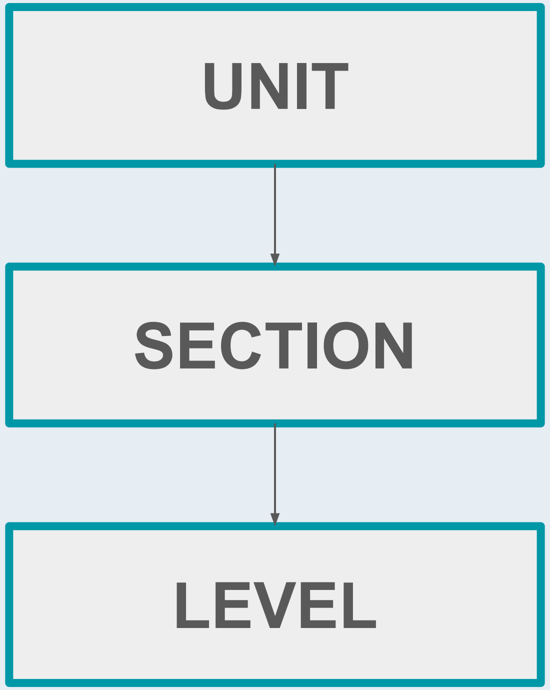
```
right
go
left
go
go
right
go
```

How do my students learn?

There are 5 units in Karel

Each course is made up of 5 sections

Each section is made up of 7 levels



UNIT

SECTION

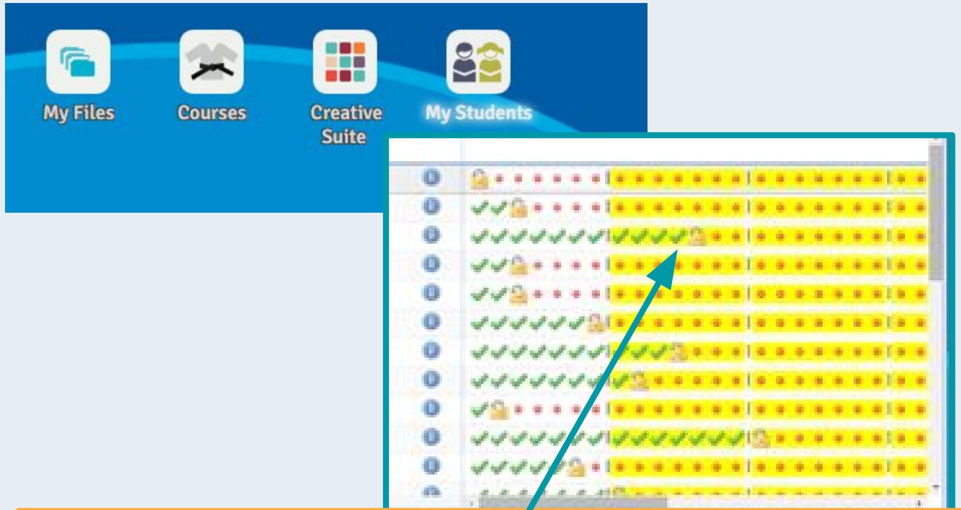
LEVEL

A section represents a concept arc.

A level represents one step of instruction or practice within the concept arc.

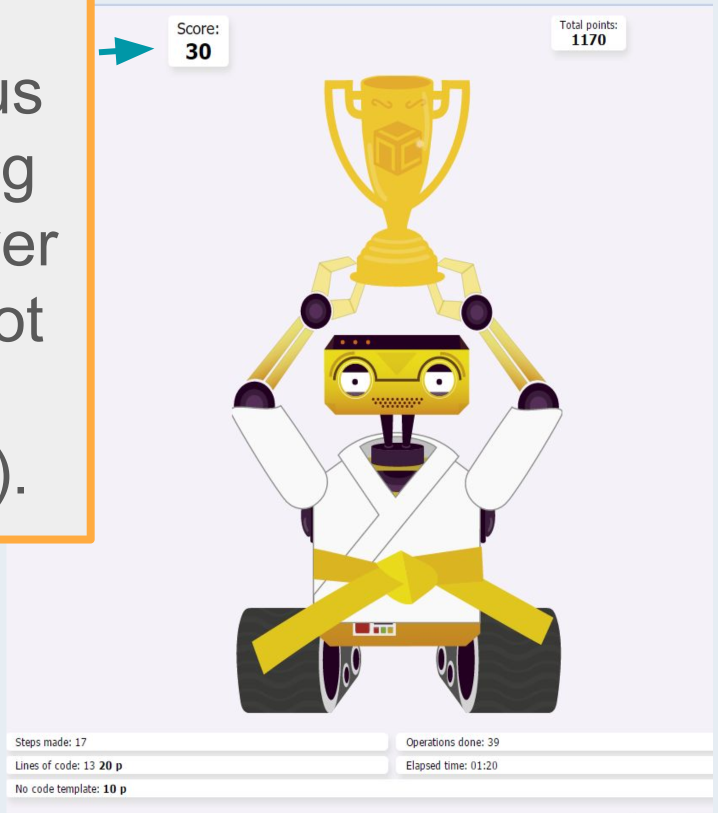
Systematic instruction and practice with gradual release - 175 levels in all.

How do I monitor and assess my students?



My Students shows how many points students have earned and what levels they are working on. You can get precise information by clicking on a completed level.

Students earn points for each level, with bonus points for writing the code in fewer lines, and for not using the code template (hints).



Score: 30

Total points: 1170

Steps made: 17

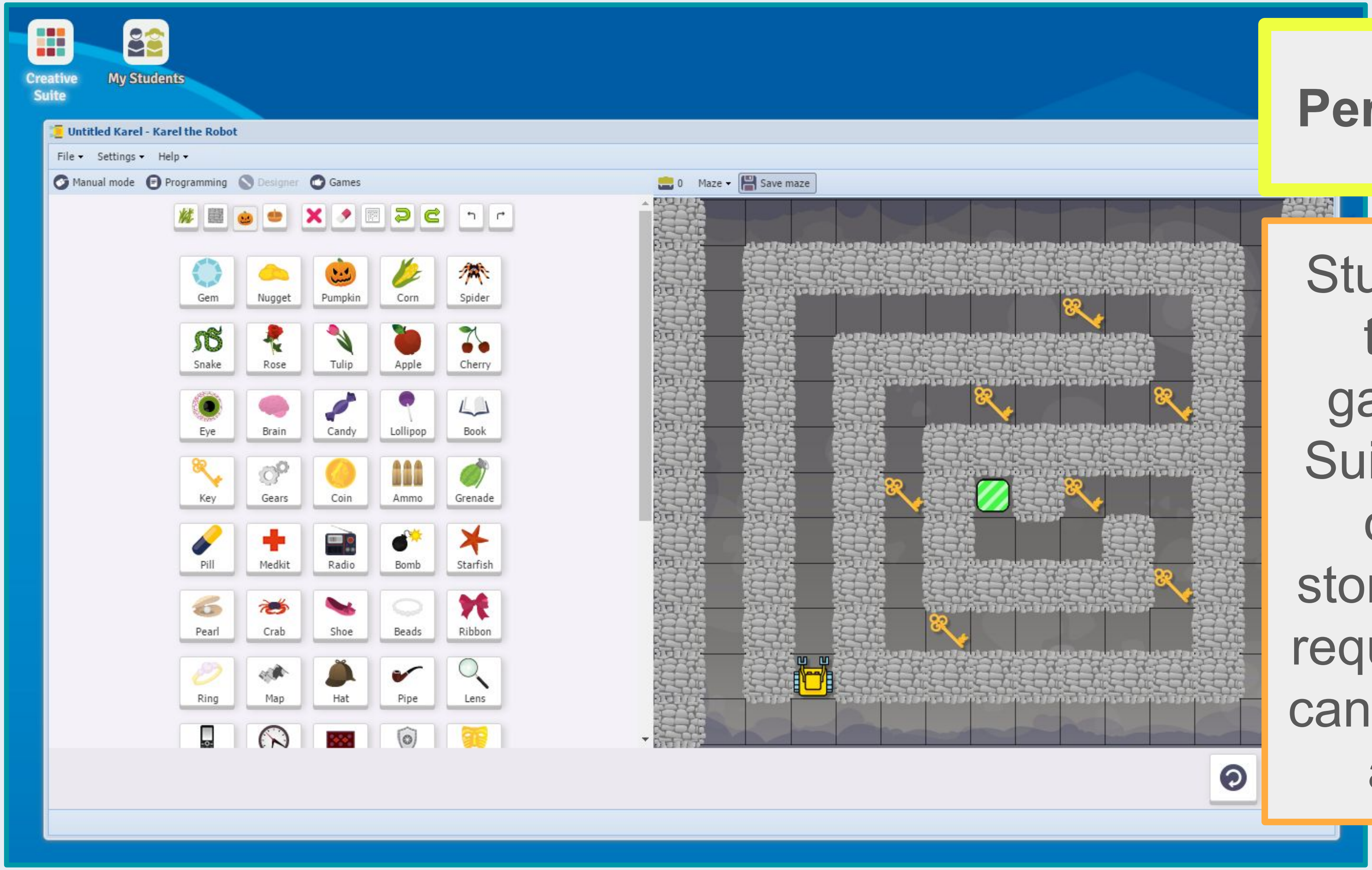
Operations done: 39

Lines of code: 13 **20 p**

Elapsed time: 01:20

No code template: **10 p**

How do I monitor and assess my students?

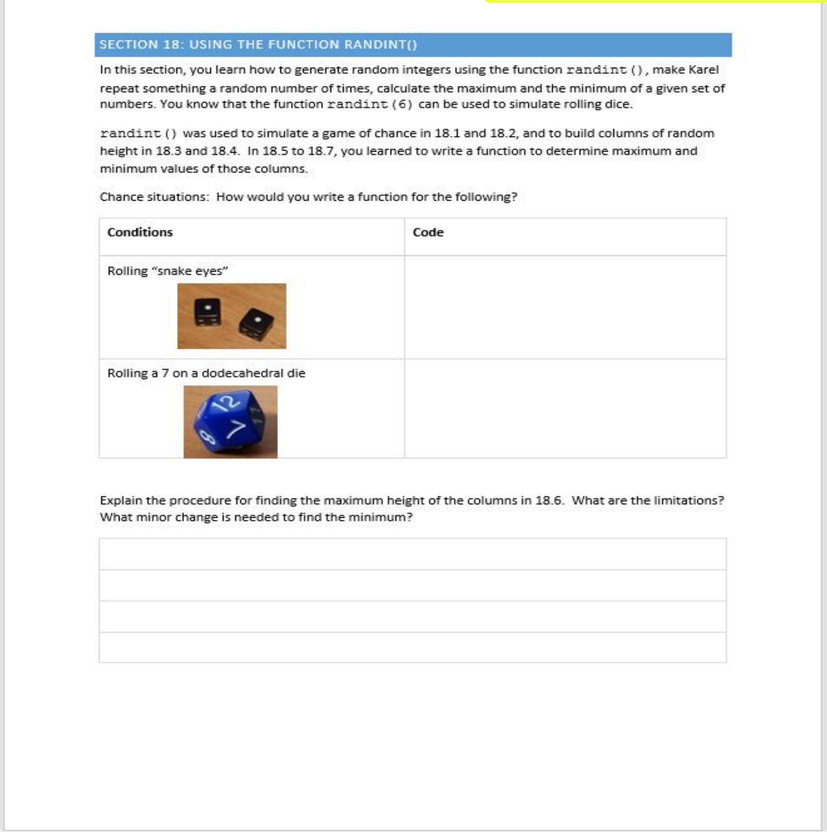


Performance Tasks

Students can create their own Karel games in Creative Suite, complete with code solutions, storylines, and game requirements. These can be saved, shared and published.

How do I monitor and assess my students?

Student Journals



SECTION 18: USING THE FUNCTION RANDINT()

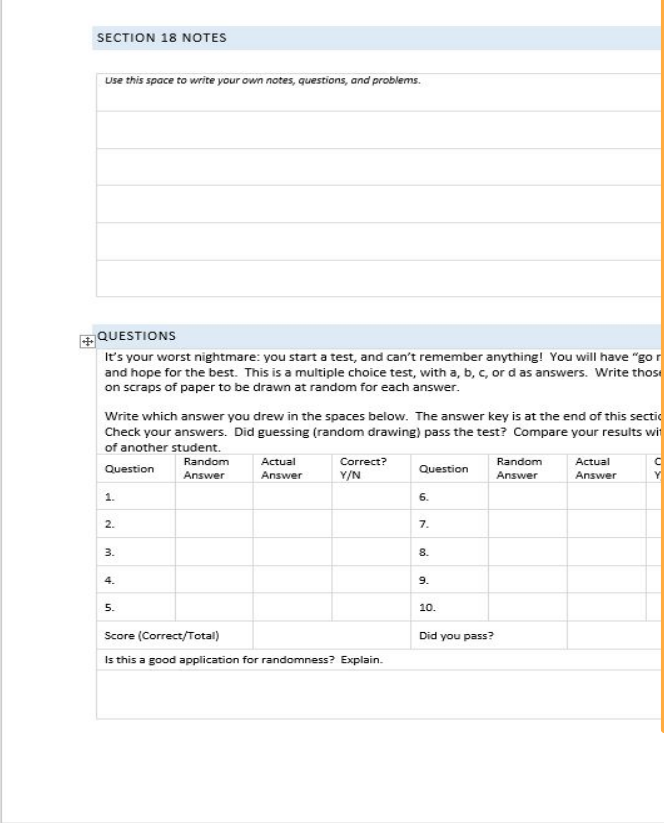
In this section, you learn how to generate random integers using the function `randint()`, make Karel repeat something a random number of times, calculate the maximum and the minimum of a given set of numbers. You know that the function `randint(6)` can be used to simulate rolling dice.

`randint()` was used to simulate a game of chance in 18.1 and 18.2, and to build columns of random height in 18.3 and 18.4. In 18.5 to 18.7, you learned to write a function to determine maximum and minimum values of those columns.

Chance situations: How would you write a function for the following?

Conditions	Code
Rolling "snake eyes"	
Rolling a 7 on a dodecahedral die	

Explain the procedure for finding the maximum height of the columns in 18.6. What are the limitations? What minor change is needed to find the minimum?



SECTION 18 NOTES

Use this space to write your own notes, questions, and problems.

QUESTIONS

It's your worst nightmare: you start a test, and can't remember anything! You will have "go r and hope for the best. This is a multiple choice test, with a, b, c, or d as answers. Write thos on scraps of paper to be drawn at random for each answer.

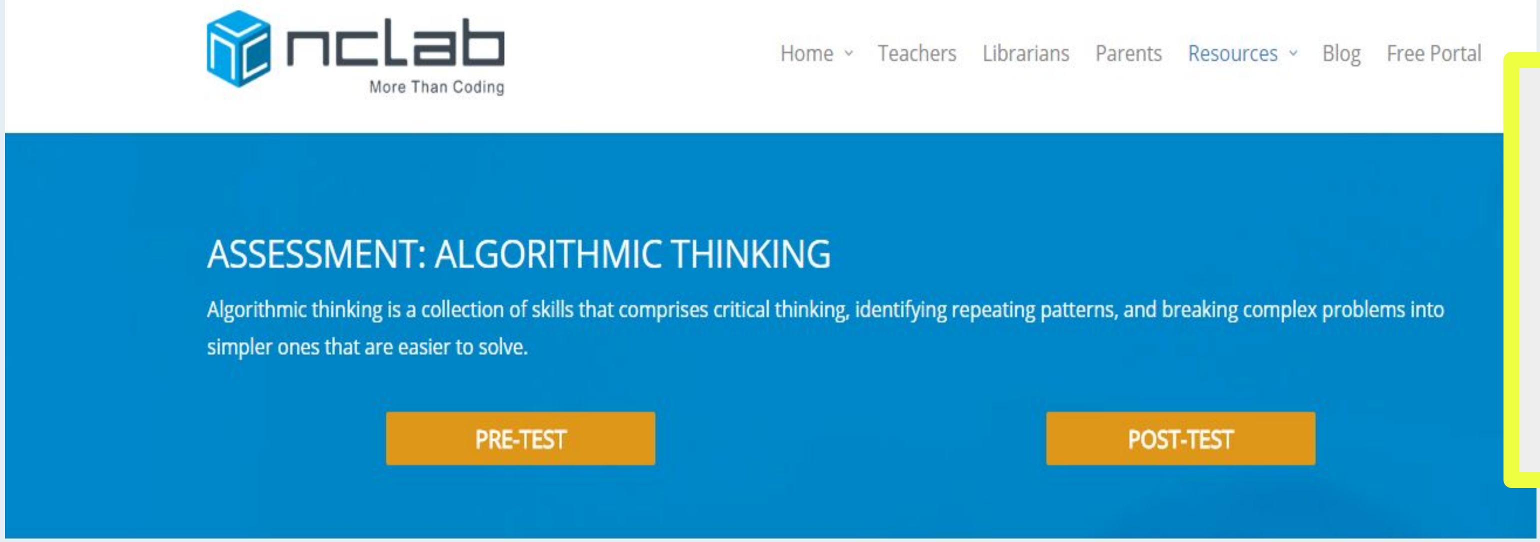
Write which answer you drew in the spaces below. The answer key is at the end of this secti Check your answers. Did guessing (random drawing) pass the test? Compare your results wi of another student.

Question	Random Answer	Actual Answer	Correct? Y/N	Question	Random Answer	Actual Answer	C Y
1.				6.			
2.				7.			
3.				8.			
4.				9.			
5.				10.			
Score (Correct/Total)				Did you pass?			
Is this a good application for randomness? Explain.							

Each Section has 4 pages:

1. Recall questions
2. Notes/Application questions
3. Bulletin Board to post links, pictures, ideas
4. Design page for designing games

How do I monitor and assess my students?



ASSESSMENT: ALGORITHMIC THINKING

Algorithmic thinking is a collection of skills that comprises critical thinking, identifying repeating patterns, and breaking complex problems into simpler ones that are easier to solve.

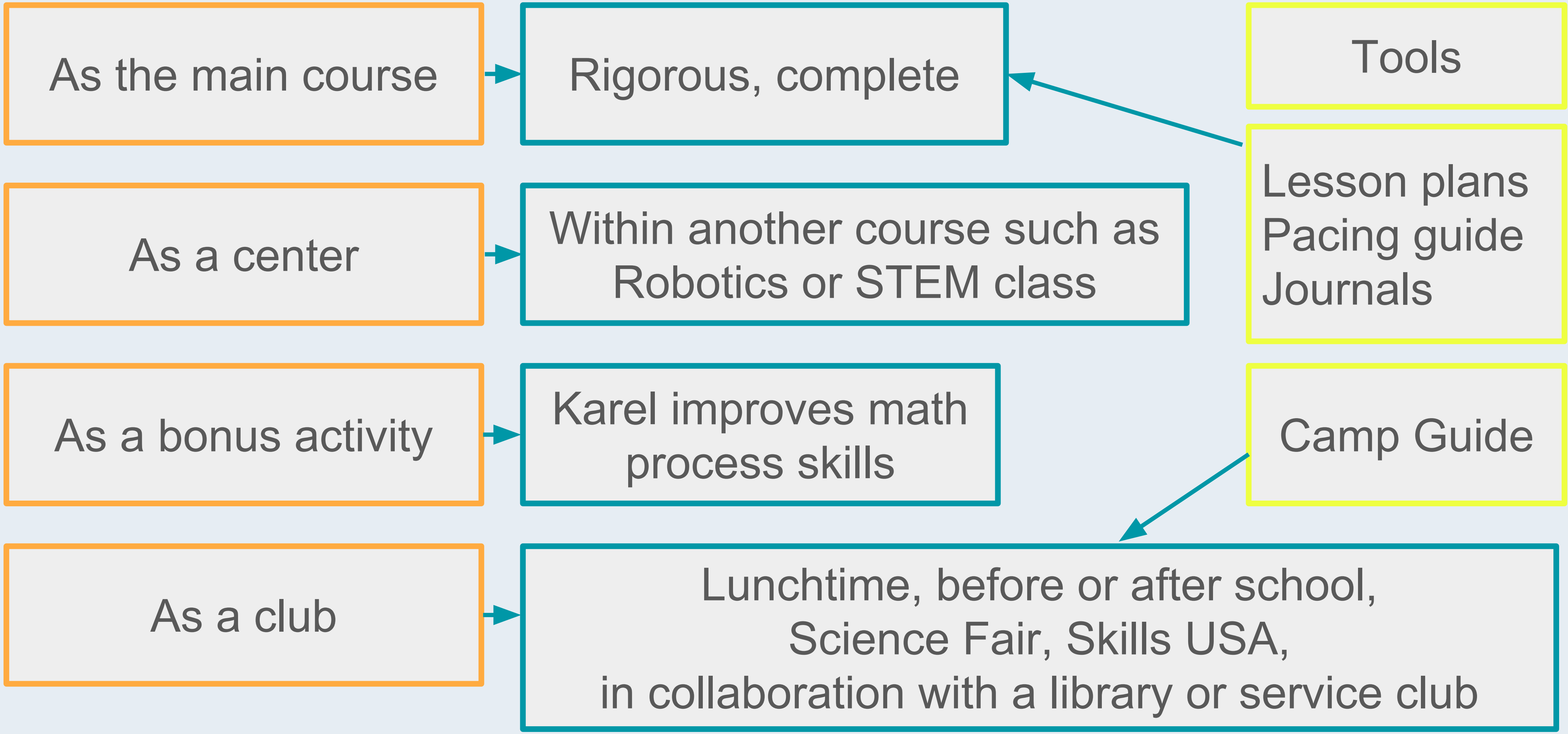
Have students take the pre-test on the NCLab website before starting Karel, then the post-test afterwards

Coming soon...

Built-in quizzes

End of unit assessments

How does Karel fit in my curriculum?



As the main course → Rigorous, complete

As a center → Within another course such as Robotics or STEM class

As a bonus activity → Karel improves math process skills

As a club → Lunchtime, before or after school,
Science Fair, Skills USA,
in collaboration with a library or service club

Tools

Lesson plans
Pacing guide
Journals

Camp Guide

What else is there beside this course?

Hour of Code Karel

For a quick taste of Karel

More Courses

Tina Turtle
3D Modeling
Python

Creative Suite

Free programming, modeling, publishing, math tools and more

Cloud Storage

Free cloud storage for files created in Creative Suite

On-line Gallery

Students can submit their games to be published

Blog

Ideas, events, and contests on our blog
(Newsletter coming soon)

How Can We Help You?

Email and phone support?

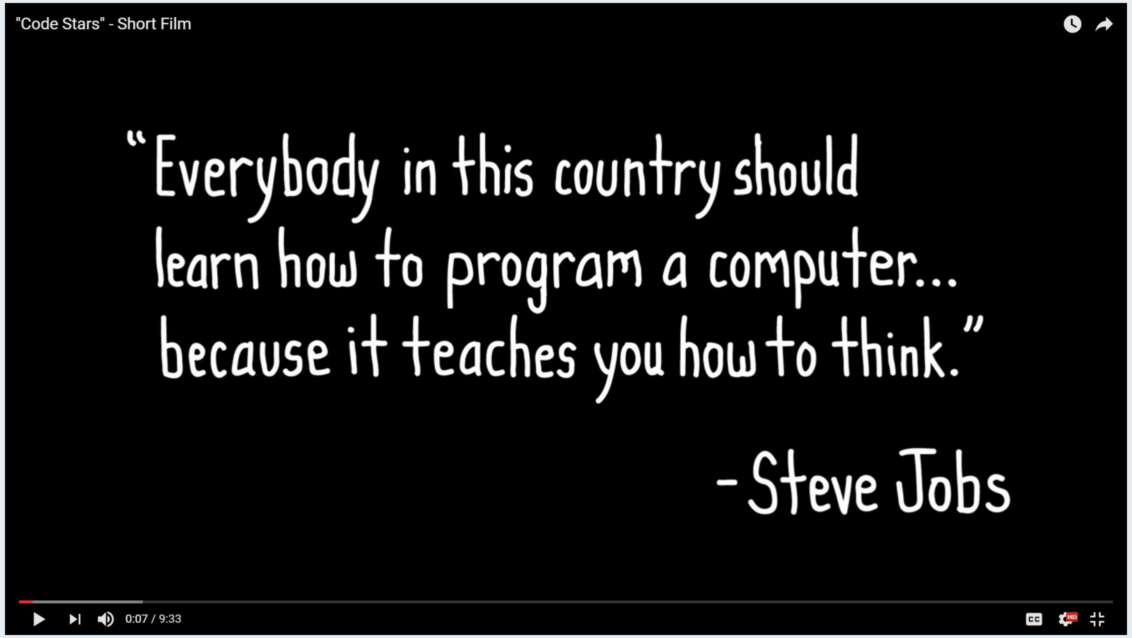
Hangout Meeting?

Live training?

Webinar (your topic)?

Monthly newsletter?

What would you like to see?



"Code Stars" - Short Film

"Everybody in this country should learn how to program a computer... because it teaches you how to think."

-Steve Jobs

0:07 / 9:33

nclab

More Than Coding

We love to hear from you!

Email: support@nclab.com

Phone: (800) 666-2024 or (775) 303-6075