

## 5. Visual Programming 3

**What did we do last time?**

**SCRATCH is a new programming language  
that let you create your own interactive  
stories, animations, games, music and art.**

# SCRATCH – visual programming



[home](#) [projects](#) [galleries](#) [support](#) [forums](#) [about](#) [my stuff](#)

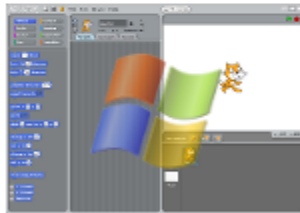
## Scratch 1.4 Download



### Scratch Installer For Mac OS X

Compatible with Mac OS X 10.4 or later

[MacScratch1.4.dmg](#)



### Scratch Installer for Windows

Compatible with Windows 2000, XP, Vista, and 7

[ScratchInstaller1.4.exe](#)

See below for additional Windows options



### Scratch Installer for Debian / Ubuntu

Compatible with Ubuntu 12.04 when backports are enabled

[Install Scratch with Software Center](#)

or [download here](#)

See the [Scratch on Linux](#) page for more information

[http://info.scratch.mit.edu/Scratch\\_1.4\\_Download](http://info.scratch.mit.edu/Scratch_1.4_Download)

# About SCRATCH

- Scratch allows the user to write programs by dragging and connecting simple programming instructions.
- The programming instructions resemble puzzle pieces and will only “fit” together in ways that make semantic sense.
- The instruction pieces are also color-coded according to what type of instruction they represent.
- The program that the user creates controls one or more objects, or sprites.

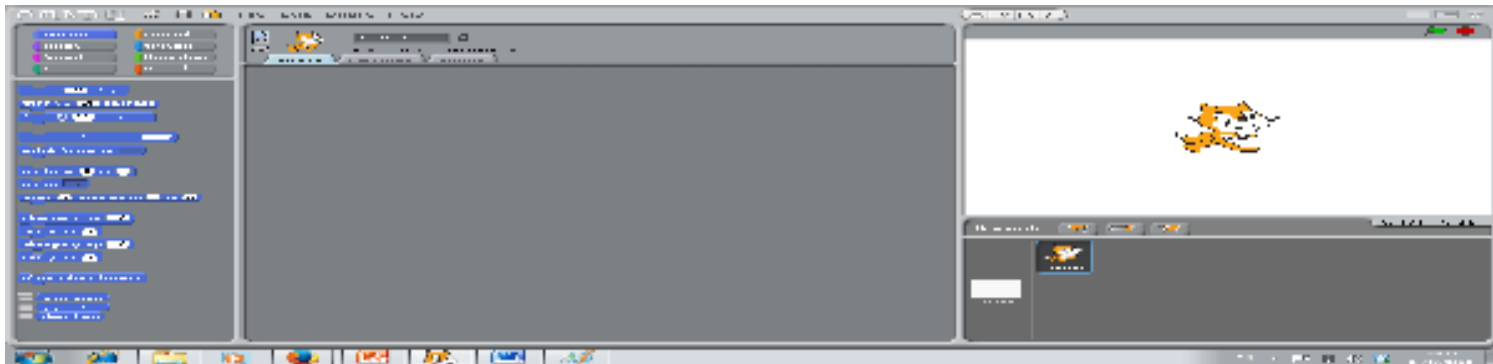


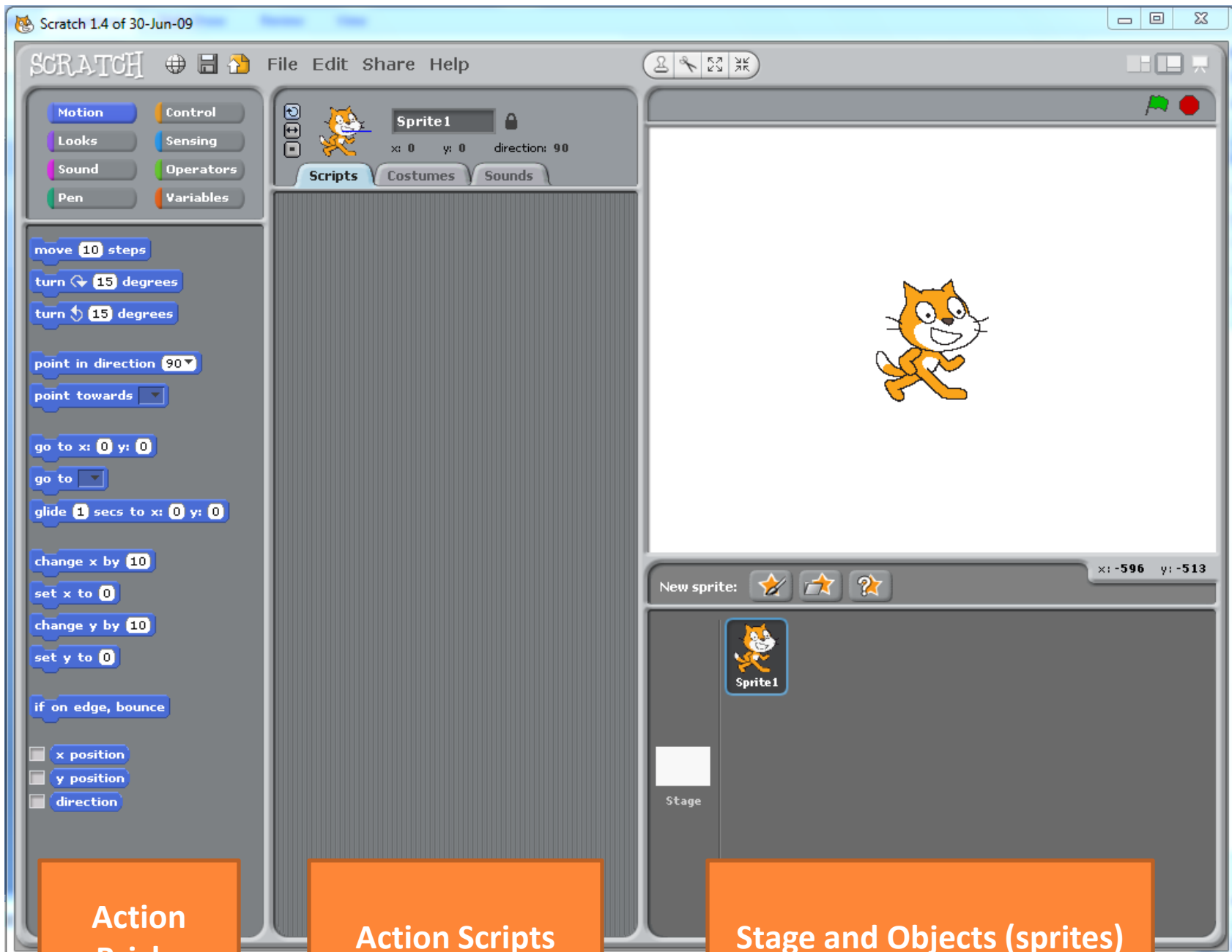
# Eight categories of programming institutions

- **Motion:** move and rotate.
- **Looks:** changing a sprite's costume and colour, and “say” and “think” messages to the user.
- **Sound:** playing drum sounds as well as 128 different instruments and sound affects.
- **Pen:** ability to draw lines under program control.
- **Control:** control structure such as while loops and if statements.
- **Sensing:** allow the user's program to test the location of a sprite or the mouse pointer.
- **Operators:** arithmetic, boolean, and string operators that can be combined to form complex expressions.
- **Variables:** allow the user to create, display and manipulate scalar and list variables.

# SCRATCH interface breaks out into 3 columns

- The **left column** contains the various instructions that the user can choose from to build a program.
- The **right column** is divided into two parts. The top part is the “stage” where all of the action takes place. The bottom part contains one or more sprites that are used in the program.
- The **center column** is where the actual programming takes place. The user simply drags programming instructions from the pallet into the center column and connects them together to build up one or more programs that control the current sprite.







- Motion
- Looks
- Sound
- Pen
- Control
- Sensing
- Operators
- Variables

Sprite 1

x: 0 y: 0 direction: 90

Scripts Costumes Sounds

switch to costume costume2

next costume

costume #

say Hello! for 2 secs

say Hello!

think Hmm... for 2 secs

think Hmm...

change color effect by 25

set color effect to 0

clear graphic effects

change size by 10

set size to 100 %

size

show

hide

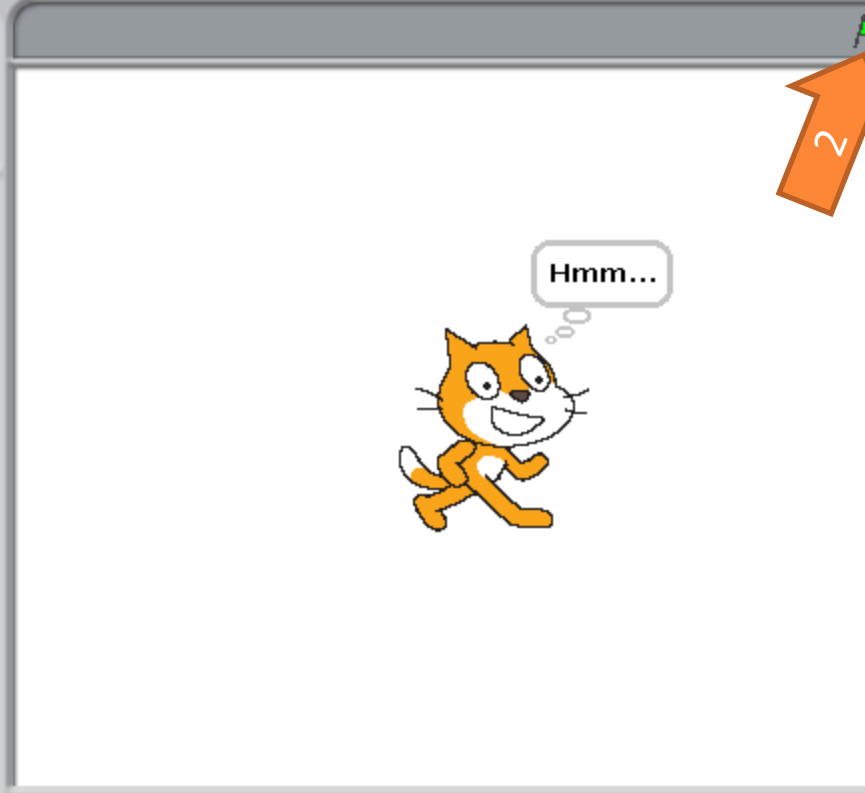
go to front

go back 1 layers



when clicked

think Hmm... for 2 secs



New sprite: Star Pencil Eraser ?

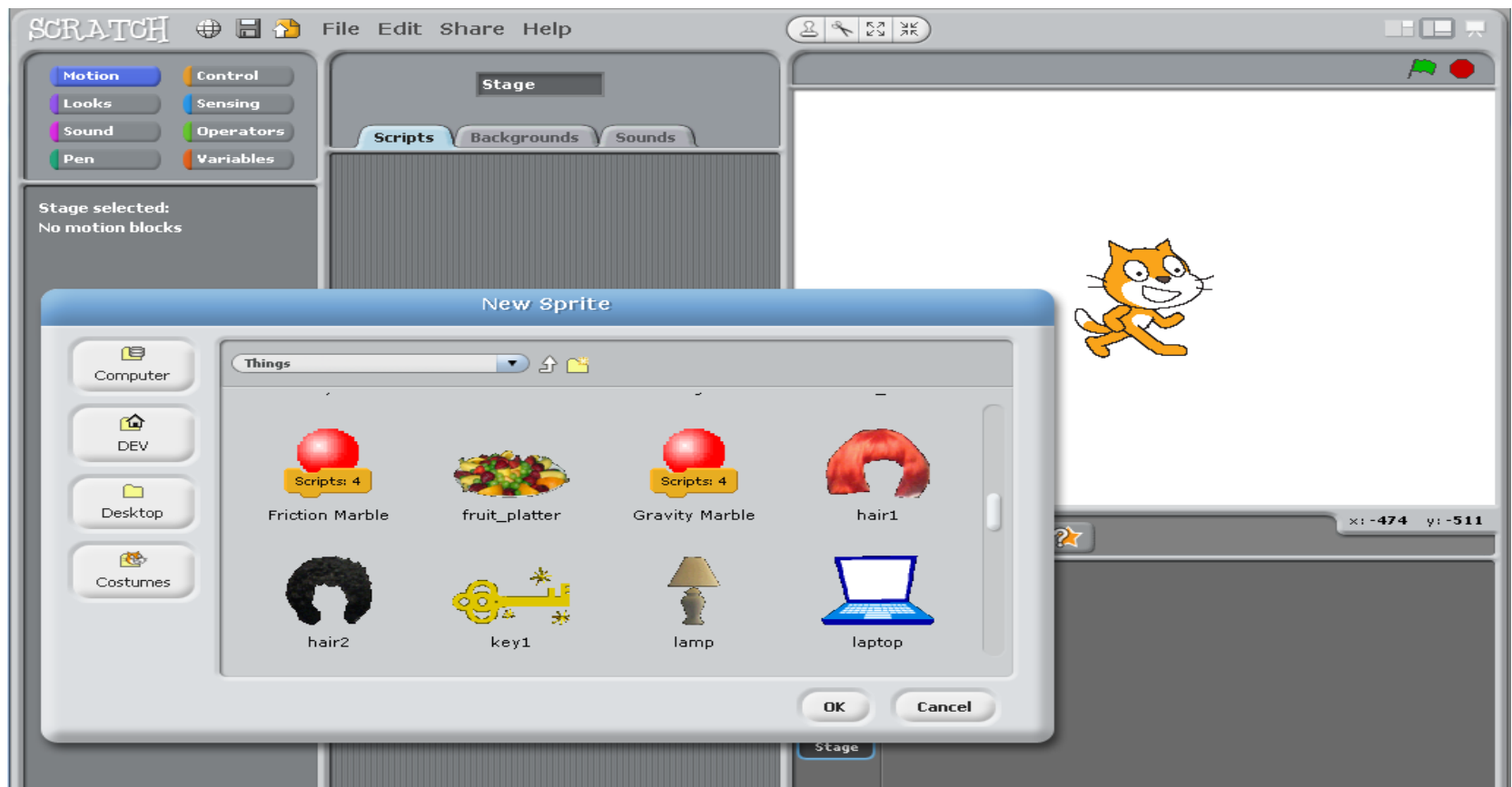
x: -605



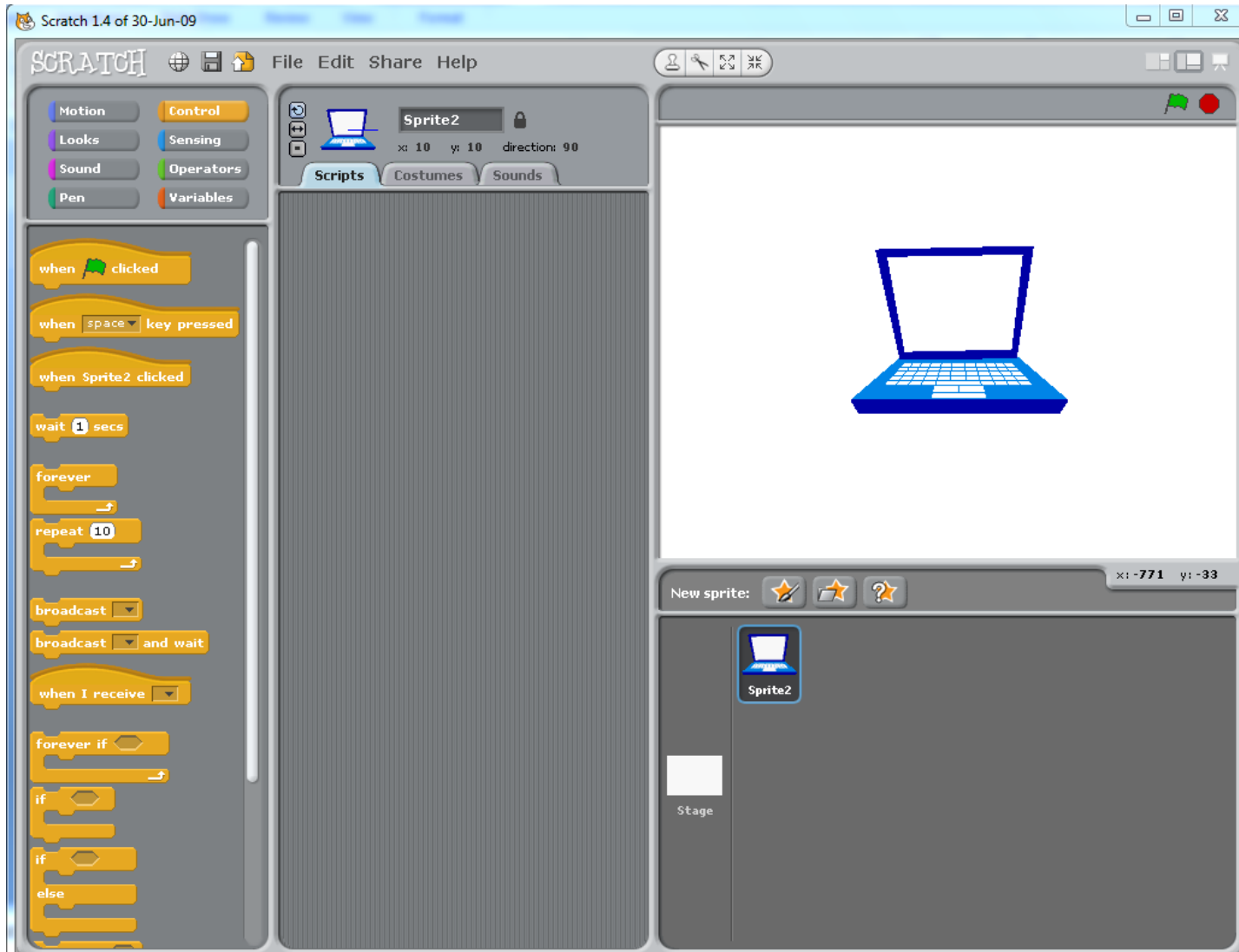
Stage

# Change Sprite

- Scratch includes several different sprites in quite a few different categories. However, the user can also import their own graphics or use the built-in sprite editor.



# Want to ask for your age and display answer



# Ask Age and Show it

The image shows the Scratch IDE interface. On the left, the 'Scripts' category is selected in the palette. The 'ask' block is highlighted with a red box. An orange callout bubble points to this block with the text: "What ever is entered goes into the holder called answer".

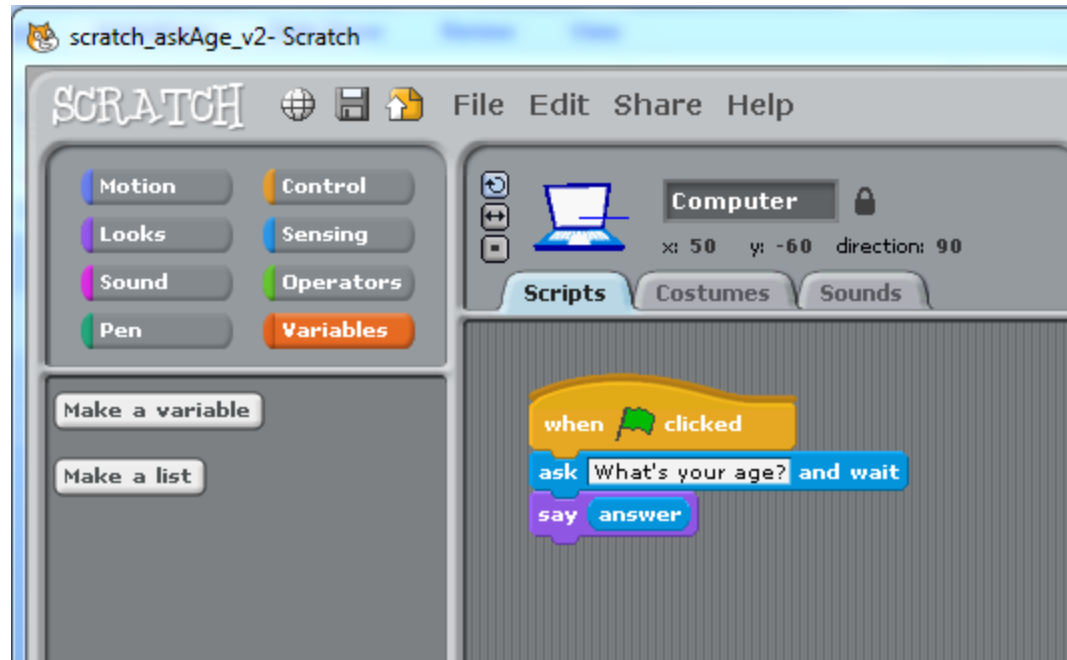
The main workspace shows a 'Computer' sprite with the following script:

- when green flag clicked
- ask "What's your age?" and wait
- say answer

The stage displays the 'Computer' sprite with a speech bubble saying "eee". Below the sprite, there is a text input field with the prompt "What's your name?".

The bottom of the IDE shows the 'New sprite' section with a 'Computer' sprite selected, labeled 'Comput...' and 'memory'. The stage area is labeled 'Stage'.

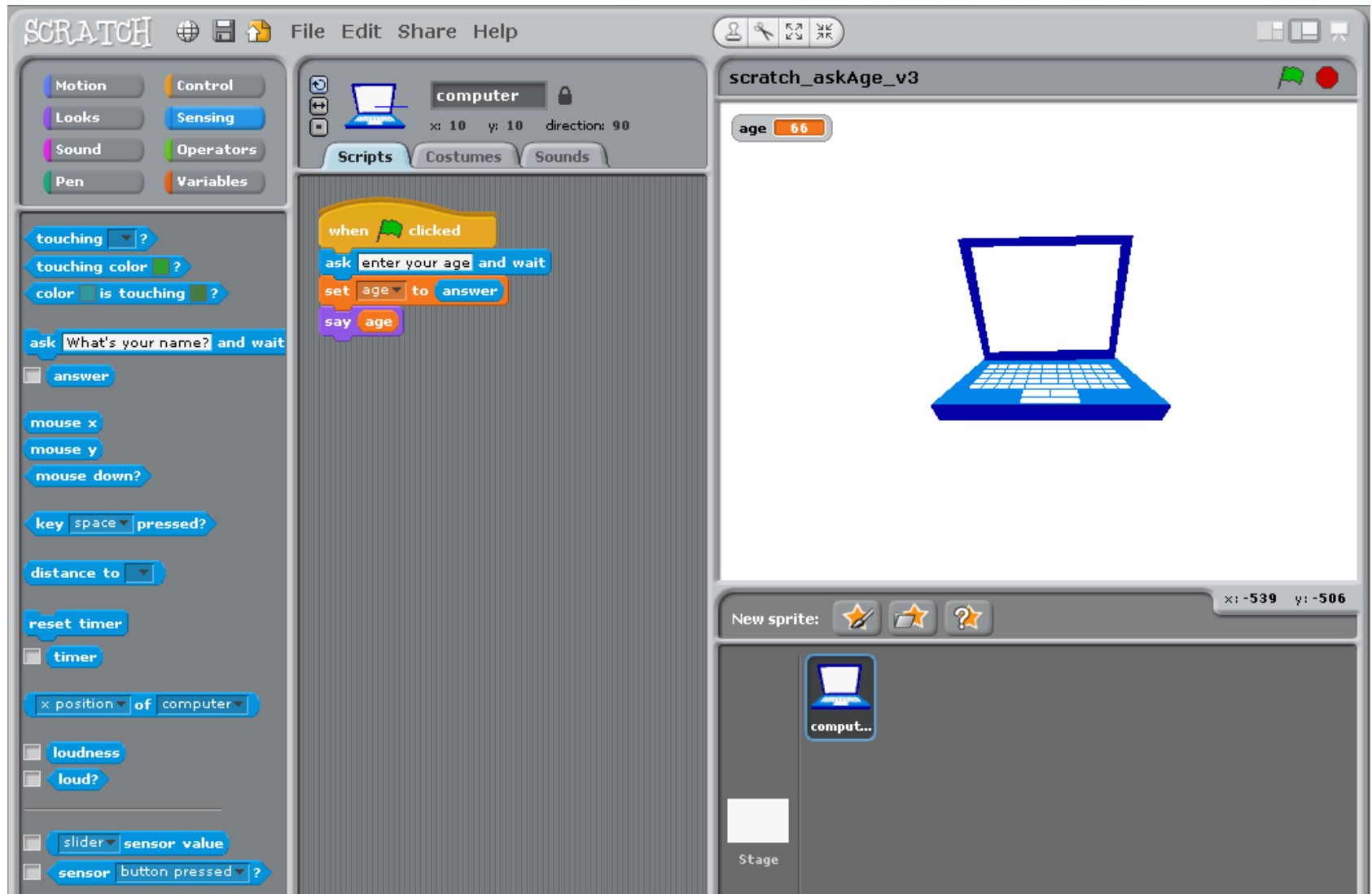
# Lets create a variable



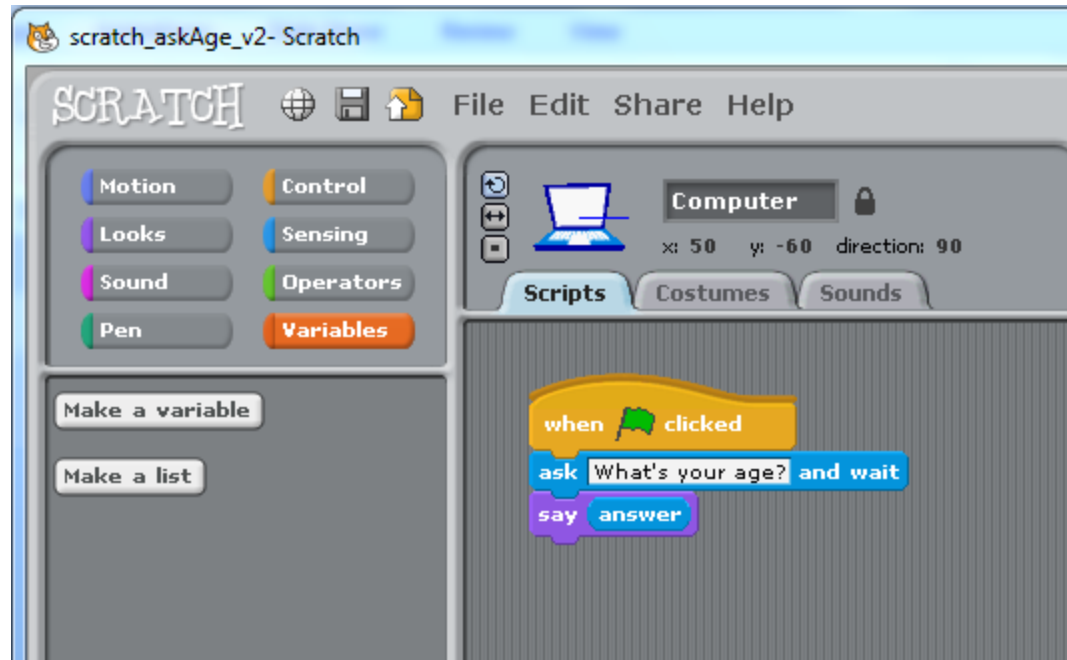
Writing Pad



# Use the variable



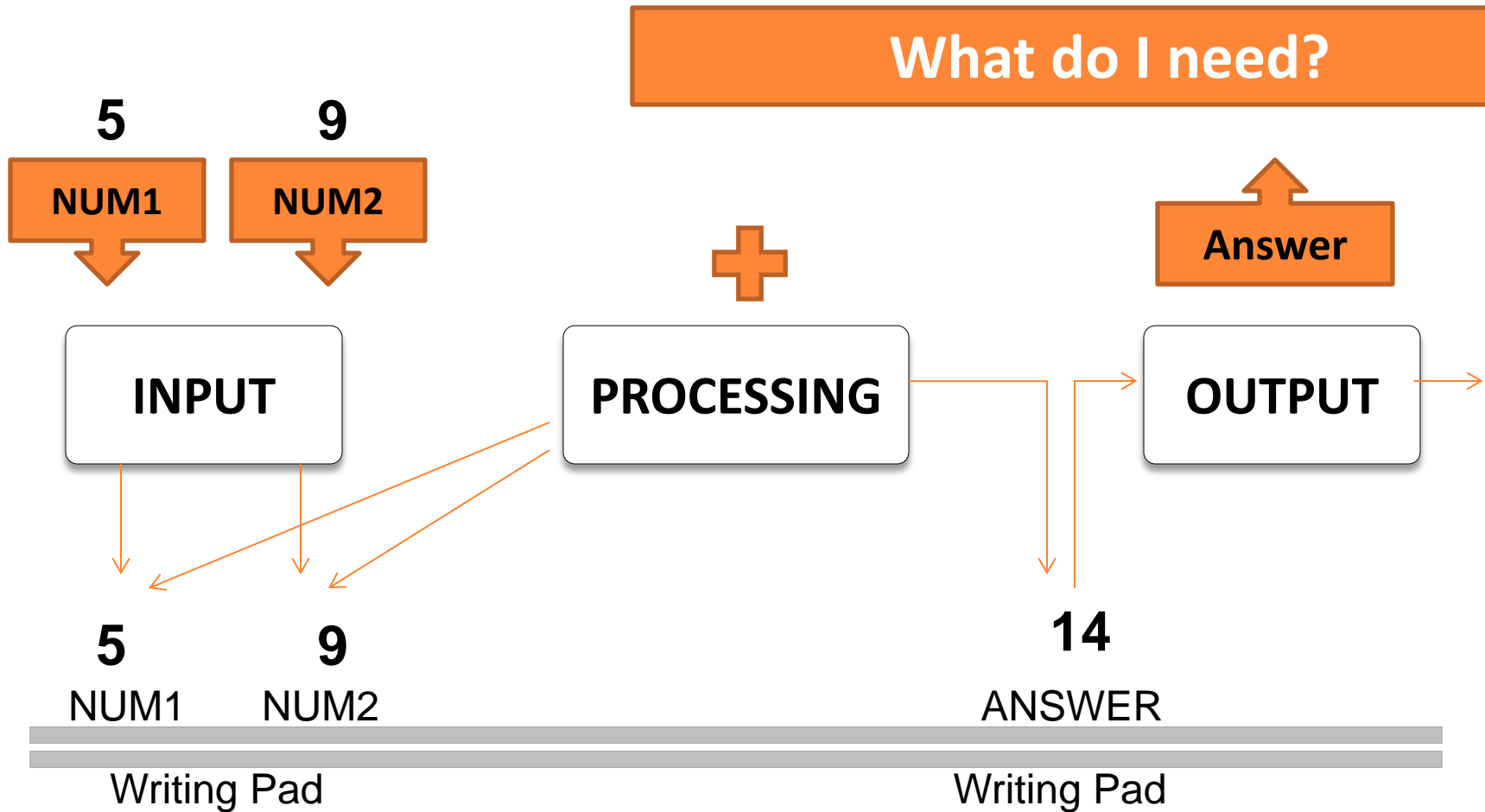
# Lets create a variable



Writing Pad



# Want to write this program





# Program Design Process

## 1. Problem Definition

- What is the objective
- What is the program to do

## 2. Design

- Draw a picture or the execution steps
- Write down in words the execution steps

## 3. Test Cases (how will you test it)

- Write what you will use for testing that it runs and creates the right answer
  - Test Case 1 :  $1 + 1 = 2$  Simple Case
  - Test Case 2:  $5 + 9 = 14$  Normal Case
  - Test Case 3:  $0 + 9 = 9$  Edge condition
  - Test Case 4 :  $5 + 0 = 5$  Edge condition

*If this was division we could have division by zero issues and very small answers*

- Test Case 5 :  $166666666666 + 78877777777777 = 788944444444443$  test the very big numbers

## 4. Write Code

- Step by step, on piece of functionality at a time, get it working, save a copy of that working version , then add the next bit of functionality.

## 5. Test Code with test cases

- Debug the code, change ONLY ONE thing at a time, KEEP SAVING VERSIONS

# Run the program

The image shows the Scratch programming environment. The top menu bar includes 'SCRATCH', a globe icon, a save icon, an upload icon, and the words 'File Edit Share Help'. On the right side of the top bar are icons for a person, a scissors, a zoom in, a zoom out, and a reset icon. Below the menu bar is a toolbar with buttons for 'Motion', 'Looks', 'Sound', 'Pen', 'Control', 'Sensing', 'Operators', and 'Variables'. The 'Variables' button is highlighted in orange. On the left side, under the 'Variables' section, there are buttons for 'Make a variable' and 'Delete a variable'. Below these are three variables: 'firstNumber', 'secondNumber', and 'theSum', each with a checked checkbox. Below the variables are several blocks: 'set firstNumber to 0', 'change firstNumber by 1', 'show variable firstNumber', and 'hide variable firstNumber'. The main workspace shows a 'computer' sprite with a lock icon. Below the sprite are tabs for 'Scripts', 'Costumes', and 'Sounds'. The 'Scripts' tab is selected, showing a script that starts with 'when clicked', followed by 'ask Enter the first number and wait', 'set firstNumber to answer', 'ask Enter the second number and wait', 'set secondNumber to answer', 'set theSum to secondNumber + firstNumber', and 'say theSum'. On the right side, there are three variable monitors: 'firstNumber' with a value of 5, 'secondNumber' with a value of 9, and 'theSum' with a value of 14. Below the monitors is a blue laptop sprite with a speech bubble saying '14'.

SCRATCH File Edit Share Help

Motion Control Looks Sensing Sound Operators Pen Variables

Make a variable Delete a variable

☒ firstNumber ☒ secondNumber ☒ theSum

set firstNumber to 0 change firstNumber by 1 show variable firstNumber hide variable firstNumber

computer

Scripts Costumes Sounds

when clicked

ask Enter the first number and wait

set firstNumber to answer

ask Enter the second number and wait

set secondNumber to answer

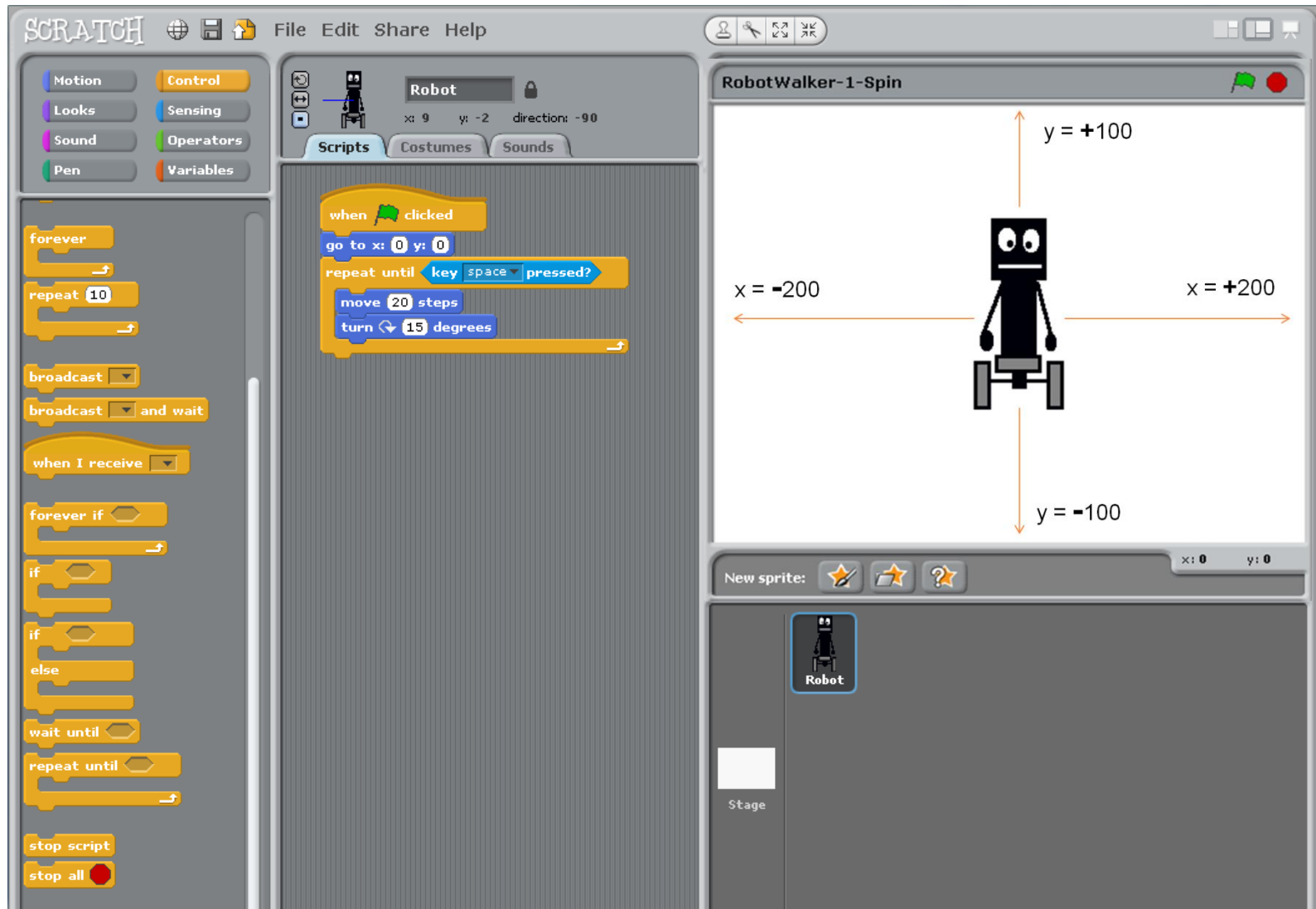
set theSum to secondNumber + firstNumber

say theSum

firstNumber 5 secondNumber 9 theSum 14

14

# Moving Scooter around the screen, forever



# Walking the block and not falling off the edge

The image shows the Scratch code editor for a project titled "RobotWalker-2-WalkTheBlock". The script is triggered by a "when clicked" event and sets the robot's initial position to (0, 0). It then defines boundaries for the robot's movement: xMax (200), xMin (-200), yMax (100), and yMin (-100). The script uses a series of "repeat until" loops to move the robot in a square path, ensuring it stays within the boundaries. A speech bubble labeled "REPEAT WHILE" points to the first "repeat until" loop.

```
when clicked
  set x to 0
  set y to 0
  set xMax to 200
  set xMin to -200
  set yMax to 100
  set yMin to -100
  repeat until (y position of Robot < yMin)
    change y by -10
  repeat until (x position of Robot > xMax)
    change x by 10
  repeat until (y position of Robot > yMax)
    change y by 10
  repeat until (x position of Robot < xMin)
    change x by -10
  repeat until (y position of Robot < yMin)
    change y by -10
  repeat until (x position of Robot = 0)
    change x by 10
  repeat until (y position of Robot = 0)
    change y by 10
```

The right side of the editor shows the stage with the robot sprite. The bottom right corner displays the coordinates x: -819 y: 66.

# Keep walking the block

The image shows the Scratch code editor for a project titled "RobotWalker-3-KeepWalkingTheBlock". The script is triggered by a "when clicked" event and sets initial coordinates and boundaries for a robot. It then enters a "repeat 10" loop that moves the robot in a square path by alternating between "repeat until" and "change by" blocks for x and y coordinates.

**Script:**

```
when clicked
  set x to 0
  set y to 0
  set xMax to 200
  set xMin to -200
  set yMax to 100
  set yMin to -100
  repeat 10
    repeat until (y position of Robot < yMin)
      change y by -10
    repeat until (x position of Robot > xMax)
      change x by 10
    repeat until (y position of Robot > yMax)
      change y by 10
    repeat until (x position of Robot < xMin)
      change x by -10
    repeat until (y position of Robot < yMin)
      change y by -10
    repeat until (x position of Robot = 0)
      change x by 10
    repeat until (y position of Robot = 0)
      change y by 10
```

**Stage:** The stage displays a robot sprite at coordinates x: -25, y: 16, direction: 60. A speech bubble points to the "repeat 10" block with the text "REPEAT 10 TIMES". The stage also shows a "New sprite:" button and a "Stage" label.

# If we use repeat with our sum two numbers

The image shows the Scratch code editor with a script for a computer sprite named 'computer'. A speech bubble points to the 'repeat' block with the text 'REPEAT 3 TIMES'.

**Code:**

```
when clicked  
repeat (3)  
  ask Enter the first number and wait  
  set firstNumber to answer  
  ask Enter the second number and wait  
  set secondNumber to answer  
  set theSum to secondNumber + firstNumber  
  say theSum
```

**Stage:**

SumTwoNumbers\_v2

firstNumber 3  
secondNumber 3  
theSum 6

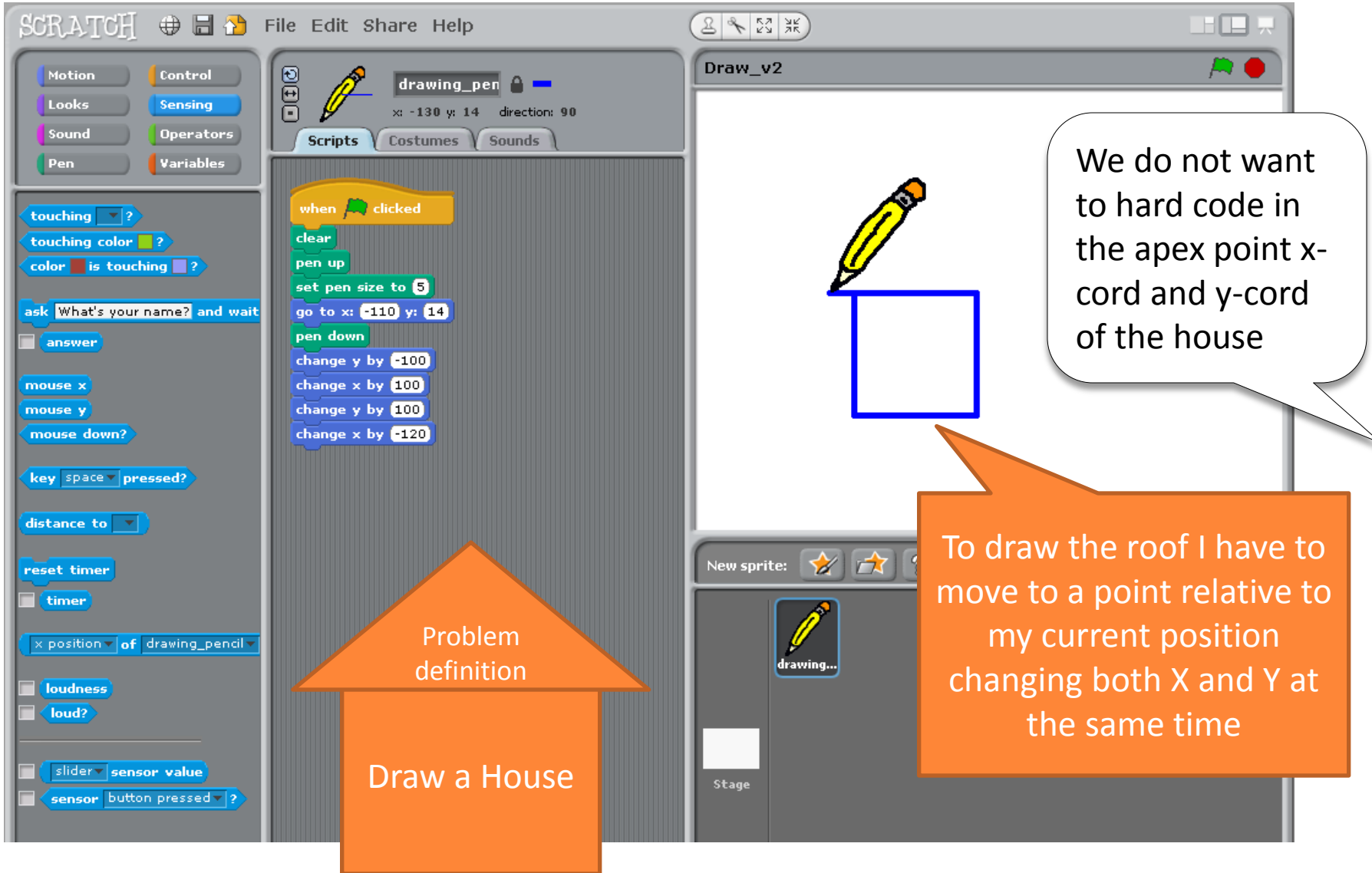
6

**Bottom Panel:**

New sprite: x: -819 y: -505

comput...

# Want to write code that will work no matter where the pen is first placed on the screen



The image shows the Scratch IDE interface. The left sidebar contains the 'Pen' category selected. The main script area shows a 'when clicked' event followed by a sequence of drawing commands: 'clear', 'pen up', 'set pen size to 5', 'go to x: -110 y: 14', 'pen down', 'change y by -100', 'change x by 100', 'change y by 100', and 'change x by -120'. The right side shows a stage with a yellow pencil sprite and a blue square. Two callout boxes provide context: a white one at the top right explains the goal of not hard-coding the starting point, and an orange one at the bottom right explains the need for relative movement for the roof.

SCRATCH File Edit Share Help

Motion Control Looks Sensing Sound Operators Pen Variables

Scripts Costumes Sounds

when clicked

clear

pen up

set pen size to 5

go to x: -110 y: 14

pen down

change y by -100

change x by 100

change y by 100

change x by -120

Draw\_v2

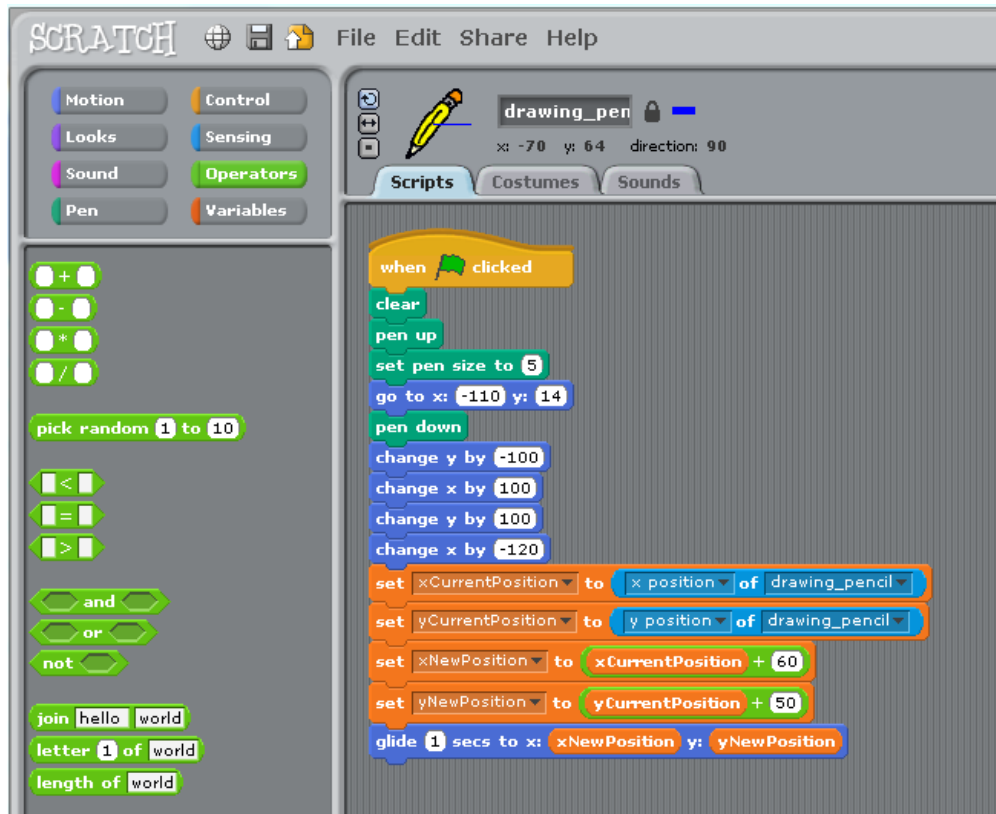
We do not want to hard code in the apex point x-cord and y-cord of the house

Problem definition

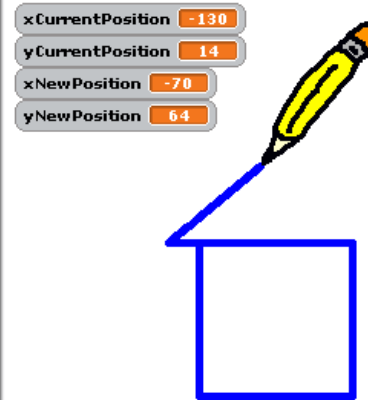
Draw a House

To draw the roof I have to move to a point relative to my current position changing both X and Y at the same time

# Need to use variables – for relative positions



We need to create variables to store our current position x-cord and y-cord



In this case we want to change x's position by 60 and y's by 50

An alternative approach is to change the direction the pen is facing by 60 degrees to the right and then let it travel a given length, then change another 60 degrees to the right and travel the same distance, turn again 60 degrees and travel 20

So we create two new variables to hold the x-cords and y-cords we want to go to.  
Then we use the action that moves the pen to a given set of x-cords and y-cords



# Algorithms

- An algorithm is a well-developed, organized approach to solving a complex problem
- It is repeatable
  - no hard coding i.e. Of co-ordinates, or values
  - can work based on current position or user inputs
- The solution for walking the block could be turned into drawing a square around the edges of the stage
- Maths equations and ideas start becoming useful for doing things – co-ordinates, the sum of all angles on a triangle : ESPECIALLY FOR GAMES – 2D and 3D

**Using SCRATCH to build a maze game**

# Step 1: Build our player

Scratch 1.4 of 30-Jun-09

SCRATCH File Edit Share Help

Motion Looks Sound Pen Control Sensing Operators Variables

player x: 14 y: -17 direction: 90

Scripts Costumes Sounds

when right arrow key pressed  
point in direction 90  
move 10 steps

when left arrow key pressed  
point in direction -90  
move 10 steps

when up arrow key pressed  
point in direction 0  
move 10 steps

when down arrow key pressed  
point in direction 180  
move 10 steps

move 10 steps  
turn 15 degrees  
turn 15 degrees  
point in direction 90  
point towards  
go to x: 4 y: -57  
go to  
glide 1 secs to x: 4 y: -57  
change x by 10  
set x to 0  
change y by 10  
set y to 0  
if on edge, bounce  
x position  
y position  
direction

x: -386 y: -132

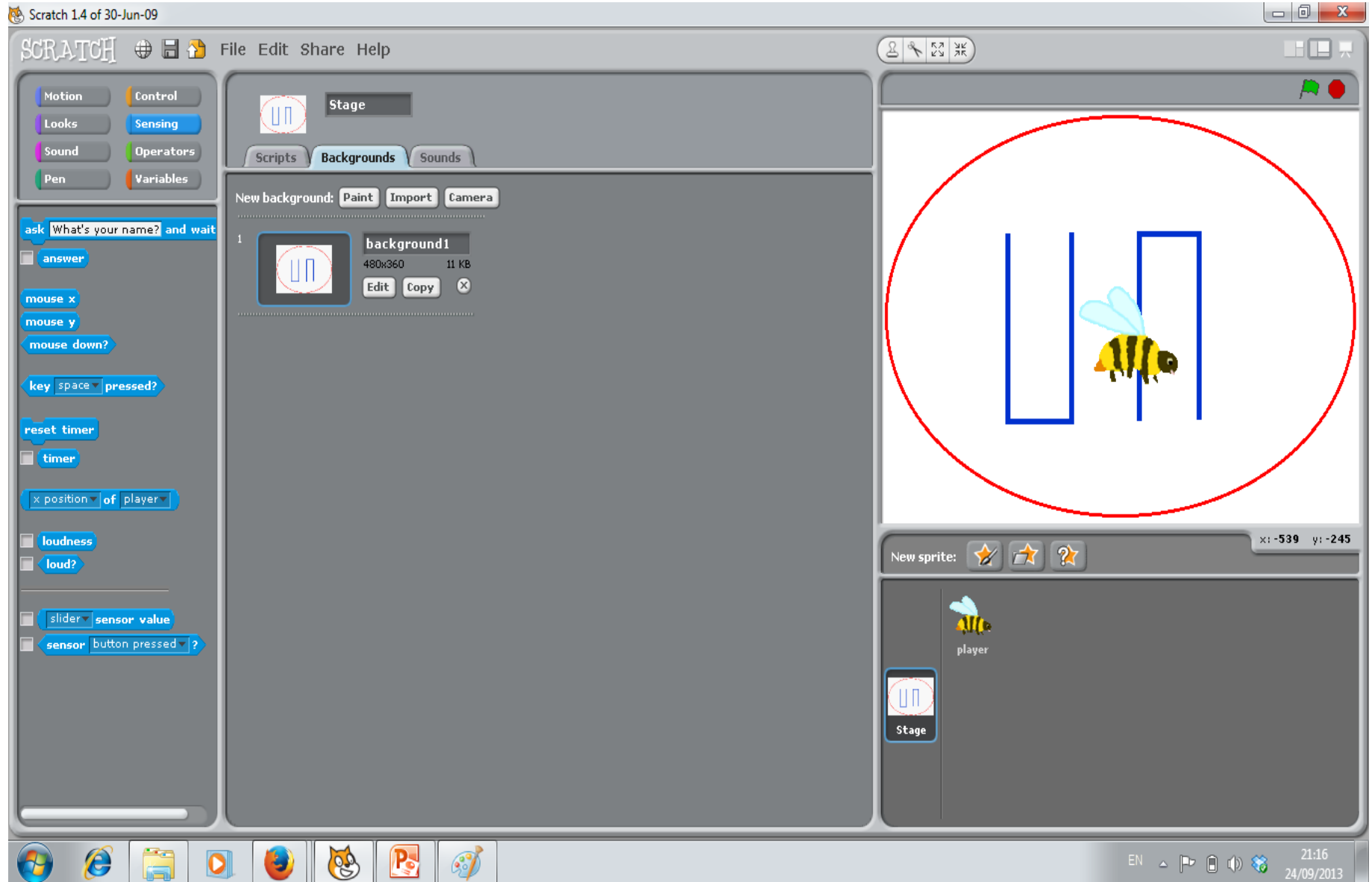
New sprite:

player

Stage

21:07 24/09/2013

# Step 2: Draw the stage



# Step 3: Create some prizes

Scratch 1.4 of 30-Jun-09

SCRATCH File Edit Share Help

Motion Control Looks Sensing Sound Operators Pen Variables

Prize 2 x: 50 y: 51 direction: 90

Scripts Costumes Sounds

move 10 steps  
turn 15 degrees  
turn 15 degrees  
point in direction 90  
point towards  
go to x: 50 y: 51  
go to  
glide 1 secs to x: 50 y: 51  
change x by 10  
set x to 0  
change y by 10  
set y to 0  
if on edge, bounce  
x position  
y position  
direction

New sprite: ☆ ☆ ?

player Prize 1 Prize 2

Stage

21:23 24/09/2013

# Step 4: Add sound to the player

Scratch 1.4 of 30-Jun-09

File Edit Share Help

Motion Control Looks Sensing Sound Operators Pen Variables

Scripts Costumes Sounds

switch to costume bee1  
next costume  
costume #  
say Hello! for 2 secs  
say Hello!  
think Hmm... for 2 secs  
think Hmm...  
change color effect by 25  
set color effect to 0  
clear graphic effects  
change size by 10  
set size to 55 %  
size  
show  
hide  
go to front  
go back 1 layers

player  
x: -171 y: 32 direction: 90

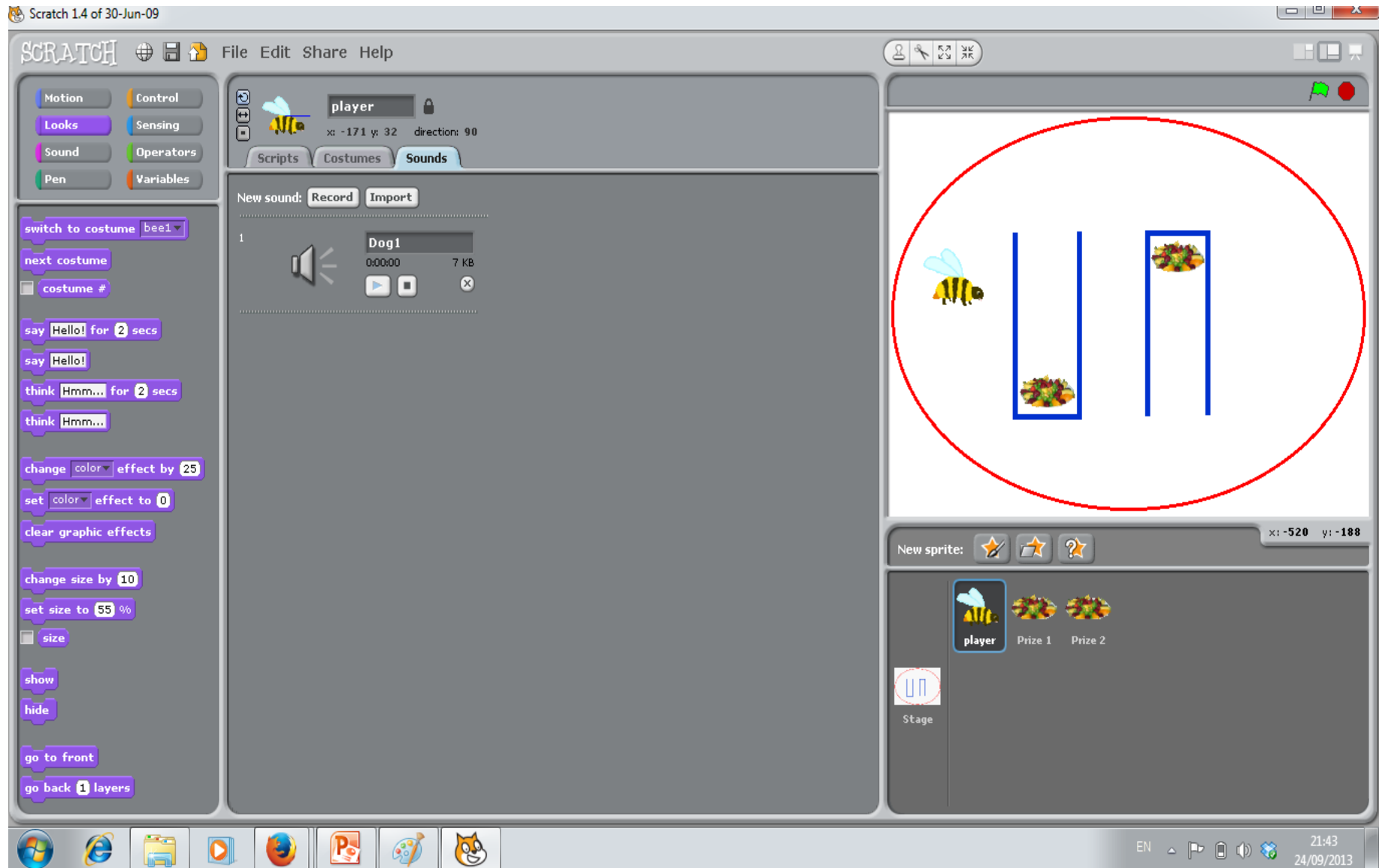
New sound: Record Import

1  
Dog1  
0:00:00 7 KB

Stage

player Prize 1 Prize 2

21:43 24/09/2013



# Step 5: Program the player

Scratch 1.4 of 30-Jun-09

SCRATCH

File Edit Share Help

Motion Looks Sound Pen Control Sensing Operators Variables

Scripts Costumes Sounds

player x: -171 y: 32 direction: 90

when right arrow key pressed

point in direction 90

move 10 steps

when left arrow key pressed

point in direction -90

move 10 steps

when up arrow key pressed

point in direction 0

move 10 steps

when down arrow key pressed

point in direction 180

move 10 steps

when clicked

forever

if touching Prize 1?

play sound Dog1 until done

if touching Prize 2?

play sound Dog1 until done

switch to costume bee1

next costume

costume #

say Hello! for 2 secs

say Hello!

think Hmm... for 2 secs

think Hmm...

change color effect by 25

set color effect to 0

clear graphic effects

change size by 10

set size to 55 %

size

show

hide

go to front

go back 1 layers

Stage

player Prize 1 Prize 2

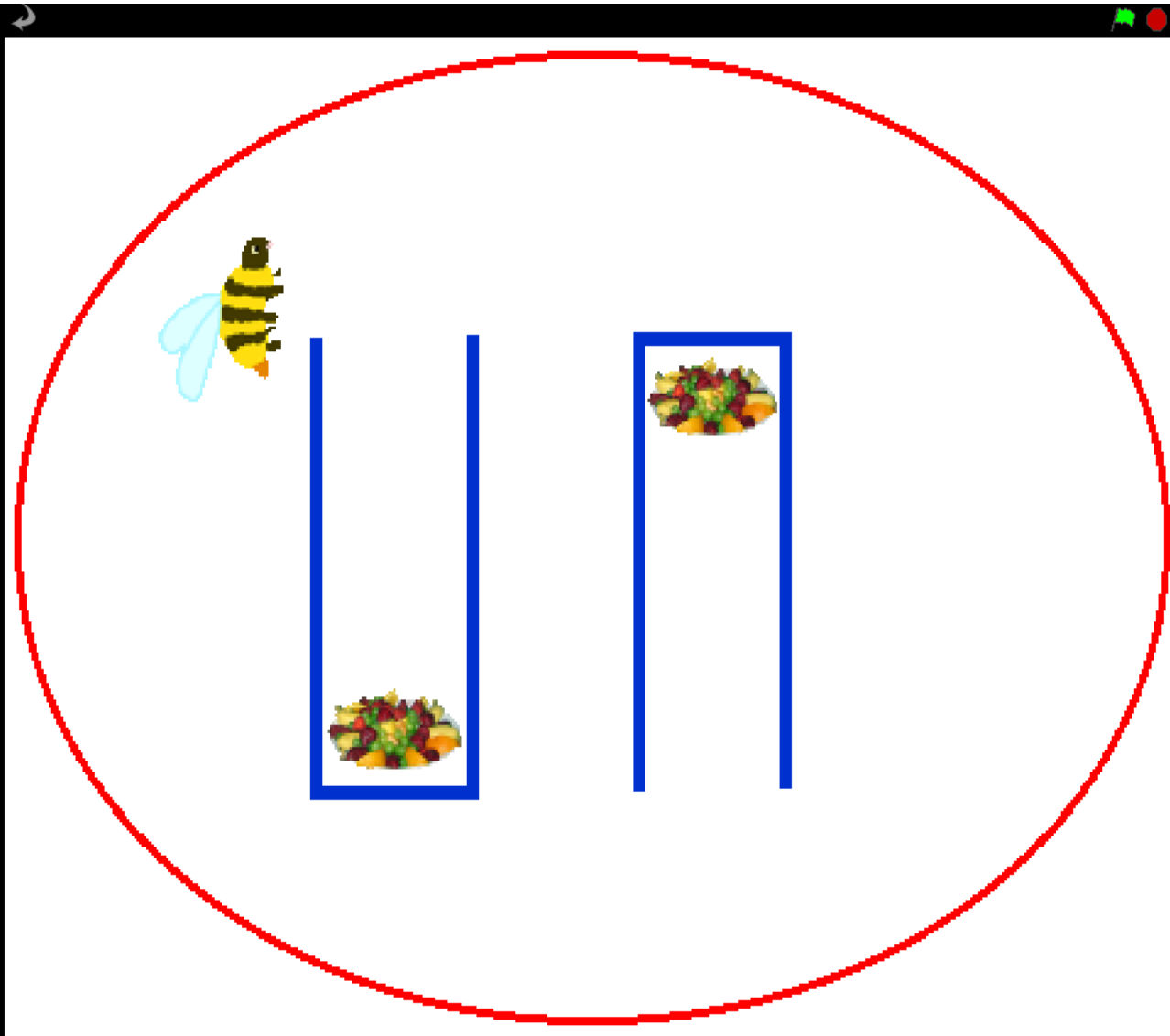
21:39 24/09/2013

## Scratch 1.4 of 30-Jun-09





## Step 7: Game completed!



# References

- 2009, Barry, Paul and Griffiths, David; Head First Programming, O'Reilly Media Inc.
- 2009, Pine,Chris ; Learn to Program, 2<sup>nd</sup> Edition, The Pragmatic Programmers