

CoderDojo Athenry

"Above all, be cool"



Every week:

- ✓ Sign in at the door

If you are new:

- ✓ Fill in Registration Form
- ✓ Ask a Mentor how to get started

Make sure you are on the Athenry Parents/Kids Google Group: email coderdojoathenry@gmail.com

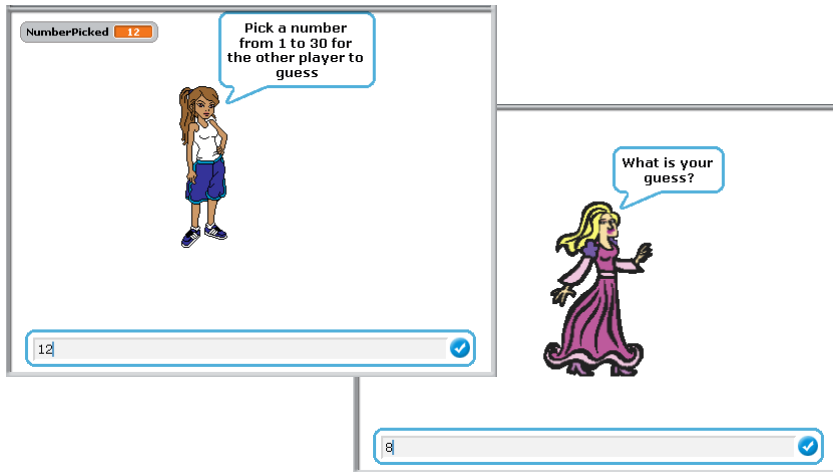
CoderDojo Athenry

Scratch Beginners



Code and notes by Michael Madden, 2013

Today's Ninja Challenge: Make a 2-Player Guessing Game!



Today's Big Ideas

Design of
2-Player Game

Communication
with variables

Also: Loops
& Decisions

Broadcasts

The Coding Plan

Write a single-player program to guess numbers like we did before

Write a second program for a "boss" to pick the number

Set up a mesh between the two programs

Use broadcasts and sensing to get them to work together

What We Need To Figure Out

Pick a random number and store it;
Ask for guess and store the answer

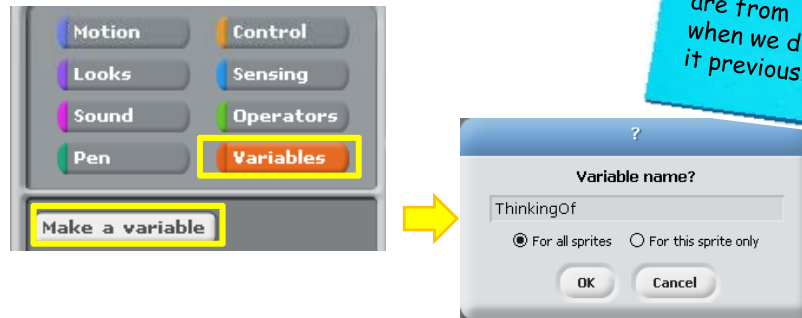
Test if it's right, too high or too low;
Display messages

Give them 5 guesses

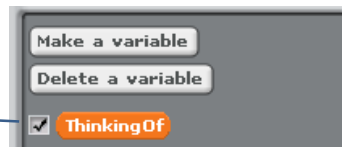
Write a "Boss" program to pick the number and broadcast a Start message

Use Mesh Network to get the two programs to work together

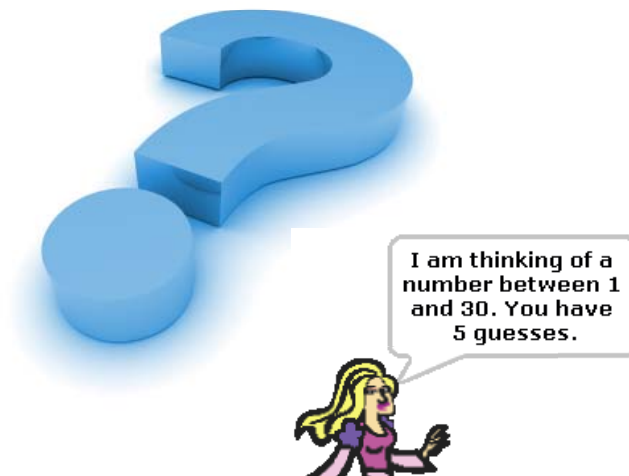
Create a Variable to Hold a Random Number



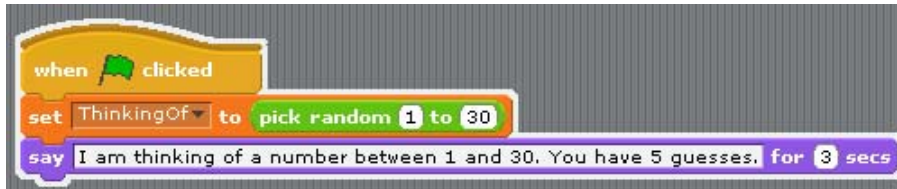
Tip:
Clear this checkbox
so it is not shown
on screen



When Flag is Clicked: Pick Random Number, Display "I am thinking of ..."



**When Flag is Clicked:
Pick Random Number,
Display "I am thinking of ..."**

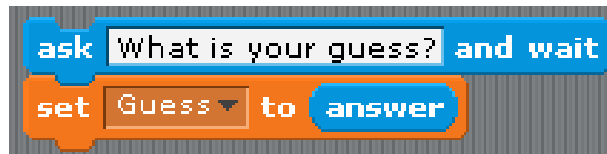


**Ask Player to Guess It,
Store Answer**



Ask Player to Guess It, Store Answer

Need another variable!
I've called it **Guess**.



Test whether Guess is Right, Too High, Too Low



Test Whether Guess is Right, Too High, Too Low

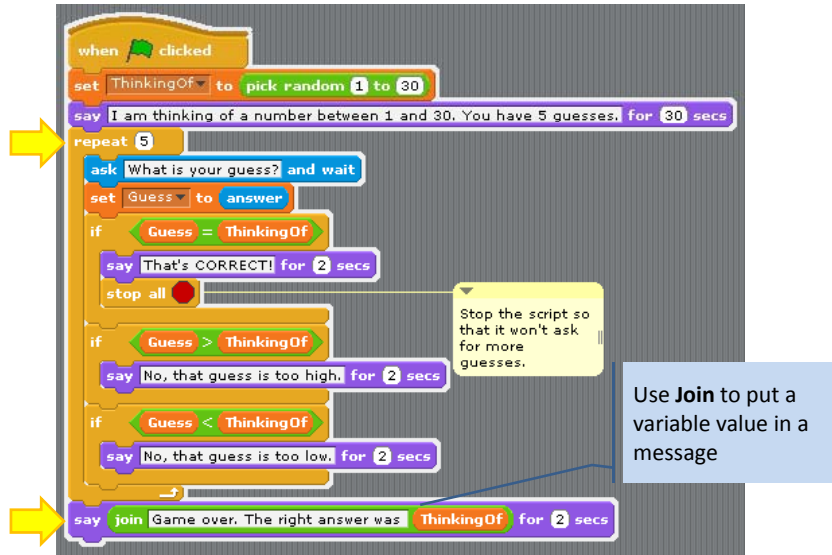
```
if <math>Guess = ThinkingOf</math>
  say That's CORRECT! for 2 secs
  stop all
if <math>Guess > ThinkingOf</math>
  say No, that guess is too high. for 2 secs
```

You need to complete this ...

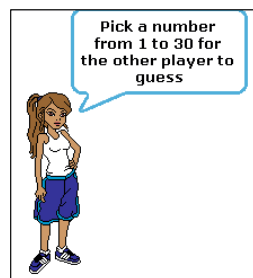
Give Player Five Guesses



Give Player Five Guesses



Add a Leader to Pick Number: The Guesser will be the Follower



The **Leader** will ask you to enter number, then **broadcast** a message so the follower can start guessing



When it **receives** the broadcast, the **Follower** will **sense the number** and then start the guessing game

First, A Recap on Networking Between Computers in Scratch



Then all of them can **communicate**:
Receive each other's **broadcasts**,
Sense each other's **variables**.

Meshes are a secret **hidden feature**!! Tricky to set up first time!

1: Shift-Click on Top of R

**2: Select "turn fill screen off":
White area appears at bottom**

**3: Click in white area;
select "open"**

4: Then select "browser"

White area

*SAVE your program first:
at the last step, Scratch
shuts down.*

Set Mesh Up, Part 2

The screenshot shows the Scratch System Browser with the following structure:

Scratch-Objects	OffscreenWorldMorph	-- all --	aboutScratch
Scratch-Blocks	ScratchFrameMorph	initialization	addServerCommandsTo:
Scratch-Execution Engine	ScratchLibraryMorph	accessing	addSpriteMorph
Scratch-Object ID	ScratchPresenterMorph	menu/button actions	allProjectMedia
Scratch-UI-Dialogs	ScratchScriptEditorMorph	geometry	canonicalizeImagesQuality
Scratch-UI-Panes	ScratchScriptsMorph	drawing	canonicalizeSoundsBits:sa
Scratch-UI-Watchers	ScratchViewerMorph	event handling	compressImages
Scratch-UI-Support	SensorBoardMorph	stepping	compressSounds
Scratch-Paint		dropping/grabbing	developersMenu
Scratch-Sound		view mode	editMenu:
Scratch-Translation		other	editNotes
Scratch-Networking		startup	enableRemoteSensors
ScratchLibraries		etc. add/delete	swtScratchControl

The code for `addServerCommandsTo:` is shown below:

```

addServerCommandsTo: t1
| t2 t3 |
t2 ← false.
t2 ifTrue: [t ← self].
t1 addLine.
(workPane scratchServer notNil and: [workPane scratchServer sessionInProgress])
ifTrue:
    [t1 add: 'Show IP Address' action: #
    t3 ← workPane scratchServer isHost
    ifTrue: ['Stop Hosting'
    ifFalse: ['Leave Mesh'
    t1 add: t3 action: #exitScratchSessio
    ifFalse:
        [t1 add: 'Host Mesh' action: #startH
        t1 add: 'Join Mesh' action: #joinScr
    
```

5: Click: **Scratch-UI-Panes** then **ScratchFrameMorph** then **menu/button actions** then **addServerCommandsTo**

6: Find line: `t2 ← true.`
Change to `t2 ← false.`

7: Ctrl-Click in this area, select "accept" to save.

Enter your initials to confirm.

8: Press X to close System Browser.

Select **Yes** to save changes.



9: Shift-click on top of R again. Select "turn fill screen on" to get rid of white area.

10: Select "save image for end-user": this is you won't have to repeat these steps every time!

Now Activate a Mesh!

Pick a computer to be the **Host**.
Shift-Click on the **Share** menu and select "**Host Mesh**" from the extra menu items.
It will display an **IP address**: write it down!



On the other computers, Shift-Click on the **Share** menu and select "**Join Mesh**". You will have to enter the **IP address** of the Host.

When meshed, Scratch programs on the two computers can receive each other's **broadcasts** and sense each other's **variables**.

Code for the **Leader**: Very Short!



Ask for a number,
store it in 'NumberPicked',
and broadcast "Guess" to tell
the Follower to start guessing.

```

when space key pressed
ask Pick a number from 1 to 30 for the other player to guess and wait
set NumberPicked to answer
broadcast Guess
  
```

Code for **Follower**: Just Modify the Start of the Guessing Game



The follower now starts when it
receives **broadcast**,
senses value of 'NumberPicked',
and **says** a different message.
Everything else is the same.

Replace code
that was here

```

when I receive Guess
set ThinkingOf to NumberPicked sensor value
say The other player has picked a number between 1 and 30. You have 5 guesses. for 5 secs
repeat 5
ask What is your guess? and wait
  
```

More Features You Can Add...

1. Sound Effects
Go to sprite's Sound tab, import sounds
Add commands to play sounds
2. Timer
Add another variable called Timer.
Add another block:
When flag is clicked, set it to 60
Repeat until 0: change by -1, wait a second
Time up: stop all scripts.
3. Display the Result to the Leader
Broadcast when Follower is finished
Add code to Leader to see if right answer guessed
4. Animation
What can you think of?

At the End ...

Upload your project to the Scratch Website
user: **cdatheny** password: _____

Access it
from home

Improve it

Show your
friends!

