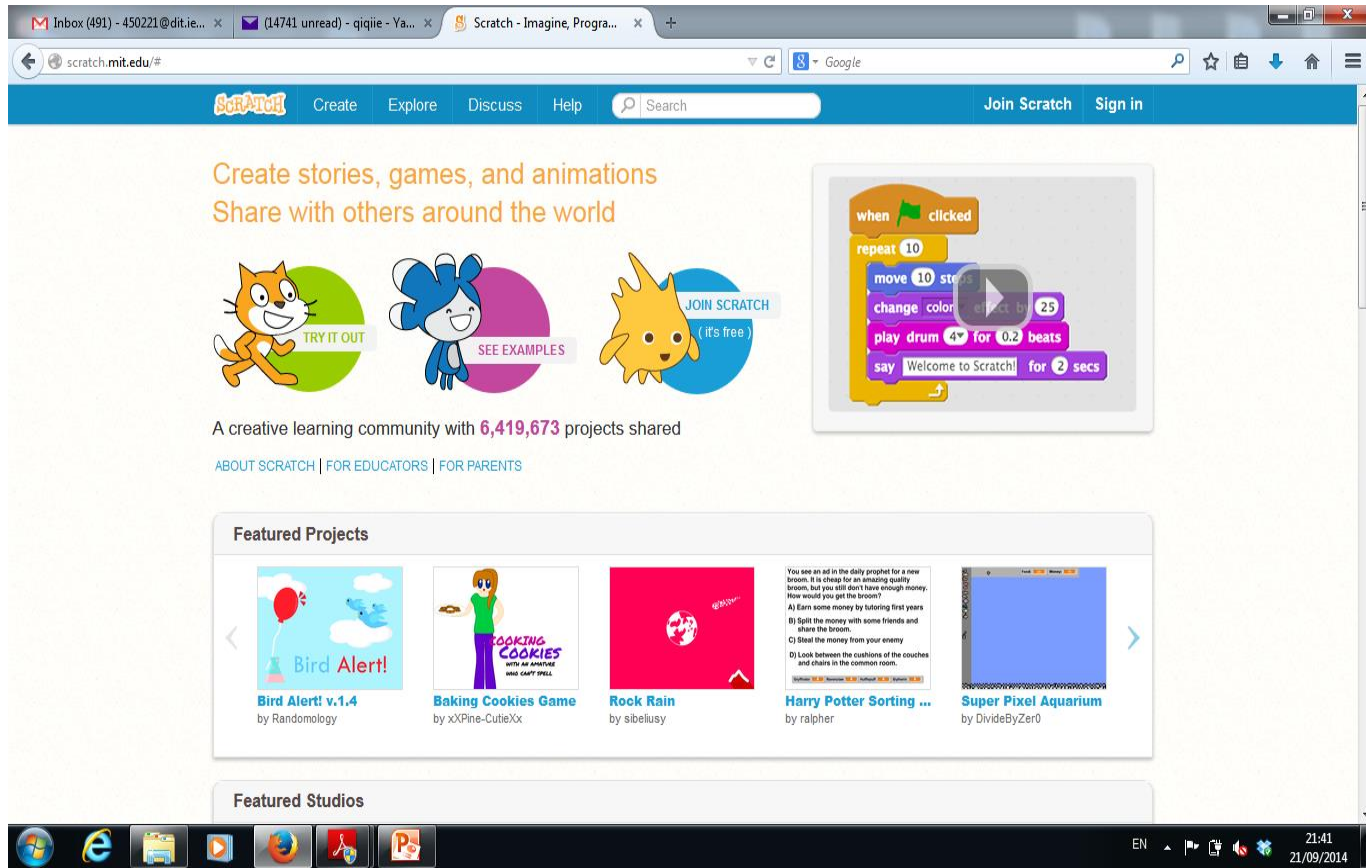


## 4. Visual Programming 2

**What did we do last time?**

# SCRATCH – visual programming

## <http://scratch.mit.edu/#>



The screenshot shows the Scratch website homepage in a web browser. The browser's address bar displays `scratch.mit.edu/#`. The website's navigation bar includes links for 'Create', 'Explore', 'Discuss', 'Help', 'Join Scratch', and 'Sign in'. The main content area features the Scratch logo and the text 'Create stories, games, and animations' and 'Share with others around the world'. Below this, there are three circular buttons: 'TRY IT OUT' with the Scratch cat, 'SEE EXAMPLES' with a blue cat, and 'JOIN SCRATCH (it's free)' with a yellow cat. A code block preview is shown on the right, featuring a 'when clicked' event, a 'repeat 10' loop, and actions like 'move 10 steps', 'change color effect by 25', 'play drum 4 for 0.2 beats', and 'say Welcome to Scratch! for 2 secs'. Below the buttons, it states 'A creative learning community with 6,419,673 projects shared' and provides links for 'ABOUT SCRATCH', 'FOR EDUCATORS', and 'FOR PARENTS'. The 'Featured Projects' section displays five project thumbnails: 'Bird Alert! v.1.4' by Randomology, 'Baking Cookies Game' by xXPine-CutieXx, 'Rock Rain' by sibelusy, 'Harry Potter Sorting ...' by ralpher, and 'Super Pixel Aquarium' by DivideByZero. The 'Featured Studios' section is partially visible at the bottom. The Windows taskbar at the bottom shows the time as 21:41 on 21/09/2014.

Create stories, games, and animations  
Share with others around the world

TRY IT OUT SEE EXAMPLES JOIN SCRATCH (it's free)

A creative learning community with 6,419,673 projects shared

ABOUT SCRATCH | FOR EDUCATORS | FOR PARENTS

Featured Projects

Bird Alert! v.1.4 by Randomology

Baking Cookies Game by xXPine-CutieXx

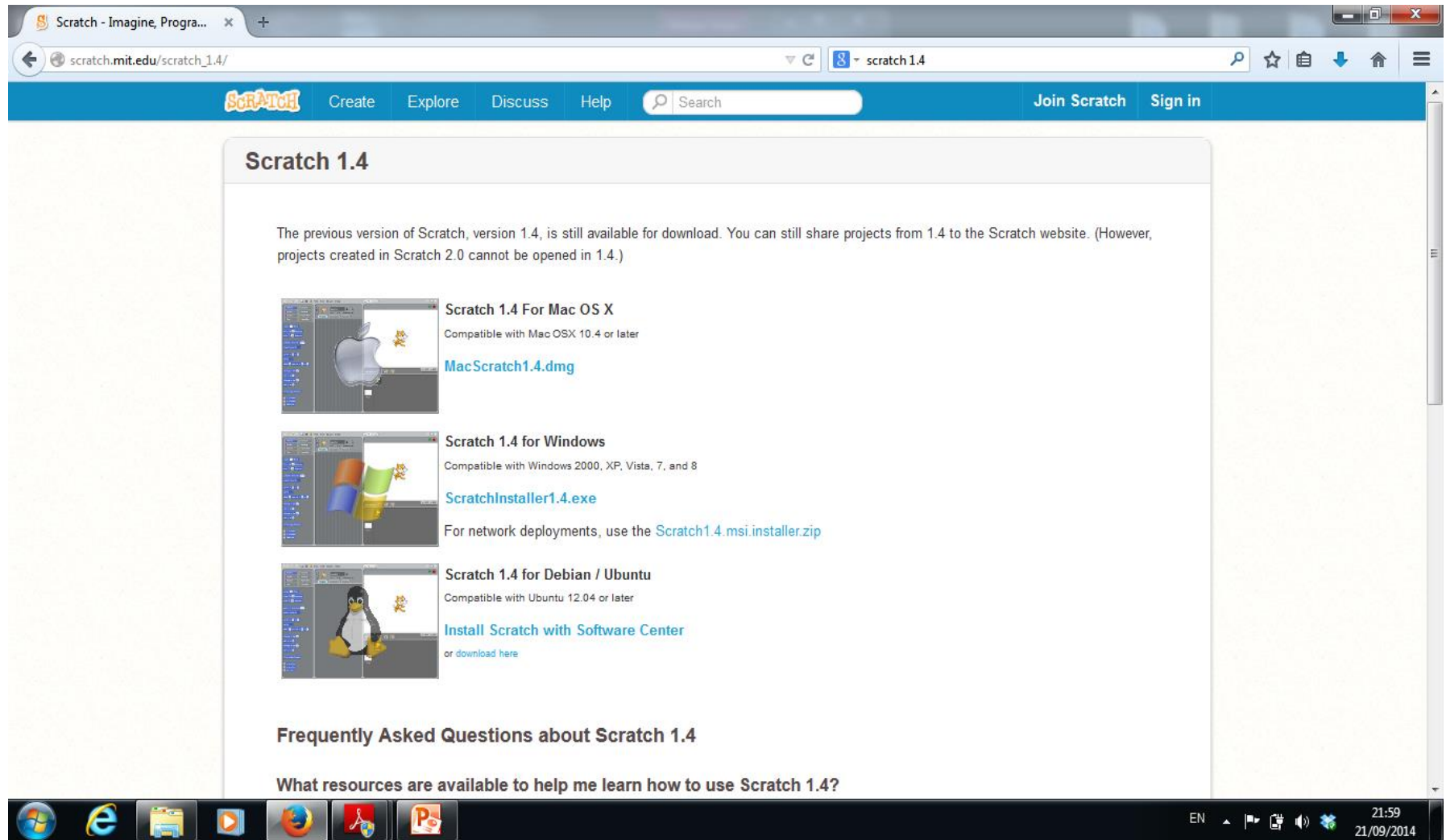
Rock Rain by sibelusy

Harry Potter Sorting ... by ralpher

Super Pixel Aquarium by DivideByZero

Featured Studios

# SCRATCH – visual programming



The screenshot shows a web browser window with the address bar displaying `scratch.mit.edu/scratch_1.4/`. The page title is "Scratch - Imagine, Program...". The navigation bar includes links for "Create", "Explore", "Discuss", "Help", a search bar, "Join Scratch", and "Sign in". The main content area is titled "Scratch 1.4" and contains the following text: "The previous version of Scratch, version 1.4, is still available for download. You can still share projects from 1.4 to the Scratch website. (However, projects created in Scratch 2.0 cannot be opened in 1.4.)". Below this text are three download options, each with a small thumbnail image of the Scratch interface and a Mac OS X logo, Windows logo, or Linux penguin logo respectively.

**Scratch 1.4**

The previous version of Scratch, version 1.4, is still available for download. You can still share projects from 1.4 to the Scratch website. (However, projects created in Scratch 2.0 cannot be opened in 1.4.)

**Scratch 1.4 For Mac OS X**  
Compatible with Mac OS X 10.4 or later  
[MacScratch1.4.dmg](#)

**Scratch 1.4 for Windows**  
Compatible with Windows 2000, XP, Vista, 7, and 8  
[ScratchInstaller1.4.exe](#)  
For network deployments, use the [Scratch1.4.msi.installer.zip](#)

**Scratch 1.4 for Debian / Ubuntu**  
Compatible with Ubuntu 12.04 or later  
[Install Scratch with Software Center](#)  
or [download here](#)

**Frequently Asked Questions about Scratch 1.4**

What resources are available to help me learn how to use Scratch 1.4?

[http://scratch.mit.edu/scratch\\_1.4/](http://scratch.mit.edu/scratch_1.4/)

# 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.

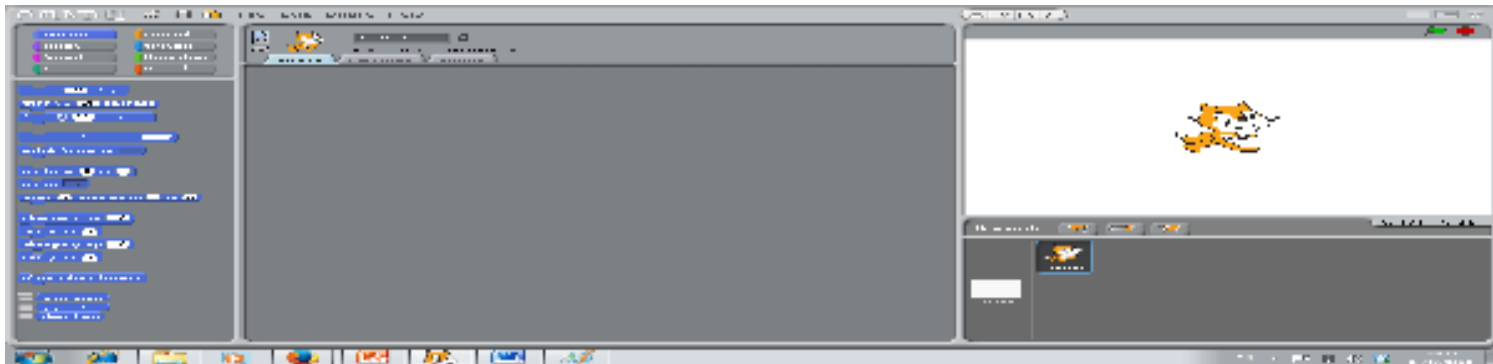


# Eight categories of programming institutions

- **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

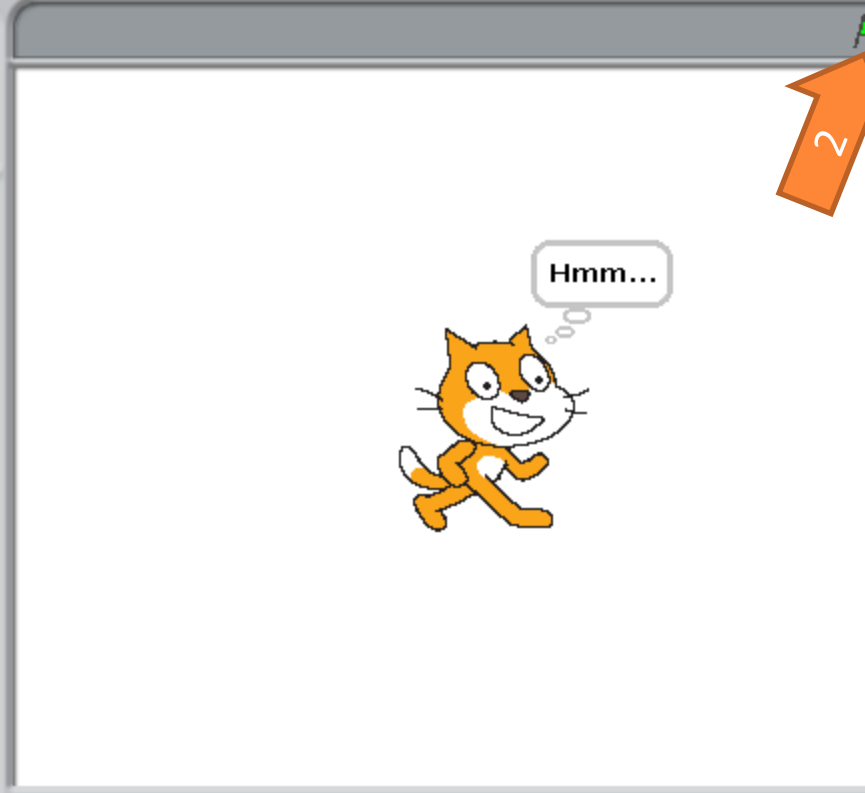
x: 0 y: 0 direction: 90

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 Star ?

x: -605

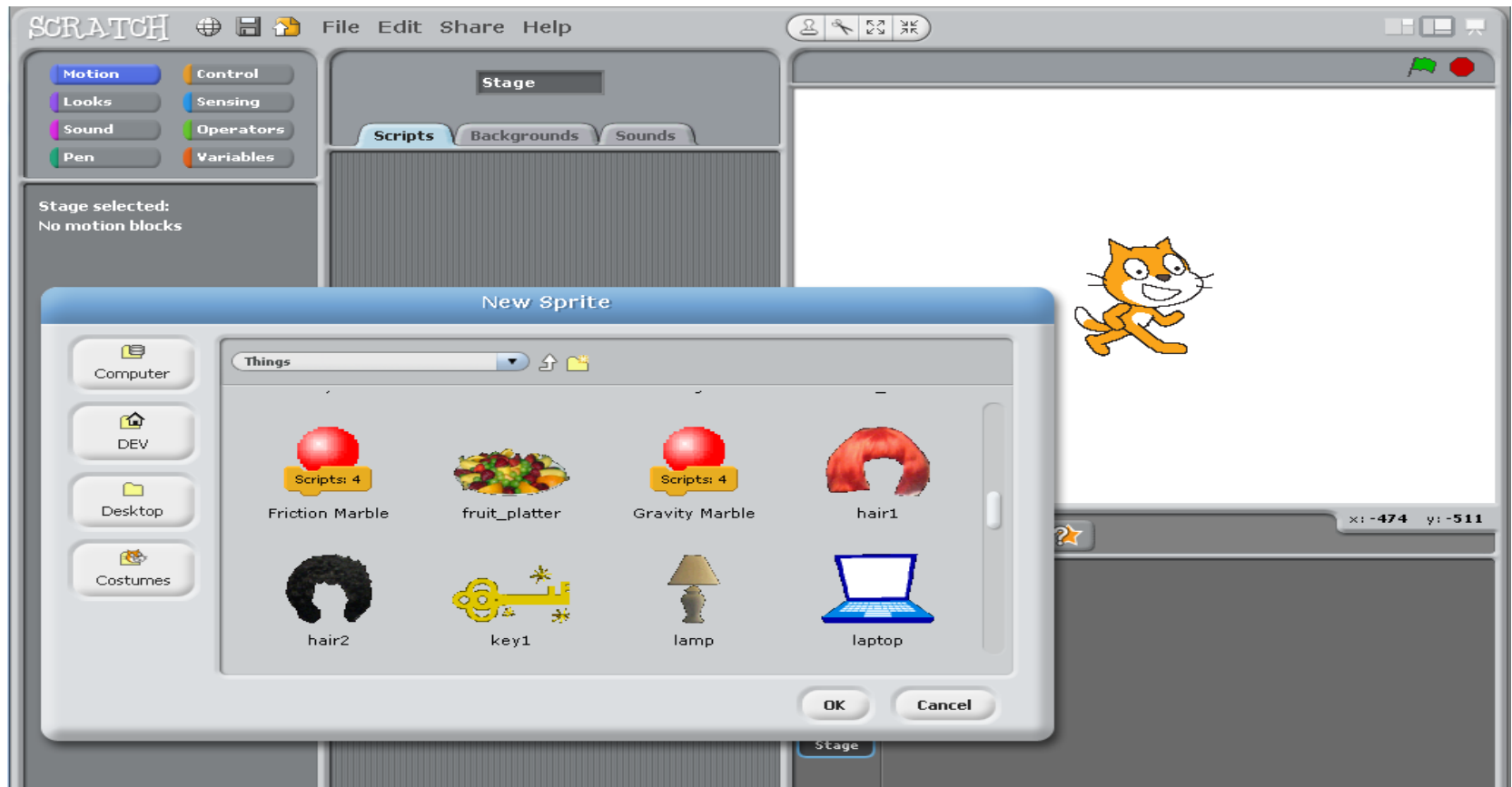


Stage

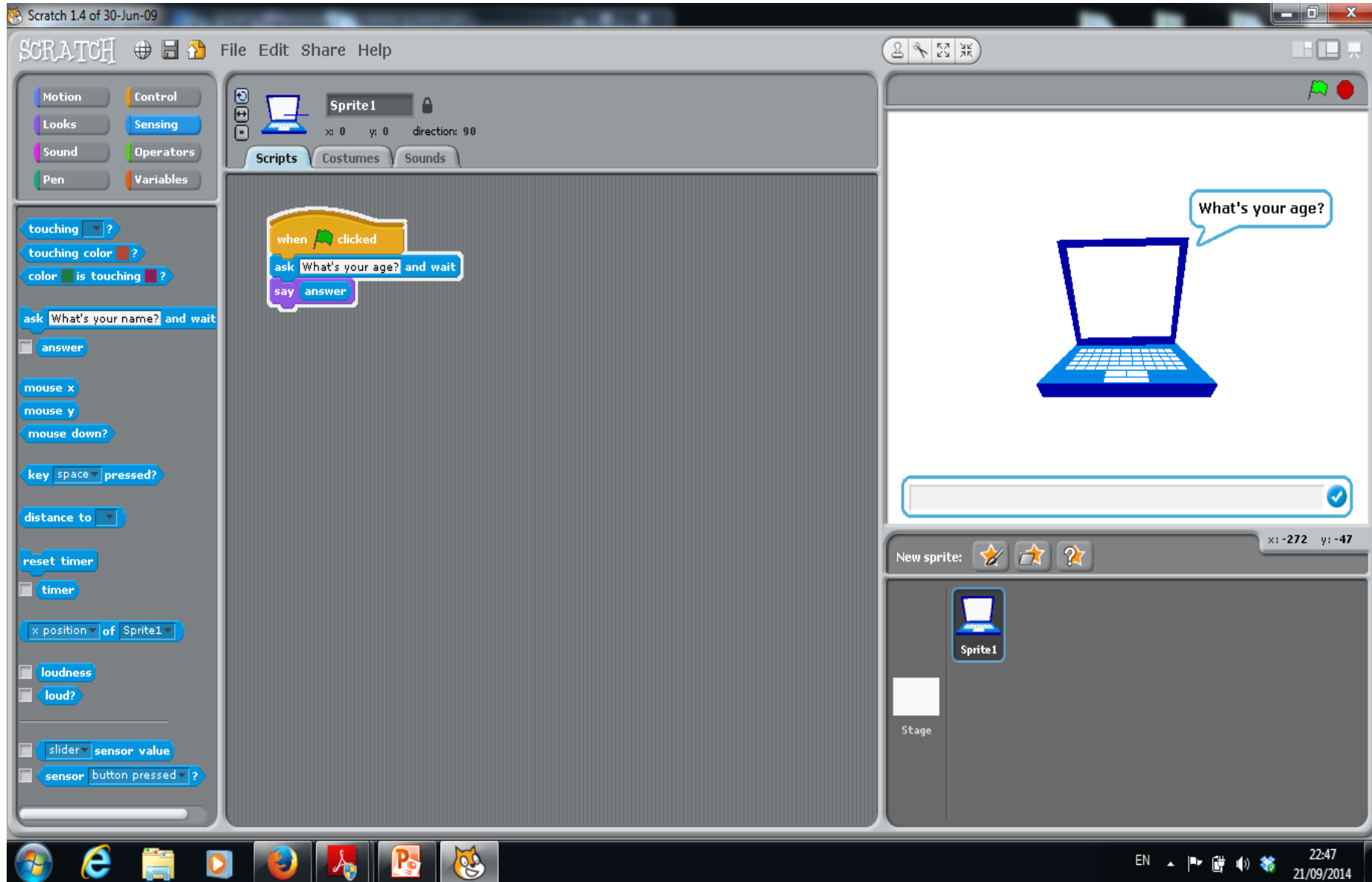


# Change Sprite

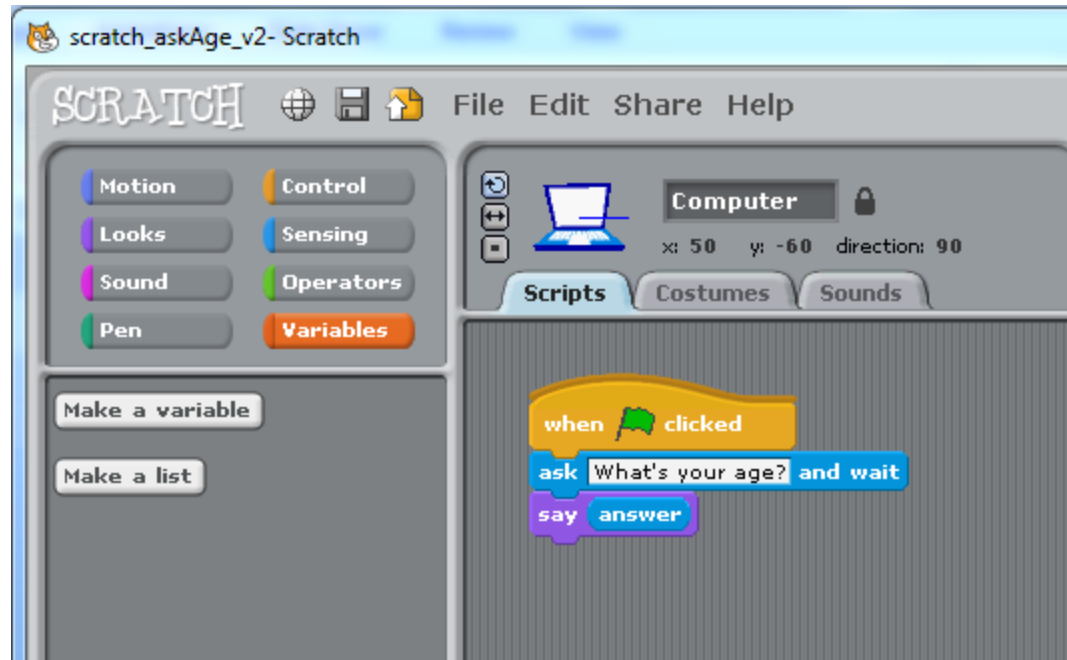
- Scratch includes several different sprites in quite a few different categories. However, the user can also import their own graphics or use the built-in sprite editor.



# Ask Age and Show it



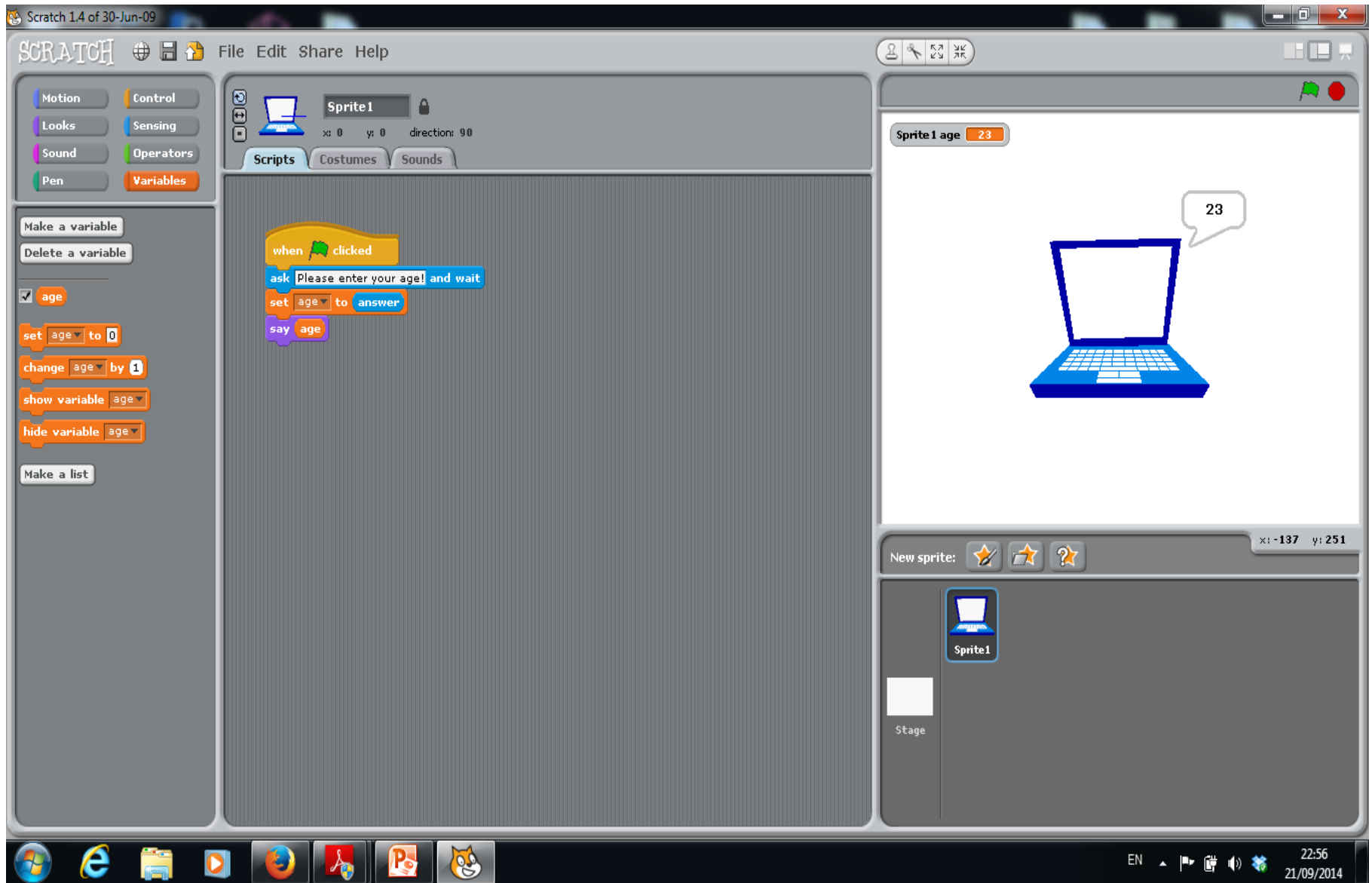
# Lets create a variable



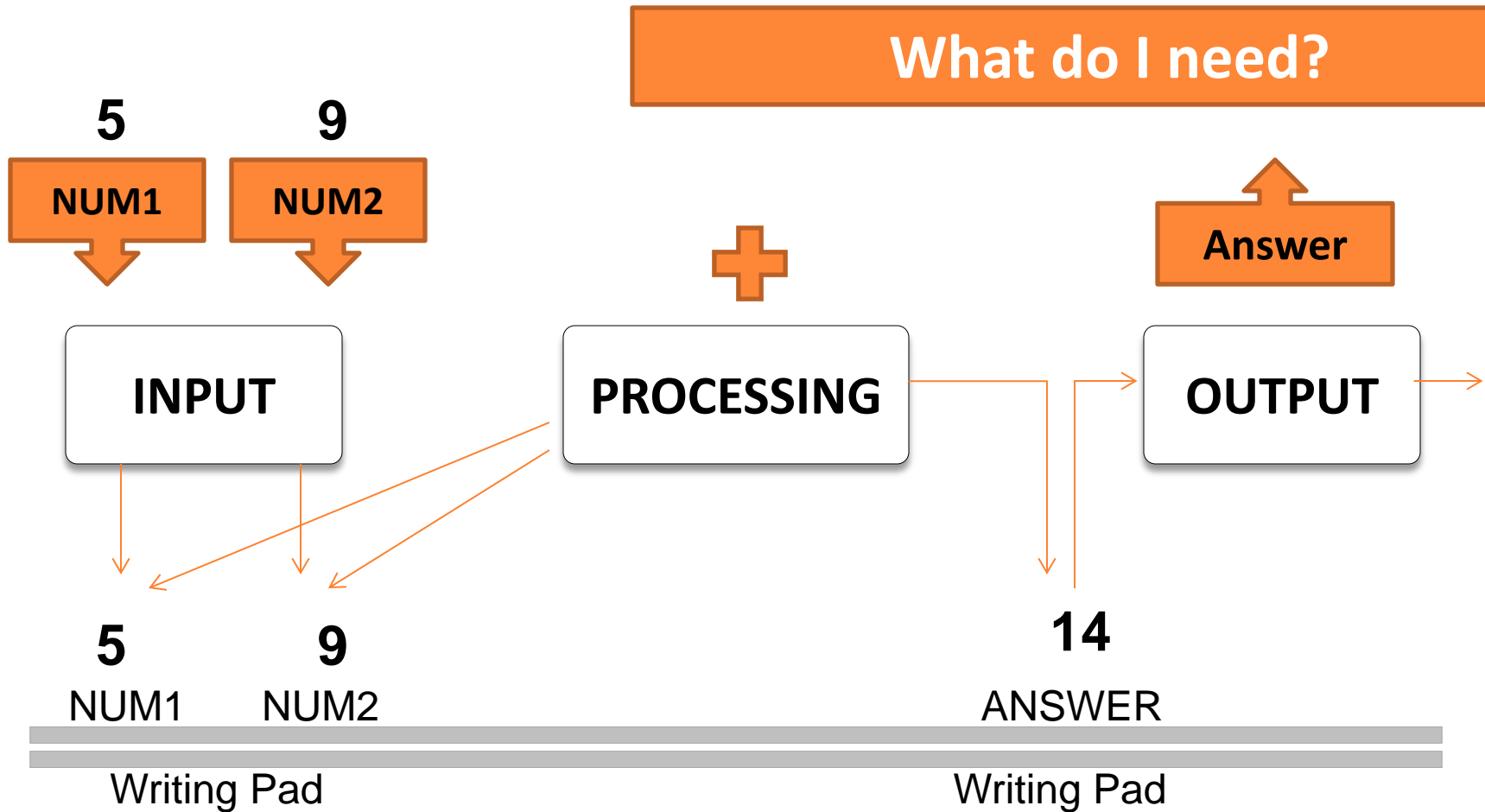
Writing Pad



# Use the variable



# Want to write this program



Need to design the program

# How?

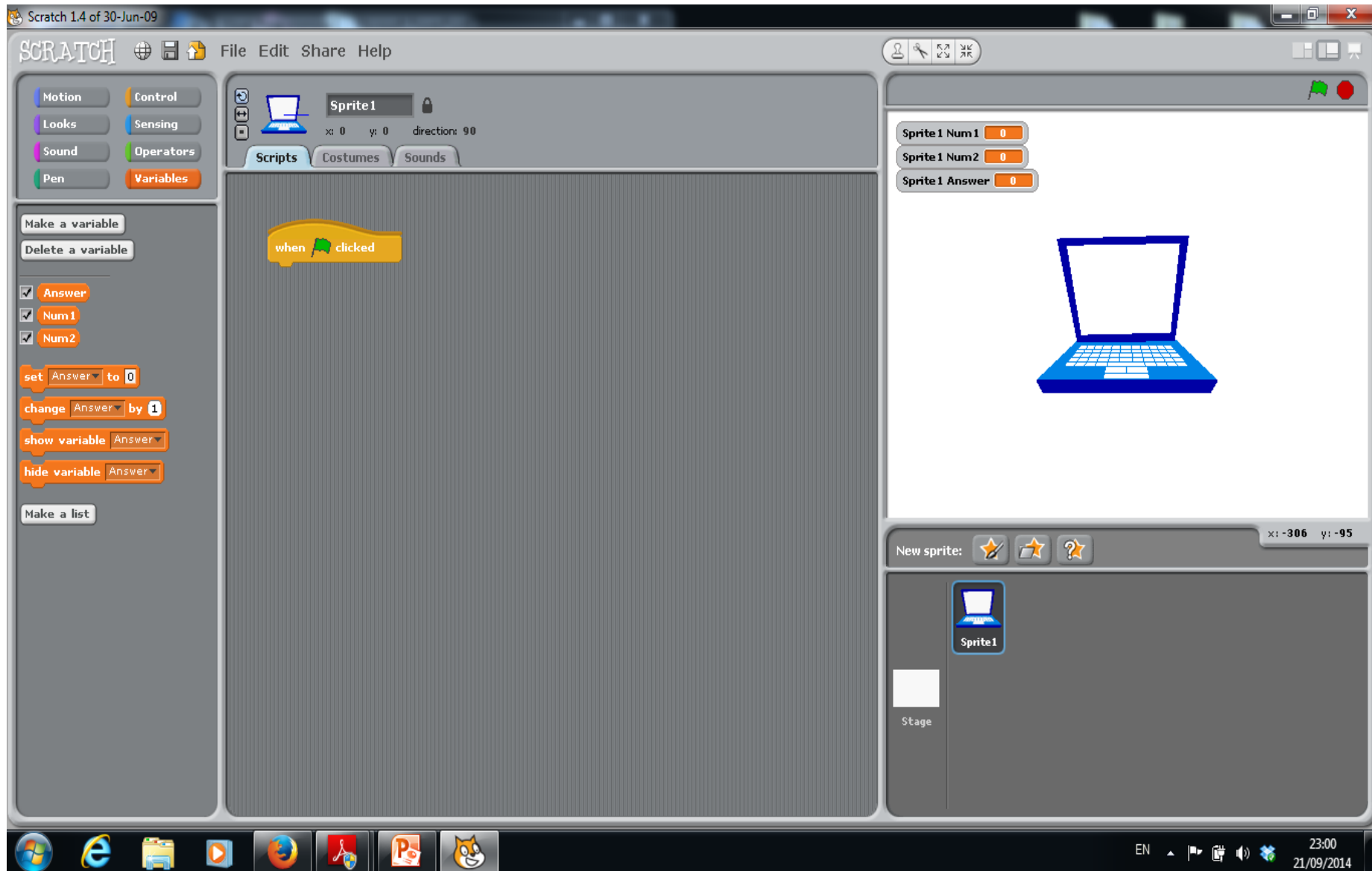
# Program Design Process

1. Problem Definition
  - What is the objective
  - What is the program to do
2. Design
3. Test Cases (how will you test it)
4. Write Code
5. Test Code with test cases

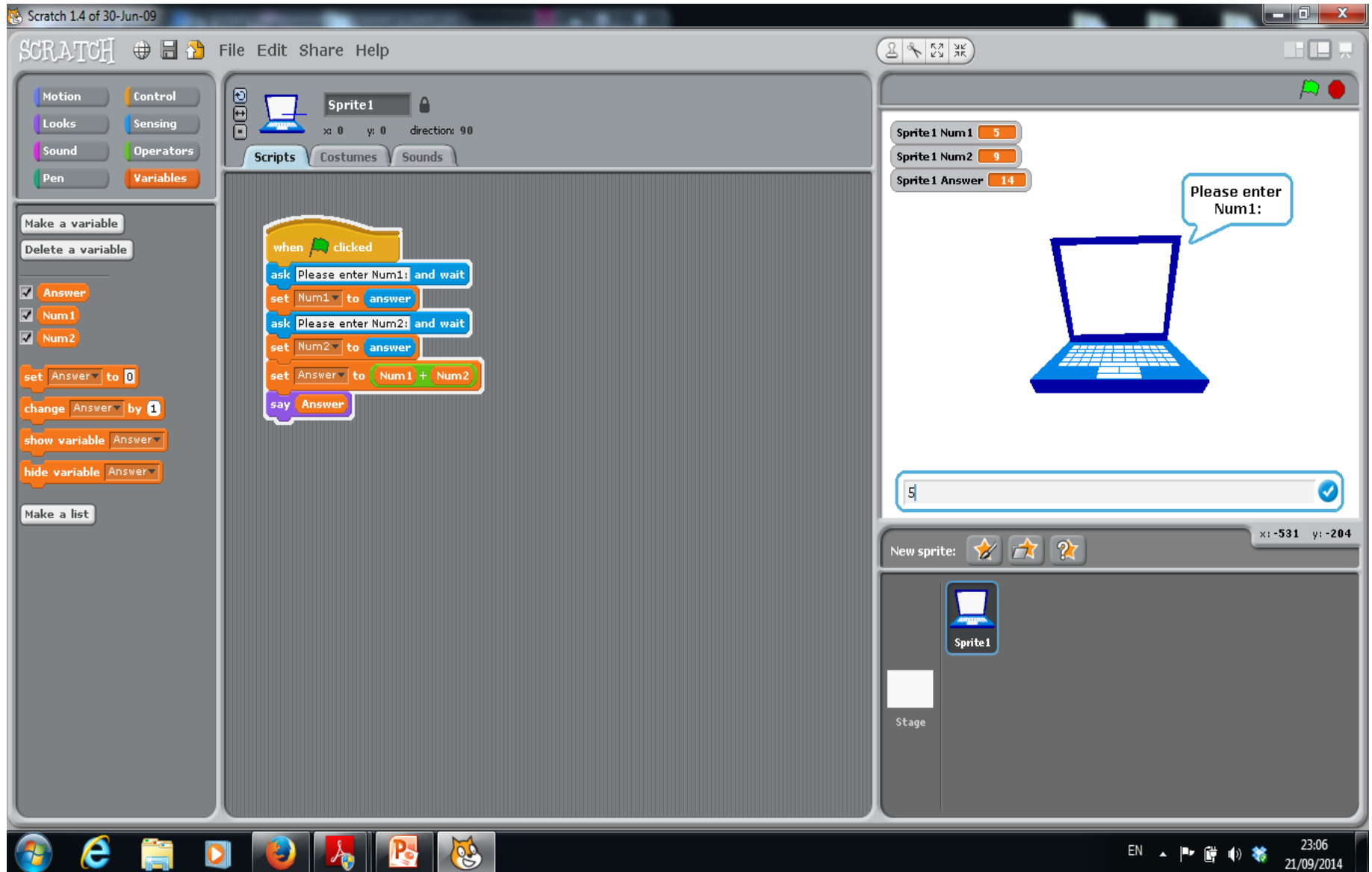
**More example programs**



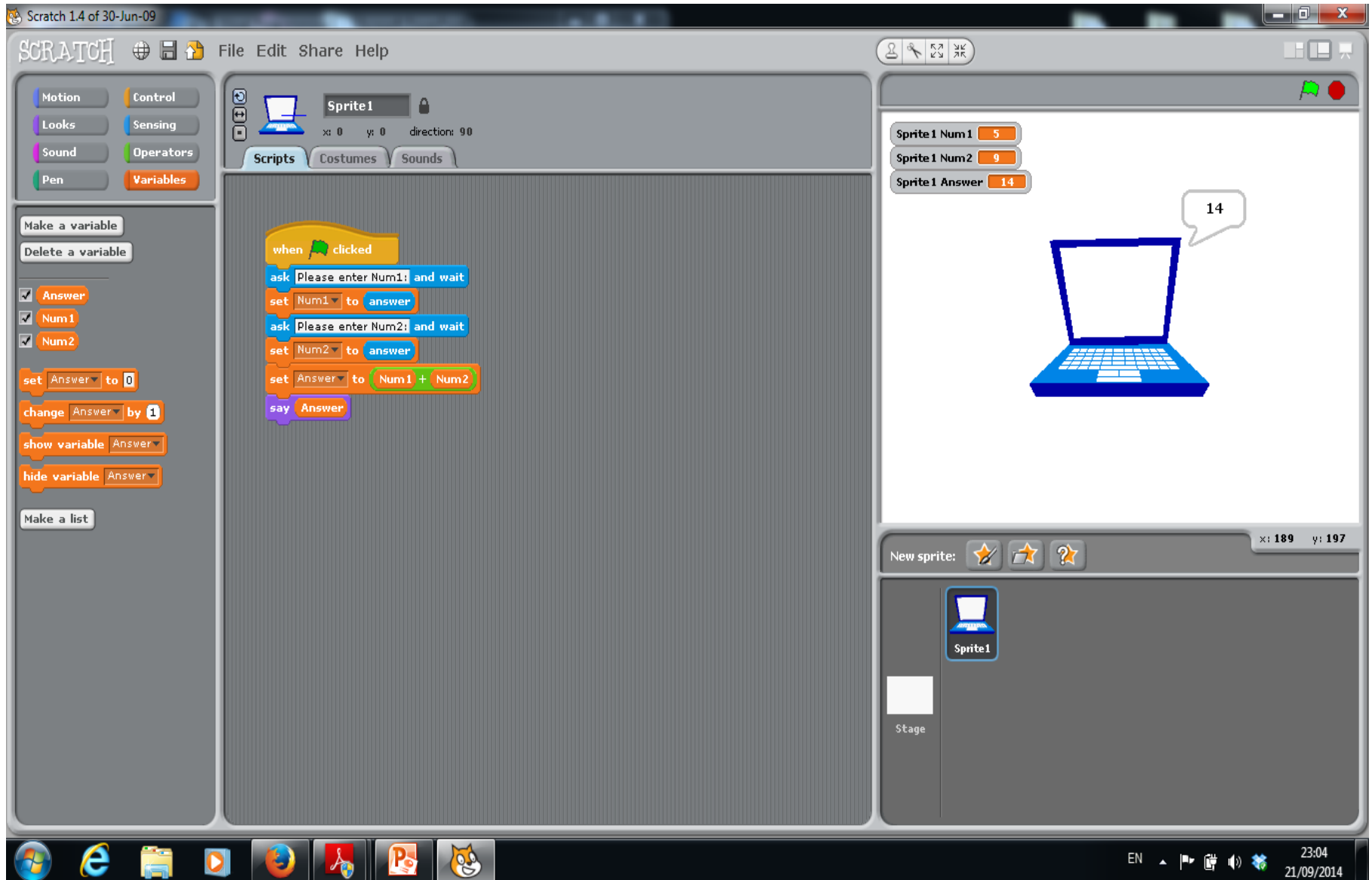
# Create the variables



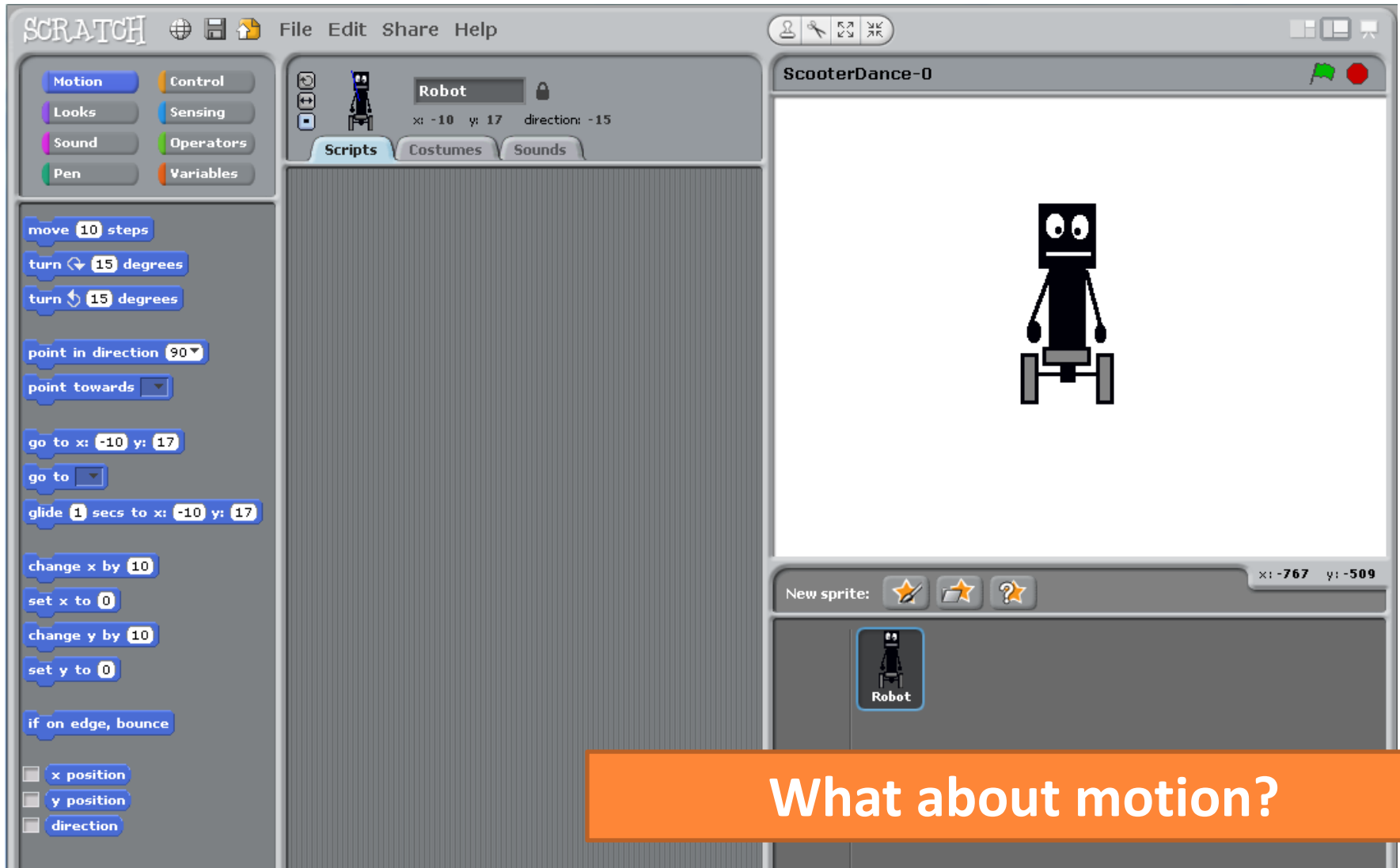
# The program



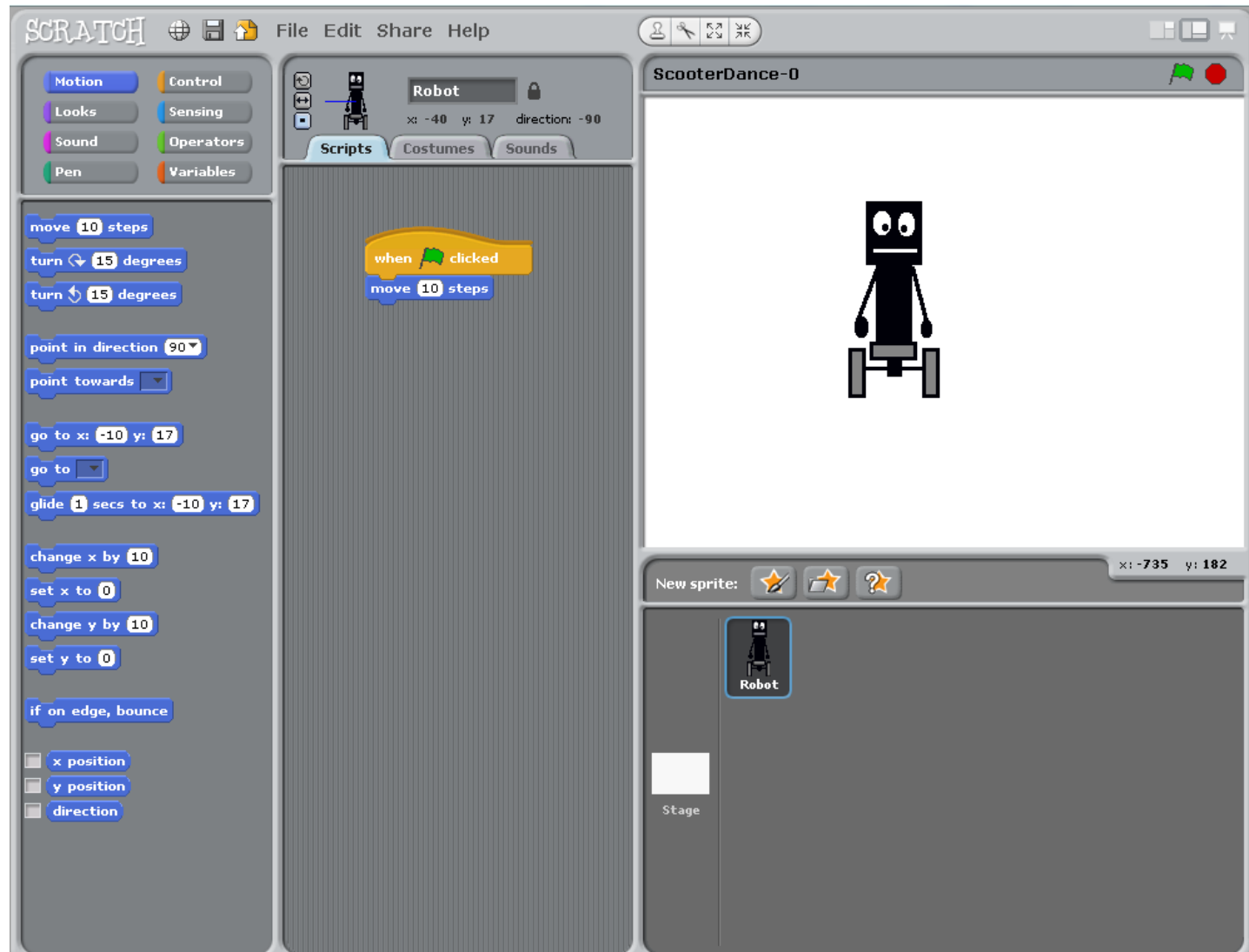
# Run the program



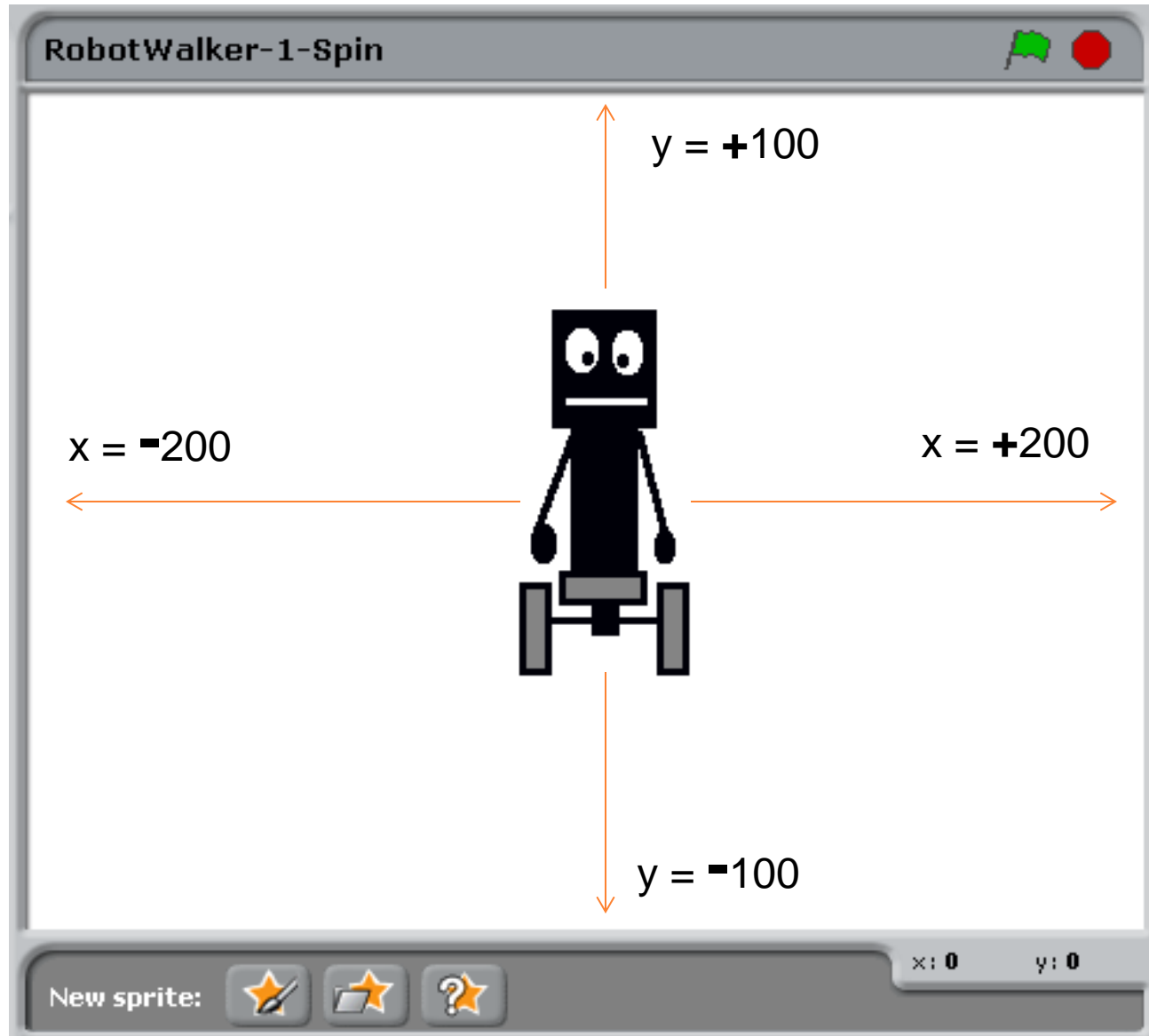
# That was just one type of program



# Move



# Co-ordinates



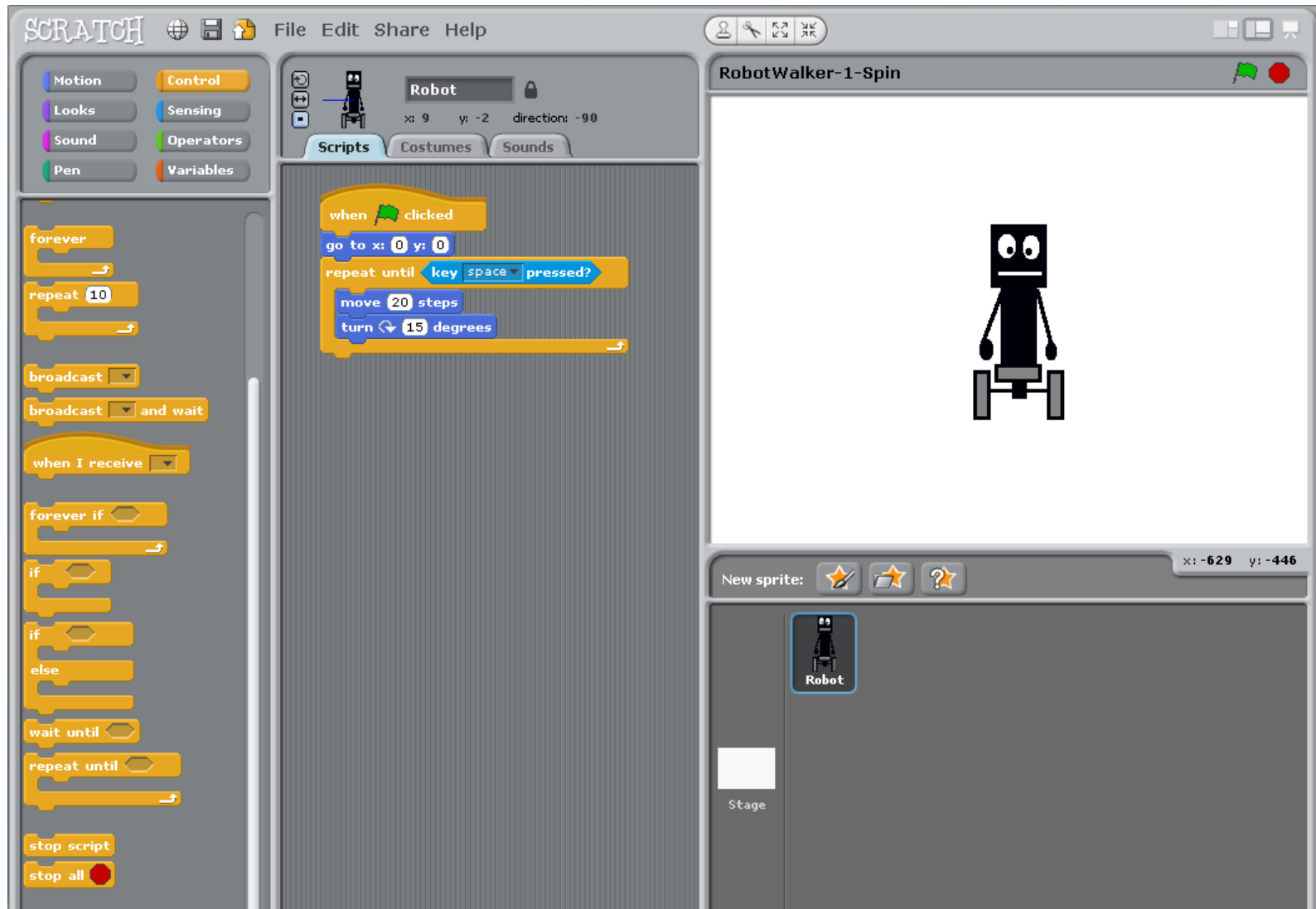


## Some interesting controls

- What do you think these controls do ?
- How might they help us move scooter around the screen?



# What does this code do?





Want to walk the block

How?

# Walking the block and not falling off the edge - 1

Robot moving - Scratch

SCRATCH File Edit Share Help

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

**Scripts** Costumes Sounds

Robot x: 0 y: 0 direction: 90

when green flag 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

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 294 y: -110

go to

glide 1 secs to x: 294 y: -110

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

Robot moving

xMax 200

xMin -200

yMax 100

yMin -100

x: -394 y: -169

New sprite: [Robot] [Stage]

20:32 23/09/2014

# Walking the block and not falling off the edge - 2

Robot moving - Scratch

SCRATCH File Edit Share Help

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

**Scripts** Costumes Sounds

Robot x: 0 y: 0 direction: 90

when green flag 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

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 294 y: -110

go to

glide 1 secs to x: 294 y: -110

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

Robot moving

xMax 200

xMin -200

yMax 100

yMin -100

x: 65 y: -49

New sprite: Robot

Stage

20:33 23/09/2014

# Walking the block and not falling off the edge - 3

RobotWalker-2-WalkTheBlock- Scratch

SCRATCH File Edit Share Help

Motion Looks Sound Pen Control Sensing Operators Variables

Robot  
x: 0 y: 0 direction: 60

Scripts Costumes Sounds

when clicked

when space key pressed

when Robot clicked

wait 1 secs

forever

repeat 10

broadcast

broadcast and wait

when I receive

forever if

if

if else

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

RobotWalker-2-WalkTheBlock

Robot

New sprite: Star Star ?

Robot

Stage

x: -819 y: 66

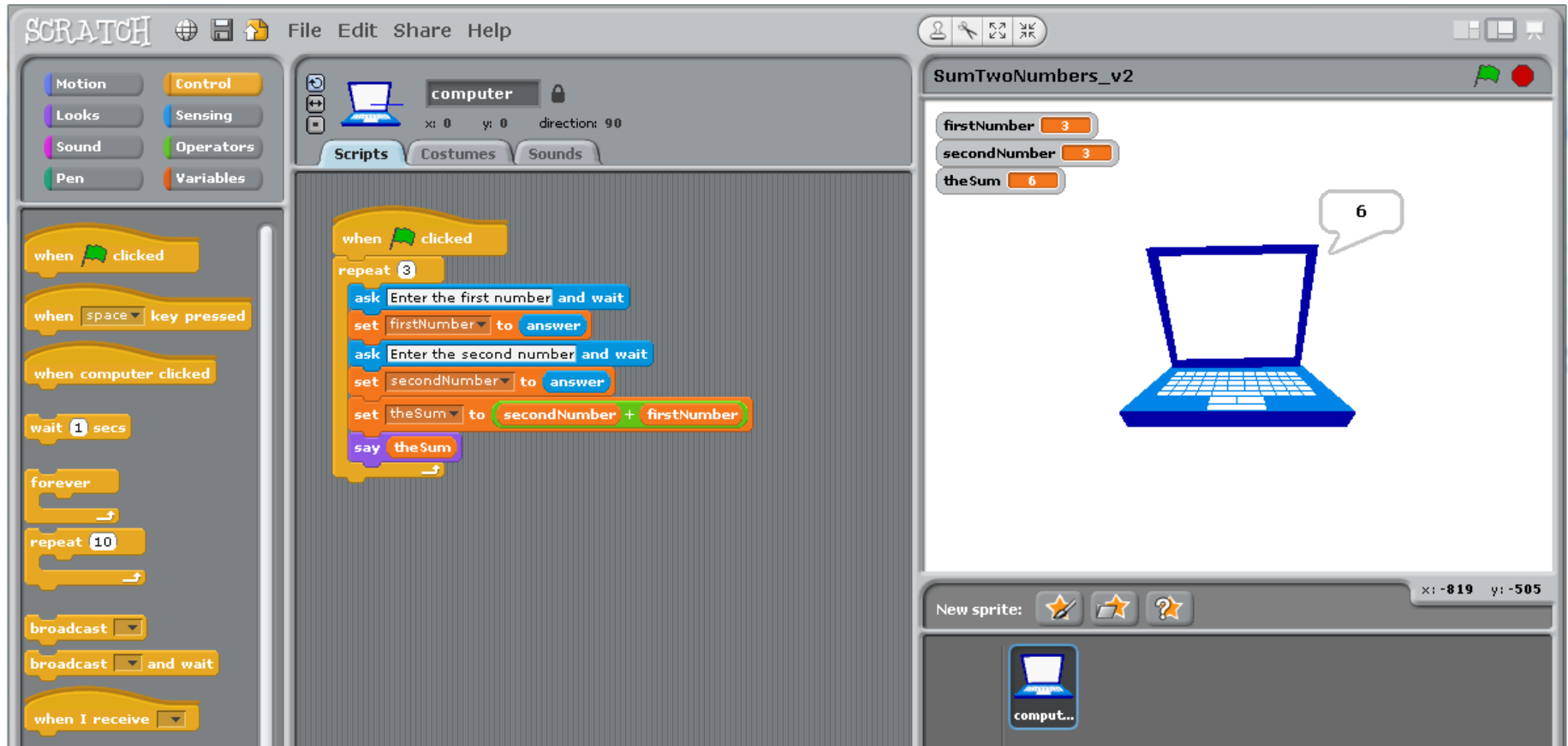
# Keep walking the block

The image shows a Scratch project window titled "RobotWalker-3-KeepWalkingTheBlock- Scratch". The project is named "RobotWalker-3-KeepWalkingTheBlock". The stage is titled "RobotWalker-3-KeepWalkingTheBlock" and shows a black robot sprite named "Robot" at coordinates x: -25, y: 16, direction: 60. The script area contains the following code:

```
when green flag 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
```

The left sidebar shows the Scratch interface with the "Scripts" tab selected. The right sidebar shows the "New sprite:" section with a "Robot" sprite selected. The stage area shows the robot sprite at its starting position.

# If we use repeat with our sum two numbers

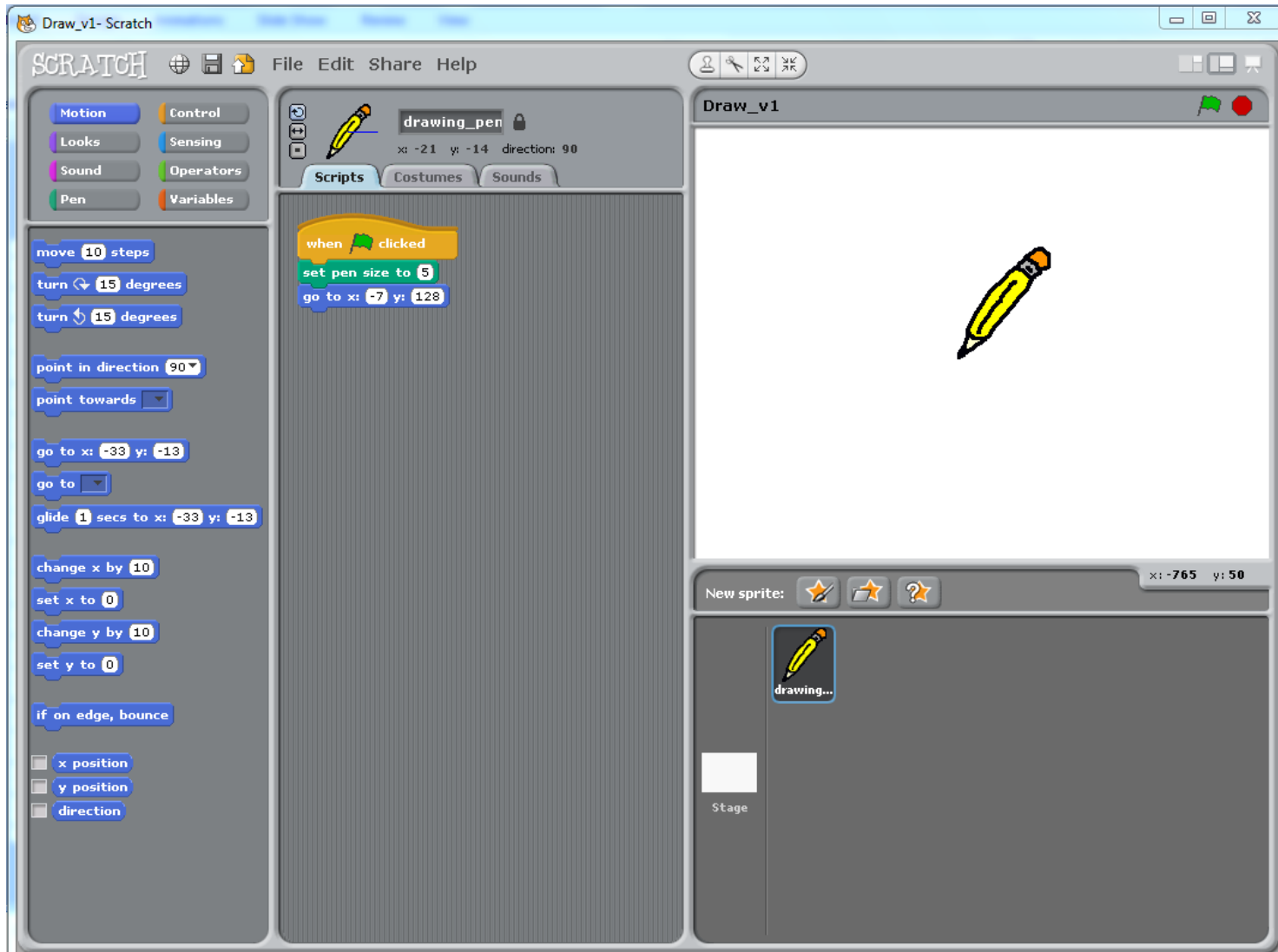


## What happens?

If we had a pen

If we could  
control a pen we  
could draw

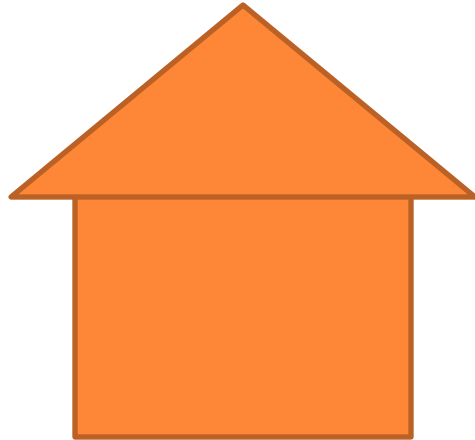
# A pen sprite





# Problem Definition

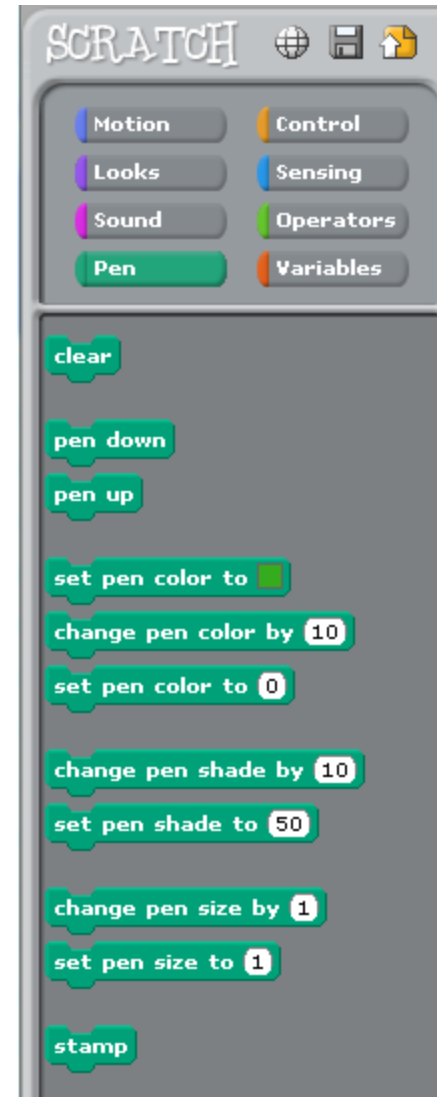
- We want to draw a house as per the picture below



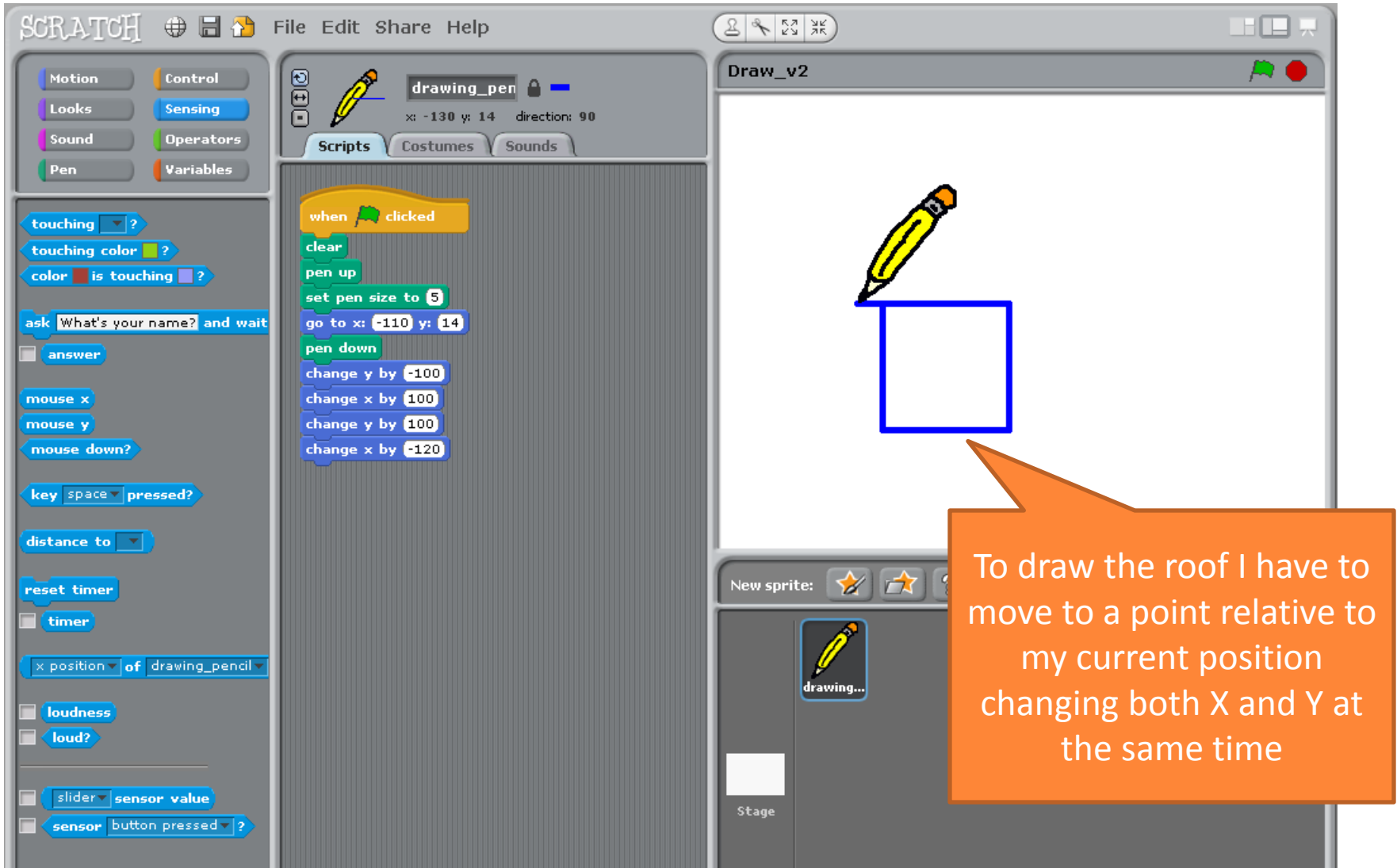
What do we need to do?

How could we do this?

What can we do with a pen?



# Can draw a square



The image shows the Scratch 2.0 interface. On the left is the 'Scripts' palette with various event, control, and sensing blocks. The center workspace shows a script for a sprite named 'drawing\_pen'. The script starts with a 'when clicked' event, followed by 'clear', 'pen up', 'set pen size to 5', 'go to x: -110 y: 14', 'pen down', and a series of 'change' blocks to draw a square: 'change y by -100', 'change x by 100', 'change y by 100', and 'change x by -120'. On the right, the 'Stage' area shows a yellow pencil sprite at the top left of a blue square. A speech bubble points to the square with the text: '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'. The bottom right shows the 'New sprite' button and a 'drawing...' sprite icon.

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

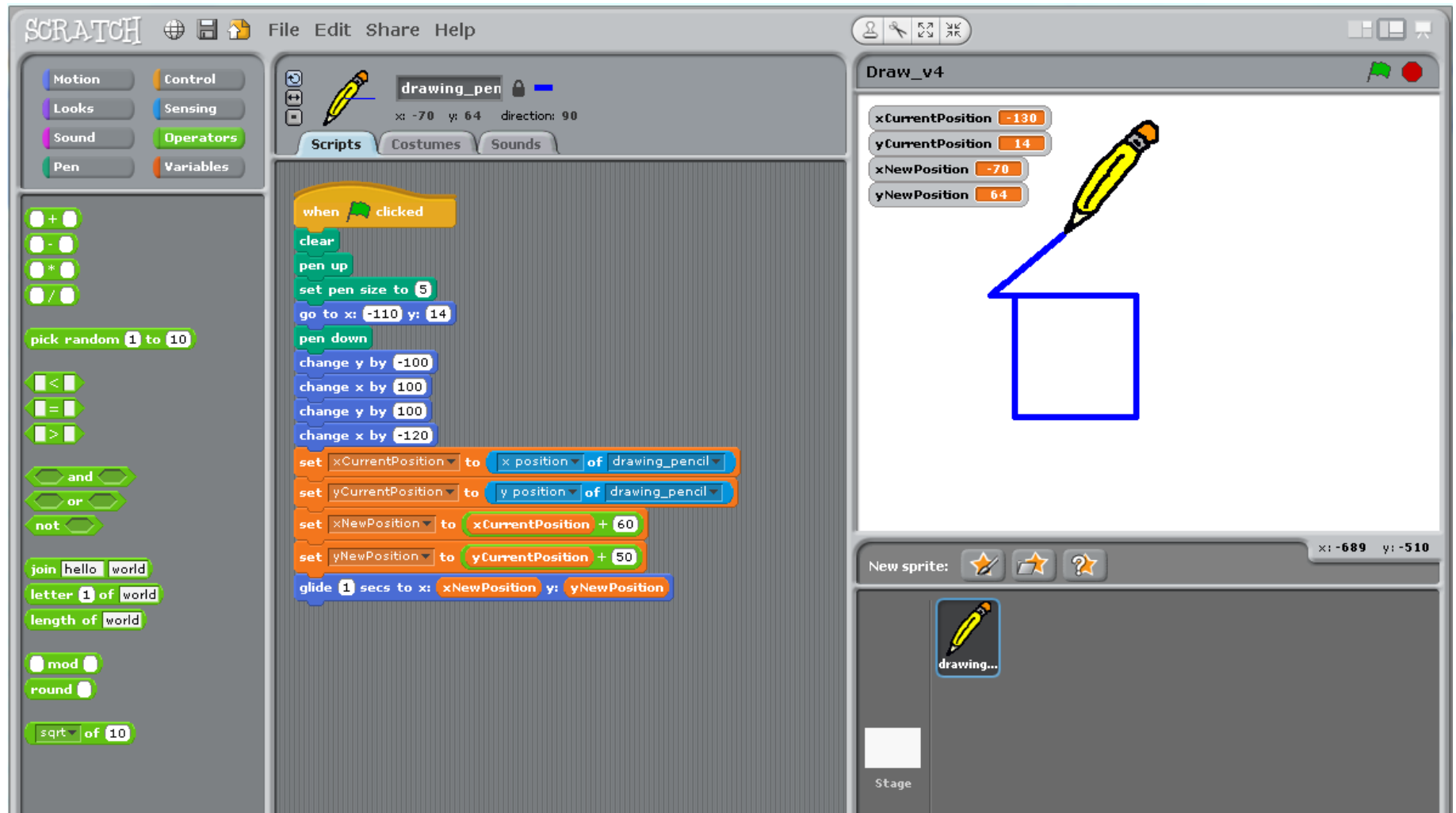
Draw\_v2

New sprite: drawing...

Stage

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



The image shows the Scratch 2.0 interface. On the left is the 'Scripts' palette with various event, control, and logic blocks. The main workspace contains a script for a drawing pencil sprite. The script starts with a 'when clicked' event, followed by 'clear', 'pen up', 'set pen size to 5', and 'go to x: -110 y: 14'. Then, 'pen down' is used to draw a square. The square is drawn by changing y by -100, x by 100, y by 100, and x by -120. To track the current and next positions, variables 'xCurrentPosition' and 'yCurrentPosition' are set to the current x and y positions of the drawing\_pencil. Then, 'xNewPosition' is set to 'xCurrentPosition + 60' and 'yNewPosition' is set to 'yCurrentPosition + 50'. Finally, a 'glide 1 secs to x: xNewPosition y: yNewPosition' block moves the pencil to the next corner of the square.

**Scratch Interface Details:**

- Top Bar:** SCRATCH logo, icons for globe, save, and share, and menu items: File, Edit, Share, Help.
- Left Palette:** Categories: Motion, Looks, Sound, Pen, Control, Sensing, Operators, Variables.
- Main Workspace:**
  - Sprite:** drawing\_pencil (pencil icon), x: -70, y: 64, direction: 90.
  - Scripts:**
    - 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
    - set xCurrentPosition to x position of drawing\_pencil
    - set yCurrentPosition to y position of drawing\_pencil
    - set xNewPosition to xCurrentPosition + 60
    - set yNewPosition to yCurrentPosition + 50
    - glide 1 secs to x: xNewPosition y: yNewPosition

**Draw\_v4 Window:**

- Variables:**
  - xCurrentPosition: -130
  - yCurrentPosition: 14
  - xNewPosition: -70
  - yNewPosition: 64
- Canvas:** A drawing of a yellow pencil with a blue line forming a square.
- Bottom Bar:** New sprite: (pencil icon), (star icon), (question mark icon). x: -689 y: -510.

# Summary

- Looked at a visual programming tool
- Created a basic add two numbers program
- Looked at the 'repeat' or loop program control for moving objects around
- How repeat can be used not just to move objects but to ask for input multiple times
- How to draw object using co-ordinates

# 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