Character Animation

(Also available for WeScheme)

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

Prerequisites	Restating the Problem
Relevant Standards OK CC-Math	*Select one or more standards from the menu on the left (⌘-click on Mac, Ctrl-click elsewhere).*
Lesson Goals	Students will be able to: • Apply the ***Design Recipe*** to create a ***function*** given the constraints of a word problem. • Explain the basics of animation.
Student-Facing Lesson Goals	• I can use the Design Recipe to make a function. • I can describe how animation works.
Materials	• Lesson slides template (Google Slides) • Danger and Target Movement (Page 36)
Preparation	• Make sure all materials have been gathered • Decide how students will be grouped in pairs
Supplemental Resources	
Key Points for the Facilitator	• Encourage students to take their time in understanding *why* we want to fix `update-danger` and `update-target`. • Students might be confused as to *how* the animation is working. The `make-game` function at the bottom of the file has many inputs - including `update-danger` and `update-target`. `make-game` 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!

For a textbook-like version of materials similar to these, you may wish 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

---

Warmup

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

Animation

45 minutes

Animation

Overview

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

Launch

- How does animation work?
- 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-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)`, `update-target(250)`, 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)`, `update-danger(39)`, 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 Template 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 36), and transfer the code to their Game Template 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 36), with their partner and transfer the code to their Game Template 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!