<table>
<thead>
<tr>
<th></th>
<th>Racket Code</th>
<th>Pyret Code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Numbers</strong></td>
<td>(define AGE 14)</td>
<td>AGE = 14</td>
</tr>
<tr>
<td></td>
<td>(define A-NUMBER 0.6)</td>
<td>A-NUMBER = 0.6</td>
</tr>
<tr>
<td></td>
<td>(define SPEED -90)</td>
<td>SPEED = -90</td>
</tr>
<tr>
<td>Two of your own:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Strings</strong></td>
<td>(define CLASS &quot;Bootstrap&quot;)</td>
<td>CLASS = &quot;Bootstrap&quot;</td>
</tr>
<tr>
<td></td>
<td>(define PHRASE &quot;Coding is fun!&quot;)</td>
<td>PHRASE = &quot;Coding is fun!&quot;</td>
</tr>
<tr>
<td></td>
<td>(define A-STRING &quot;2500&quot;)</td>
<td>A-STRING = &quot;2500&quot;</td>
</tr>
<tr>
<td>Two of your own:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Images | SHAPE = triangle(40, "outline", "red")
(define SHAPE (triangle 40 "outline" "red"))
(define OUTLINE (star 80 "solid" "green"))
(define SQUARE (rectangle 50 50 "solid" "blue"))

| Booleans | BOOL = true
(define BOOL true)
(define BOOL2 false)

| Functions | double : Number -> Number
; Given a number, multiply by 2 to double it

(EXAMPLE (double 5) (* 2 5))
(EXAMPLE (double 7) (* 2 7))

(fun double(n):
  2 * n
end)

# double :: Number -> Number
# Given a number, multiply by 2 to double it

elements:
  double(5) is 2 * 5
double(7) is 2 * 7
end

One of your own:
Fast Functions!

Fill out the contract for each function, then try to write two examples and the definition by yourself.

# _________________: _______________ → _______________

<table>
<thead>
<tr>
<th>name</th>
<th>domain</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>double</td>
<td>Number</td>
<td>Number</td>
</tr>
</tbody>
</table>

examples:

- `double (5)` is `2 * 5`
- `double (7)` is `2 * 7`

end

fun `double (n)`: 

- `2 * n`

end

Fast Functions!

Fill out the contract for each function, then try to write two examples and the definition by yourself.

# _________________: _______________ → _______________

<table>
<thead>
<tr>
<th>name</th>
<th>domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>double</td>
<td>Number</td>
</tr>
</tbody>
</table>

examples:

- `double (5)` is `2 * 5`
- `double (7)` is `2 * 7`

end

fun `double (n)`: 

- `2 * n`

end
Fast Functions!

Fill out the contract for each function, then try to write two examples and the definition by yourself.

```
# _____________________::_________________ → _____________________
  | name               | domain | range |
  |___________________|________|_______|
examples:
    _______________(________) is _______________________
    ______________(________) is _______________________
end

fun ____________(_________________):


end
```

```
# _____________________::_________________ -> _____________________
  | name               | domain | range |
  |___________________|________|_______|
examples:
    ______________(________) is _______________________
    ______________(________) is _______________________
end

fun ____________(_________________):


end
```
Fast Functions!

Fill out the contract for each function, then try to write two examples and the definition by yourself.

<table>
<thead>
<tr>
<th>name</th>
<th>domain</th>
<th>range</th>
</tr>
</thead>
</table>

examples:

____________(________) is ______________________

____________(________) is ______________________

end

fun ____________(_________________):

____________________________________________________

end

<table>
<thead>
<tr>
<th>name</th>
<th>domain</th>
<th>range</th>
</tr>
</thead>
</table>

examples:

____________(________) is ______________________

____________(________) is ______________________

end

fun ____________(_________________):

____________________________________________________

end
| #1 | SECONDS = (7) |
|    | STRING = my string |
| #2 | SHAPE1 = circle(50 “solid” “blue”) |
|    | SHAPE2 = triangle(75, outline, yellow) |
| #3 | # triple :: Number -> Number |
|    | # Multiply a given number by |  |
|    | # 3 to triple it |  |
|    | examples: |  |
|    | triple(5) = 3 * 5 |  |
|    | triple(7) = 3 * 7 |  |
|    | end |  |
| #4 | fun triple(n): |
|    | 3 * n |
| #5 | # ys :: Number -> Number |
|    | # Given a number, create a solid |
|    | # yellow star of the given size |
|    | examples: |  |
|    | ys(99) is star(99, “solid”, “yellow”) |  |
|    | ys(33) is star(99, “solid”, “yellow”) |  |
|    | ys(size): |  |
|    | star(size “solid” “yellow”) |  |
|    | end |  |
Write a function double-radius, which takes in a radius and a color. It produces an outlined circle of whatever color was passed in, whose radius is twice as big as the input.

**Contract+Purpose Statement**

Every contract has three parts:

\[
\begin{array}{ccc}
\text{name} & : & \text{Domain} \\
\text{ } & \rightarrow & \text{Range} \\
\end{array}
\]

What does the function do?

**Give Examples**

Write examples of your function in action

**examples:**

\[\text{___________} (\text{___________________}) \]

the user types...

\[\text{is} \]

...which should become

\[\text{___________} (\text{___________________}) \]

the user types...

\[\text{is} \]

...which should become

**Function**

Circle the changes in the examples, and name the variables.

Write the code, copying everything that isn’t circled, and using names where you find variables.

\[\text{fun} \]

\[\text{___________________} (\text{___________________}) : \]

end
Word Problem: double-width

Write a function double-width, which takes in a number (the length of a rectangle) and produces a rectangle whose width is twice the given length.

Contract+Purpose Statement
Every contract has three parts:

# ___________ : : __________________________ → ___________
   name       Domain       Range

# _______________________________________________________

What does the function do?

Give Examples
Write examples of your function in action

examples:

_____________(___________________)
   the user types...

is ____________________________
   ...which should become

_____________(___________________)
   the user types...

is ____________________________
   ...which should become

end

Function
Circle the changes in the examples, and name the variables.
Write the code, copying everything that isn't circled, and using names where you find variables!

fun ________________(__________________) : 

________________________________________________________

end
Word Problem: next-position

Write a function `next-position`, which takes in two numbers (an x and y-coordinate) and returns a DeliveryState, increasing the x-coordinate by 5 and decreasing the y-coordinate by 5.

Contract+Purpose Statement
Every contract has three parts:

```
# _____________ :: ___________________________  →  ___________
     name                               Domain            Range
# __________________________________________
What does the function do?
```

Give Examples
Write examples of your function in action

```
examples:________________________________________________________
              ____________________________
the user types...
              ___________
is __________________________________________
               ...which should become

          ____________________________
the user types...

              ___________
is __________________________________________
               ...which should become
end
```

Function
Circle the changes in the examples, and name the variables.
Write the code, copying everything that isn't circled, and using names where you find variables!

```
fun ___________________(__________________) :

          _________________________________________________________
end
```
# A CakeType is a flavor, layers, & is-iceCream

data CakeType:
    cake(
        ___________________________________
        ___________________________________
        ___________________________________
    )

to make instances of this structure, I would write:

    cake1 = ___________________________________

    cake2 = ___________________________________

To access the fields of cake2, I would write:

    ___________________________________
    ___________________________________
    ___________________________________
    ___________________________________
Word Problem: taller-than

Write a function called *taller-than*, which consumes two CakeTypes, and produces true if the number of layers in the first CakeType is greater than the number of layers in the second.

Contract+Purpose Statement

# ___________ : : _____________________________________________________________________ → ___________

# ___________________________________________________________________________________

Give Examples

Write examples of your function in action

examples:

_________ (______________________)

the user types...

is __________________________________________________________

...which should become

_________ (______________________)

the user types...

is __________________________________________________________

end

...which should become

Function

Circle the changes in the examples, and name the variables.
Write the code, copying everything that isn't circled, and using names where you find variables!

fun _________________(______________________) :

______________________________________________________________________________

end
Word Problem: will-melt

Write a function called \textit{will-melt}, which takes in a \texttt{CakeType} and a temperature, and returns true if the temperature is greater than 32 degrees, AND the \texttt{CakeType} is an ice cream cake.

**Contract+Purpose Statement**

\begin{verbatim}
# ____________________ :: ____________________________ → __________
#
\end{verbatim}

**Give Examples**

Write examples of your function in action

\textbf{examples:}

\begin{verbatim}
________________________(________________________)

the user types...

is __________________________

...which should become

________________________(________________________)

the user types...

is __________________________

...which should become
end
\end{verbatim}

**Function**

Circle the changes in the examples, and name the variables.

Write the code, copying everything that isn't circled, and using names where you find variables!

\begin{verbatim}
fun __________________________(________________________) :

def________________________
end
\end{verbatim}
Below is a new structure definition:

```plaintext
data MediaType:
    | book(
        title :: String,
        author :: String,
        pubyear :: Number)
end
```

# an example book:

Fill in the blanks below with the vocabulary term that applies to each name. Here are the terms to choose from:

- contract
- header
- datatype
- constructor
- name
- example
- field
- instance
- data block
- purpose

**author** is a __________________________

**book** is a __________________________

**MediaType** is a ______________________

**book1** is a _________________________

**title** is a __________________________

**data ... end** is a _____________________
Identifying Animation Data Worksheet: Sunset

Draw a sketch for three distinct moments of the animation

Sketch A

Sketch B

Sketch C

What things are changing?

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What fields do you need to represent the things that change?

<table>
<thead>
<tr>
<th>Field name (dangerX, score, playerIMG...)</th>
<th>Datatype (Number, String, Image, Boolean...)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(worksheet continues on the next page)
Define the Data Structure

# a _________State is _________________________

data _________State:

| _________(____________________________________ |
| ____________________________________________ |
| ____________________________________________ |
| ____________________________________________ |
| ____________________________________________ |
| ____________________________________________ |

end

Make a sample instance for each sketch from the previous page:

_________ = ____________________________________________

_________ = ____________________________________________

_________ = ____________________________________________

_________ = ____________________________________________
Word Problem: draw-state

Write a function called draw-state, which takes in a SunsetState and returns an image in which the sun (a circle) appears at the position given in the SunsetState. The sun should be behind the horizon (the ground) once it is low in the sky.

Contract+Purpose Statement

# draw-state :: _____________________________ → Image

# _____________________________

Write an expression for each piece of your final image

SUN =
GROUND =
SKY =

Write the draw-state function, using put-image to combine your pieces

fun _____________________________(______________________) :

______________________________________
______________________________________
______________________________________

end
Word Problem: next-state-tick

Write a function called `next-state-tick`, which takes in a SunsetState and returns a SunsetState in which the new x-coordinate is 8 pixels larger than in the given SunsetState and the y-coordinate is 4 pixels smaller than in the given SunsetState.

Contract+Purpose Statement

```plaintext
# ____________ :: _____________________________________________________________________ → ____________  
# __________________________________________________________________________________
```

Give Examples

Write examples of your function in action

```plaintext
examples:  
_____________(____________________)  
the user types...  

is ____________________________________________________________________________  
...which should become  
_____________(____________________)  
the user types...  

is ____________________________________________________________________________  
...which should become  
end
```

Function

Circle the changes in the examples, and name the variables.
Write the code, copying everything that isn’t circled, and using names where you find variables!

```plaintext
fun ________________(____________________) :  

___________________________________________________________  
end
```
Identifying Animation Data Worksheet

Draw a sketch for three distinct moments of the animation

<table>
<thead>
<tr>
<th>Sketch A</th>
<th>Sketch B</th>
<th>Sketch C</th>
</tr>
</thead>
</table>

What things are changing?

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What fields do you need to represent the things that change?

<table>
<thead>
<tr>
<th>Field name (dangerX, score, playerIMG...)</th>
<th>Datatype (Number, String, Image, Boolean...)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(worksheet continues on the next page)
Define the Data Structure

# a _______ State is _______________________

data _______ State:

| _______ (_____________________________________

| ___________________________________________

| _______________________________________

end

Make a sample instance for each sketch from the previous page:

_________ = __________________________________

_________ = __________________________________

_________ = __________________________________

_________ = __________________________________
# Identifying Animation Data Worksheet

Draw a sketch for three distinct moments of the animation:

<table>
<thead>
<tr>
<th>Sketch A</th>
<th>Sketch B</th>
<th>Sketch C</th>
</tr>
</thead>
</table>

**What things are changing?**

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**What fields do you need to represent the things that change?**

<table>
<thead>
<tr>
<th>Field name (dangerX, score, playerIMG…)</th>
<th>Datatype (Number, String, Image, Boolean…)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Worksheet continues on the next page)
Define the Data Structure

# a __________State is ____________________________

data __________State:

| __________ (______________________________

| ______________________

| ______________________

| ______________________)

data State:

| ______________________

| ______________________

| ______________________

| ______________________

| ______________________

end

Make a sample instance for each sketch from the previous page:

__________ = ________________________________

__________ = ________________________________

__________ = ________________________________

__________ = ________________________________
## Identifying Animation Data Worksheet

**Draw a sketch for three distinct moments of the animation**

<table>
<thead>
<tr>
<th>Sketch A</th>
<th>Sketch B</th>
<th>Sketch C</th>
</tr>
</thead>
</table>

### What things are changing?

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### What fields do you need to represent the things that change?

<table>
<thead>
<tr>
<th>Field name (dangerX, score, playerIMG…)</th>
<th>Datatype (Number, String, Image, Boolean…)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(worksheet continues on the next page)
Define the Data Structure

```plaintext
# a _________State is ____________________________
data _________State:
 | _________(____________________________________
 | ___________________________
 | ___________________________
 | ___________________________
 | ___________________________
 | ___________________________
end
```

Make a sample instance for each sketch from the previous page:

```
__________ = _________________________________

__________ = _________________________________

__________ = _________________________________

__________ = _________________________________
```
## Identifying Animation Data Worksheet

**Draw a sketch for three distinct moments of the animation**

<table>
<thead>
<tr>
<th>Sketch A</th>
<th>Sketch B</th>
<th>Sketch C</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="sketch_a.png" alt="Sketch A" /></td>
<td><img src="sketch_b.png" alt="Sketch B" /></td>
<td><img src="sketch_c.png" alt="Sketch C" /></td>
</tr>
</tbody>
</table>

### What things are changing?

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### What fields do you need to represent the things that change?

<table>
<thead>
<tr>
<th>Field name (dangerX, score, playerIMG…)</th>
<th>Datatype (Number, String, Image, Boolean…)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Worksheet continues on the next page)
Define the Data Structure

# a ________State is __________________________

data ________State:

| ________((______________________________

| ________________________________

| ________________________________

| ________________________________

end

Make a sample instance for each sketch from the previous page:

_______ = __________________________________________________________________

_______ = __________________________________________________________________

_______ = __________________________________________________________________

_______ = __________________________________________________________________
Word Problem: location

Write a function called `location`, which consumes a DeliveryState, and produces a String representing the location of a box: either “road”, “delivery zone”, “house”, or “air”.

Contract+Purpose Statement

# ______________ :: ___________________________ → __________

# ________________________________

Give Examples

examples:

________(__________________) is ________________

________(__________________) is ________________

________(__________________) is ________________

________(__________________) is ________________

________(__________________) is ________________

end

(Worksheet continues next page)
## Syntax and Style Bug Hunting: Piecewise Edition

<table>
<thead>
<tr>
<th>Round</th>
<th>Buggy Code</th>
<th>Correct Code / Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round 1</td>
<td><code>fun piecewisefun(n):</code>&lt;br&gt;<code>  if (n &gt; 0): n</code>&lt;br&gt;<code>  else: 0</code></td>
<td></td>
</tr>
<tr>
<td>Round 2</td>
<td><code>fun cost(topping):</code>&lt;br&gt;<code>  if string-equal(topping, &quot;pepperoni&quot;): 10.50</code>&lt;br&gt;<code>  else string-equal(topping, &quot;cheese&quot;): 9.00</code>&lt;br&gt;<code>  else string-equal(topping, &quot;chicken&quot;): 11.25</code>&lt;br&gt;<code>  else string-equal(topping, &quot;broccoli&quot;): 10.25</code>&lt;br&gt;<code>  else: &quot;That's not on the menu!&quot;</code></td>
<td></td>
</tr>
<tr>
<td>Round 3</td>
<td><code>fun absolute-value(a b):</code>&lt;br&gt;<code>  if a &gt; b: a - b</code>&lt;br&gt;<code>  b - a</code></td>
<td></td>
</tr>
<tr>
<td>Round 4</td>
<td><code>fun best-function(f):</code>&lt;br&gt;<code>  if string-equal(f, &quot;blue&quot;): “you win!”</code>&lt;br&gt;<code>  else if string-equal(f, &quot;blue&quot;): “you lose!”</code>&lt;br&gt;<code>  else if string-equal(f, &quot;red&quot;): “Try again!”</code>&lt;br&gt;<code>  else: “Invalid entry!”</code></td>
<td></td>
</tr>
</tbody>
</table>
**Animation Extension Worksheet**

Describe the goal of your change: what new feature or behavior will it add to your animation?

## Draw a sketch for three distinct moments of the animation

<table>
<thead>
<tr>
<th>Sketch</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
</table>

## What things are changing?

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## What fields do you need to represent the things that change?

<table>
<thead>
<tr>
<th>Field name</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>dangerX</td>
<td></td>
</tr>
<tr>
<td>score</td>
<td></td>
</tr>
<tr>
<td>playerIMG</td>
<td></td>
</tr>
<tr>
<td>number</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td></td>
</tr>
<tr>
<td>image</td>
<td></td>
</tr>
<tr>
<td>boolean</td>
<td></td>
</tr>
</tbody>
</table>

## Make a To-Do List, and check off each as “Done” when you finish each one.

<table>
<thead>
<tr>
<th>Component</th>
<th>When is there work to be done?</th>
<th>To-Do</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structure</td>
<td>If any new field(s) were added, changed or removed</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>draw-state</td>
<td>If something is displayed in a new way or position</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-tick</td>
<td>If the Data Structure changed, or the animation happens automatically</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-key</td>
<td>If the Data Structure changed, or a keypress triggers the animation</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>reactor</td>
<td>If either next-state function is new</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
Make a sample instance for each sketch from the previous page:

_________ = __________________________________
_________ = _______________________________
_________ = __________________________________
_________ = _______________________________

Write at least one NEW example for one of the functions on your To-Do list

________________________________________
________________________________________
________________________________________
________________________________________

If you have another function on your To-Do list , write at least one NEW example

________________________________________
________________________________________
________________________________________
________________________________________
Word Problem: draw-sun

Write a function called `draw-sun`, which consumes a `SunsetState`, and produces an image of a sun (a solid, 25 pixel circle), whose color is "yellow", when the sun’s y-coordinate is greater than 225, "orange", when its y-coordinate is between 150 and 225, and "red" otherwise.

Contract+Purpose Statement

```latex
# __________________ :: _____________________________ \rightarrow ____________
# __________________________
```

Give Examples

```latex
examples:
    ______(__________________) is ________________
    ______(__________________) is ________________
    ______(__________________) is ________________
    ______(__________________) is ________________
```

end

(worksheet continues next page)
Animation Extension Worksheet

Describe the goal of your change: what new feature or behavior will it add to your animation?

Decrease the cat’s hunger level by 2 and sleep level by 1 on each tick.

Draw a sketch for three distinct moments of the animation, focusing on the new behavior:

![Sketch A](image1)
![Sketch B](image2)
![Sketch C](image3)

What things are changing?

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
</table>

What fields do you need to represent the things that change?

<table>
<thead>
<tr>
<th>Field name (dangerX, score, playerIMG…)</th>
<th>Datatype (Number, String, Image, Boolean…)</th>
</tr>
</thead>
</table>

Make a To-Do List, and check off each as “Done” when you finish each one.

<table>
<thead>
<tr>
<th>Component</th>
<th>When is there work to be done?</th>
<th>To-Do</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structure</td>
<td>If any new field(s) were added, changed or removed</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>draw-state</td>
<td>If something is displayed in a new way or position</td>
<td>✓</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-tick</td>
<td>If the Data Structure changed, or the animation happens automatically</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-key</td>
<td>If the Data Structure changed, or a keypress triggers the animation</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>reactor</td>
<td>If either next-state function is new</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
Make a sample instance for each sketch from the previous page:

**FULLPET** = pet(100, 100)

**MIDPET** = pet(50, 75)

**LOSEPET** = pet(0, 0)

Write at least one NEW example for one of the functions on your To-Do list

next-state-tick(FULLPET) is pet(FULLPET.hunger - 2, FULLPET.sleep - 1)
next-state-tick(MIDPET) is pet(MIDPET.hunger - 2, MIDPET.sleep - 1)
next-state-tick(LOSEPET) is LOSEPET

If you have another function on your To-Do list, write at least one NEW example
Animation Extension Worksheet

Describe the goal of your change: what new feature or behavior will it add to your animation?

Draw a sketch for three distinct moments of the animation

Sketch A

Sketch B

Sketch C

What things are changing?

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What fields do you need to represent the things that change?

<table>
<thead>
<tr>
<th>Field name (dangerX, score, playerIMG…)</th>
<th>Datatype (Number, String, Image, Boolean…)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Make a To-Do List, and check off each as “Done” when you finish each one.

<table>
<thead>
<tr>
<th>Component</th>
<th>When is there work to be done?</th>
<th>To-Do</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structure</td>
<td>If any new field(s) were added, changed or removed</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>draw-state</td>
<td>If something is displayed in a new way or position</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-tick</td>
<td>If the Data Structure changed, or the animation happens automatically</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-key</td>
<td>If the Data Structure changed, or a keypress triggers the animation</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>reactor</td>
<td>If either next-state function is new</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
Make a sample instance for each sketch from the previous page:

__________ = __________________________________

__________ = __________________________________

__________ = __________________________________

Write at least one NEW example for one of the functions on your To-Do list

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

If you have another function on your To-Do list, write at least one NEW example

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________
### Animation Extension Worksheet

Describe the goal of your change: what new feature or behavior will it add to your animation?

**Draw a sketch for three distinct moments of the animation**

<table>
<thead>
<tr>
<th>Sketch A</th>
<th>Sketch B</th>
<th>Sketch C</th>
</tr>
</thead>
</table>

**What things are changing?**

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**What fields do you need to represent the things that change?**

<table>
<thead>
<tr>
<th>Field name (dangerX, score, playerIMG…)</th>
<th>Datatype (Number, String, Image, Boolean…)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Make a To-Do List, and check off each as “Done” when you finish each one.**

<table>
<thead>
<tr>
<th>Component</th>
<th>When is there work to be done?</th>
<th>To-Do</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structure</td>
<td>If any new field(s) were added, changed or removed</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>draw-state</td>
<td>If something is displayed in a new way or position</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-tick</td>
<td>If the Data Structure changed, or the animation happens automatically</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-key</td>
<td>If the Data Structure changed, or a keypress triggers the animation</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>reactor</td>
<td>If either next-state function is new</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
Make a sample instance for each sketch from the previous page:

__________  = __________________________________________

__________  = __________________________________________

__________  = __________________________________________

Write at least one NEW example for one of the functions on your To-Do list

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________

If you have another function on your To-Do list, write at least one NEW example

__________________________________________________________________________

__________________________________________________________________________

__________________________________________________________________________
**Animation Design Worksheet**

**Draw a sketch for three distinct moments of the animation**

<table>
<thead>
<tr>
<th>Sketch A</th>
<th>Sketch B</th>
<th>Sketch C</th>
</tr>
</thead>
</table>

**What things are changing?**

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**What fields do you need to represent the things that change?**

<table>
<thead>
<tr>
<th>Field name (e.g., dangerX, score, playerIMG...)</th>
<th>Datatype (Number, String, Image, Boolean...)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Make a To-Do List, and check off each as “Done” when you finish each one.**

<table>
<thead>
<tr>
<th>Component</th>
<th>When is there work to be done?</th>
<th>To-Do</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structure</td>
<td>If any new field(s) were added, changed or removed</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>draw-state</td>
<td>If something is displayed in a new way or position</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-tick</td>
<td>If the Data Structure changed, or the animation happens automatically</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-key</td>
<td>If the Data Structure changed, or a keypress triggers the animation</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>reactor</td>
<td>If either next-state function is new</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
Define the Data Structure

# a ________State is ______________________

data ________State:

| ________(_________________________________
| _______________________________
| _______________________________
| _______________________________
| _______________________________
end

Make a sample instance for each sketch from the previous page:

_______ = _______________________________
_______ = _______________________________
_______ = _______________________________
_______ = _______________________________

Write an example for one of the functions on the previous page:

_________________________________________________________________________________
_________________________________________________________________________________
_________________________________________________________________________________
_________________________________________________________________________________
_________________________________________________________________________________
Distance:

The Player is at (4, 2) and the Target is at (0, 5). Distance takes in the player’s x, player’s y, character’s x and character’s y.

Use the formula below to fill in the EXAMPLE:

\[ \sqrt{(4 - 0)^2 + (2 - 5)^2} \]

Convert it into a Circle of Evaluation. (We’ve already gotten you started!)

Convert it into Pyret code:
Word Problem: distance

Write a function `distance`, which takes FOUR inputs:
- `px`: The x-coordinate of the player
- `py`: The y-coordinate of the player
- `cx`: The x-coordinate of another game character
- `cy`: The y-coordinate of another game character

It should return the distance between the two, using the Distance formula:

\[
\text{Distance}^2 = (px - cx)^2 + (py - cy)^2
\]

Contract+Purpose Statement

# __________ :: ____________________________________________________________________ -> __________
# __________________________________________________________________________________

Give Examples

Write examples of your function in action

examples:

___________(__________)

is____________________________________________________________________________________

___________(__________)

is____________________________________________________________________________________

end

Function

fun _______________(______________):

____________________________________________________________________________________

____________________________________________________________________________________

end
Word Problem: is-collision

Write a function is-collision, which takes FOUR inputs:

- px: The x-coordinate of the player
- py: The y-coordinate of the player
- cx: The x-coordinate of another game character
- cy: The y-coordinate of another game character

It should return true if the coordinates of the player are within **50 pixels** of the coordinates of the other character. Otherwise, false.

Contract+Purpose Statement

```
# ____________::: ____________________________ -> ____________
# ____________________________
```

Give Examples

Write examples of your function in action

examples:

```
__________________________(____________________)

is________________________(____________________)

__________________________(____________________)

is________________________(____________________)
```

end

Function

```
fun _____________(___________________):

______________________________

______________________________

end
```
DESIGN RECIPE

Contract+Purpose Statement
Every contract has three parts:

# __________________ :: ____________________________________ - > __________
  name                     Domain                        Range

# _______________________________________________________
  What does the function do?

Give Examples
Write examples of your function in action

examples:

  ______________(_____________)
    the user types...
    is ______________________________________________________________
    ...which should become

  ______________(_____________)
    the user types...
    is ______________________________________________________________
    ...which should become

end

Function
Circle the changes in the examples, and name the variables.

fun __________________(______________):

                           ______________________________________

end
**DESIGN RECIPE**

**Contract+Purpose Statement**

Every contract has three parts:

# __________________ :: ___________________________ - > __________________

name          Domain             Range

# __________________

What does the function do?

**Give Examples**

Write examples of your function in action

**examples:**

(__________________)  

the user types...

is __________________

...which should become

(__________________)  

the user types...

is __________________

...which should become

end

**Function**

Circle the changes in the examples, and name the variables.

fun __________________(__________________):

_________________________________________

end
Animation Design Worksheet

Draw a sketch for three distinct moments of the animation

<table>
<thead>
<tr>
<th>Sketch A</th>
<th>Sketch B</th>
<th>Sketch C</th>
</tr>
</thead>
</table>

What things are changing?

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What fields do you need to represent the things that change?

<table>
<thead>
<tr>
<th>Field name (dangerX, score, playerIMG…)</th>
<th>Datatype (Number, String, Image, Boolean…)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Make a To-Do List, and check off each as “Done” when you finish each one.

<table>
<thead>
<tr>
<th>Component</th>
<th>When is there work to be done?</th>
<th>To-Do</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structure</td>
<td>If any new field(s) were added, changed or removed</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>draw-state</td>
<td>If something is displayed in a new way or position</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-tick</td>
<td>If the Data Structure changed, or the animation happens automatically</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-key</td>
<td>If the Data Structure changed, or a keypress triggers the animation</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>reactor</td>
<td>If either next-state function is new</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
Define the Data Structure

# a ________State is ______________________

data ________State:

| _________(_________________________________

       _________________________________________

       _________________________________________

       _________________________________________

end

Make a sample instance for each sketch from the previous page:

_________ = _______________________________________

_________ = _______________________________________

_________ = _______________________________________

_________ = _______________________________________

Write an example for one of the functions on the previous page:

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________

________________________________________________________________________
**Animation Design Worksheet**

Draw a sketch for three distinct moments of the animation

<table>
<thead>
<tr>
<th>Sketch A</th>
<th>Sketch B</th>
<th>Sketch C</th>
</tr>
</thead>
</table>

What things are changing?

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What fields do you need to represent the things that change?

<table>
<thead>
<tr>
<th>Field name (dangerX, score, playerIMG...)</th>
<th>Datatype (Number, String, Image, Boolean...)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Make a To-Do List, and check off each as “Done” when you finish each one.

<table>
<thead>
<tr>
<th>Component</th>
<th>When is there work to be done?</th>
<th>To-Do</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structure</td>
<td>If any new field(s) were added, changed or removed</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>draw-state</td>
<td>If something is displayed in a new way or position</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-tick</td>
<td>If the Data Structure changed, or the animation happens automatically</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>next-state-key</td>
<td>If the Data Structure changed, or a keypress triggers the animation</td>
<td>☐</td>
<td>☐</td>
</tr>
<tr>
<td>reactor</td>
<td>If either next-state function is new</td>
<td>☐</td>
<td>☐</td>
</tr>
</tbody>
</table>
Define the Data Structure

# a ________State is __________________________

data ________State:
  | ________(_________________________________
  | _________________________________________
  | _________________________________________
  | _________________________________________
end

Make a sample instance for each sketch from the previous page:

_________ = __________________________________________
_________ = __________________________________________
_________ = __________________________________________
_________ = __________________________________________

Write an example for one of the functions on the previous page:

_____________________________________________________
_____________________________________________________
_____________________________________________________
_____________________________________________________
_____________________________________________________
_____________________________________________________
_____________________________________________________
_____________________________________________________
_____________________________________________________
_____________________________________________________
**Animation Design Worksheet**

Draw a sketch for three distinct moments of the animation

<table>
<thead>
<tr>
<th>Sketch A</th>
<th>Sketch B</th>
<th>Sketch C</th>
</tr>
</thead>
</table>

What things are changing?

<table>
<thead>
<tr>
<th>Thing</th>
<th>Describe how it changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

What fields do you need to represent the things that change?

<table>
<thead>
<tr>
<th>Field name (dangerX, score, playerIMG...)</th>
<th>Datatype (Number, String, Image, Boolean...)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Make a To-Do List, and check off each as “Done” when you finish each one.

<table>
<thead>
<tr>
<th>Component</th>
<th>When is there work to be done?</th>
<th>To-Do</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structure</td>
<td>If any new field(s) were added, changed or removed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>draw-state</td>
<td>If something is displayed in a new way or position</td>
<td></td>
<td></td>
</tr>
<tr>
<td>next-state-tick</td>
<td>If the Data Structure changed, or the animation happens automatically</td>
<td></td>
<td></td>
</tr>
<tr>
<td>next-state-key</td>
<td>If the Data Structure changed, or a keypress triggers the animation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reactor</td>
<td>If either next-state function is new</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Define the Data Structure

# a _________ State is ____________________________

data _________ State:
| _________(_________________________________________________________________________

|_________________________________________________
|_________________________________________________
|_________________________________________________
|_________________________________________________
|_________________________________________________
|_________________________________________________
end

Make a sample instance for each sketch from the previous page:

_______ = ____________________________________

_______ = ____________________________________

_______ = ____________________________________

_______ = ____________________________________

Write an example for one of the functions on the previous page:

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________
Make a sample instance for each sketch from the previous page:

\[
___ = \ldots
\]

\[
___ = \ldots
\]

\[
___ = \ldots
\]

Write at least one NEW example for one of the functions on your To-Do list

\[
\ldots
\]

\[
\ldots
\]

\[
\ldots
\]

\[
\ldots
\]

If you have another function on your To-Do list, write at least one NEW example

\[
\ldots
\]

\[
\ldots
\]

\[
\ldots
\]

\[
\ldots
\]
Make a sample instance for each sketch from the previous page:

_________ = __________________________________

_________ = __________________________________

_________ = __________________________________

Write at least one NEW example for one of the functions on your To-Do list

_________________________________________________

_________________________________________________

_________________________________________________

_________________________________________________

If you have another function on your To-Do list, write at least one NEW example

_________________________________________________

_________________________________________________

_________________________________________________

________________________________________________