



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:

- 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 our [Standards Document](#) provided as part of the Bootstrap curriculum.

Length: 90 minutes

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
- *OPTIONAL:* Bug Hunting [Eugs.rkt from [source-files.zip](#) | [WeScheme](#)] preloaded on students' machines, in front
- Class posters
- Seating arrangements: ideally clusters of desks/tables

Agenda

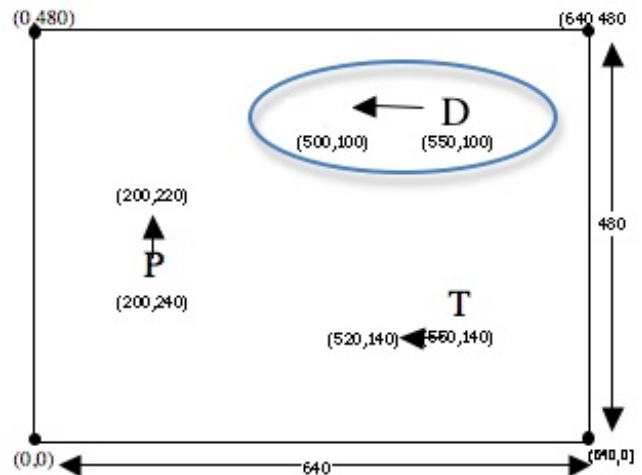
25 min	Introduction
20 min	Danger Movement
20 min	Target Movement
20 min	Optional: Fun With Movement
15 min	Projectile Movement (OPTIONAL)
15 min	Closing

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

- When tackling the bug-hunting activity, be aware that the file contains several bugs! Running the file will always return an error until each bug is fixed.
- This is by design, to encourage both reading the error message and using it to debug.

Danger Movement

- In our last class, you learned how to use the Design Recipe to solve word problems. In this lesson, all of the word problems will be about making your videogame characters 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? (Answer: Danger's x-coordinate is 500 and its y-coordinate is 100)
- Suppose my Danger is moving to the left. What is changing, the x- or the y-coordinate? (Answer: The x-coordinate is changing since you are moving horizontally.)
- Suppose it moved to the right, by 50 pixels. What are the new coordinates? (Answer: The new coordinates are (550, 100))
- 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)? (Answer: (50, 100), (350, 600), (600,50))
- What we have here is a **word problem**, which describes how the danger should move! Given any x-coordinate, `update-danger` should produce the next x-coordinate, which is fifty pixels to the left.
- Turn to [Page 15](#) for `update-danger`.
Have a student read the Word Problem out loud.
- Can you find the function name in this word problem? Underline it. (Answer: "The function name is " `update-danger`)
- Can you find the Domain and Range? Underline them, too. (Answer: The input is the x-coordinate of Danger and the output is the x-coordinate of Danger minus 50. Both of these are Numbers, which means the Domain and Range are both `@code{Number}`.)
- *Who would like to act out `update-danger`? Take a volunteer.*



Exercise:

According to the word problem, what's your name? What's your Domain? Your Range? Given an x-coordinate for the danger, you'll need to tell me what work gets done in order to calculate the new x-coordinate. For example, if I say "update-danger 75", you'll need to say "minus 75 50", because that's the work you'll do to move the danger to 25. Let's try it out: "update-danger 200!"

Have the class do some practice call-and-response with the volunteer.

- 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

- We also need to animate our target. Guess what? It's just another **Word Problem!**
- How fast should your target move?
- Which direction?
- Next on [Page 16](#) is `update-target`. Using a Design Recipe Worksheet again, solve this word problem.
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 20 min)

- 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?

Projectile Movement (OPTIONAL)

(Time 15 min)

- *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 causes the projectile to "fire".*
- 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 15 min)

- 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?*
- *Write down a few things you learned today*
- *Who saw someone else in the class do something great?*
- *Cleanup, dismissal.*



Bootstrap by [Emmanuel Schanzer](https://www.EmmanuelSchanzer.com) is licensed under a [Creative Commons 3.0 Unported License](https://creativecommons.org/licenses/by-nc-nd/3.0/). Based on a work at www.BootstrapWorld.org. Permissions beyond the scope of this license may be available at schanzer@BootstrapWorld.org.