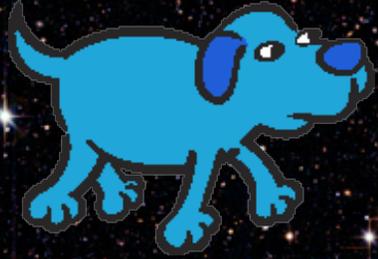
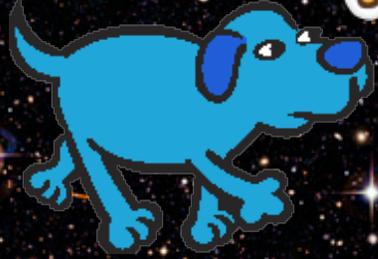


# Creating a Game with

# SCRATCH



LITTLE CRAB GAME



# FREE COMPUTER SCIENCE TOOLS

FEATURES

## Simplest tool Ages 5-15

- Drag and drop code blocks
- 2D graphics frameworks interaction

## Simpler tool Ages 8-22

- Drag and drop code blocks
- 3D graphics frameworks interaction

## Less simple Ages 12-22

- Interactive interpreter code typing
- Media computation helper classes

## Less simple Ages 13-25

- Type, compile, run, debug
- 2D gaming framework interaction via 5 Java classes

## Less simple Ages 15-25

- Type, compile, run, debug
- No default graphics framework environment

## Complex tool Ages 15-25

- Type, compile, run, debug
- No default graphics framework environment

## No/Any tool Ages 16+

- Type, compile, run, debug
- No default graphics framework environment

TOOLS



TEACHING CONCEPTS

- Sequence
- Iteration
- Conditional Logic
- Variables
- Data Structures (dynamic lists)
- Events Handling
- Parallel Execution
- Synchronization
- Random Numbers
- Boolean Logic
- Dynamic Interaction
- User Interface Design
- Publish projects as Java applets on [scratch.mit.edu](http://scratch.mit.edu)

- All Scratch concepts plus..
- Procedures and Functions
- Parameter Passing & Return Values
- Recursion
- Defining Classes of Objects
- Inheritance
- Text Input

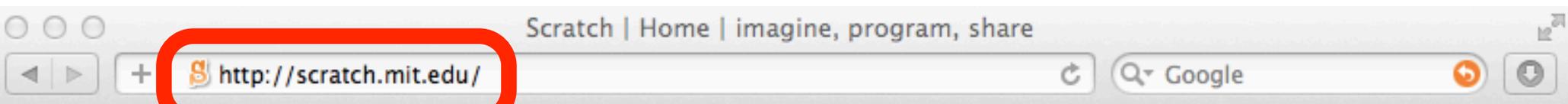
- All previous Alice concepts, excluding user interface design
- Manipulation of audio, images, video through media computation helper classes – see [mediacomputation.org](http://mediacomputation.org)
- All Java Programming Language Features Available
- Interpreted mode is great step from drag and drop code blocks (no errors possible) to type – compile – run – debug (all errors possible)

- All previous Alice concepts plus..
- 5 Java classes encapsulate 2D gaming and simulation concepts
- All Java Programming Language Features Available

- Simplest Java IDE
- All Java Programming Language Features Available

- Complex IDE
- Multiple programming languages available (Java, Ruby, Groovy, Python, PHP, JavaScript, etc.)
- Tooling covers mobile and embedded, enterprise, all avenues

- Programming Language common to Alice, Dr. Java, Greenfoot, BlueJ and NetBeans environments
- All Java Programming Language Features Available
- Tool / environment agnostic



# SCRATCH

imagine • program • share

[home](#) [projects](#) [galleries](#) [support](#) [forums](#) [about](#) [Language](#)

[Login](#) or [Signup](#) for an account

search

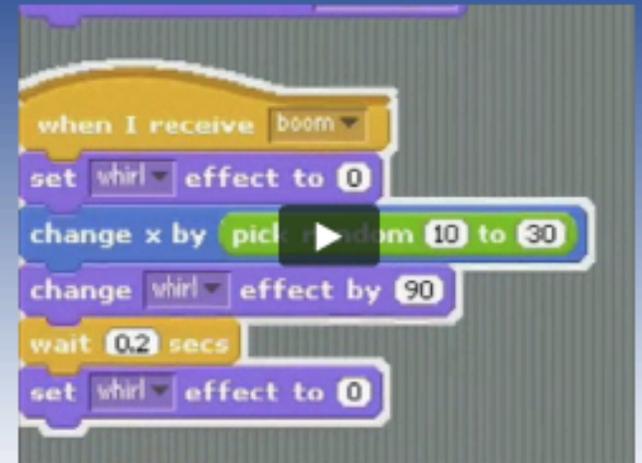
Create and share your own interactive stories, games, music, and art

[Check out](#) the 2,097,036 projects from around the world!

To create your own projects:



[Download Scratch](#)



# SCRATCH

is **free** at [http://](http://scratch.mit.edu)

[scratch.mit.edu](http://scratch.mit.edu) and runs on

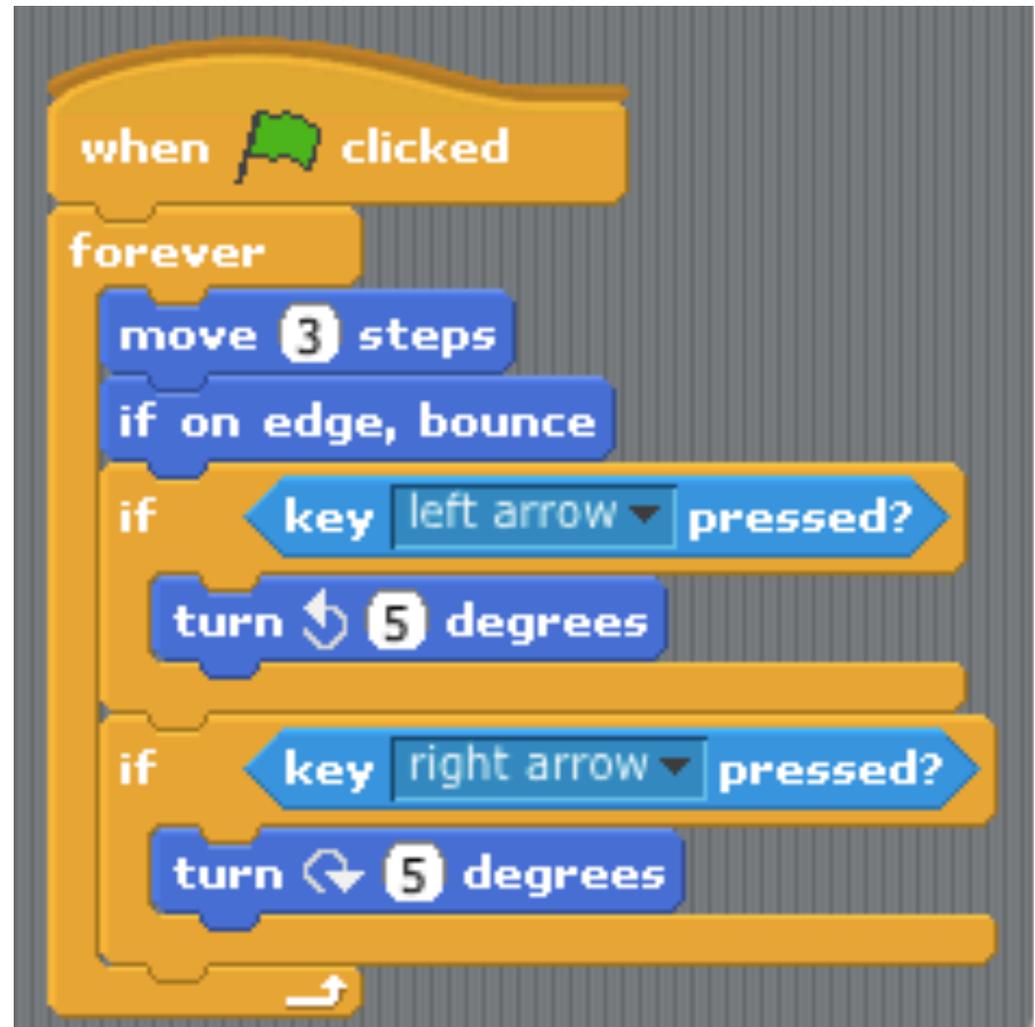
**Windows, Mac and Linux**

# Scratch Project: Little Crab

Phase	 Scratch	Concepts	Game
1. Get Moving	<b>Motion</b> , <b>Control</b>	movement, turning, event handling	get that crab moving!
2. Worms	<b>Looks</b> , <b>Sound</b> , <b>Control</b> , <b>Sensing</b>	collision, visibility, audio	give the crab something to eat!
3. Lobsters	<b>Motion</b> , <b>Looks</b> , <b>Sound</b> , <b>Control</b> , <b>Sensing</b> , <b>Operators</b>	collision, visibility, audio	add some crab predators!
4. Animation	<b>Looks</b> , <b>Control</b> , <b>Operators</b> , <b>Variables</b>	animation	make it look good!
5. Game Over	<b>Motion</b> , <b>Looks</b> , <b>Sound</b> , <b>Control</b> , <b>Sensing</b> , <b>Operators</b> , <b>Variables</b>	messaging, variables	when crab eats the worms or lobsters eats the crab - game over!

**The images and sounds  
used in this project are  
downloadable as a zip file  
from <http://bit.ly/littlecrab>**

# Scratch Phase 1: Get Moving



<http://scratch.mit.edu/projects/dang/2103788>

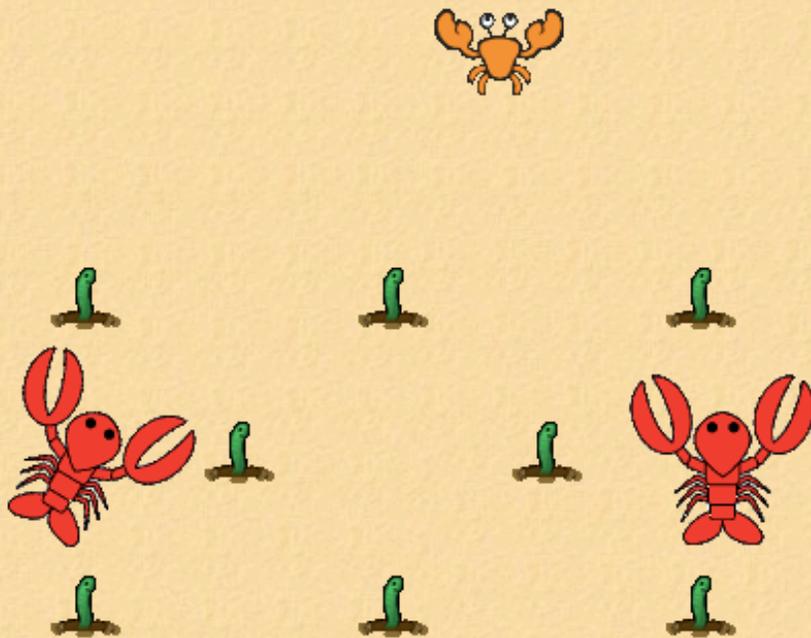
# Scratch Phase 2: Add Food



<http://scratch.mit.edu/projects/dang/2105788>

# Scratch Phase 3: Predators

LittleCrab\_Phase03



x: 32 y: -446

New sprite:



Stage



when  clicked

go to x: 150 y: -150

point in direction 0

forever

move 2 steps

turn  pick random -15 to 15 degrees

if on edge, bounce

<http://scratch.mit.edu/projects/dang/1400765>

# Scratch Phase 3: Predators

The image shows a Scratch project window titled "LittleCrab\_Phase04". The stage is a light brown textured area with a small orange crab at the top center, several green worms scattered across the bottom, and two red lobsters on the left and right sides. The script area on the right contains the following code:

```
when green flag clicked
  go to x: 0 y: 120
  point in direction 90
  show
  forever loop
    move 3 steps
    if on edge, bounce
    if key left arrow pressed?
      turn 5 degrees
    if key right arrow pressed?
      turn 5 degrees
    if touching lobster01?
      play sound au
      hide
    if touching lobster02?
      play sound au
      hide
```

The code is designed to move the crab around the stage, bounce off edges, and respond to keyboard input. It also includes collision detection with two lobster sprites, labeled "lobster01" and "lobster02". When the crab touches either lobster, it plays a sound and hides. The collision detection blocks are highlighted with a red box in the image.

At the bottom of the interface, the "New sprite" area shows a selection of sprites: a crab, several worms, and two lobsters. The "Stage" area shows the current scene with worms and lobsters.

<http://scratch.mit.edu/projects/dang/2106008>

# Scratch Phase 4: Animation



<http://scratch.mit.edu/projects/dang/2108057>

# Scratch Phase 4: Animation

The screenshot displays the Scratch interface for a project titled "LittleCrab\_Phase...". The character is named "lobster01" and is currently at coordinates (150, -77) facing direction -2. The "Costumes" tab is active, showing four costumes: "lobster1-a", "lobster1-b", "lobster1-c", and "lobster1-d". The stage shows a crab character and several worms. The script area contains the following code:

```
when green flag clicked
  go to x: 150 y: -150
  set animation to 1
  forever loop
    if animation mod 4 = 0
      next costume
      set animation to 1
    else
      change animation by 1
  turn pick random -15 to 15 degrees
  if on edge, bounce
```

The code block for the animation logic is highlighted with a red border. It uses a "when green flag clicked" event to initialize the character's position and an "animation" variable to 1. A "forever" loop follows, where an "if" statement checks if the animation variable modulo 4 equals 0. If true, it changes the costume and resets the animation variable to 1. If false, it increments the animation variable by 1. The loop also includes a "turn" block with a random angle and an "if on edge, bounce" block.

<http://scratch.mit.edu/projects/dang/2108057>

# Scratch Phase 5: Game Over

The image shows the Scratch IDE interface for a game titled "LittleCrab\_Phase...". The stage displays the text "GAME OVER" in large red letters, with small crab icons integrated into the letters. The score is 0. The script area shows the following code:

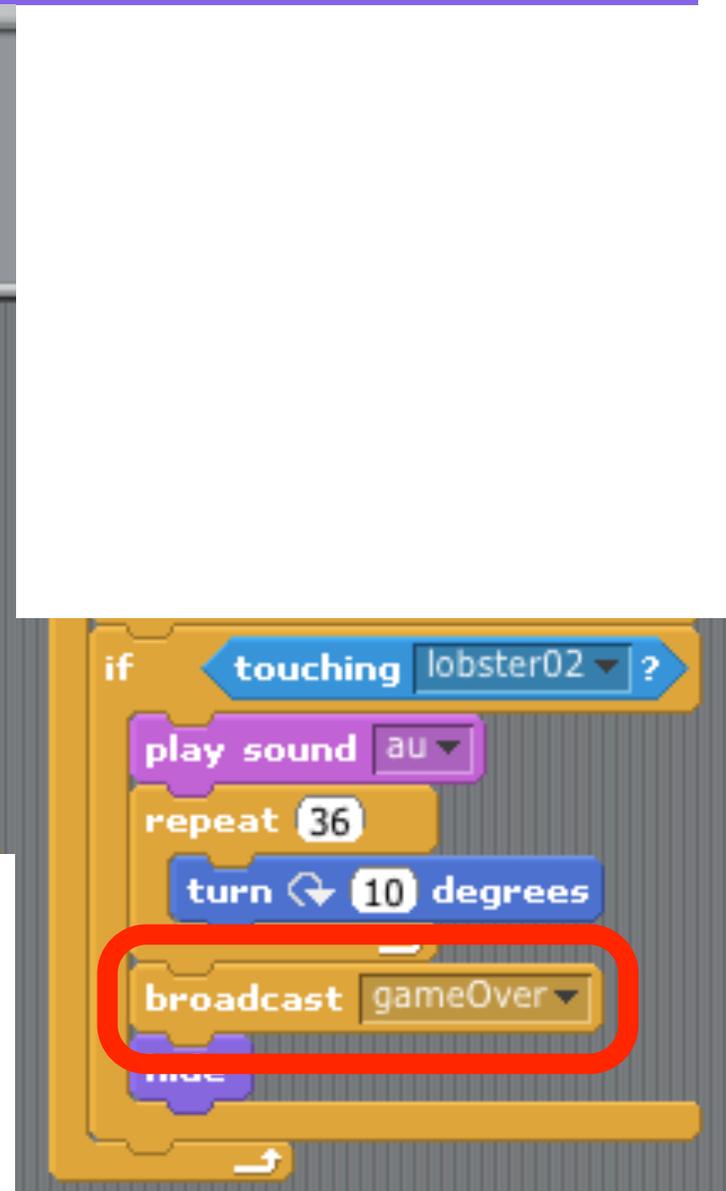
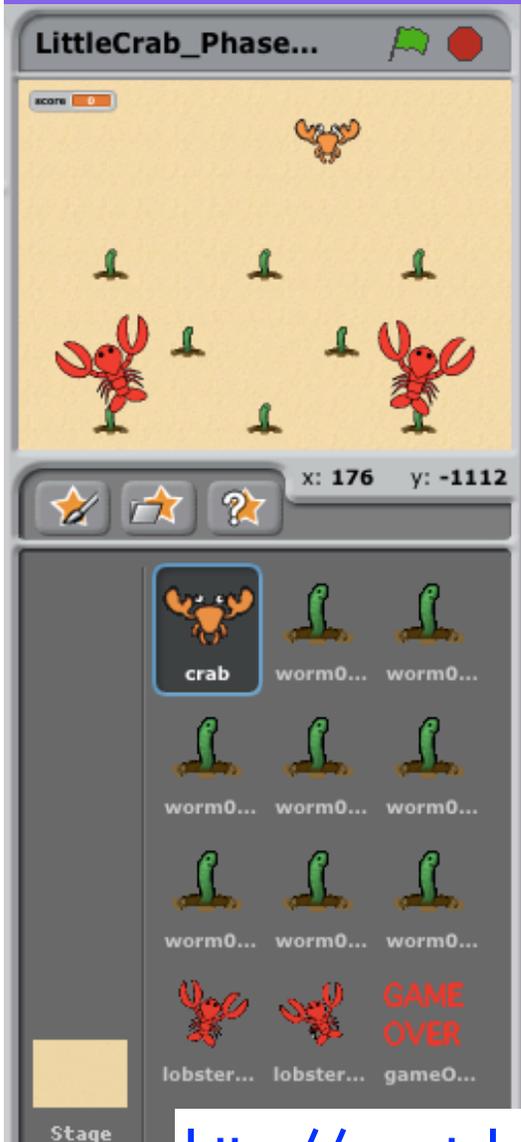
```
when green flag clicked
  hide

when I receive gameOver
  show
  forever loop
    change color effect by 3
```

The costume area shows a "gameOver" costume selected, which is a red "GAME OVER" text with a crab icon. The stage area shows a grid of costumes: a crab, two worms, three worms, three worms, and two lobsters, with the "gameOver" costume selected.

<http://scratch.mit.edu/projects/dang/2108271>

# Scratch Phase 5: Game Over



<http://scratch.mit.edu/projects/dang/2108271>

# Scratch Phase 5: Game Over

The image shows a Scratch project titled "LittleCrab\_Phase...". The stage displays a game scene with a worm character, several green worms, and two red crabs. A score indicator shows 0. The worm character's properties are visible: name "worm01", x: 0, y: 0, direction: 90. The code block for the worm character is as follows:

```
when green flag clicked
  show
  go to x: 0 y: 0
  forever loop
    if touching crab?
      play sound slurp
      hide
      change score by 1
```

The "change score by 1" block is highlighted with a red circle. The sprite palette shows a crab, worm01, worm02, worm03, worm04, worm05, worm06, worm07, worm08, worm09, lobster01, lobster02, and game01.

<http://scratch.mit.edu/projects/dang/2108271>