Player Animation
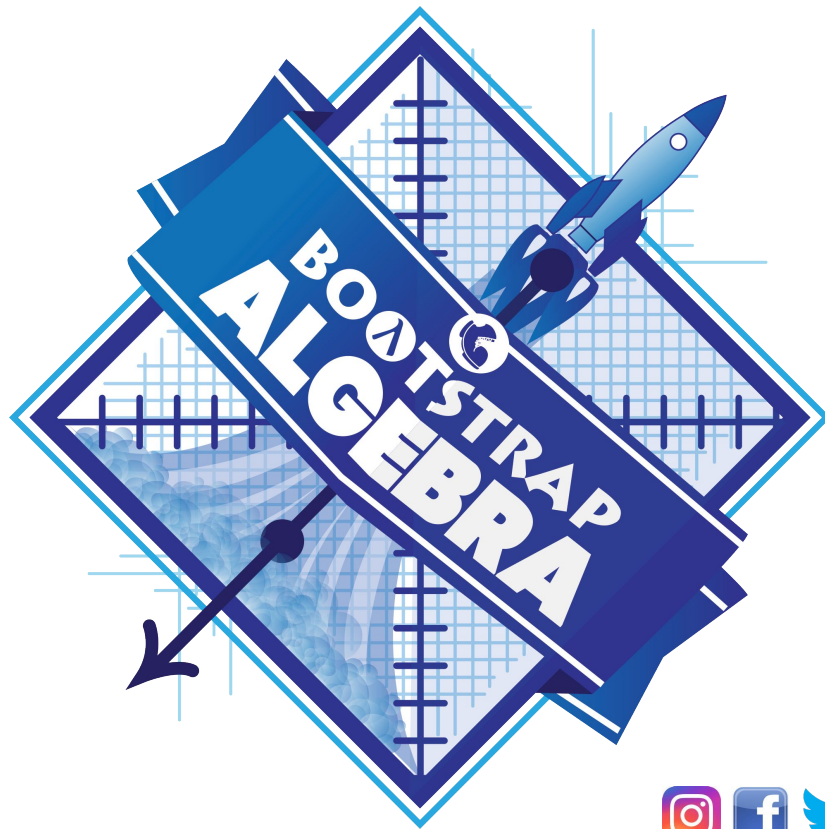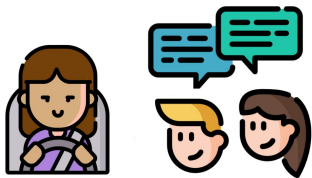
Defining Piecewise Functions

Open your saved Game file

Scroll to find your definitions for `update-danger` and `update-target`.

What controlled the *speed* of these characters?

What controlled the *direction* of these characters?
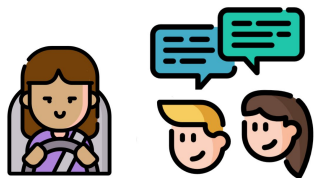
Defining Piecewise Functions

Open your saved Game file, and find your definition for `update-player`.

What is the contract for `update-player`?

What does each part of the domain and range represent?

How does the y-coordinate of `PLAYER` change when the user presses the `"up"` key?

Use the **Design Recipe to write update-player**

Defining Piecewise Functions

1. How is this function similar to the piecewise functions you've seen before? How is it different?

2. If we wanted the `"w"` key to make the player go up, _instead_ of the arrow key, what would change?

3. What should happen if you hit a key that you _don't_ have a condition for? What happens now?

4. Does this function need `otherwise`? If so, how would it be used?

Cheat Codes & Customizations

1. The second thing in the domain is a String that represents the key pressed.

2. Some players like to use "w" for up and "s" for down. What would need to change? What if we wanted both to work?

3. What if we wanted "w" and "s" to move the player faster than "up" and "down"?
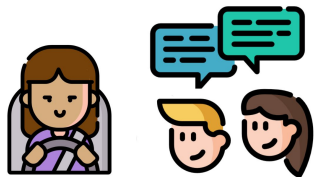
Cheat Codes & Customizations

- **Warping** - program one key to "warp" the player to a set location, such as the center of the screen

- **Boundaries** - change update-player such that PLAYER cannot move off the top or bottom of the screen

- **Wrapping** - add code to update-player such that when PLAYER moves to the top of the screen, it reappears at the bottom, and vice versa

- **Hiding** - add a key that will make PLAYER seem to disappear, and reappear when the same key is pressed again

Complete Challenges to get you started!

Cheat Codes & Customizations

Share back what you tried!

What would it take to make the player move left and right?
Why can't we do this without changing the contract?