Code For Kids Curriculum

Prepared by:

Baran Saleh
2017

Table of Content

Lecture [1]

1. Activity 1: Count the dots—Binary numbers

1.1 OVERVIEW

All data in a modern digital computer is ultimately stored and transmitted as a series of zeros and ones. This activity demonstrates how numbers and text can be represented using just these two symbols. Focusing on representing numbers in base two.

1.2 Technical terms

Binary number representation; binary to decimal conversion.

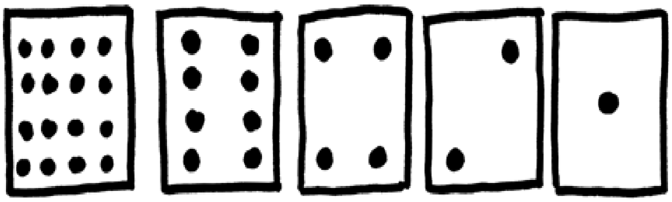
1.3 Materials

- Sets of five cards as in figure 1.1
- Worksheet1: Binary numbers
- Worksheet2: sending secret massage

Start Introduction:

Demonstrate the principles to the whole group.

Choose five children to hold the demonstration cards at the front of the class. The cards should be in the following order:



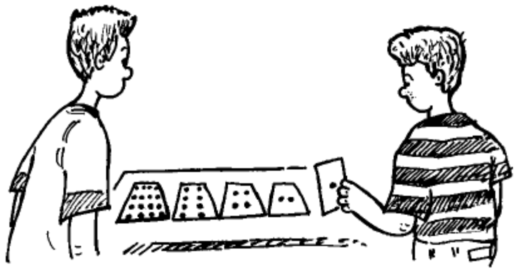


Figure 1.1: Initial layout of the binary cards

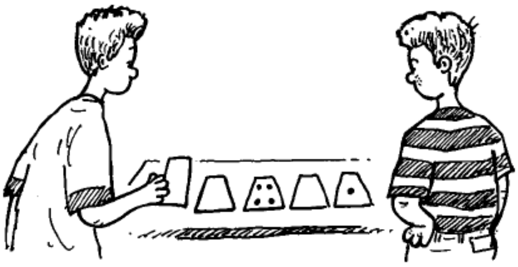
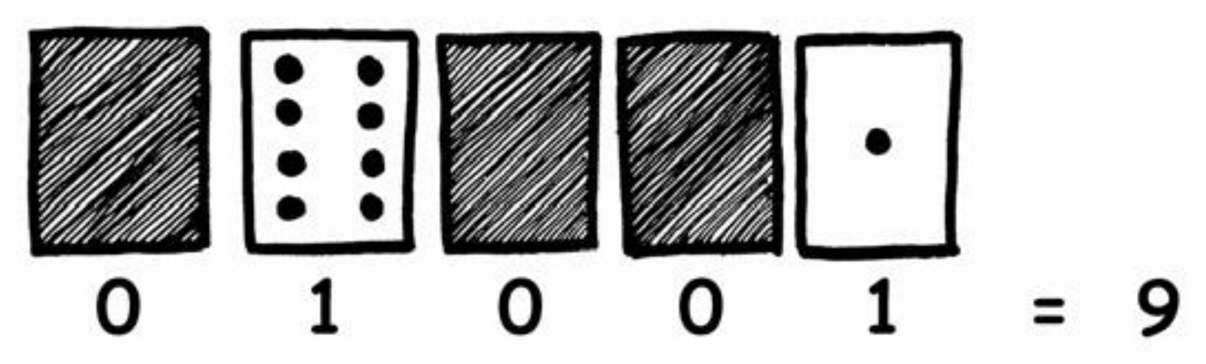


Figure 1.2: Flipping the cards to show five dots

Worksheet Activity: Working With Binary

The binary system uses **zero** and **one** to represent whether a card is face up or not. **0** shows that a card is hidden, and **1** means that you can see the dots. For example:



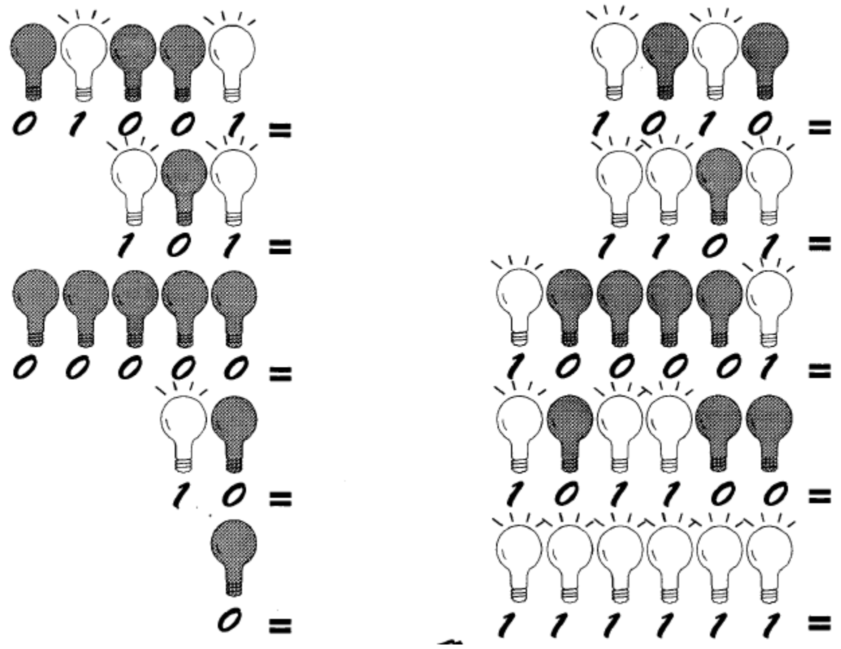
- Can you work out what **10101** is? What about **11111**?

Worksheet 1:

BINARY NUMBERS

Try to work out these binary numbers

TOP SECRET



☒☑☒☒☑ =
(☑=1, ☒=0)

⇧⇩⇧ =
(⇧=1, ⇩=0)

○○○○○ =
(⊙=1, ○=0)

📫📪 =
(📫=1, 📪=0)

☹ =
(☺=1, ☹=0)

👍👎👍👎 =
(👍=1, 👎=0)

✚✚✖✚ =
(✚=1, ✖=0)

↻↺↺↺↻ =
(↻=1, ↺=0)

▲▼▲▼▼ =
(▲=1, ▼=0)

♠♠♠♠♠ =
(♠=1, ♣=0)

Lecture [2]

Worksheet 2(A): Sending Secret Messages

1	2	3	4	5	6	7	8	9	10	11	12	13
a	b	c	d	e	f	g	h	i	j	k	l	m
14	15	16	17	18	19	20	21	22	23	24	25	26
n	o	p	q	r	s	t	u	v	w	x	y	z

	🎄				1
		🎄		🎄	2
	🎄	🎄			3
🎄					4
					5
	🎄			🎄	6
	🎄	🎄		🎄	7
					8
🎄		🎄			9
🎄			🎄		10
				🎄	11
🎄					12
🎄					13
		🎄		🎄	14
		🎄			15

- **(B) Write your own secret massages** ☺

					1						25
					2						26
					3						27
					4						28
					5						29
					6						30
					7						31
					8						32
					9						33
					10						34
					11						35
					12						36
					13						37
					14						38
					15						39
					16						40
					17						41
					18						42
					19						43
					20						44
					21						45
					22						46
					23						47
					24						48

2. Activity (2): You Can Say That Again! -Text Compression

2.1 OVERVIEW

Since computers only have a limited amount of space to hold information, they need to represent information as efficiently as possible. This is called compression. By coding data before it is stored, and decoding it when it is retrieved, the computer can store more data, or send it faster through the Internet.

2.2 Technical terms

- Coding
- Compression
- Memory space

2.3 Materials

- Worksheet Activity3: patterns of letters in this poem
- Worksheet Activity4: letters are missing
- Worksheet Activity5: How many words do you really need here?

Worksheet 3:

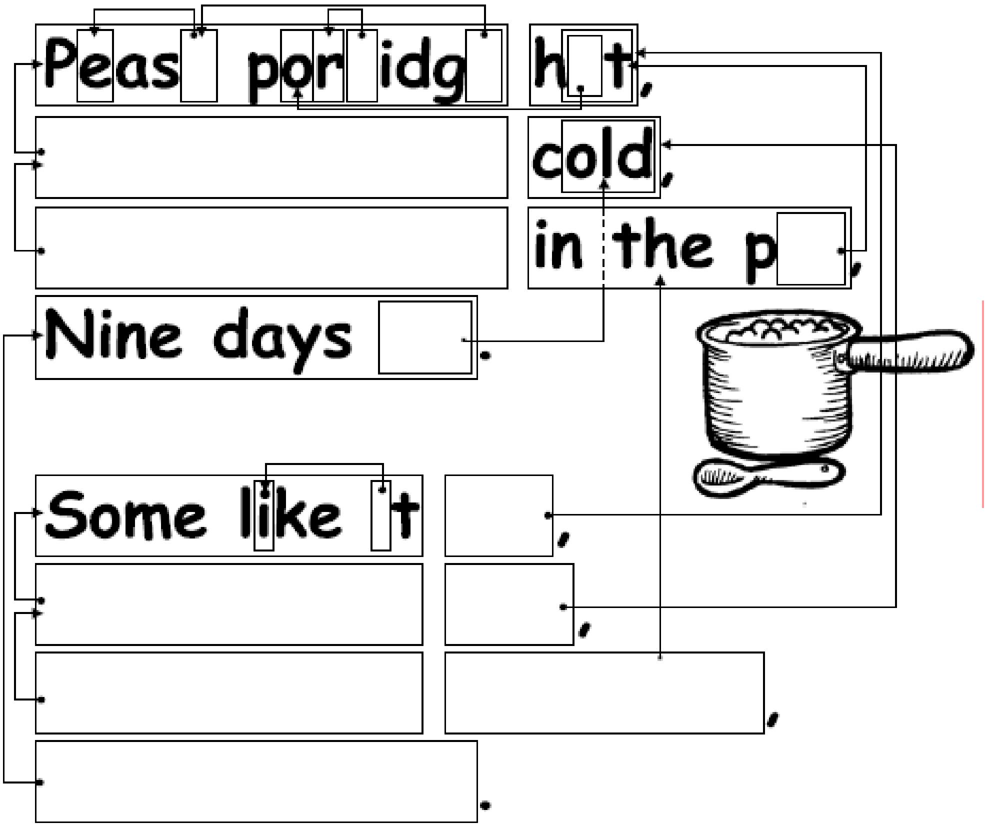
Look for the patterns of letters in this poem. Can you find groups of 2 or more letters that are repeated, or even whole words or phrases?

The Rain

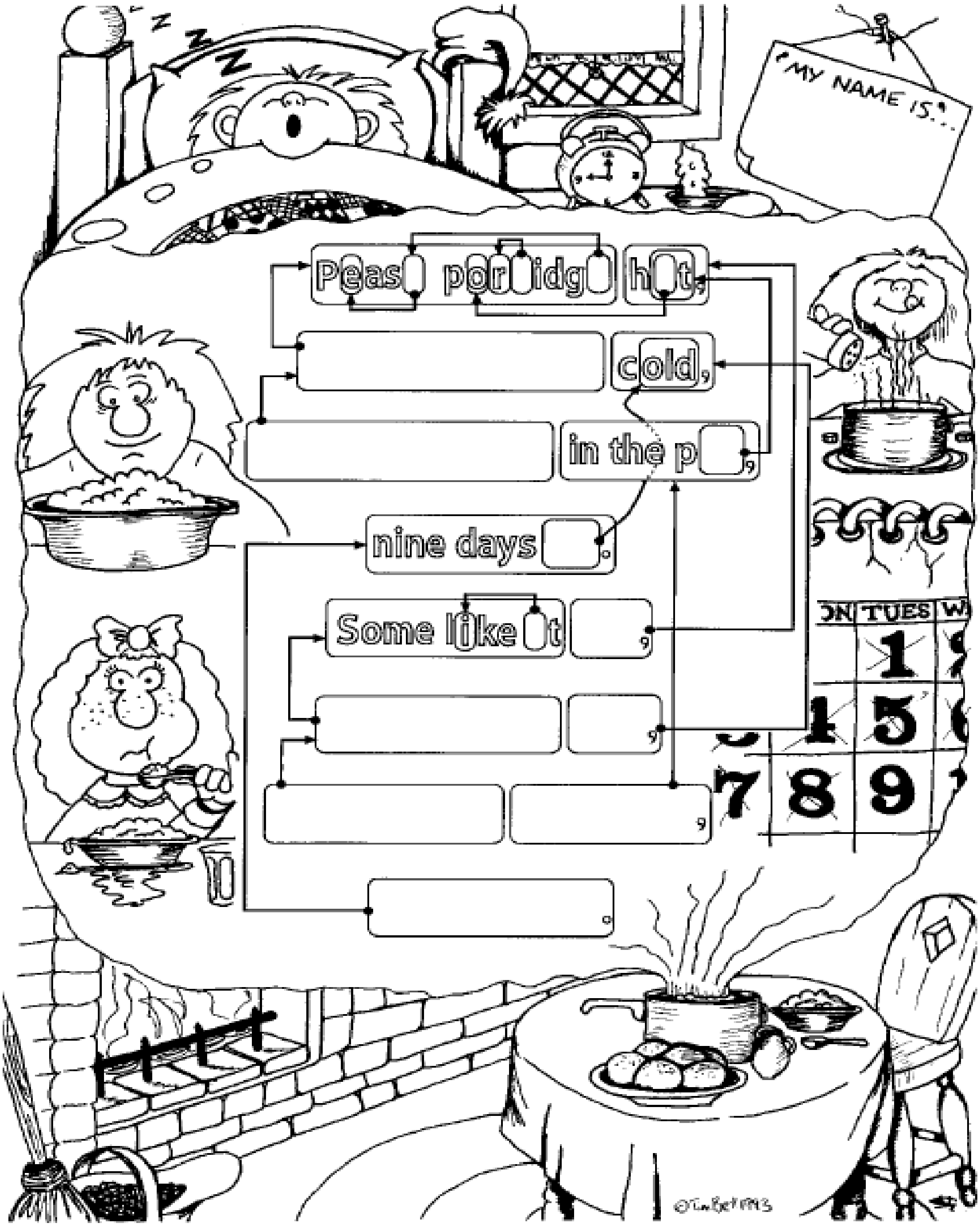
Pitter patter
Pitter patter
Listen to the rain
Pitter patter
Pitter patter
On the window pane

Worksheet 4: (A)

Many of the words and letters are missing in this poem. Can you fill in the missing letters and words to complete it correctly? You will find these in the box that the arrow is pointing to.



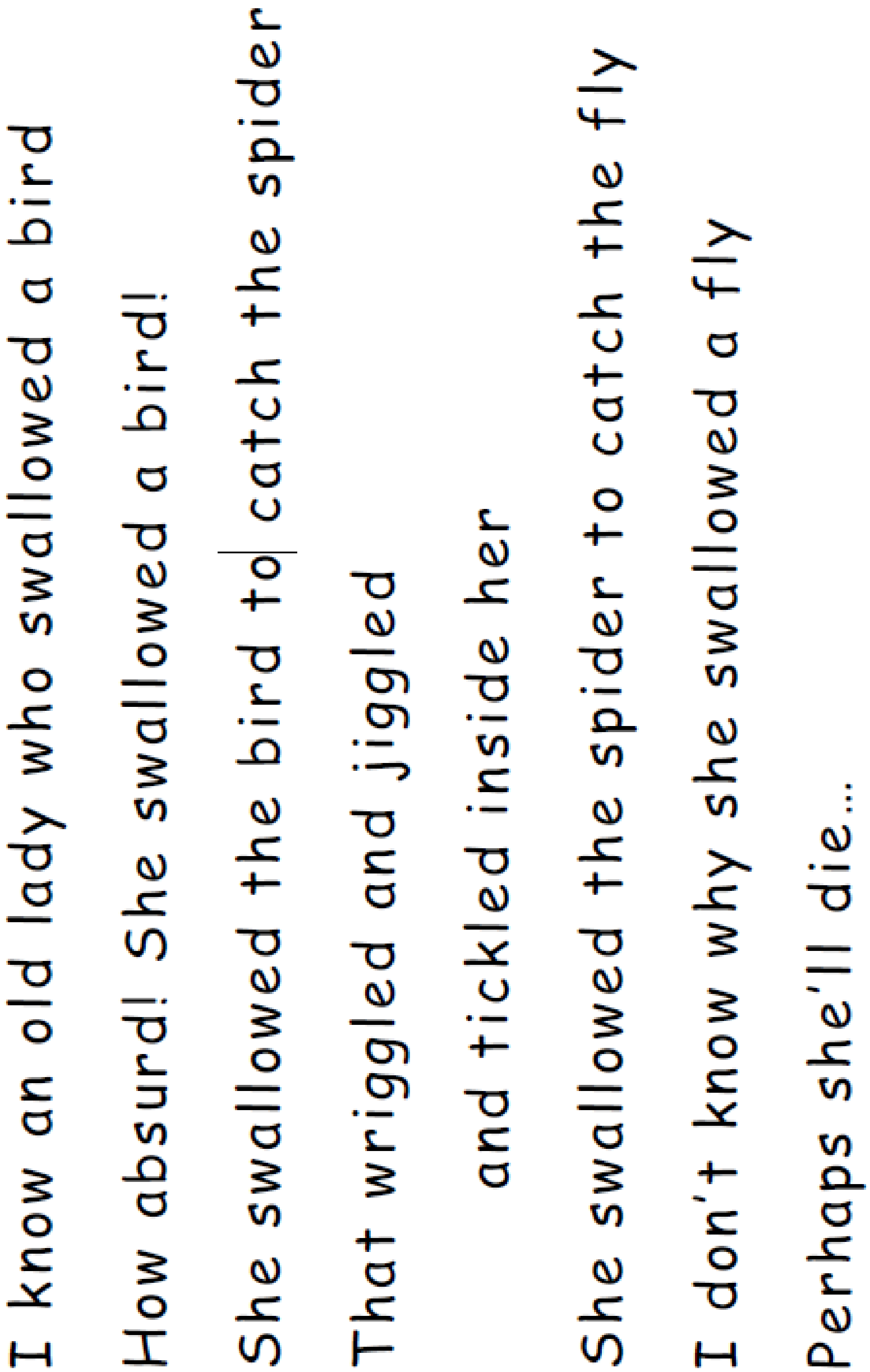
Worksheet 4: (B)



Instructions: *Fill in each blank box by following the arrow attached to the box and copying the letters in the box that it points to.*

Worksheet 5:

How many words do you really need here?
Pretend you are a computer trying to fit as much into your disk as possible. Cross out all the groups of two or more letters that have already occurred.



I know an old lady who swallowed a bird
How absurd! She swallowed a bird!
She swallowed the bird to catch the spider
That wriggled and jiggled
and tickled inside her
She swallowed the spider to catch the fly
I don't know why she swallowed a fly
Perhaps she'll die...

➢

➢

Lecture [3]

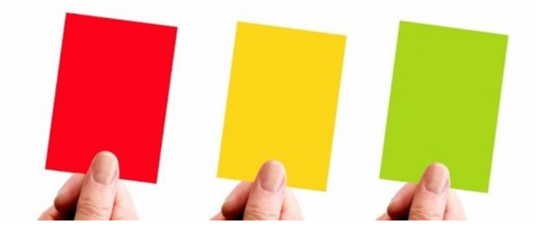
3. Activity 3: Conditionals with Cards

3.1 OVERVIEW

We don’t always know ahead of time what things will be like when we run our computer programs. Different users have different needs, and sometimes you will want to do something based off of one user’s need that you don’t want to do with someone else. That is where conditionals come in. This lesson demonstrates how conditionals can be used to tailor a program to specific information.

3.2 Technical terms

Algorithm, conditionals, loops.



3.3 Materials

Cards: red & green or others.

To begin, your class will need to understand the idea of an algorithm (a list of steps that you can follow to finish a task) as well as loops (the action of doing something over and over again) and eventually conditionals (statements that only run under certain conditions).
It is not required that you define each of those terms before you present them with examples. It can be easier to comprehend the definitions if they have had experience with the concepts.

- **<u>If/else aspect: (Play a game)</u>**

Get the classroom’s attention. Let them know that you’re going to play a game.
“If I raise my arms, I want you all to stand up. Then, if I put them down, I want you to sit. Ready?”
Raise your arms and watch the class stand. Put them down and make sure they sit. Run through the combination a couple of times, using whatever order you like. When you’re certain that they have it, add some more detail.
“If I clap when I raise my hands, I want you to clap when you stand. Else, stand quietly.”
And write the conditions in sentences on the board in case they forget.
Go through this action a few times to see how the class deals with the If/else aspect.
When you’re comfortable that they get it, go one step further and add more instructions to the conditions.

Here is a sample algorithm:

```
If (card is red)
     Say your name

Else if (card is green)
     Say next

Else,
     Stand up
```

➢ Loop aspect

1. Start with the student at the first desk.
2. If student shouts “Ready!” I give direction. Else, I move to next student.
* If I raise my hand, the student shouts their name.
* Else, student shouts my name.

When you’re done, you will have shown the students what an algorithm is (by listing out the steps of the game), what a loop is (by repeating the steps over and over for each student) and what a conditional is (by doing one thing “if” you raise your hand, and something else otherwise).

Lecture [4]

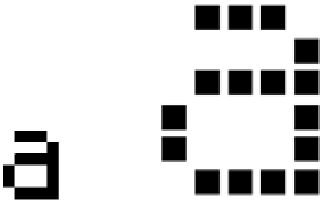
4. Activity 4: Colour by numbers—Image Representation

1.4 4.1 OVERVIEW

Computers store drawings, photographs and other pictures using only numbers. The following activity demonstrates how they can do this.

Computer screens are divided up into a grid of small dots called *pixels* (**pic**ture **el**ements).

In a black and white picture, each pixel is either black or white.



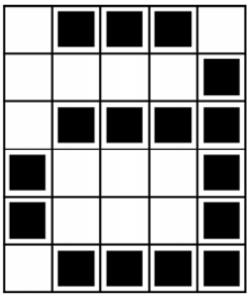
4.2 Technical terms

Raster images; pixels; image compression; run-length coding; facsimile machines.

4.3 Materials

- Worksheet 6: Kid Fax
- Worksheet Activity 7: Make Your Own Picture

The letter "a" has been magnified above to show the pixels. When a computer stores a picture, all that it needs to store is which dots are black and which are white.



1, 3, 1
4, 1
1, 4
0, 1, 3, 1
0, 1, 3, 1
1, 4

The picture above shows us how a picture can be represented by numbers. The first line consists of one white pixel, then three black, then one white. Thus the first line is represented as 1, 3, 1.

The first number always relates to the number of white pixels. If the first pixel is black the line will begin with a zero.

Worksheet 6: Kid Fax

The first picture is the easiest and the last one is the most complex. It is easy to make mistakes and therefore a good idea to use a pencil to colour with and have a rubber handy!

4, 11
4, 9, 2, 1
4, 9, 2, 1
4, 11
4, 9
4, 9
5, 7
0, 17
1, 15

6, 5, 2, 3
4, 2, 5, 2, 3, 1
3, 1, 9, 1, 2, 1
3, 1, 9, 1, 1, 1
2, 1, 11, 1
2, 1, 10, 2
2, 1, 9, 1, 1, 1
2, 1, 8, 1, 2, 1
2, 1, 7, 1, 3, 1
1, 1, 1, 1, 4, 2, 3, 1
0, 1, 2, 1, 2, 2, 5, 1
0, 1, 3, 2, 5, 2
1, 3, 2, 5

6, 2, 2, 2
5, 1, 2, 2, 2, 1
6, 6
4, 2, 6, 2
3, 1, 10, 1
2, 1, 12, 1
2, 1, 3, 1, 4, 1, 3, 1
1, 2, 12, 2
0, 1, 16, 1
0, 1, 6, 1, 2, 1, 6, 1
0, 1, 7, 2, 7, 1
1, 1, 14, 1
2, 1, 12, 1
2, 1, 5, 2, 5, 1
3, 1, 10, 1
4, 2, 6, 2
6, 6

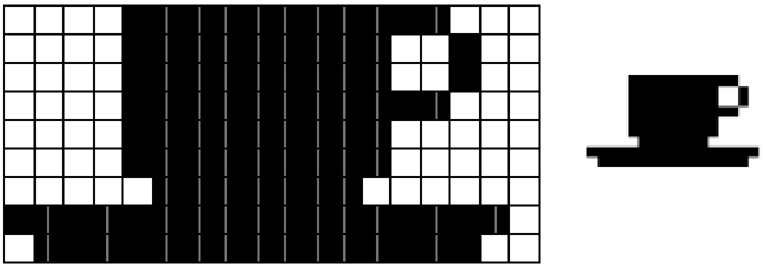
Worksheet Activity 7: Make Your Own Picture

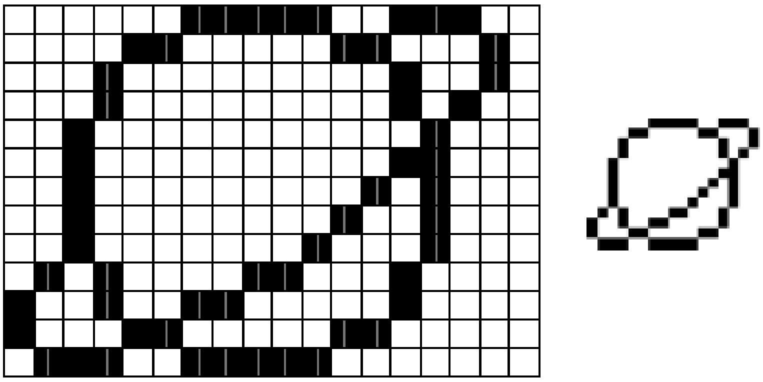
Now that you know how numbers can represent pictures, why not try making your own coded picture for a friend? Draw your picture on the top grid, and when you've finished, write the code numbers beside the bottom grid.

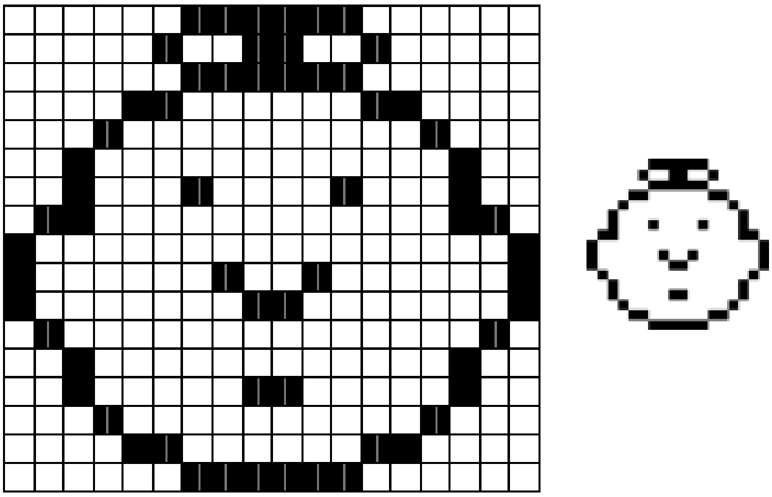
Cut along the dotted line and give the bottom grid to a friend to colour in.

Solutions and hints

Answers to Kid Fax Worksheet







LECTURE [5]

5. Scratch

5.1 Introduction to scratch and a video about it

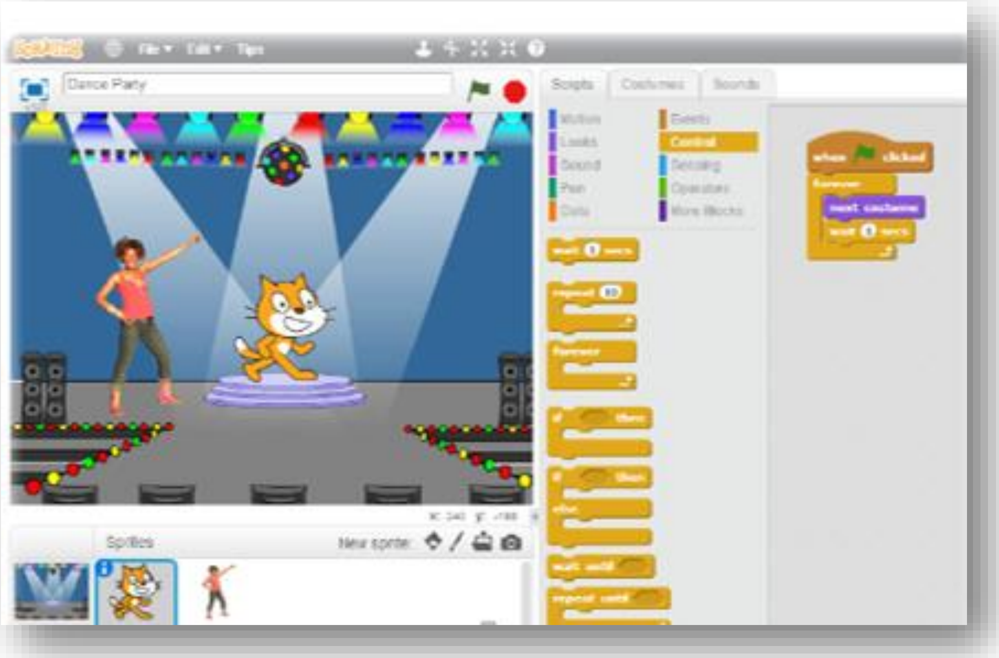
Scratch is a graphical programming language that you can download for free. By simply dragging and dropping coloured blocks, you can create interactive stories, games, animation, music, art, and presentations. You can even upload your creations to the Internet to share with others from around the world. Scratch is designed for play, self-learning, and design.

https://www.youtube.com/watch?v=ywG6lv9mFLI

5.2 Basic of scratch:

- Getting started of scratch and make account
- Introduce the UI of scratch & button functionality
- Create a simple dummy game/moving objects.

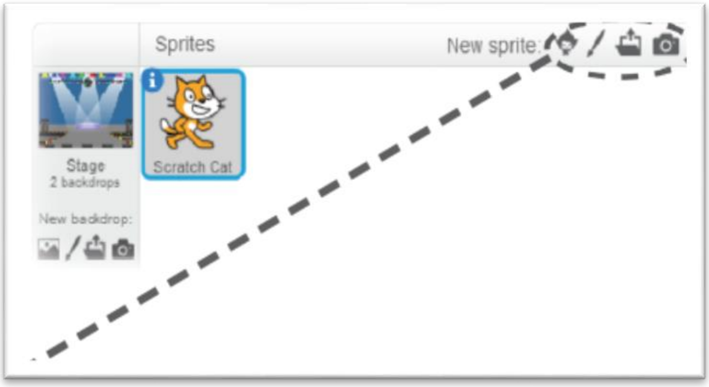
5.3 Building the scratch tutorial with the students



Lecture [6]

Scratch,, Continue

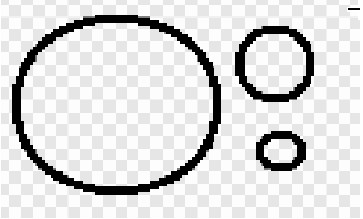
5.4 Add sprites and change backgrounds.



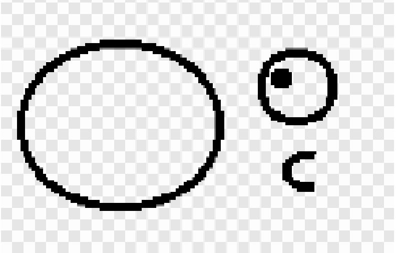
5.5 Draw a new sprite and organize them

Delete the cat sprite by right click on "sprite1" and select "Delete". Then click the ("Create new sprite" button) to open the Paint Editor. I created the following guy using mostly Paintbrush Eraser Fill Tool □ Rectangle Tool ○ Ellipse Tool Line Tool T Text Tool Select Tool Stamp Tool Eyedropper

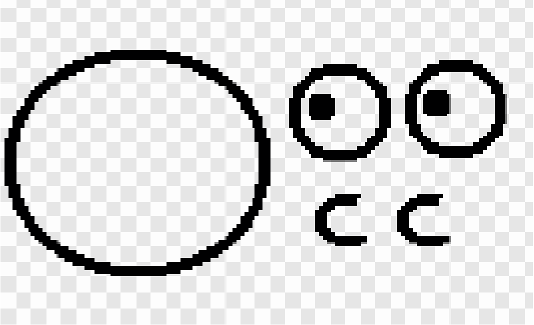
Click Ellipse Tool (○) and click hollow mode (● ○). Draw three circles.



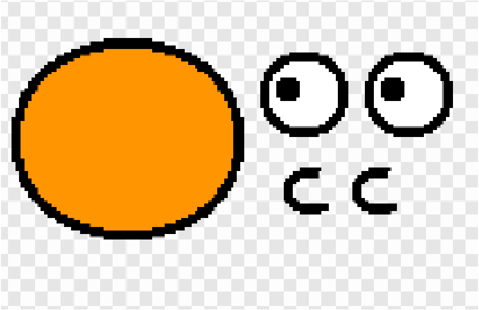
Click Ellipse Tool (○) and click solid mode (● ○); draw a little dark solid circle inside the medium circle, which will be the eye. Use Eraser tool ( ) to trim the smallest circle; this will be the ear.



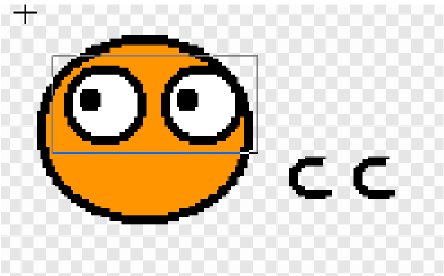
Click Stamp Tool ( ) button and select the eyeball to copy. Drag the eyeball copy to where you want the new eyeball to be. Do the same for the ear to make two ears.



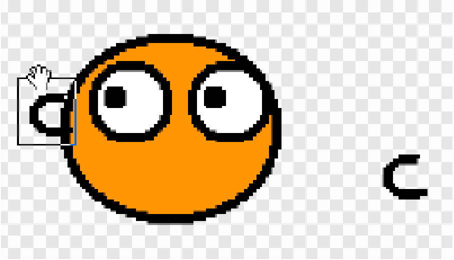
Use Fill Tool ( ) to fill the face and the eye ball.



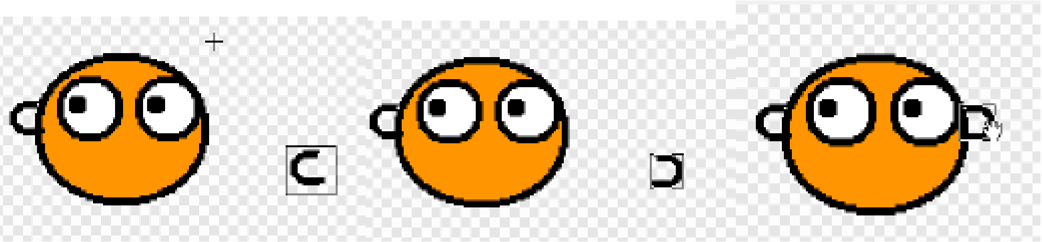
Click Select Tool ( ) button and select both eyes; move them to the face.



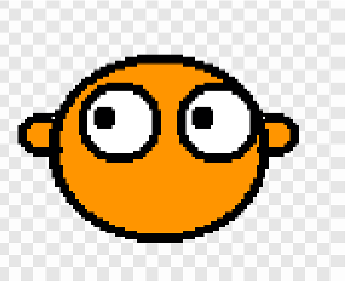
Click Select Tool ( ) button and select left ear; drag it to its place.



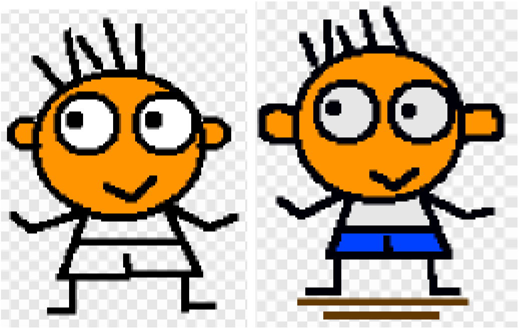
Click Select Tool () button and select the right ear; click to flip it. Then drag the right ear to its place.



Click Eyedropper Tool () and click the face to copy the face color. Click Fill Tool () and fill both ears with the face color.



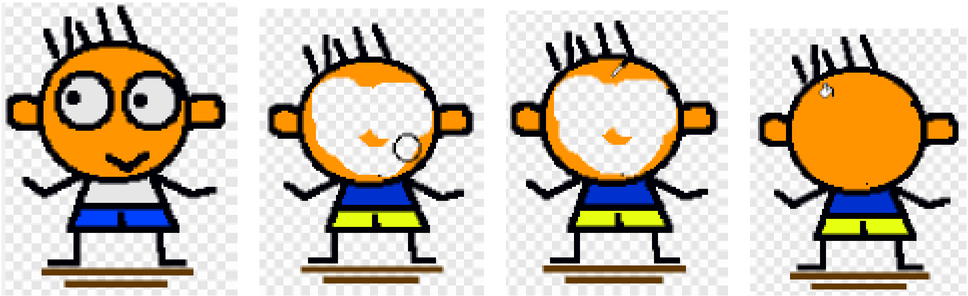
Use Line Tool () to add hair, body, arms, legs, and cloth. Fill in color with Fill Tool ().



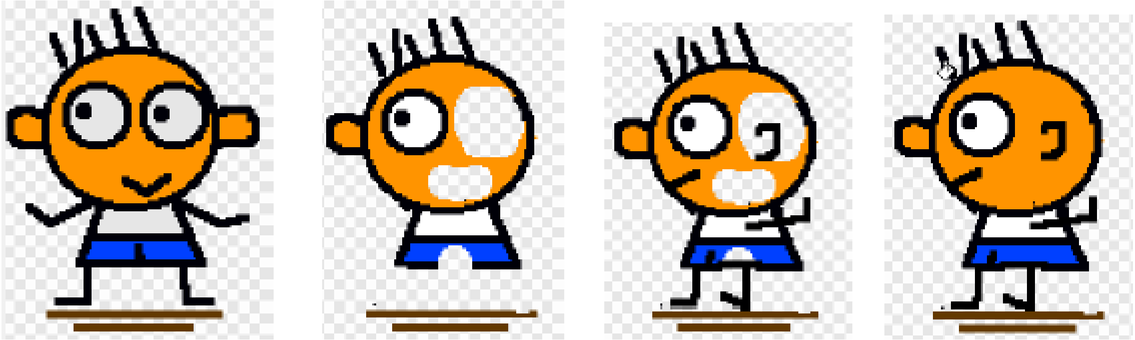
Once you are happy with your own Virtual Me, click OK to save. Rename the costume to “front”. This is the front view.

STEP 3 CREATE BACK, LEFT SIDE, and RIGHT SIDE VIEWS

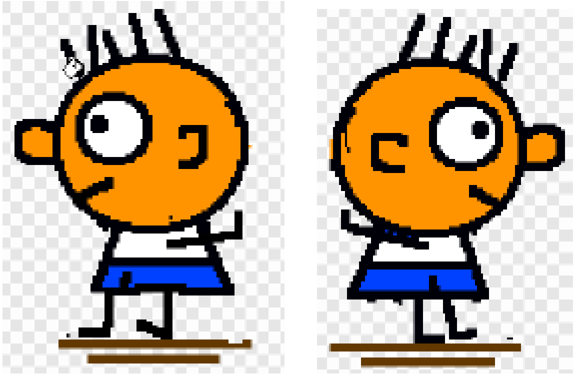
To create back view, make a copy of costume “front” (click “Copy” button next to costume “front”). Use Erase tool to erase eyes and mouth. Use Eyedropper tool to copy the face color. Then use Fill Tool to paste the color in the empty area. Click OK to save. Rename this costume as “back”.



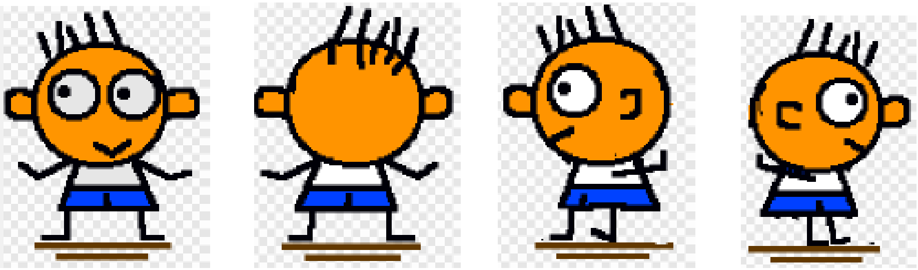
Then copy the costume "back". Erase extra body parts. Redraw body parts. Refill color of the face and the pant using Eyedropper Tool and Fill Tool. Click OK to save and rename this costume as "facing left"



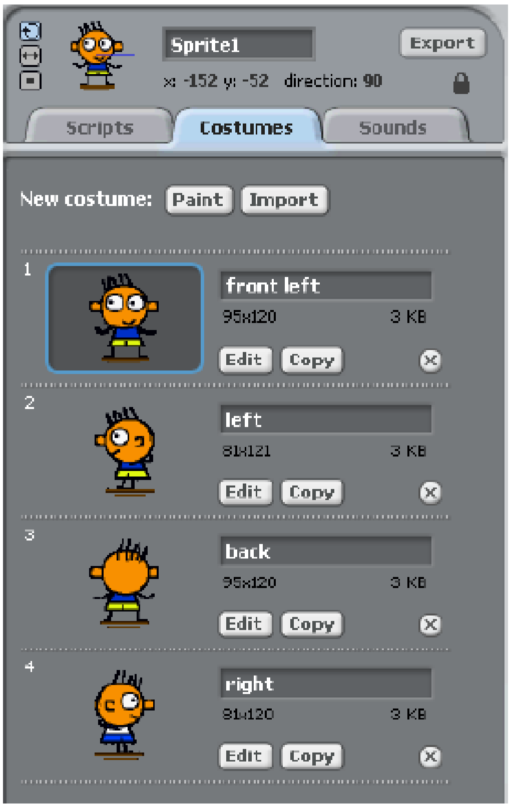
Copy the costume "facing left" and click ⇆ to flip the figure horizontally. Click OK to save and rename the costume as "facing right".



We've just created four costumes for the same sprite: "front", "back", "facing right" and "facing left".



Your Sprite Editor should looks like this:



Sprite1

x: -152 y: -52 direction: 90

New costume:

1 front left 95x120 3 KB

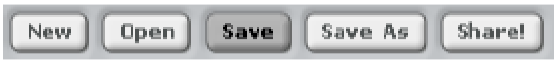
2 left 81x121 3 KB

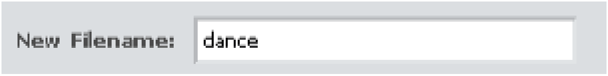
3 back 95x120 3 KB

4 right 81x120 3 KB

5.6 Saving the project & Reopen it.

Click Save button and, at the bottom of the “Save Project” window, enter “dance” as file name. Then click OK.

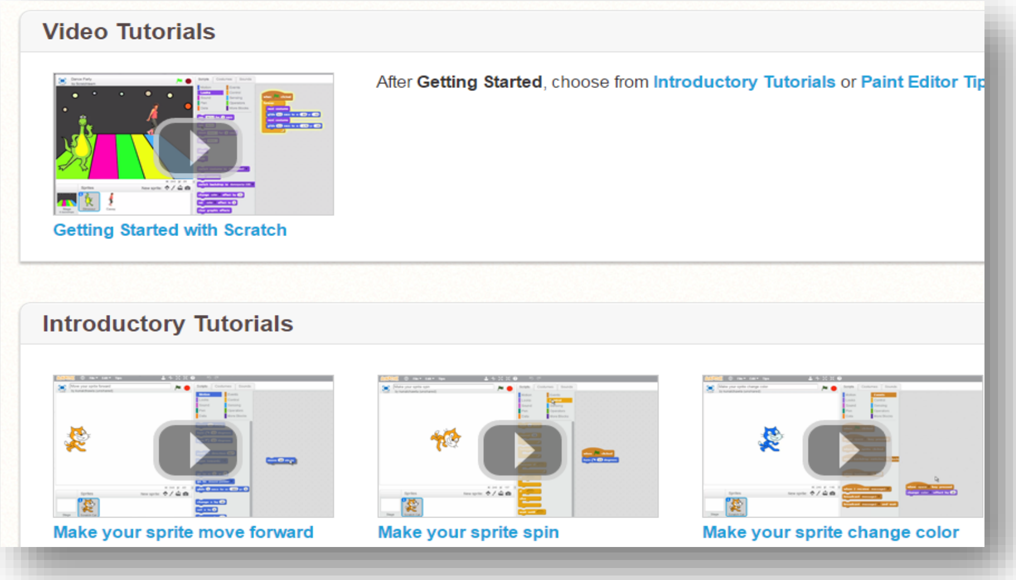




New Filename: dance

5.7 Benefits from the video tutorial exists on the website

Students will build similar projects by themselves.



Video Tutorials

After **Getting Started**, choose from Introductory Tutorials or Paint Editor Tip

Getting Started with Scratch

Introductory Tutorials

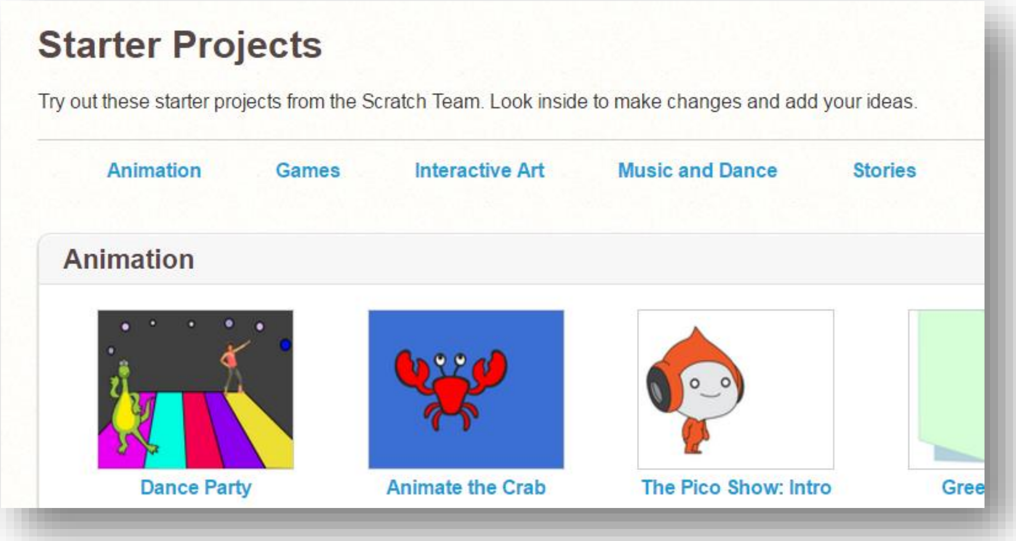
Make your sprite move forward

Make your sprite spin

Make your sprite change color

5.8 Introduce the students to use

Starter projects created by the Scratch team, and add more scripts to it.



Starter Projects

Try out these starter projects from the Scratch Team. Look inside to make changes and add your ideas.

Animation Games Interactive Art Music and Dance Stories

Animation

Dance Party

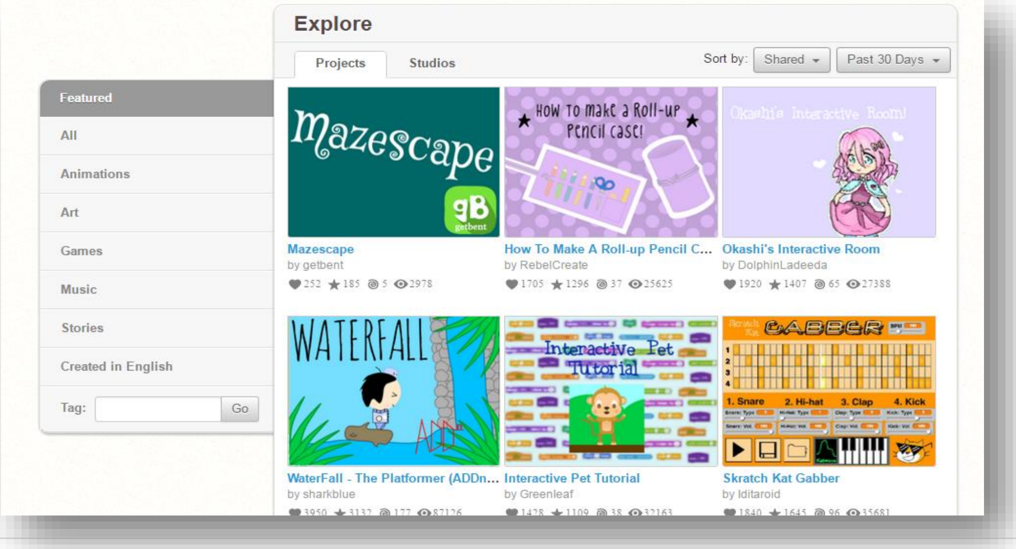
Animate the Crab

The Pico Show: Intro

Gree

5.9 Explore the existent projects

Explore the existent projects that other others have made with Scratch.



Explore

Projects Studios

Sort by: Shared Past 30 Days

Featured
All
Animations
Art
Games
Music
Stories
Created in English
Tag: Go

Mazescape
by getbent
♥ 252 ★ 185 5 2978

How To Make A Roll-up Pencil C...
by RebelCreate
♥ 1705 ★ 1296 37 25625

Okashi's Interactive Room
by DolphinLadeeda
♥ 1920 ★ 1407 65 27388

WaterFall - The Platformer (ADDn...
by sharkblue
♥ 3950 ★ 3132 177 87126

Interactive Pet Tutorial
by Greenleaf
♥ 1428 ★ 1109 38 32163

Skratch Kat Gabber
by Iditaroid
♥ 1840 ★ 1645 96 35681

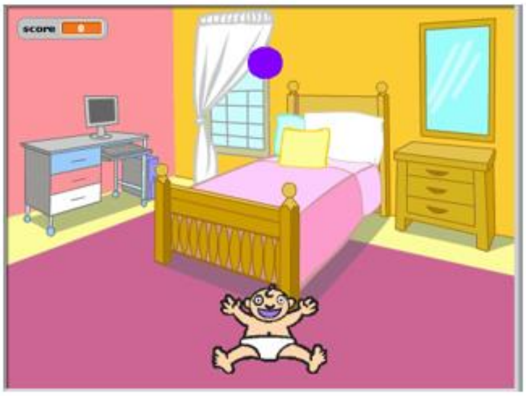
Lecture [7]

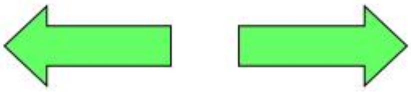
6. Building full Scratch game

Create Baby Catch Game in Scratch

6.1 Game Description

We will make a game where the player controls a baby using the arrow keys to catch a falling ball. If the baby misses the ball the game is over.





6.2 Goals

To learn about

- event handling
- simple sequential execution
- loops
- variables
- conditionals
- parallel execution
- message passing

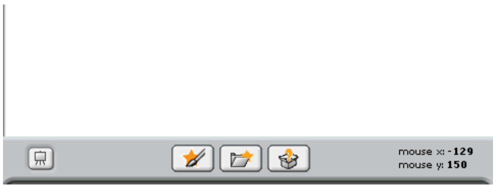
6.3 Building the game

Start-up Scratch.

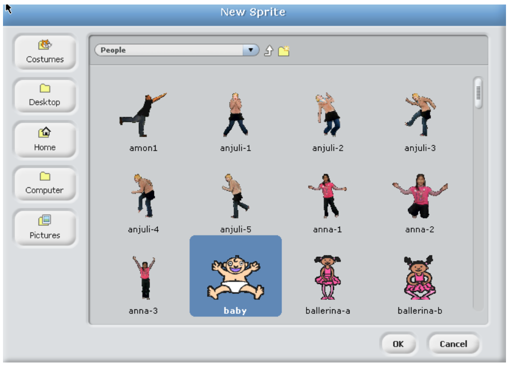
- Delete the Cat.

Add the Sprite,

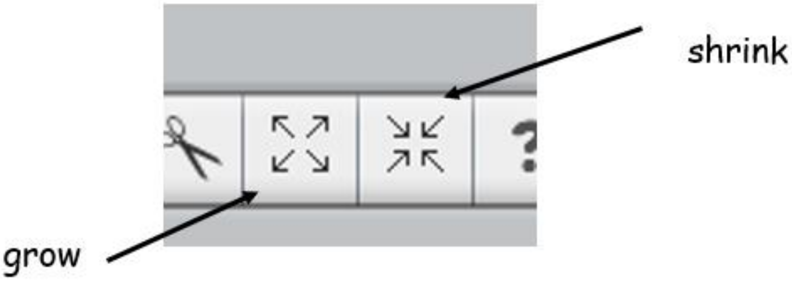
Click on the button with the picture of folder with a star in it.



- Add the Baby sprite

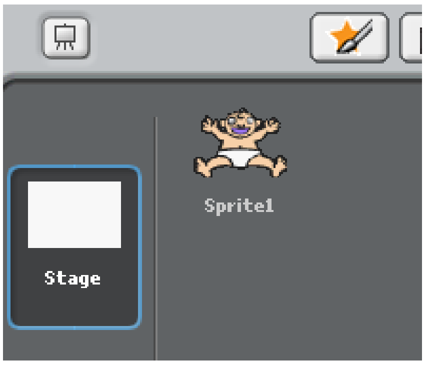


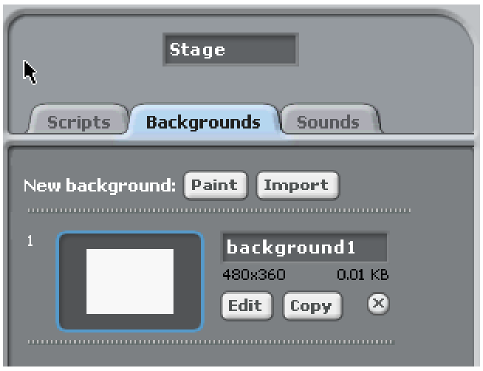
- Resize Your Sprite:



Add a Background

- Click on the Stage, in the bottom right area
- Click on the Backgrounds tab, in the center, click on the import button
- Pick a background, like bedroom1 in indoors



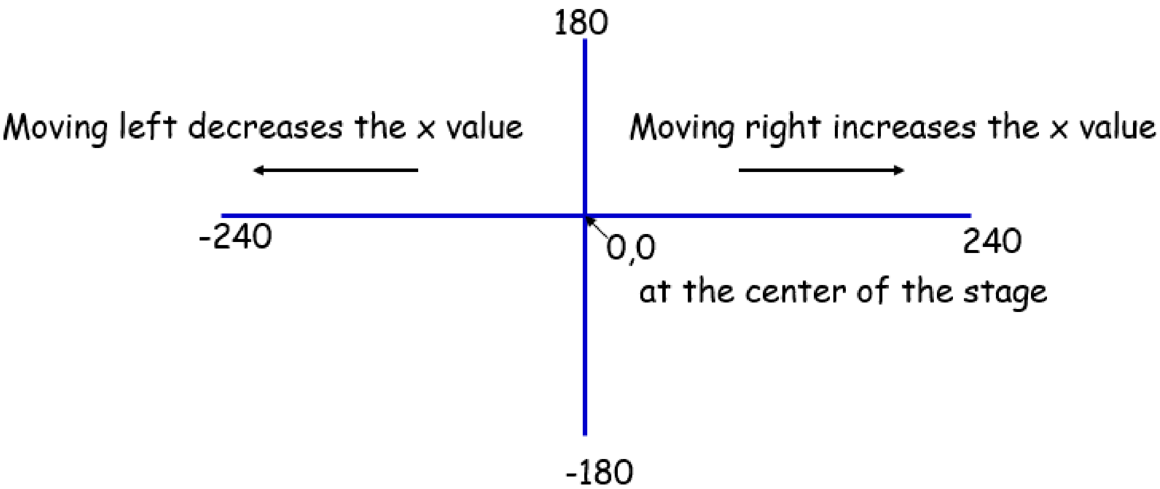


6.4 Event Handling

- We want to control the baby using the arrow keys
  - When we click the left arrow the baby should move left
  - When we click the right arrow the baby should move right
- This is a form of event handling
  - Responding to user actions like mouse clicks and key presses

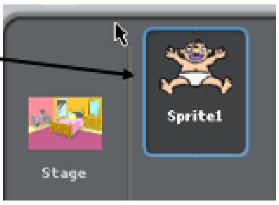
6.5 The Scratch Stage

The Scratch stage is 480 pixels wide and 360 pixels high.


180
Moving left decreases the x value
Moving right increases the x value
-240
0,0
240
at the center of the stage
-180

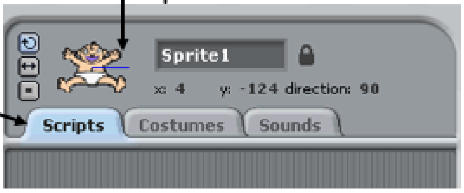
Programming the Sprite

- Click on the baby sprite in the view of the sprites and stage
  - Bottom right section



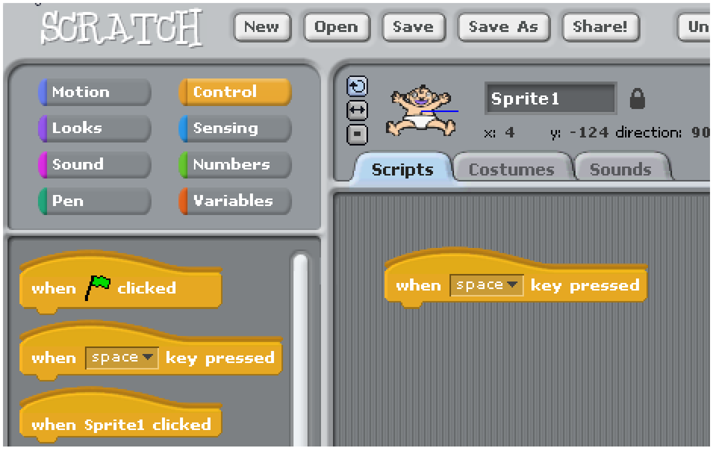
- Click on the Scripts tab
  - In the center area
  - This allows us to create scripts (programs) for the baby
  - Each sprite can have several scripts

This area shows the current sprite

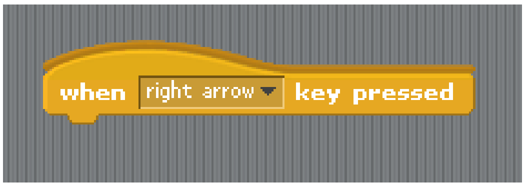


Respond to Arrow Keys

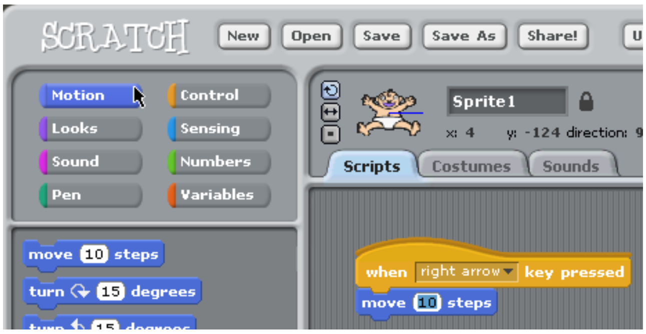
Click on Control (orange) and then drag out "when space key pressed"



- Respond to Right Arrow



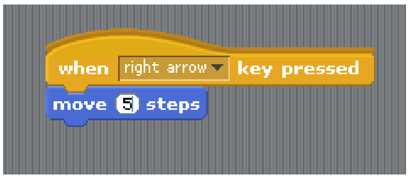
```
when right arrow key pressed
```



```
when right arrow key pressed
move 10 steps
```

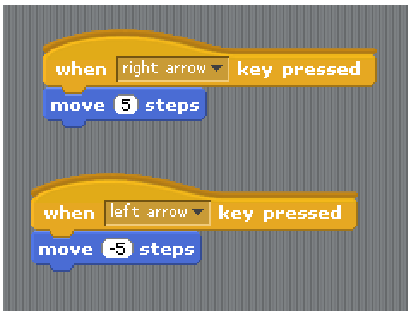
- Change the move amount

Click on the 10, it will highlight in blue, type 5 and press enter.



```
when right arrow key pressed
move 5 steps
```

- Respond to Arrow Keys
- Click on Control (orange)
- Drag out "when space key pressed"
- Change "space" to "left arrow"
- Click on Motion
- Drag out "move 10 steps"
- Change it to -5 (to move left)
- Click on the stage and try out the left and right arrow keys
    - Does the sprite leave the window?

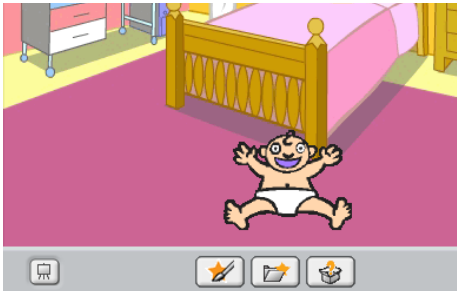


```
when right arrow key pressed
move 5 steps

when left arrow key pressed
move -5 steps
```

Paint a Ball

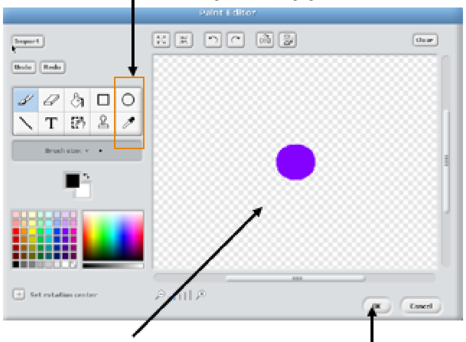
Click on the paint brush and star, It will say "Paint new sprite" if you hover over it.



Draw the Ball

- Click the circle tool
  - and then use the eyedropper to pick a color
  - and then click in the drawing area and drag to create the ball

Circle tool and eyedropper

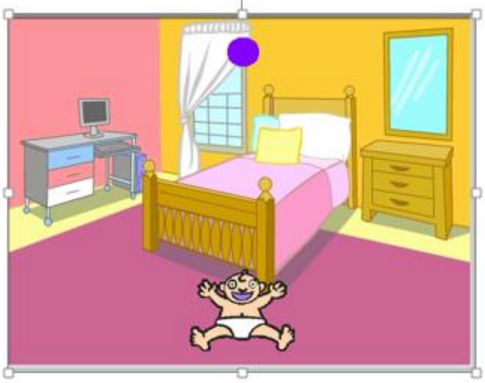


Drawing area

Click ok when done

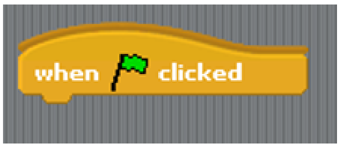
Size the ball as desired and move it to the top

- Click and drag the ball to the top of the window



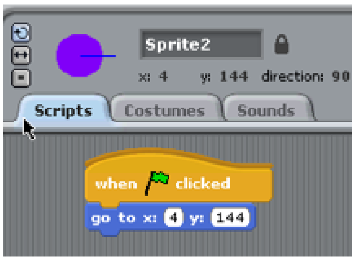
Make the Ball Fall

- When the green flag is clicked we want the ball to always start at the top and fall down
  - Click on Control (orange)
  - Drag out "When green flag clicked"



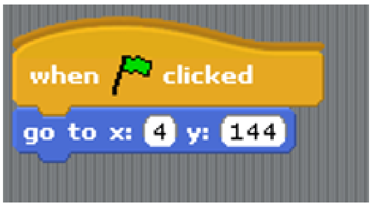
Start the Ball

- Click on Motion (blue)
- Drag out “go to x # y # “
  - this will always start the ball at its current position (Scratch doesn’t automatically put it back for you).



Sequential Execution

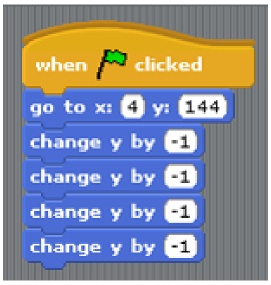
- One block is executed after the other
- In order from top to bottom
- When the green flag is clicked
  - the ball will go to the specified x and y location



```
when ⚑ clicked
go to x: 4 y: 144
```

Loops

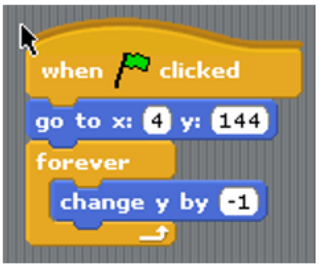
- We want the ball to continue to move down unless the baby catches it
- How do we make this happen?
  - We could use lots of blocks one after the other
  - But, that would be slow and repetitive
- We need a way to repeat a block or set of blocks
  - This is called a loop or iteration



```
when ⚑ clicked
go to x: 4 y: 144
change y by -1
change y by -1
change y by -1
change y by -1
```

Make the ball fall

- Click on Control (orange)
  - drag out "forever"
- Click on Motion (blue)
  - drag out “change y by 10”
  - Change it to -1



```
when ⚑ clicked
go to x: 4 y: 144
forever
    change y by -1
```

Catch the ball!

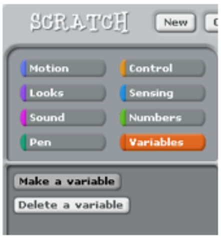
- If the ball touches the baby then it is caught
- Let's track how many balls we have caught with a score
  - We will increment the score each time we catch a ball

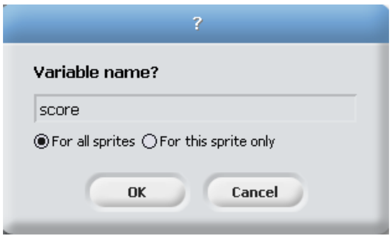
Variables

If we are going to keep track of the score, we want something to hold the current score and we want to be able to change the score. We want the value to change or vary, this is called a variable.

Track the score

- When we start the game set the score to 0
- Click on Variables (red)
- Click on Make a Variable
- Name it score





?

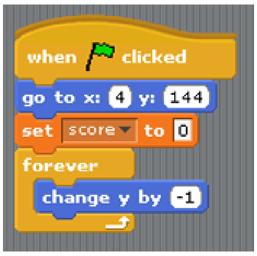
Variable name?

score

● For all sprites ○ For this sprite only

Set score to 0

- In the ball script
- Drag the “forever” down
- Drag out "set score to 0“
- Drag the “forever” back up
- Notice the score showing on the window



```
when ⚑ clicked
go to x: 4 y: 144
set score to 0
forever
    change y by -1
```

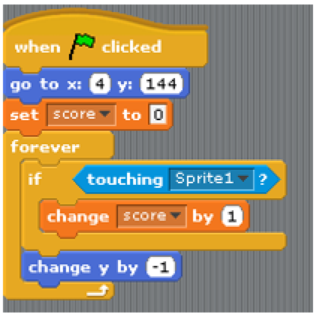
```
score 0
```

Conditionals

We want to increase the score if the baby caught the ball, so this action will only occur only if some condition is true. This is called a conditional or an “if”.

Did we catch the ball?

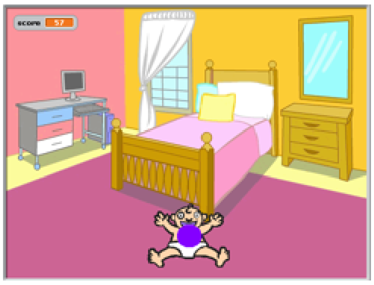
- From Control drag out an “if”
- Check if the ball is touching the baby
    - Get this in Sensing
- If this is true increment the score
    - From Variables



```
when ⚑ clicked
go to x: 4 y: 144
set score to 0
forever
    if touching Sprite1 ?
        change score by 1
    change y by -1
```

Increment the score

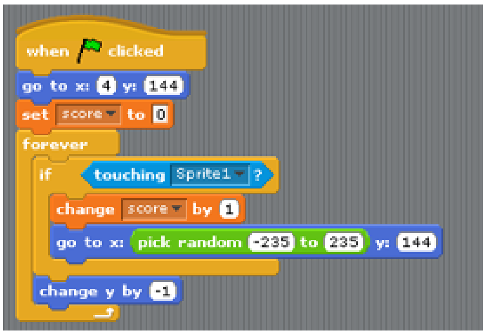
- Try it out!
    - is this what you expected?
- Computers do what you tell them to
    - Not what you want them to



Reset the Ball

- If we caught the ball
    - Increment the score
    - And move the ball to some random spot at the top of the window
    - So we don’t keep increasing the score

- Click on Number
    - drag out "pick random 1 to 10"
    - drop on the x value after “go to x:”
    - change the 1 to -235 and change 10 to 235
    - change the y value to match the y in the first “go to x # y #”



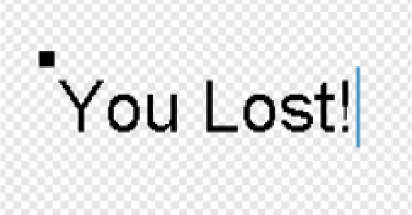
```
when ⚑ clicked
go to x: 4 y: 144
set score to 0
forever
    if touching Sprite1 ?
        change score by 1
        go to x: pick random -235 to 235 y: 144
    change y by -1
```

Adding Losing

- If the baby doesn't catch the ball it just gets stuck at the bottom of the screen
- Let's tell the player that he or she lost

Add a text sprite

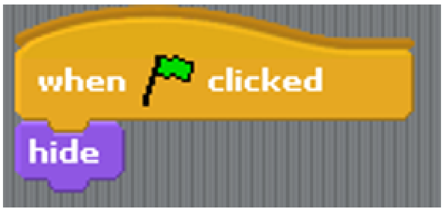
- Click on the Paint new sprite button
  - Click on the T for text
  - Pick the color
  - Modify the font size
  - Move the square to where you want the text
  - Type You Lost!



You Lost!

Hide the sprite

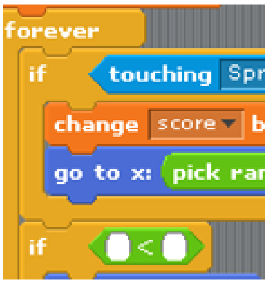
- We don’t want to tell the player that she lost when the game starts
  - So hide the message when the game starts
- Click on Control
  - drag out "when green flag clicked"
- Click on Looks
  - drag out “hide”



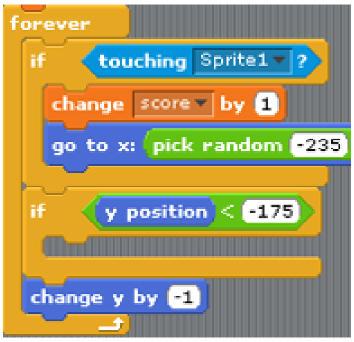
```
when 🏁 clicked
hide
```

Check if Lost

- If the y position gets near the bottom (near -180)
  - Drag out an if ,from Control
  - Drag out a blank < blank, From Numbers
  - Add a y position, from Motion
  - Type in -175



```
forever
  if touching Spr
    change score b
    go to x: pick ran
  if ( ) < ( )
```



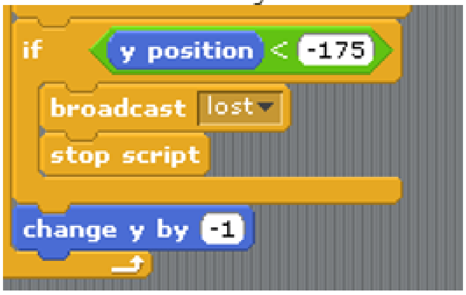
```
forever
  if touching Sprite1 ?
    change score by 1
    go to x: pick random -235
  if y position < -175

  change y by -1
```

Sprites pass messages

Broadcast a message

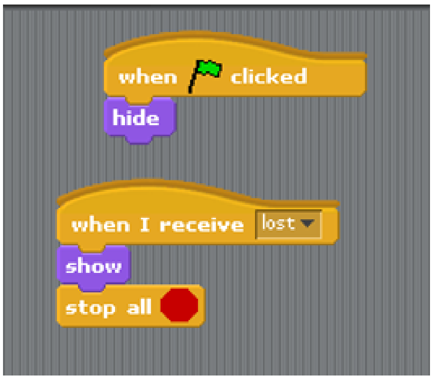
- Sprites communicate by passing messages
  - One sprite broadcasts the message
  - Other sprites can listen for it and react to it when they receive it
  - Click on Control
    - drag out "broadcast blank"
    - click on the drop down arrow next to new – name it lost
    - Add “stop script”
      - to stop the forever loop



```
if <(y position) < -175>
    broadcast lost
    stop script
change y by -1
```

Receive Lost

- Click on the text sprite
- Click on Control
  - drag out "when I receive blank"
  - click on the down arrow and select lost
- Click on Looks
  - drag out “show”
- Click on Control
  - drag out “stop all”
    - to stop all scripts

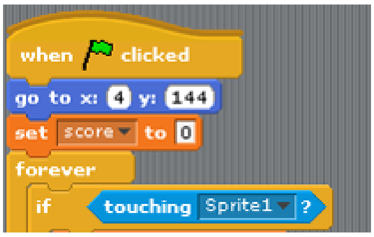


```
when ⚑ clicked
hide

when I receive lost
show
stop all
```

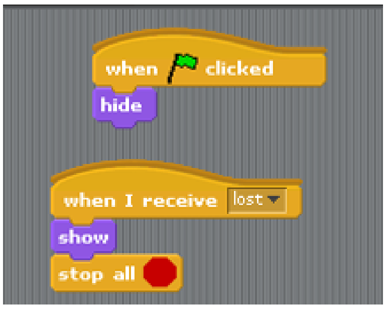
Parallel Execution

- We have several things happening at the same time
  - when the green flag is clicked



```
when ⚑ clicked
go to x: 4 y: 144
set score to 0
forever
    if <touching Sprite1 ?>
```

- This is called parallel execution
  - More than one thing happening at a time



```
when ⚑ clicked
hide

when I receive lost
show
stop all
```

Test your game

- Click the green flag
- If you want, adjust the speed of the ball, Increase the amount it changes in y
- Modify the sprites using the “Costume” tab
- Save your game with the “Save” button

6.6 Concept Summary

- **Variables**
  - can hold values and can change value
- **Forever loops**
  - repeat all the commands inside of them one at a time until the script is stopped or all scripts are stopped
- **Conditionals – ifs**
  - only execute the body of the if when the condition is true
- **Sprites can pass messages**
  - and receive them
- **Sprites can react to events**
  - like clicking the green flag and pressing the left or right arrow keys
- **Sprites can have several scripts**, costumes, and sounds
- Things can happen one after the other – **sequential execution** or at the same time – **parallel execution**

LECTURE [8]

7. App Inventor

7.1 Introduce App Inventor & showing a video about it.

- Give students a chance to talk amongst themselves to figure out what it does.
- Talk about the apps that can be built with App Inventor.

Why Learn App Inventor?

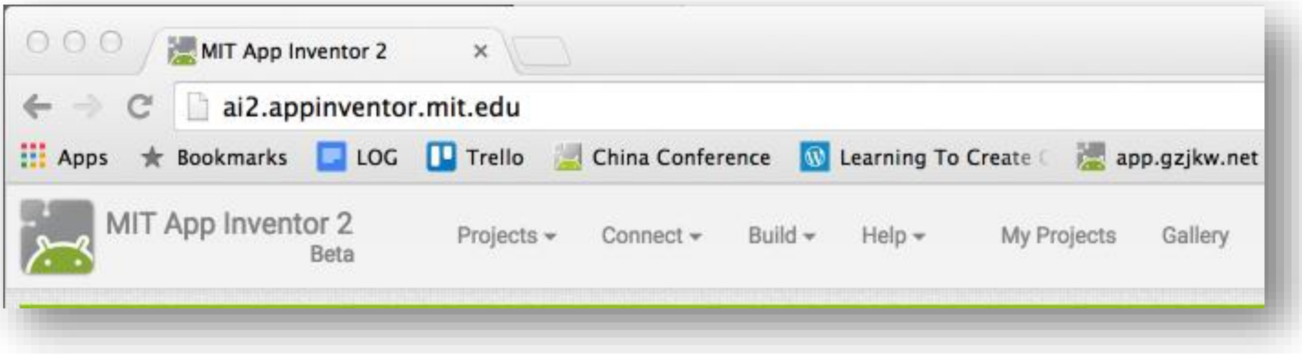
1. Software is in every *walk* of life. Programming is becoming part of many jobs.
2. Explore mobile computing
3. Practical Skills -- web, math, media
4. Prep for learning Java, Python, Javascript, etc.
5. Creativity, entrepreneurship, idea formulation
6. Problem-Solving skills

Video: App Inventor- I Have a Dream

https://www.youtube.com/watch?v=HbntNGGXamw

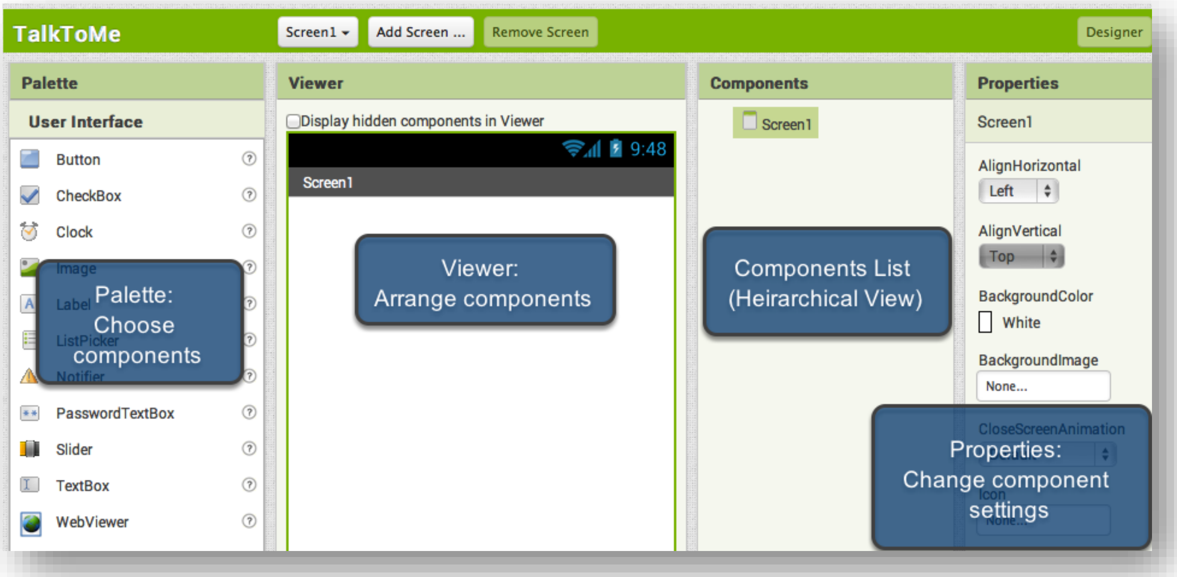
Other video: https://youtu.be/ThZ5ZkY0IIU

6.2 Prepare the lab, Setup stage.



6.3 Introduce UI of app inventor

Introduce UI of appinventor.mit.edu & simple tutorial to build the first app.



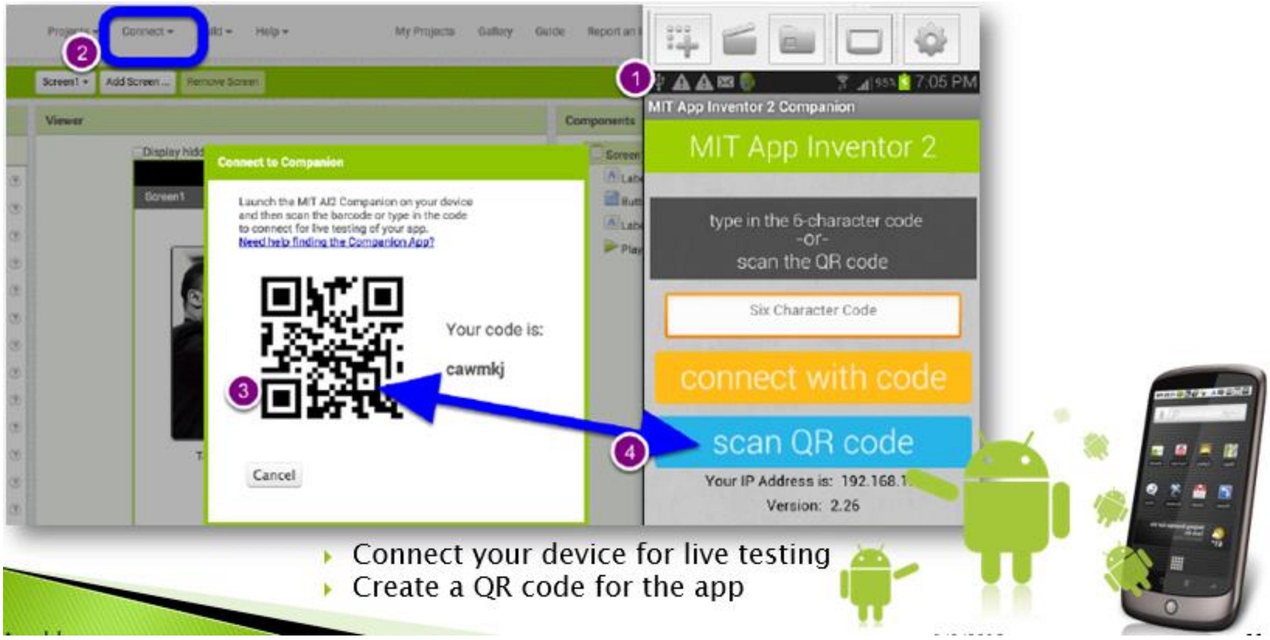
Palette: Choose components

Viewer: Arrange components

Components List (Heirarchical View)

Properties: Change component settings

6.4 How students will test their apps on a device or emulator



Connect to Companion

Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app.
Need help finding the Companion App?

Your code is:
cawmkj

type in the 6-character code
-or-
scan the QR code

Your IP Address is: 192.168.1.

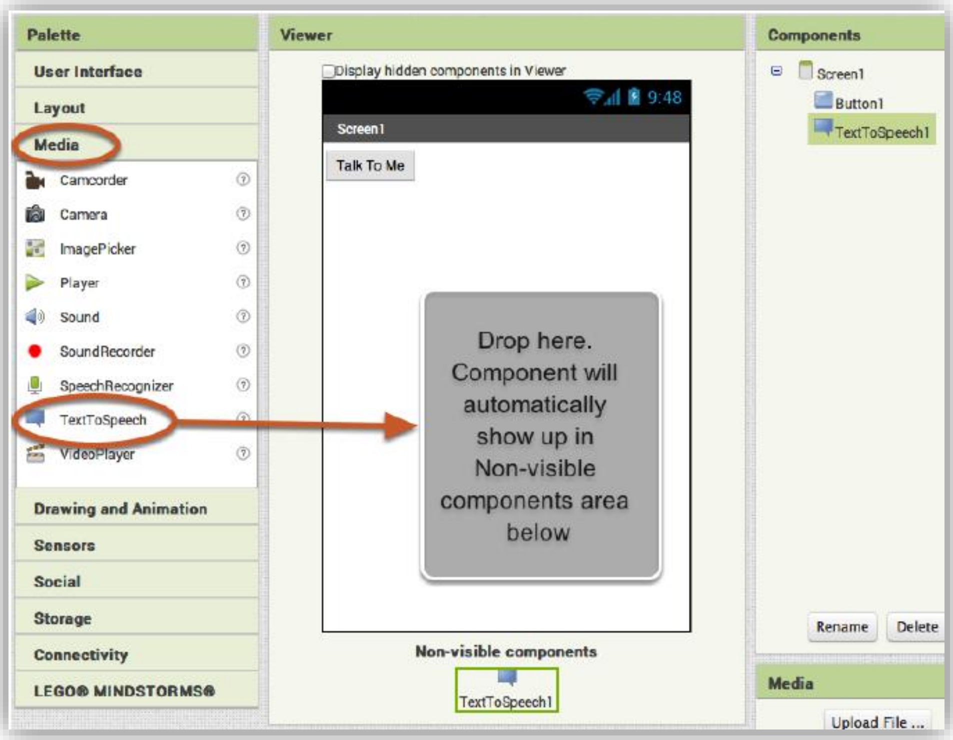
Version: 2.26

- Connect your device for live testing
- Create a QR code for the app

Lecture [9]

App Inventor, Continue

6.5 Build more mobile Apps and run on deceives.


Palette
User Interface
Layout
Media
Camcorder
Camera
ImagePicker
Player
Sound
SoundRecorder
SpeechRecognizer
TextToSpeech
VideoPlayer
Drawing and Animation
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Viewer
☐Display hidden components in Viewer
9:48
Screen1
Talk To Me
Drop here. Component will automatically show up in Non-visible components area below
Non-visible components
TextToSpeech1
Components
Screen1
Button1
TextToSpeech1
Rename
Delete
Media
Upload File ...