



Unit 5 Game Animation

Unit Overview

Students define functions that map position n to position $n+1$, allowing them to move their dangers, targets and projectiles.

Learning Objectives

Students will:

- Apply algebra and geometry to solve a real world problem.
- Generalize their understanding of functions to create animation.

Product Outcomes

- Students identify and fix bugs in Racket expressions
- Students write simple, linear functions
- Students write update-target and update-danger

State Standards

See [Bootstrap Standards Matrix](#) provided as part of the Bootstrap curriculum.

Length: 90min

Materials and Equipment

- ☐ Computers w/DrRacket or WeScheme
- ☐ Student [workbook](#) folders - in pairs! - with names on covers
- ☐ Pens/pencils for the students, fresh whiteboard markers for teachers
- ☐ Class posters (List of rules, basic skills, course calendar)
- ☐ Language Table (see below)

Preparation

- ☐ Write agenda on board
- ☐ Student games open on their computers, in back
- ☐ OPTIONAL: Bug Hunting [[DrRacket file](#) | [WeScheme](#)]
- ☐ Class posters
- ☐ Seating arrangements: ideally clusters of desks/tables

Pre-Class Assessment: [Review of Unit 4](#)

- *This data helps us track student performance. All data collected is anonymous, and participation is optional.*
- If you would like us to see the assessment results for your class, [contact us](#).

Language Table

Types	Functions
Number	+ - * / sq sqrt expt
String	string-append string-length
Image	rectangle circle triangle ellipse radial-star scale rotate put-image

Agenda

- 25min [Introduction](#)
- 20min [Danger Movement](#)
- 20min [Target Movement](#)
- 15min [Opt1: Fun with Movement](#)
- 15min [Opt2: Projectile Movement](#)
- 5 min [Closing](#)

Introduction

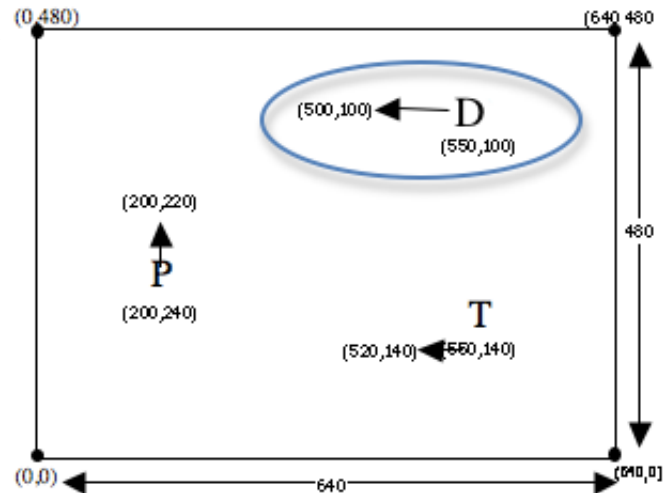
Time: 25 minutes

- Have students take the online assessment
- If you have students tackle the bug-hunting activity, be aware that the file contains several bugs! Until students have fixed all of them, they will always get an error message when they attempt to run the file. This is by design, to encourage students to both read the error and to look at it's location.

Danger Movement

Time: 20 minutes

- In our last class, you learned how to use the Design Recipe to write functions. Now it's time to write functions to make your game characters move!
- Who wants to start making their game move?
- Draw a 640x480 "game screen" on the board, and have students help you label the axes and corners.
- Here I've drawn a really boring game: my Danger is the letter "D", my Target is the letter "T", and my Player is just the letter "P". I've also labeled all the coordinates for where they are on the screen.
- What is the Danger's x-coordinate? Its y-coordinate?
- Suppose my Danger is moving to the left. What is changing, the x- or the y-coordinate?
- Draw an arrow to the NEW location, and label the coordinates.
- Suppose it moved to the right, by 50 pixels. What are the new coordinates?
- I want my Danger to always move **left by 50 pixels**, each time the screen is redrawn. So it goes from (550, 100) to (500, 100).
- What if it starts at (100, 100)? Where will go next? What if it's at (400, 600)? What if it's off the screen, at (650, 50)?
- You will need to write a function that takes an x-coordinate and produces the *next* x-coordinate, to make your danger move.



Current x-coordinate	Racket	Next x-coordinate
50	(- 50 50)	100
75	(- 75 50)	125
30	(- 30 50)	80

- Turn to **Page 14** for [update-danger](#). Have a student read the problem statement out loud.
- Can you find the function name in this problem statement? Underline it.
- Can you find the inputs and outputs? Underline them, too.
- As a team, complete the Design Recipe for [update-danger](#). Raise your hand when you're done with each step, so I can check your contracts and examples before you move on to code. Don't forget to circle the things that change between your examples!

Target Movement

Time: 20 minutes

- We also need to animate our target. Guess what? We need to write another function!
- How fast should your target move?
- Which direction?
- Next on **Page 15** is [update-target](#). Using a Design Recipe Worksheet again, complete the design recipe. Once again, make sure you come up to the board and write down your example. I will give you the "OK" to start typing.

Optional: Fun with movement

Time: 15 minutes

- Can you make your danger move slower? faster?
- Can you make your target move the other way? (just to try)

- It'll break your examples, but there's a function called (`random n`) that gives you a random number less than n. Can you make your target move with random speed?

OPTIONAL: Projectile Movement

Time: 15 minutes

- *NOTE FOR INSTRUCTORS: Many teachers find it useful to add projectiles only at the end of the class, as a "treat" for kids. It also tends to help break the "shooting-style" bias that a lot of students have.*
- *If you want to introduce projectiles this early, then go ahead and change `mystery` and `update-mystery` near the end of the game to `projectile` and `update-projectile`, and move them up to near the danger and target. The space bar triggers projectiles if the updater moves them. Some teachers will want to save this for later in the class, and have students rename the functions themselves.*
- Now we need to animate our projectile. If our dangers are flying in from the left, which way should our projectile fire? To the right!
- Use another Design Recipe worksheet to animate your projectile! Once again, make sure you write down your examples. I will give you the "OK" to start typing.
- Scroll to the bottom of the screen, and look for the semicolon before the word "start". Delete that semicolon.
- If your animation functions are correct, you should see your game start running, and your danger and target will move!

Closing

Time: 5 minutes

- Congratulations - you've got the beginnings of a working game!
- What's still missing?
 - The Player doesn't move
 - When the Target and Danger move offscreen, they never come back
 - Nothing happens when the Player collides with the Danger or Target
- We're going to fix these over the next few lessons, and also work on the artwork and story for our games, so stay tuned!
- Who can tell us one thing we learned today?
- Who saw someone else in the class do something great?
- Cleanup, dismissal.