

# 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 1st avenue & 1st street

# Karel the Robot

## Walls

- Made of neutronium
- Obstacles
  - Karel cannot pass through a wall
- Located
  - Between streets
  - Between avenues
- From origin, Karel's World has a huge immoveable wall on his west side & south side

# Karel the Robot

## Beepers

- Pinging beeper
  - Found at intersections
- Karel can do things with the beepers
  - pick up one at a time
  - carry
  - put down

# Karel the Robot

**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 the Robot

## 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  $\frac{1}{2}$  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 the Robot

## 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?

- Karel only does what we tell it to do.
- We write a program that gives Karel its instructions.

# Karel the Robot

## Karel's Programming Language

- vocabulary
- punctuation marks
- rules of grammar

Simple, yet powerful



# Karel the Robot

## 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

# Karel the Robot

## Primitive Instructions

- move
- turnleft
- pickbeeper
- putbeeper
- turnoff
- They allow Karel to move through world & handle beepers
- Provide for avoiding obstacles & transporting beepers

# Karel the Robot

## Program & Instruction Execution

- An instruction is **Executed** (Run) when Karel performs the task it is told to perform.
- A program is **Executed** when the instructions in a program are carried out.

# Karel the Robot

## 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

When Karel turns off due to an error this is called an

### Error shutoff

- turnleft
  - Karel pivots 90 degrees to left
  - Location does not change
  - Cannot cause error shutoff

### Why not?

# Karel the Robot

## 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

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**

# Karel the Robot

## 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

# Karel the Robot

## 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

instructions

{ END-OF-EXECUTION

END-OF-PROGRAM

# Karel the Robot

## 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



# Karel the Robot

## 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.
- 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

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

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

# Karel the Robot

## Using Karel the Robot in Lab

### Different “Windows”

- **World View** (Window)
- **Program View** (Window)
- **Execution Window**

# Karel the Robot

Create a World

There is a beeper inside a box

Karel needs to retrieve beeper and Return

Write a Program

# Karel the Robot

**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

# Karel the Robot

**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

# Karel the Robot

## 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 5th street
  - 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

# Karel the Robot

## 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

End