Workbook v0.9

Brought to you by the Bootstrap team:
- Emma Youndsmith
- Emmanuel Schanzer
- Kathi Fisler
- Joe Politz
- Shriram Krishnamurthi

Visual Design: Colleen Murphy

Bootstrap is licensed under a Creative Commons 3.0 Unported License. Based on a work from www.BootstrapWorld.org. Permissions beyond the scope of this license may be available at schanzer@BootstrapWorld
## Unit 1

<table>
<thead>
<tr>
<th>Racket Code</th>
<th>Pyret Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>(define AGE 14)</td>
<td>AGE = 14</td>
</tr>
<tr>
<td>(define A–NUMBER 0.6)</td>
<td>A–NUMBER = 0.6</td>
</tr>
<tr>
<td>(define SPEED –90)</td>
<td>SPEED = –90</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Two of your own:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>(define CLASS “Bootstrap”)</td>
<td>CLASS = &quot;Bootstrap&quot;</td>
</tr>
<tr>
<td>(define PHRASE &quot;Coding is fun!&quot;)</td>
<td>PHRASE = “Coding is fun!”</td>
</tr>
<tr>
<td>(define A–STRING &quot;2500&quot;)</td>
<td>A–STRING = &quot;2500&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Two of your own:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

1
<table>
<thead>
<tr>
<th>Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>(define SHAPE (triangle 40 &quot;outline&quot; &quot;red&quot;))</td>
</tr>
<tr>
<td>(define OUTLINE (star 80 &quot;solid&quot; &quot;green&quot;))</td>
</tr>
<tr>
<td>(define SQUARE (rectangle 50 50 &quot;solid&quot; &quot;blue&quot;))</td>
</tr>
</tbody>
</table>

SHAPE = triangle(40, "outline", "red")
OUTLINE = star(80, "solid", "green")
SQUARE = rectangle(50, 50, "solid", "blue")

One of your own:

<table>
<thead>
<tr>
<th>Booleans</th>
</tr>
</thead>
<tbody>
<tr>
<td>(define BOOL true)</td>
</tr>
<tr>
<td>(define BOOL2 false)</td>
</tr>
</tbody>
</table>

BOOL = true

One of your own:

<table>
<thead>
<tr>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>; double : Number -&gt; Number</td>
</tr>
<tr>
<td>; Given a number, multiply by</td>
</tr>
<tr>
<td>; 2 to double it</td>
</tr>
<tr>
<td>(EXAMPLE (double 5) (* 2 5))</td>
</tr>
<tr>
<td>(EXAMPLE (double 7) (* 2 7))</td>
</tr>
<tr>
<td>(define (double n) (* 2 n))</td>
</tr>
</tbody>
</table>

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

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>
<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**

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

---

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

**Examples:**

- **_______(_______)** is **_______________**
- **_______(_______)** is **_______________**

```
fun _________(_______________):
    _______________________
end
```
Fast Functions!

Fill out the contract for each function, then try to write two examples and the definition by yourself.
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></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

examples:

________(______) is ___________________

________(______) is ___________________

definition

fun ___________(__________________):

end

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

examples:

________(______) is ___________________

________(______) is ___________________

definition

fun ___________(__________________):

end
# _________________::_________________ → ____________________

name domain range

text examples:

_______(_______) is _______________________

_______(_______) is _______________________

end

fun _________(_________________):

_________________________________________

end

# _________________::_________________ → ____________________

name domain range

examples:

_______(_______) is _______________________

_______(_______) is _______________________

end

fun _________(_________________):

_________________________________________

end

Syntax and Style Bug Hunting: Pyret Edition
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>#1</strong></td>
<td>SECONDS = (7)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STRING = my string</td>
<td></td>
</tr>
<tr>
<td><strong>#2</strong></td>
<td>SHAPE1 = circle(50 “solid” “blue”)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SHAPE2 = triangle(75, outline, yellow)</td>
<td></td>
</tr>
<tr>
<td><strong>#3</strong></td>
<td># triple :: Number -&gt; Number</td>
<td></td>
</tr>
<tr>
<td></td>
<td># Multiply a given number by</td>
<td></td>
</tr>
<tr>
<td></td>
<td># 3 to triple it</td>
<td></td>
</tr>
<tr>
<td></td>
<td>examples:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>triple(5) = 3 * 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>triple(7) = 3 * 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>end</td>
<td></td>
</tr>
<tr>
<td><strong>#4</strong></td>
<td>fun triple(n):</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 * n</td>
<td></td>
</tr>
<tr>
<td><strong>#5</strong></td>
<td># ys :: Number -&gt; Number</td>
<td></td>
</tr>
<tr>
<td></td>
<td># Given a number, create a solid</td>
<td></td>
</tr>
<tr>
<td></td>
<td># yellow star of the given size</td>
<td></td>
</tr>
<tr>
<td></td>
<td>examples:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ys(99) is star(99, “solid”, “yellow”)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ys(33) is star(99, “solid”, “yellow”)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ys(size):</td>
<td></td>
</tr>
<tr>
<td></td>
<td>star(size “solid” “yellow”)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>end</td>
<td></td>
</tr>
</tbody>
</table>
Word Problem: double-radius

Write a function \textit{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.

\textbf{Contract+Purpose Statement}

Every contract has three parts:

\#
name : Domain -> Range

\#

\textbf{What does the function do?}

\textbf{Give Examples}

Write examples of your function in action

\textbf{examples:}

\_
\_
\_
\_
\_
\_
\_
\_
\_
\_

the user types...

\_
\_
\_
\_
\_
\_
\_
\_
\_
\_

...which should become

\_
\_
\_
\_
\_
\_
\_
\_
\_
\_

the user types...

\_
\_
\_
\_
\_
\_
\_
\_
\_
\_

...which should become

\textbf{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!

\textbf{fun}

\_
\_
\_
\_
\_
\_
\_
\_
\_
\_

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!

```plaintext
fun (__________________________) : 

```

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

data CakeType:
    | cake(____________________________________

    _______________________________________

    _______________________________________

  )

data CakeType:
    | cake(____________________________________

    _______________________________________

    _______________________________________

  )

end

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 will-melt, which takes in a CakeType and a temperature, and returns true if the temperature is greater than 32 degrees, AND the CakeType is an ice cream cake.

Contract+Purpose Statement

```python
# ____________________ :: ____________________________ → ____________
# ___________________________ ______________
```

Give Examples

Write examples of your function in action

```python
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!

```python
fun ______________________________(__________________________):

    ____________________________

end
```
Vocabulary Practice

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

<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:
| __________(________________________________
  ________________________________________
  ________________________________________
  ________________________________________
  ________________________________________
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

```plaintext
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

# __________________ : ___________________________ → __________
# __________________

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
### 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></td>
<td></td>
<td></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>
</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>
</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>
</tbody>
</table>

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

# a _________State is ____________________________

data _________State:
|

# State is ____________________________

| __________(_________________________________
| _________(_________________________________
| _____________________________
| _____________________________
  _____________________________)
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:

| _________(_________________________________

_________________________________

_________________________________

_________________________________

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:

| _________(_________________________________

________________________________

| __________(_________________________________

________________________________

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**

```plaintext
# ______________ :: ____________________________ → __________
# __________________________
```

**Give Examples**

Examples:

- `________(__________________)` is `____________`
- `________(__________________)` is `____________`
- `________(__________________)` is `____________`
- `________(__________________)` is `____________`

end

(worksheet continues next page)
fun ________________(____________________) :
    if ____________________________________________:
        ________________________________
    else if ____________________________________:
        ________________________________
    else if ____________________________________:
        ________________________________
    else: ____________________________________

    end

end end
## Syntax and Style Bug Hunting: Piecewise Edition

<table>
<thead>
<tr>
<th>Round 1</th>
<th>Buggy Code</th>
<th>Correct Code / Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td></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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Round 2</th>
<th>Buggy Code</th>
<th>Correct Code / Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td></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>&lt;br&gt;<code>end</code>&lt;br&gt;<code>end</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Round 3</th>
<th>Buggy Code</th>
<th>Correct Code / Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td></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>&lt;br&gt;<code>end</code>&lt;br&gt;<code>end</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Round 4</th>
<th>Buggy Code</th>
<th>Correct Code / Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>fun best-function(f):</code>&lt;br&gt;<code>if string-equal(f, &quot;blue&quot;): &quot;you win!&quot;</code>&lt;br&gt;<code>else if string-equal(f, &quot;blue&quot;): &quot;you lose!&quot;</code>&lt;br&gt;<code>else if string-equal(f, &quot;red&quot;): &quot;Try again!&quot;</code>&lt;br&gt;<code>else: &quot;Invalid entry!&quot;</code>&lt;br&gt;<code>end</code>&lt;br&gt;<code>end</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 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

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________
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

# ______________ :: ___________________________ → __________
# __________________________________________________________________

Give Examples

examples:
    ________ (__________________) is ______________
    ________ (__________________) is ______________
    ________ (__________________) is ______________

end

(worksheet continues next page)
fun _______________(____________________) :
    if _____________________________________:
        _____________________________________
    else if ___________________________________:
        _____________________________________
    else: _____________________________________
end
end
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
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:

**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>
</tbody>
</table>
If either next-state function is new

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>
<tbody>
<tr>
<td></td>
<td></td>
<td></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>
</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 (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

```plaintext
# 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:
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
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

describe:

examples:

_____________(_____________)

the user types...

is ________________________________

...which should become

_____________(_____________)

the user types...

is ________________________________

...which should become

describe

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:

________________________________________________

________________________________________________

________________________________________________

________________________________________________

________________________________________________

________________________________________________

________________________________________________

________________________________________________
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>
</tbody>
</table>
reactor | If either next-state function is new

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

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>
</tbody>
</table>
reactor | If either next-state function is new
--------|-----------------------------------

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:

_________________________________________________
_________________________________________________
_________________________________________________

_________________________________________________

63