

## Karel the Robot

### Karel's World

Plain made of streets & avenues

Corners or intersection

Location

Determined by avenue & street numbers

Positive integers

Origin/Start is 1<sup>st</sup> avenue & 1<sup>st</sup> street

### Walls

Made of neutronium

Obstacles -- Karel cannot pass through a wall

Located

Between streets

Between avenues

From origin, Karel's World has a huge immovable wall on his west side & south side

### Beepers

Pinging beeper

Found at intersections

Karel can do things with the beepers

pick up one at a time

carry

put down

### Karel is a Robot

#### Karel cannot think

Karel is remarkable at his ability to follow instructions

As long as these instructions are VERY detailed

Karel has capabilities

move forward

turn in place

knows which direction it is facing – compass

is equipped with three cameras

forward

right

left

These cameras have a ½ block range

Karel can hear

Can detect a beeper on the same corner Karel is standing

Karel's has a Beeper Bag

Stores beepers

Soundproof – Karel cannot hear beepers in the bag

Karel can put beepers in the bag and remove beepers from the bag

### Karel's Tasks

Not isn't very smart

But, Karel is very good at following instructions

The algorithm or step-by-step set of instructions Karel follows is called a program

Karel understands a very simple programming language

By computer standards, it is a high-level language

The computer inside Karel doesn't actually "speak" the language of our programs

Our programs will have to be translated into the machine code (0,1) that the

computer understands. To do this we will **compile** our programs.

## Karel the Robot

How does Karel know what to do?

We write a program that gives Karel its instructions

Karel's Programming Language

vocabulary

punctuation marks

rules of grammar

Simple, yet powerful

### Tasks & Situations

What is a task?

Something we want Karel to do

What is a situation?

Description of Karel's world

Includes

Size & location of wall(s)

Location & number of beepers

Karel's location & direction Karel is facing

Initial Situation (Initialize)

Situation when Karel is assigned a task

Final Situation

After task is completed

### Primitive Instructions

move

turnleft

pickbeeper

putbeeper

turnoff

They allow Karel to move through world & handle beepers

Provide for avoiding obstacles & transporting beepers

### Program & Instruction Execution

An instruction is executed when Karel performs the task it is told to

A program is executed when the instructions in a program are carried out

### Primitive Instruction Details

Changing Position

move

Karel moves forward one block

Karel remains facing in the same direction

If a wall is in Karel's way & a move instruction is executed, Karel shuts off

• Error shutoff

turnleft

Karel pivots 90 degrees to left

Location does not change

Cannot cause error shutoff

Why not?

Handling Beepers

pickbeeper

Karel picks up a beeper from the corner on which it is standing and deposits beeper in the beeper bag

The pickbeeper command work on one beeper at a time

There may be more than one beeper on a corner

## Karel the Robot

What if Karel tries to pick up a beeper when no beeper is at the corner?

Error shutoff

putbeeper

Karel takes a beeper out of beeper bag and places beeper on corner

What if there are not any beepers in the bag?

Error shutoff

Completing a Task – Telling Karel it's done for the task

turnoff

Karel must be shut down after task is completed

Last instruction in every program

### What does a Karel the Robot Program look like?

BEGINNING-OF-PROGRAM

BEGINNING-OF-EXECUTION

*Instructions, ending with a semicolon (;)*

END-OF-EXECUTION

END-OF-PROGRAM

The instructions are made up of the **Primitive** commands Karel understands.

Karel's Language has **Reserved Words** that structure the program.

BEGINNING-OF-PROGRAM

BEGINNING-OF-EXECUTION  
END-OF-EXECUTION

END-OF-PROGRAM

### Errors, Bugs, Debugging

Four kinds of errors can occur in Karel's Language

Lexical Errors

When Karel tries to read a word it doesn't understand

Spelling

Syntax Errors

Errors in Grammar and Punctuation, Ex.

Reserved words in the wrong order

Missing semicolon (;)

Execution Errors

When Karel is asked to do something it cannot do

Pick up a beeper where none exists

Walk into a wall

Logic Errors

Hardest to find and recognize

Where program has no obvious errors but it doesn't solve the problem

Or Karel executes part way and turns off

### Running/Executing a Program

Instructions between BEGINNING-OF-EXECUTION and END-OF-EXECUTION are acted on in order (from top to bottom) until a turnoff instruction or error shutoff is encountered.

## Karel the Robot

How do we know what to ask Karel to do?

Simulate – we model our solution on paper before we ever write a line in Karel's language

We **trace** the program to make sure we have given Karel all the instructions needed

Graph paper might be a good idea

Verification

If we run Karel's program, and our goal is not achieved?

We again trace the program seeing if we can find where he gave Karel incorrect information.

Verification

### Using Karel the Robot in Lab

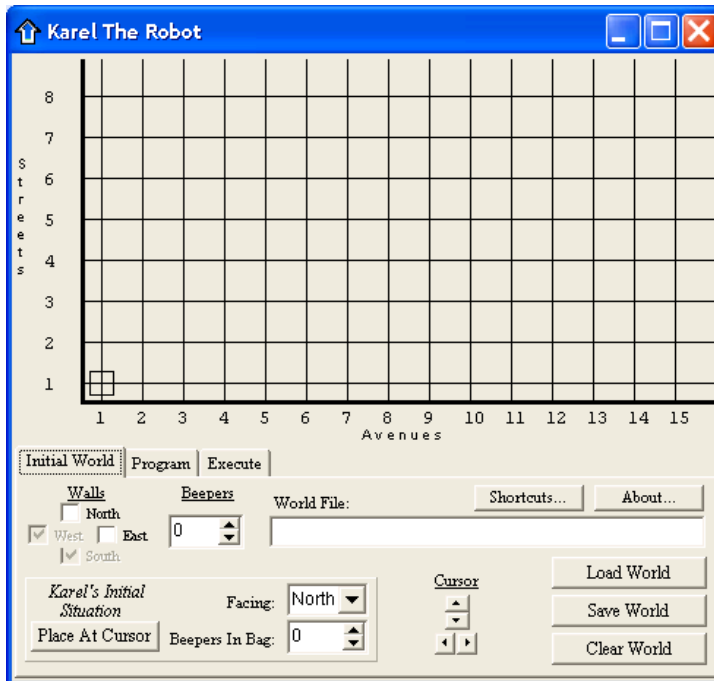
#### Different "Windows"

**World View** (Window)

**Program View** (Window)

**Execution** Window

#### World View



While there appears to be nothing in Karel's World, we have quite a lot of information in the bottom portion of the screen

Notice: under Walls only North and East are black. Karel can only move North or East

Under the section called Karel's Initial Situation, we are told

Karel is facing North

Karel has 0 beepers in the beeper-bag.

On the right hand side of the screen are three buttons:

Load World

Inputs a World that has been already created for Karel

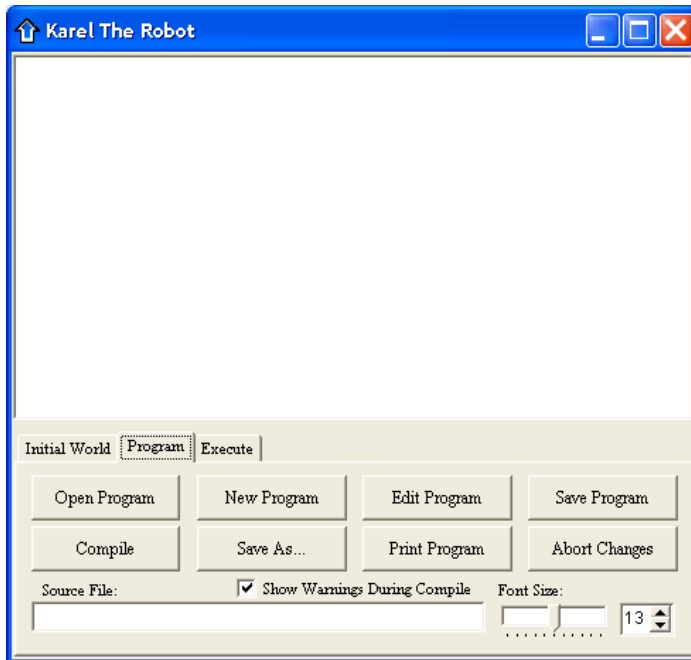
Save World

Saves a World that has been created and is visible on the Initial World screen

## Karel the Robot

Worlds end in .kw (for Karel World)  
 Clear World  
 Erases whatever is on the screen without saving it

### Program View



Unlike the Initial World screen, this screen is really empty  
 Buttons lists tasks that the Programmer is able to perform

#### Open Program

Opens up a Karel program that has been created and Saved by the Karel environment. These programs end in .kp (for Karel Program)

#### New Program

Provides a place to enter a new Karel program, and provides the necessary beginning and ending commands.

#### Edit Program

Take a program that you are working with and make changes

#### Save Program

**VERY IMPORTANT**

Programs and Worlds **MUST** be SAVED

Magic doesn't happen! The computer inside Karel doesn't know you want to save things unless you actually do so!

#### Compile

Translate the Karel Programming Language into Machine-code

#### Save As...

You can use this to change the name of a program

When might you want to do this?

#### Print Program

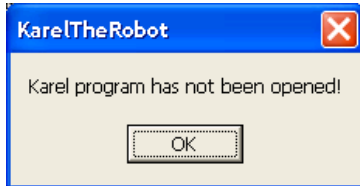
Print out the program on the printer

#### Abort Program

STOP! Halt! End. Terminate

You can also enter a source file in at the bottom rather than hitting open and finding one

## Karel the Robot

**Execution View**

You cannot move into the Execution view unless you have created and saved or loaded a program.

Execution = Run

You can't run a program that hasn't been entered somehow into the Program View.

Create a World

    Beeper inside a box

Write a Program

**Problem statement:** Karel is to go to the open side of the box, go inside and retrieve the beeper, then go home.

Remember our Algorithm Planner

Define the output

Define the input

Define the initial algorithm

Refine the algorithm

Define the program

**Define the output:** Karel ends up at the origin having retrieved a beeper from the box

**Define the input:** Karel starts at the origin with no beepers in the beeper-bag

**Define the initial algorithm**

Karel is at the origin

Karel is to go to the open side of the box

Karel goes inside and retrieve the beeper

Karel goes home

This is a pretty good initial algorithm. It describes in English everything Karel is supposed to do. There are two problems with this algorithm.

    It isn't specific enough

    Too many things are done at once

    It doesn't always give enough information

**Refine the algorithm**

Karel is initialized at the origin with an empty beeper-bag

Karel is to go to the open side of the box

    Go up to 5<sup>th</sup> street

## Karel the Robot

```

    Turnright
    Go another 3 blocks
    Make another right
Karel goes inside and retrieve the beeper
    Go one more block
    Pickup the beeper (and put it in the beeper-bag)
Karel goes home
    Travel the reverse of the directions given before

```

This is a lot closer to what our refined final algorithm should be. The biggest problem is the last of the directions which makes a lot of assumptions about Karel's abilities that aren't true. We have to tell Karel how to go home. Reverse the directions isn't sufficient.

**Refine Again**

```

Karel is initialized at the origin with an empty beeper-bag
Go up to 5th street
Turnright
Go another 3 blocks
Make another right
Go one more block
Pickup the beeper (and put it in the beeper-bag)
Turn around
Go one block
Make a left
Go 3 blocks
Turnleft again
Go 5 blocks
Turnaround

```

**Define the program**

Now we have to translate our English into Karel's language

```

move;
move;
move;
move;
move;
turnleft;
turnleft;
turnleft;
move;
move;
move;
turnleft;
turnleft;
turnleft;
move;
pickbeeper
turnleft;
turnleft;
move;
turnleft;
move;
move;
move;

```

## Karel the Robot

```
turnleft  
move;  
move;  
move;  
move;  
move;  
turnleft;  
turnleft;
```

After correction syntax errors, we find that the program compiles perfectly. However, it doesn't do what we expected it to do.

If we look at our refined algorithm, we said **Go up to 5<sup>th</sup> street**. Our program does 5 initial move; commands. This takes us to 6<sup>th</sup> street instead. As a result, everything that follows doesn't turn out as expected. Karel follows the instructions as best it can, but when it tries to pick up a beeper at the corner of 5<sup>th</sup> Street and 4<sup>th</sup> Avenue, it turns itself off, because there is no beeper at that location. The beeper was at 4<sup>th</sup> Street and 4<sup>th</sup> Avenue.