

Functions for Character Animation

(Also available for WeScheme)

Students define functions that control the movement of the target and danger in their games.

Lesson Goals	<p>Students will be able to:</p> <ul style="list-style-type: none">• Apply the <i>Design Recipe</i> to create a <i>function</i> given the constraints of a word problem.• Explain the basics of animation.															
Student-Facing Lesson Goals	<ul style="list-style-type: none">• I can use the Design Recipe to make a function.• I can describe how animation works.															
Materials	<ul style="list-style-type: none">• Lesson Slides• Danger and Target Movement (Page 60)															
Preparation	<ul style="list-style-type: none">• Make sure all materials have been gathered• Decide how students will be grouped in pairs															
Key Points for the Facilitator	<ul style="list-style-type: none">• Encourage students to take their time in understanding <i>why</i> we want to fix <code>update-danger</code> and <code>update-target</code> .• Students might be confused as to <i>how</i> the animation is working. The <code>make-game</code> function at the bottom of the file has many inputs - including <code>update-danger</code> and <code>update-target</code> . <code>make-game</code> takes in all those inputs, including the functions we'll write, and creates the interactive window that we see when we press the Run button!															
Language Table	<table><tr><th>Types</th><th>Functions</th><th>Values</th></tr><tr><td>Number</td><td><code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>num-expt</code> , <code>num-sqr</code> , <code>num-sqrt</code></td><td><code>4</code> , <code>-1.2</code> , <code>2/3</code> , <code>pi</code></td></tr><tr><td>String</td><td><code>string-length</code> , <code>string-repeat</code> , <code>string-contains</code></td><td><code>"hello"</code> , <code>"91"</code></td></tr><tr><td>Boolean</td><td><code><</code> , <code><></code> , <code><=</code> , <code>>=</code> , <code><</code> , <code>></code> , <code>==</code> , <code><></code> , <code>>=</code></td><td><code>true</code> , <code>false</code></td></tr><tr><td>Image</td><td><code>star</code> , <code>triangle</code> , <code>circle</code> , <code>square</code> , <code>rectangle</code> , <code>rhombus</code> , <code>ellipse</code> , <code>regular-polygon</code> , <code>radial-star</code> , <code>text</code> , <code>overlay</code> , <code>above</code> , <code>beside</code> , <code>rotate</code> , <code>scale</code> , <code>flip-horizontal</code> , <code>flip-vertical</code></td><td></td></tr></table>	Types	Functions	Values	Number	<code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>num-expt</code> , <code>num-sqr</code> , <code>num-sqrt</code>	<code>4</code> , <code>-1.2</code> , <code>2/3</code> , <code>pi</code>	String	<code>string-length</code> , <code>string-repeat</code> , <code>string-contains</code>	<code>"hello"</code> , <code>"91"</code>	Boolean	<code><</code> , <code><></code> , <code><=</code> , <code>>=</code> , <code><</code> , <code>></code> , <code>==</code> , <code><></code> , <code>>=</code>	<code>true</code> , <code>false</code>	Image	<code>star</code> , <code>triangle</code> , <code>circle</code> , <code>square</code> , <code>rectangle</code> , <code>rhombus</code> , <code>ellipse</code> , <code>regular-polygon</code> , <code>radial-star</code> , <code>text</code> , <code>overlay</code> , <code>above</code> , <code>beside</code> , <code>rotate</code> , <code>scale</code> , <code>flip-horizontal</code> , <code>flip-vertical</code>	
Types	Functions	Values														
Number	<code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>num-expt</code> , <code>num-sqr</code> , <code>num-sqrt</code>	<code>4</code> , <code>-1.2</code> , <code>2/3</code> , <code>pi</code>														
String	<code>string-length</code> , <code>string-repeat</code> , <code>string-contains</code>	<code>"hello"</code> , <code>"91"</code>														
Boolean	<code><</code> , <code><></code> , <code><=</code> , <code>>=</code> , <code><</code> , <code>></code> , <code>==</code> , <code><></code> , <code>>=</code>	<code>true</code> , <code>false</code>														
Image	<code>star</code> , <code>triangle</code> , <code>circle</code> , <code>square</code> , <code>rectangle</code> , <code>rhombus</code> , <code>ellipse</code> , <code>regular-polygon</code> , <code>radial-star</code> , <code>text</code> , <code>overlay</code> , <code>above</code> , <code>beside</code> , <code>rotate</code> , <code>scale</code> , <code>flip-horizontal</code> , <code>flip-vertical</code>															

Click here to see the prior unit-based version

[Click here to see the prior unit-based version](#)

Glossary

coordinate :: a number or set of numbers describing an object's location

design recipe :: a sequence of steps that helps people document, test, and write functions

function :: a mathematical object that consumes inputs and produces an output

Animation

45 minutes

Overview

Students connect the behavior of functions with changing coordinate values, ultimately leading to animation.

Launch

Students should have their computer, contracts page, and pencil. Students should have their own **game file** open in a separate window or tab.

- How does a flip-book animation work? *Each page of the book is slightly different, and the pages go so fast that the motion looks smooth.*
- Why do we see movement from still images?
Our eyes fill in the gaps between rapidly changing images.
- How might this apply to our game?
*If we change image **coordinates** a little bit at a time, they will appear to move.*

Draw a number line on the board, running from 0 to 1000 (you can also lay tape on the floor, or use a tile floor as a coordinate plane!). Select 2 student volunteers - one to be **TARGET**, one to be **DANGER**. Start with just **TARGET**.

- Have the class select a starting x- and y-coordinate for the **TARGET**, and have the volunteer move to that position on the number line or coordinate plane.
- The **TARGET** character moves by 50 (pixels) on each frame of the game.
- When they hear "update target" followed by their current location, the **TARGET** takes a step in the negative direction, moving down the x-axis by 50 (pixels).
- We make **TARGET** move by calling out `update-target(300, 200)`, `update-target(250, 100)`, etc.

How quickly could I get **TARGET** to move across the classroom?

After practicing with **TARGET**, add **DANGER** in.

- **DANGER** takes a step in the *positive direction* when they hear "update danger" followed by their current x-coordinate.
- We make **DANGER** move by calling out `update-danger(40, 92)`, `update-danger(39, 18)`, etc.
- On a standard number line, if the **DANGER** is moving to the right, is its x-coordinate increasing or decreasing?

Practice this a few times with your volunteer, asking the class what their new x-coordinate is each time. Then have the other students call the update-danger function.

- **What did you notice about the movement of **TARGET** and **DANGER**? What was changing about them?**
Answers will vary: they were moving horizontally, their x-coordinates were changing, they were not moving smoothly, etc.
- **What jobs could we hand over to the computer to make it possible for us to play the game?** *The computer could handle automatically moving **TARGET** and **DANGER**, then we could control the movement of **PLAYER**.*

Investigate

- Have students examine the `update-danger` function in their Game Starter File, identify the contract, and interpret what the function is currently doing.

- Guide students as they complete the first word problem on [Danger and Target Movement \(Page 60\)](#), and transfer the code to their Game Starter File.

When students press the Run button, the working `update-danger` function should automatically move the `DANGER` image across the screen!

Have students complete the second word problem on [Danger and Target Movement \(Page 60\)](#), with their partner and transfer the code to their Game Starter File. Press Run to see `DANGER` and `TARGET` move across the screen independently!

Extension Activities

Once students have successfully gotten `update-target` and `update-danger` working, they can change the functions to make the characters move in whichever direction and whatever speed they want! They should be sure to modify their purpose statements and examples if they change their functions.

Want 2-D movement? A supplemental lesson [linked here](#) provides information on how to modify these functions to allow movement in the *x* and *y* directions!

These materials were developed partly through support of the National Science Foundation, (awards 1042210, 1535276, 1648684, and 1738598). Bootstrap:Algebra by the [Bootstrap Community](#) is licensed under a [Creative Commons 4.0 Unported License](#).

This license does not grant permission to run training or professional development. Offering training or professional development with materials substantially derived from Bootstrap must be approved in writing by a Bootstrap Director. Permissions beyond the scope of this license, such as to run training, may be available by contacting contact@BootstrapWorld.org.

