Name: _____

Bootstrap:Reactive

www.bootstrapworld.org

Class: _____

Workbook v0.9

Brought to you by the Bootstrap team:

- Emma Youndtsmith
- Emmanuel Schanzer
- Kathi Fisler
- Joe Politz
- Shriram Krishnamurthi

Visual Design: Colleen Murphy

Unit 1

	Racket Code	Pyret Code
Numbers	(define AGE 14) (define A-NUMBER 0.6) (define SPEED -90)	AGE = 14 A-NUMBER = 0.6 SPEED = -90 Two of your own: **WIDTH = 640** **YEAR = 2017**
Strings	(define CLASS “Bootstrap”) (define PHRASE "Coding is fun!") (define A-STRING "2500")	CLASS = “Bootstrap” PHRASE = “Coding is fun!” A-STRING = “2500” Two of your own: **NAME = “Elizabeth”** **CITY = “Philadelphia”**

Images	(define SHAPE
	  (triangle 40 "outline" "red"))
	
	(define OUTLINE
	  (star 80 "solid" "green"))
	
	(define SQUARE
	  (rectangle 50 50 "solid" "blue"))
		SHAPE =
		  triangle(40, "outline", "red")
		
		OUTLINE =
		  star(80, "solid", "green")
		
		SQUARE =
		  rectangle(50, 50, "solid", "blue")
		
		One of your own:
		
		MY-SHAPE =
		** rhombus(90, 60, "solid", "red")**

Booleans	(define BOOL true) (define BOOL2 false)	BOOL = true One of your own: **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))
	
	(define (double n) (* 2 n))
		# 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.

```
# double : Number → Number
  name   domain   range

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

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

```
# triple : Number -> Number
  name   domain   range

examples:
   triple ( 9 ) is 9 * 3
   triple ( 40 ) is 40 * 3
end

fun triple ( n ):
    n * 3
end
```

Fast Functions!

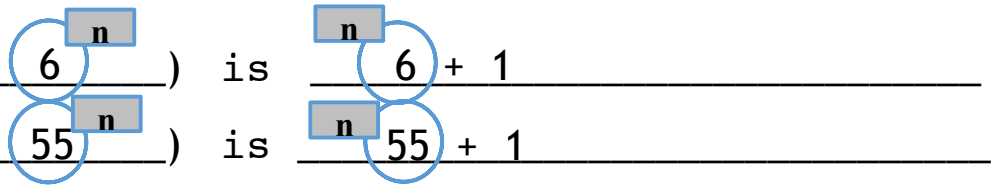
Fill out the contract for each function, then try to write two examples and the definition by yourself.

```
# plus1 : Number -> Number
  name    domain   range
```

examples:

plus1 (6) is 6 + 1

plus1 (55) is 55 + 1



end

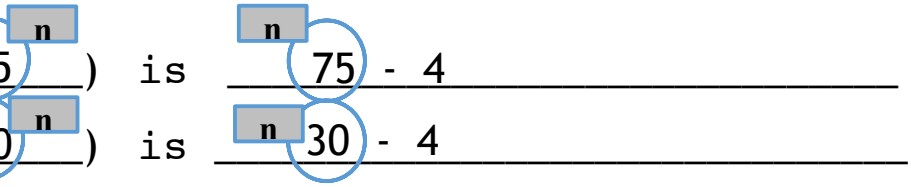
```
fun plus1 ( n ):
    n + 1
end
```

```
# mystery : Number -> Number
  name      domain   range
```

examples:

mystery (75) is 75 - 4

mystery (30) is 30 - 4



end

```
fun mystery ( n ):
    n - 4
end
```

Fast Functions!

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

```
# red-spot : Number -> Image
  name     domain    range
```

```
examples:
   red-spot(20) is circle(20, "solid", "red")
   red-spot(99) is circle(99, "solid", "red")
end

fun red-spot(radius):
   circle(radius, "solid", "red")
end
```

```
# _____ : _____ -> _____
  name    domain    range
```

```
examples:
   _____(_____) is _____
   _____(_____) is _____
end

fun _____(_____):
   _____
end
```

Syntax and Style Bug Hunting: Pyret Edition

#1	`SECONDS = (7)`
	
	`STRING = my string`
		SECONDS = 7
		
		STRING = “my string”

#2	`SHAPE1 = circle(50 “solid” “blue”)`
	
	`SHAPE2 = triangle(75, outline, yellow)`
		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`
		# triple : Number -> Number
		# Multiply a given number by 3 to triple it
		
		examples:
		** triple(5) is 3 * 5**
		** triple(7) is 3 * 7**
		end

#4	`fun triple(n):` `  3 * n`	**fun triple(n) :** ** 3 * n** **end**

#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`
		# ys : Number -> Number
		# Given a number, create a solid yellow star of the given size
		
		examples:
		** ys(99) is star(99, “solid”, “yellow”)**
		** ys(99) is star(99, “solid”, “yellow”)**
		end
		
		ys(size) :
		** star(size, “solid”, “yellow”)**
		end

Unit 2

Word Problem: double-radius

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:

```
# double-radius : Number, String -> Image
    name          Domain           Range
# Consumes a number and a string, produces an outlined circle of the given color, whose
  radius is twice the given number
                What does the function do?
```

Give Examples

Write examples of your function in action

```
examples:
    double-radius (50, "pink") is
                  the user types...
    circle(50 * 2, "outline", "pink")
                  ...which should become

    double-radius (918, "orange") is
                  the user types...
    circle(918 * 2, "outline", "orange")
                  ...which should become
end
```

(radius: 50, color: "pink"; radius: 50, color: "pink"; radius: 918, color: "orange"; radius: 918, color: "orange")

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 double-radius (radius, color) :

    circle(radius * 2, "outline", color)
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:

```
# double-width : Number -> Image
  name           Domain    Range
# Consumes a length and produces a solid green rectangle whose width is twice
  the given length
  What does the function do?
```

Give Examples

Write examples of your function in action

```
examples:
  double-width (45) is
    rectangle(45, 45 * 2, "solid", "green")

  double-width (8) is
    rectangle(8, 8 * 2, "solid", "green")
end
```

(length: the user types... 45 ...which should become 45 and 45 * 2; length: the user types... 8 ...which should become 8 and 8 * 2)

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 double-width (length):

  rectangle(length, length * 2, "solid", "green")
end
```

Word Problem: next-position

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

Contract+Purpose Statement

Every contract has three parts:

```
# next-position : Number, Number -> JumperState
  name            Domain           Range

#Consumes x and y coordinates and produces a JumperState by adding 5 to x
 and subtracting 5 from y   What does the function do?
```

Give Examples

Write examples of your function in action

```
examples:
    next-position(30, 250)        is
      the user types...
                  x   y
    jumper(30 + 5, 250 - 5)
      ...which should become

    next-position(65, 800)        is
      the user types...
                  x   y
    jumper(65 + 5, 800 - 5)
      ...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 next-position(x, y):

    jumper(x + 5, y - 5)
end
```

Data Structure

```
# a CakeType is a flavor, layers, & is-iceCream
data CakeType:
     |  cake(      flavor :: String,
                  layers :: Number,
                  is-iceCream :: Boolean      )
end
```

To make examples of this structure, I would write:

cake1 = _____

cake2 = cake("Red Velvet", 2, true)

To access the fields of **cake2**, I would write:

cake2.flavor

cake2.layers

cake2.is-iceCream

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

```
# taller-than : CakeType, CakeType → Boolean
# Consumes 2 CakeTypes, produces true if the first CakeType has
  more layers than the second
```

Give Examples

Write examples of your function in action

```
examples:
  taller-than (birthday-cake, chocolate-cake) is
```

a-cake1 → birthday-cake; a-cake2 → chocolate-cake

the user types...

```
  birthday-cake.layers > chocolate-cake.layers
```

...which should become

```
  taller-than (strawberry-cake, pb-cake) is
```

a-cake1 → strawberry-cake; a-cake2 → pb-cake

the user types...

```
  strawberry-cake.layers > pb-cake.layers
```

...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 taller-than (a-cake1, a-cake2) :

  a-cake1.layers > a-cake2.layers

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

```
# will-melt : CakeType, Number -> Boolean
# Consumes a CakeType and a temperature, produces true if the temp is greater than 32
  degrees, AND the CakeType is an ice cream cake
```

Give Examples

```
examples:
  will-melt(cake3, 75)        is
    cake3.is-iceCream and (75 > 32)

  will-melt(cake4, 10)        is
    cake4.is-iceCream and (10 > 32)
end
```

(a-cake: cake3, cake4; temp: 75, 10)

Function

```
fun will-melt(a-cake, temp) :
  a-cake.is-iceCream and (temp > 32)
end
```

Vocabulary Practice

Below is a new structure definition:

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

# an example book:
book1 = book("1984", "Orwell", 1949)
```

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 field

book is a constructor

MediaType is a datatype

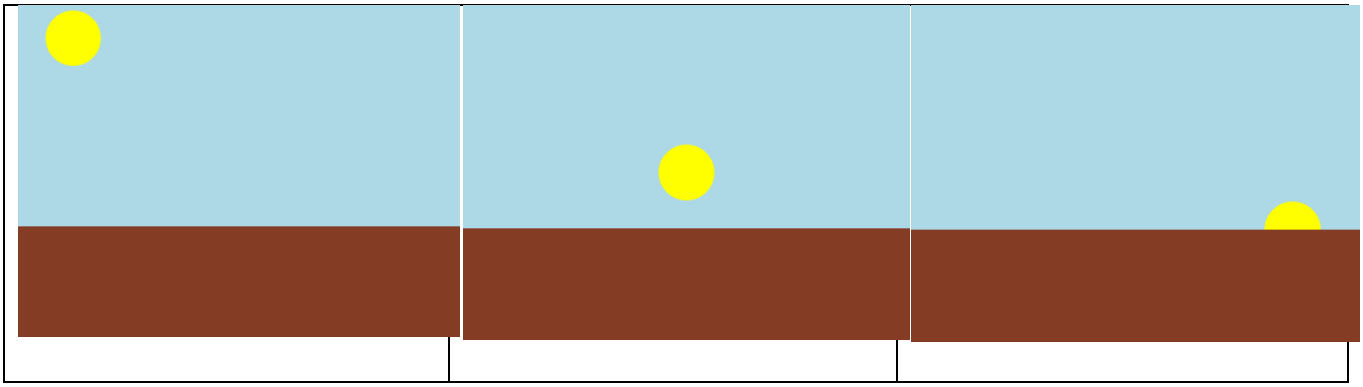
book1 is a instance

title is a field

data ... end is a data block

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?

Thing	Describe how it changes
Sun's x-coordinate	Location increases – moves to the right
Sun's y-coordinate	Location decreases – moves downward

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)
xpos	Number
ypos	Number

(worksheet continues on the next page)

Define the Data Structure

```
# a SunsetState is the x and y location of a sunset

data SunsetState:

   |    sunset( xpos :: Number,
              ypos :: Number )

end
```

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

sunsetA = sunset(50, 270)

sunsetB = sunset(200, 150)

sunsetC = sunset(340, 100)

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 : SunsetState → Image

# Consumes a SunsetState, and produces an image of a sunset
```

Write an expression for each piece of your final image

SUN =	circle(25, "solid", "yellow")
GROUND =	rectangle(400, 100, "solid" "brown")
SKY =	rectangle(400, 300, "solid", "light-blue")

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

```
fun draw-state(a-sunset):

    put-image(GROUND, 200, 50,

        put-image(SUN, a-sunset.xpos, a-sunset.ypos,

                  SKY)))

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

```
# next-state-tick : SunsetState → SunsetState

# Consumes a SunsetState and produces the next SunsetState by moving the
  sun 8 pixels to the right and 4 pixels down
```

Give Examples

Write examples of your function in action

```
examples:
  next-state-tick ( sunset(50, 330) ) is
```

a-sunset

the user types...

```
    sunset(50 + 8, 330 - 4)
```

...which should become

```
  next-state-tick ( sunset(240, 120) ) is
```

a-sunset

the user types...

```
    sunset(240 + 8, 120 - 4)
end
```

...which should become

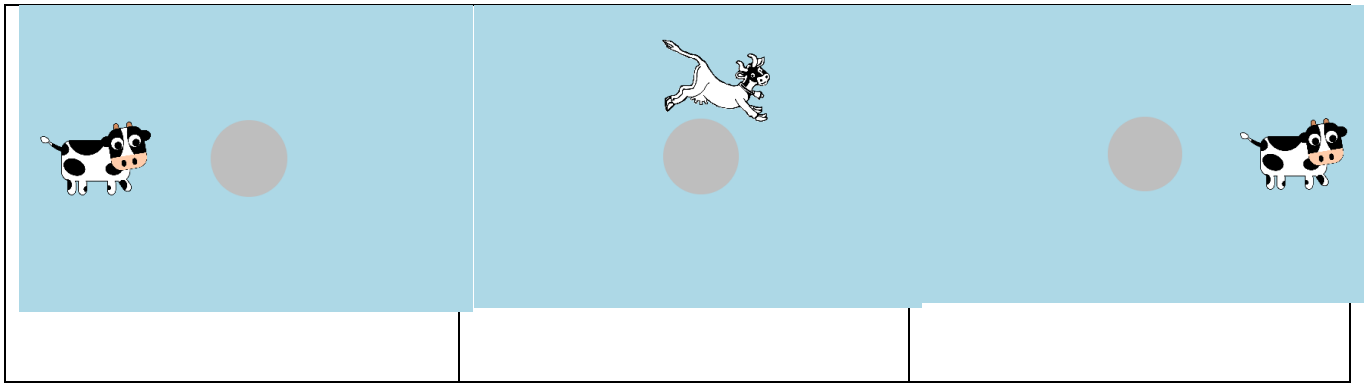
Function

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 next-state-tick ( a-sunset ) :

    sunset(a-sunset.x + 8, a-sunset.y - 4)

end
```

Identifying Animation Data Worksheet

Draw a sketch for three distinct moments of the animation



Sketch A | Sketch B | Sketch C

What things are changing?

Thing	Describe how it changes
Cow's x-coordinate	Location increases – moves to the right
Cow's y-coordinate	Stays consistent until cow reaches the moon, then increases, decreases when cow has finished jumping
Cow image	Switches from normal to jumping cow image when cow is jumping over the moon

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)
xpos	Number
ypos	Number
cow-img	Image

(worksheet continues on the next page)

Define the Data Structure

```
# a CowState is the x and y-position of the cow,
and the cow's image

data CowState:

    | cow(    xpos :: Number,

            ypos :: Number,

            cow-img :: Image)
end
```

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

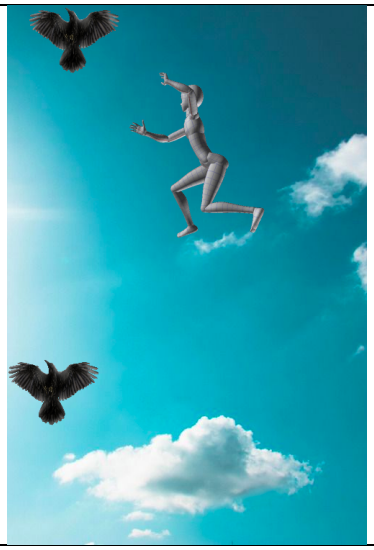
CowA = `cow(100, 200, NORMAL-COW)`

CowB = `cow(300, 300, FLYING-COW)`

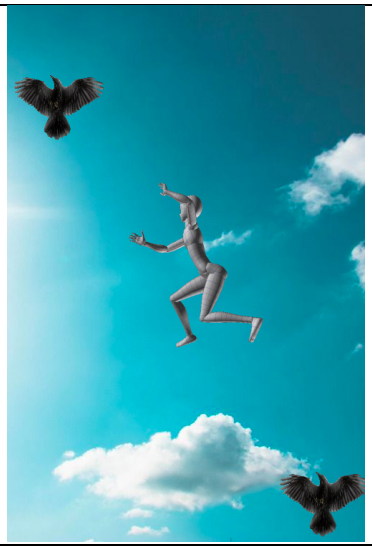
CowC = `cow(400, 200, NORMAL-COW)`

Identifying Animation Data Worksheet

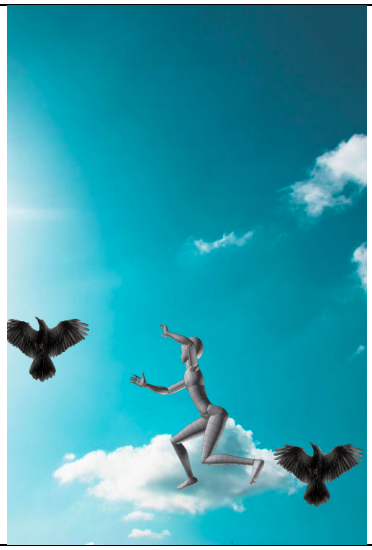
Draw a sketch for three distinct moments of the animation



Sketch A



Sketch B



Sketch C

What things are changing?

Thing	Describe how it changes
Person's y-coordinate	Decreases consistently
Bird1's x coordinate	Increases consistently when onscreen, resets at random when it resets
Bird1's y-coordinate	Increases consistently
Bird2's x-coordinate	Increases consistently when onscreen, resets at random when it resets
Bird2's y-coordinate	Increases consistently

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)
persony	Number
bird1x	Number
bird1y	Number
bird2x	Number
bird2y	Number

(worksheet continues on the next page)

Define the Data Structure

```
# a FallingState is a person's x-coordinate, and
the x and y coordinates of 2 flying birds

data FallingState:

      | fall(   persony :: Number,
              bird1x :: Number,
              bird1y :: Number,
              bird2x :: Number,
              bird2y :: Number )
end
```

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

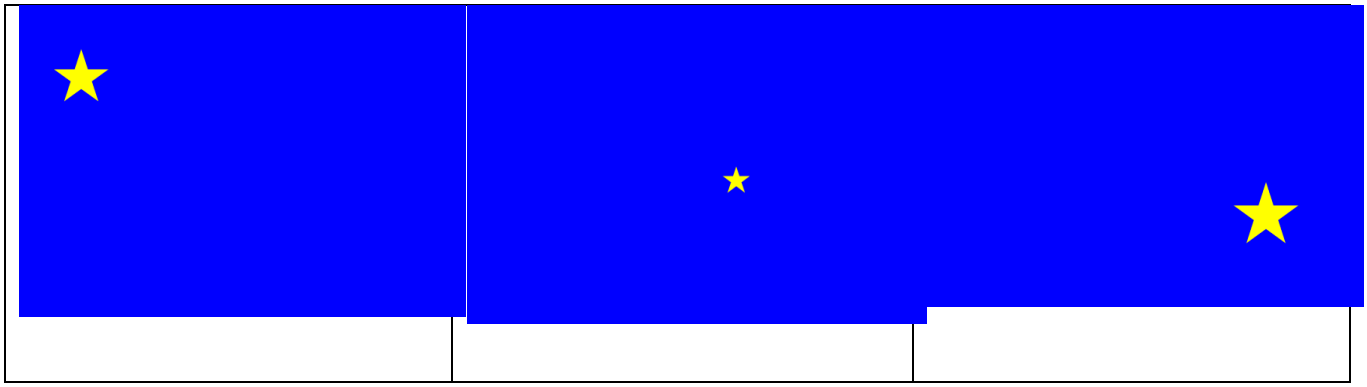
FallingA = `fall(600, 100, 700, 75, 250)`

FallingB = `fall(450, 100, 650, 330, 100)`

FallingC = `fall(300, 440, 97, 60, 270)`

Identifying Animation Data Worksheet

Draw a sketch for three distinct moments of the animation



Sketch A Sketch B Sketch C

What things are changing?

Thing	Describe how it changes
Star's x-coordinate	Increases consistently
Star's y-coordinate	Decreases consistently
Star's size	Increases until a set size, then decreases
Star's growth rate	Either negative or positive, based on size of the star

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)
xpos	Number
ypos	Number
star-size	Number
growth-rate	Number

(worksheet continues on the next page)

Define the Data Structure

```
# a StarState is the x and y-coordinate of the
star, its size, and growth rate

data StarState:

     | pstar(   xpos :: Number,
              ypos :: Number,
              star-size :: Number,
              growth-rate :: Number        )
end
```

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

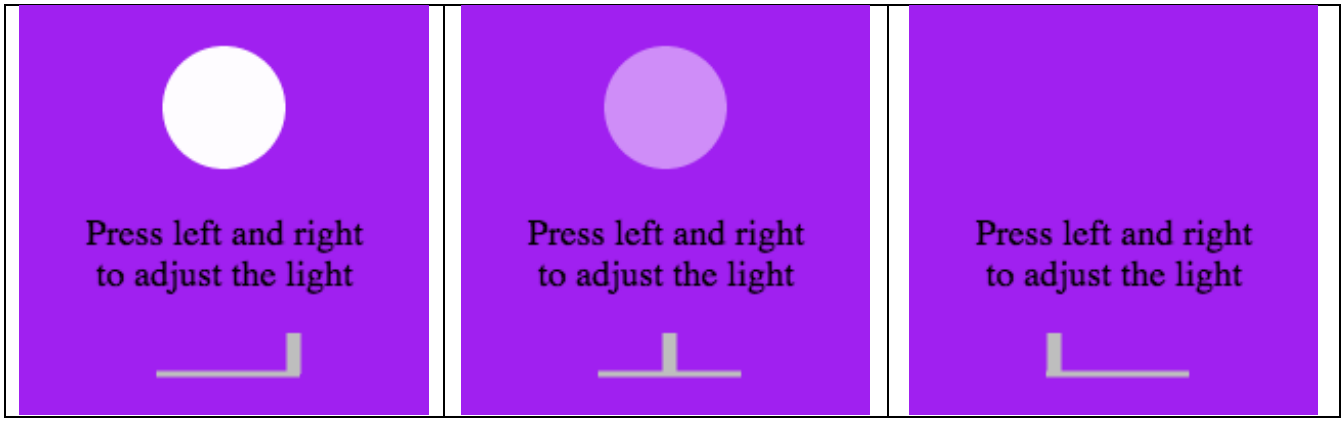
PulsingStarA = `pstar(100, 400, 60, 2)`

PulsingStarB = `pstar(450, 200, 30, -2)`

PulsingStarC = `pstar(600, 130, 60, 2)`

Identifying Animation Data Worksheet

Draw a sketch for three distinct moments of the animation



Sketch A Sketch B Sketch C

What things are changing?

Thing	Describe how it changes
Dimmer switch x-coordinate	Increases and decreases when arrow keys are pressed
Light brightness/circle opacity	Increases and decreases based on dimmer switch's x-coordinate

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)
xpos	Number

(worksheet continues on the next page)

Define the Data Structure

```
# a DimmerState is the x-coordinate of a light
dimmer switch

data DimmerState:

    | dimmer(    xpos :: Number
                 _____________________________________
                 _____________________________________
                 _____________________________________)
end
```

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

LightDimmerA = `dimmer(60)`

LightDimmerB = `dimmer(30)`

LightDimmerC = `dimmer(0)`

Word Problem: location

Write a function called *location*, which consumes a JumperState, and produces a String representing the jumper's location: either "cliff", "beach", "water", or "air".

Contract+Purpose Statement

```
# location : JumperState → String
# Consumes a JumperState and produces the location of the
  jumper: either "cliff", "beach", "water", or "air"
```

Give Examples

```
examples:
   location( jumper(100, 400) ) is "air"
   location ( jumper(321, 200) ) is "beach"
   location ( jumper(50, 75) ) is "water"
   location ( jumper(600, 240) ) is "cliff"

end
```

a-jumper

location-name

(worksheet continues next page)

Function

```
fun location(a-jumper):
    if (a-jumper.x <= 320) and (a-jumper.y <= 240):
        "water"
    else if (a-jumper.x > 320) and (a-jumper.y <= 240):
        "beach"
    else if (a-jumper.x >= 300) and (a-jumper.y <= 380):
        "cliff"
    else: "air"

    end
end
```

Syntax and Style Bug-Hunting: Piecewise Edition

	Buggy Code	Correct Code / Explanation

Round 1	`fun piecewisefun(n):`
	`  if (n > 0): n`
	`  else: 0`
		No 'end' statements
		
		`fun piecewisefun(n):`
		`  if (n > 0): n`
		`  else: 0`
		`  end`
		`end`

Round 2	`fun cost(topping):`
	`  if string-equal(topping, "pepperoni"): 10.50`
	`  else string-equal(topping, "cheese"): 9.00`
	`  else string-equal(topping, "chicken"): 11.25`
	`  else string-equal(topping, "broccoli"): 10.25`
	`  else: "That's not on the menu!"`
	`  end`
	`end`
		Need to use 'else if' for all but the final condition
		
		`fun cost(topping):`
		`  if string-equal(topping, "pepperoni"): 10.50`
		`  else if string-equal(topping, "cheese"): 9.00`
		`  else if string-equal(topping, "chicken"): 11.25`
		`  else if string-equal(topping, "broccoli"): 10.25`
		`  else: "That's not on the menu!"`
		`  end`
		`end`

Round 3	`fun absolute-value(a b):`
	`  if a > b: a - b`
	`  b - a`
	`  end`
	`end`
		No 'else' statement before final condition
		
		`fun absolute-value(a b):`
		`  if a > b: a - b`
		`  else: b - a`
		`  end`
		`end`

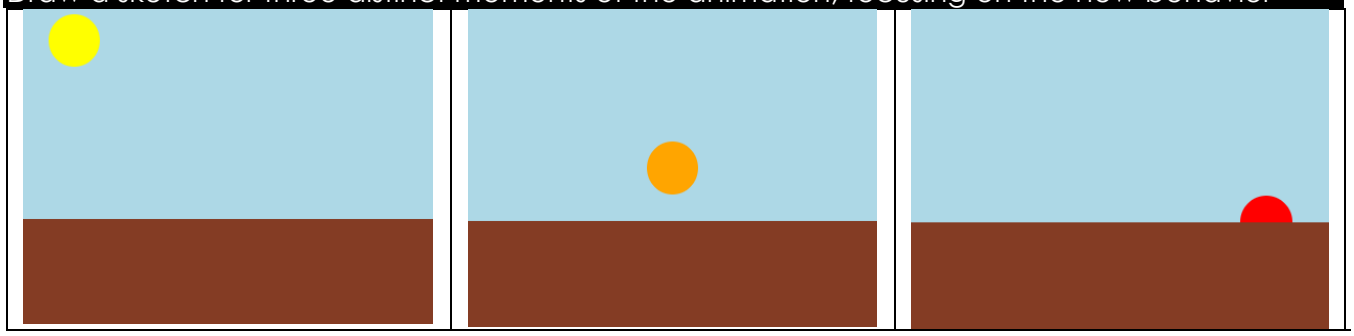
Round 4	`fun best-function(f):`
	`  if string-equal(f, "blue"):`
	`  "you win!"`
	`  else if string-equal(f, "blue"):`
	`  "you lose!"`
	`  else if string-equal(f, "red"):`
	`  "Try again!"`
	`  else: "Invalid entry!"`
	`  end`
	`end`
		First and second condition are the same
		(This program will run, but the second condition will never be evaluated)
		
		`if string-equal(f, "blue"): "you win!"`
		`else if string-equal(f, "green"): "you lose!"`

Animation Extension Worksheet

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

Make the sun change colors- yellow at the top of the screen, orange in the middle, and red at the bottom (the horizon)

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



Sketch A | Sketch B | Sketch C

What NEW things are changing? Are they independent of existing fields?

Thing	Describe how it changes
Sun's color	Changes from yellow to orange to red, based on sun's y-coordinate

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)

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

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

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

top-left = sunset(50, 270)

middle = sunset(200, 150)

bottom-right = sunset(340, 100)

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

```
# draw-sun : SunsetState → Image
# Consumes a SunsetState and produces a circle whose color
  ("yellow", "orange", or "red") is based on the sunset's y coordinate
```

Give Examples

a-sunset

sun-color

```
examples:
  draw-sun ( sunset(50, 270)    ) is circle(25, "solid", "yellow")
  draw-sun ( sunset(200, 150)   ) is circle(25, "solid", "orange")
  draw-sun ( sunset(340, 100)   ) is circle(25, "solid", "red")
  draw-sun ( sunset(400, 0)     ) is circle(25, "solid", "red")

end
```

(worksheet continues next page)

Function

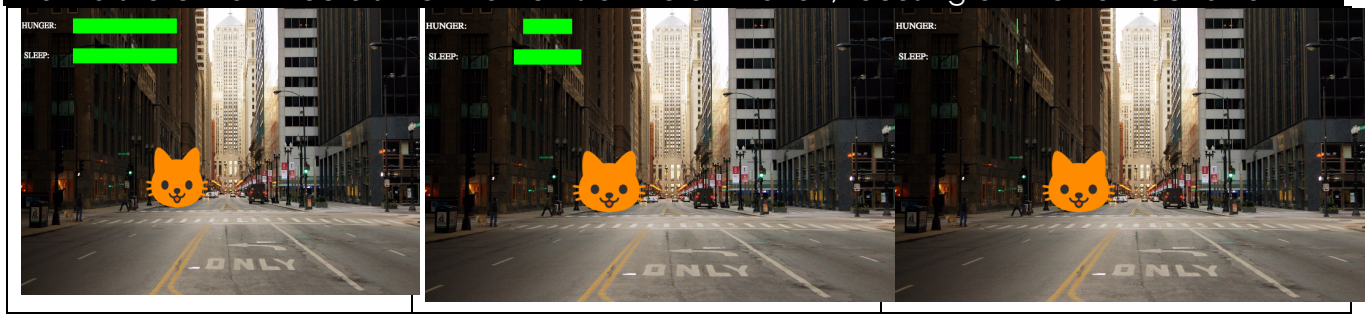
```
fun ____draw-sun____(____a-sunset____) :

    if ___a-sunset.y >= 225_____:

        __circle(25, “solid”, “yellow”)_____

    else if
_____a-sunset.y >= 150_____:

        ___circle(25, “solid”, “orange”)_____

    else: __circle(25, “solid”, “red”)_____

    end
end
```

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 | Sketch B | Sketch C

What NEW things are changing? Are they independent of existing fields?

Thing	Describe how it changes
Hunger level	Decreases by 2 each tick
Sleep level	Decreases by 1 each tick

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)

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

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

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

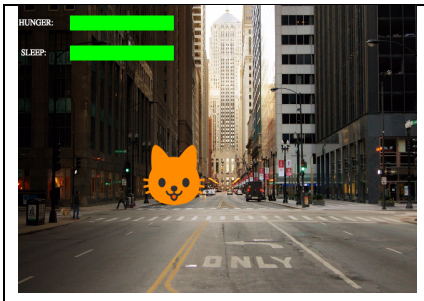
If you have another function on your To-Do list , write at least one NEW example

Animation Extension Worksheet

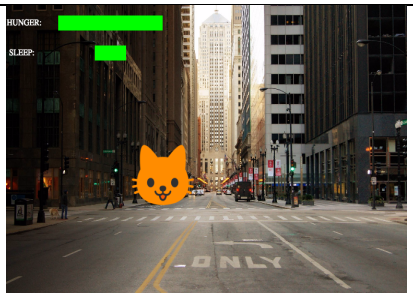
Describe the goal of your change: what new feature or behavior will it add to your animation?

On a keypress, if the user pressed "f" (for "feed"), hunger should increase by 10. If the user pressed "s" (for "sleep"), sleep should increase by 5. If the user presses any other keys, nothing should change.

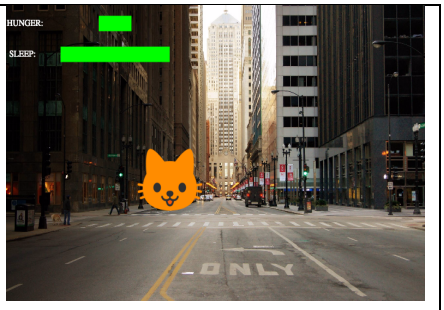
Draw a sketch for three distinct moments of the animation, focusing on the new behavior



Sketch A



Sketch B



Sketch C

What NEW things are changing? Are they independent of existing fields?

Thing	Describe how it changes
Hunger	Increases by 10 if 'f' key is pressed
Sleep	Increases by 5 if 's' key is pressed

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)

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

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

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-key(FULLPET, “f”) is FULLPET

next-state-key(FULLPET, “s”) is FULLPET

next-state-key(MIDPET, “f”) is pet(50 + 10, 75)

next-state-key(MIDPET, “s”) is pet(50, 75 + 5)

next-state-key(LOSEPET, “s”) is LOSEPET

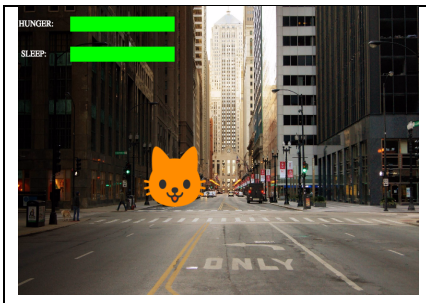
If you have another function on your To-Do list , write at least one NEW example

Animation Extension Worksheet

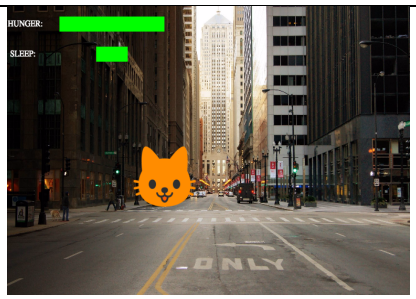
Describe the goal of your change: what new feature or behavior will it add to your animation?

When either of the pet's hunger or sleep levels reaches 0, the game is lost and the pet is sad- the happy pet image is replaced with a sad pet image

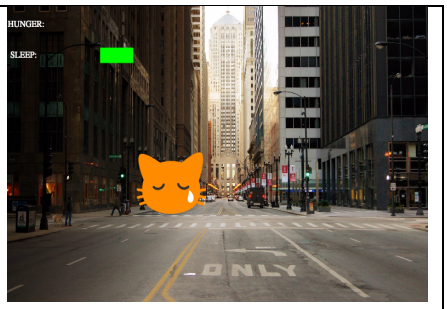
Draw a sketch for three distinct moments of the animation, focusing on the new behavior



Sketch A



Sketch B



Sketch C

What NEW things are changing? Are they independent of existing fields?

Thing	Describe how it changes
Cat's image	Changes from happy to sad cat image when either of the cat's levels reaches 0

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)

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

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

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

draw-state(LOSEPET) is put-image(...

[put hunger and sleep bars at current position on screen] …

put-image(SADCAT, 300, 200, BACKGROUND)))))

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

has-lost : PetState -> Boolean

helper function which returns true when hunger or sleep reaches 0

examples:

has-lost(FULLPET) is false

has-lost(LOSEPET) is true

end

Build Your Own Animation

Animation Design Worksheet

Draw a sketch for three distinct moments of the animation

Sketch A Sketch B Sketch C

What things are changing?

Thing	**Describe how it changes**

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)

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

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

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:

_____

_____

_____

_____

_____

Collision

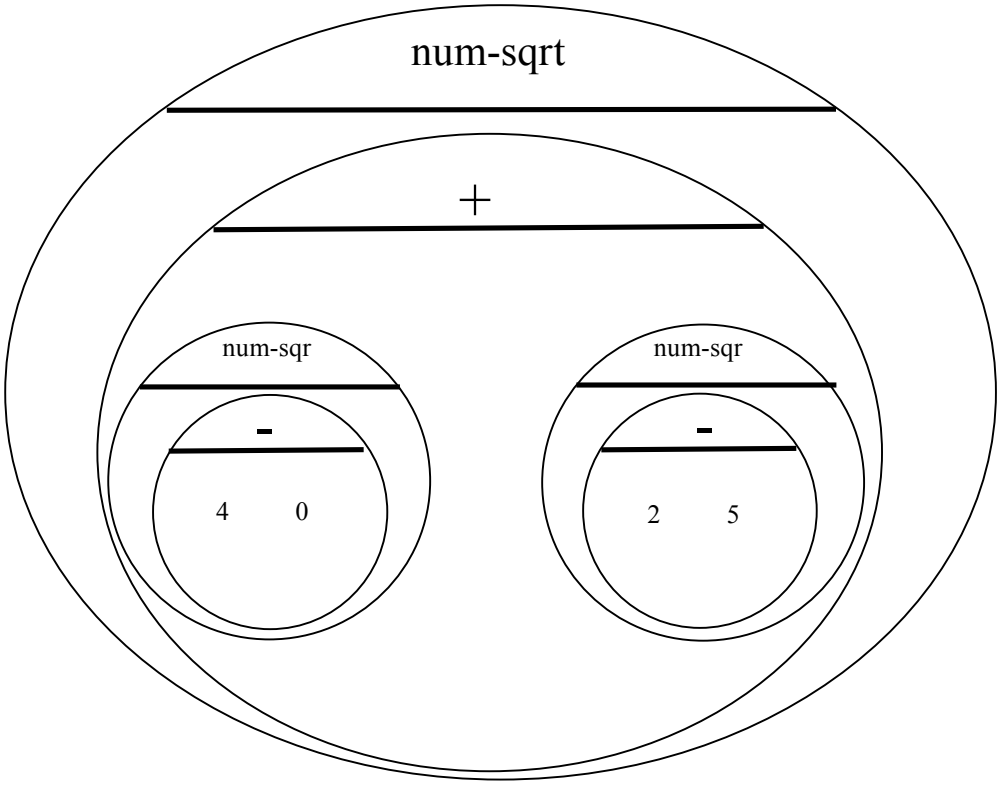
Distance:

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:

```
num-sqrt(num-sqr(4 - 0) + num-sqr(2 - 5))
```

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:

$$Distance^2 = (px - cx)^2 + (py - cy)^2$$

Contract+Purpose Statement

```
# distance : Number, Number, Number, Number -> Number
  Consumes the coordinates of 2 characters: px, py, cx, and cy, produces
# the distance between them using the distance formula
```

Give Examples

Write examples of your function in action

```
examples:
    distance(4, 2, 0, 5)        is
      num-sqrt(num-sqr(4 - 0) + num-sqr(2 - 5))

    distance(80, 33, 6, 50)     is
      num-sqrt(num-sqr(80 - 6) + num-sqr(33 - 50))
end
```

px py cx cy

Function

```
fun distance(px, py, cx, cy):

    num-sqrt(num-sqr(px - cx) + num-sqr(py - cy))


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

```
# is-collision : Number, Number, Number, Number -> Boolean
# Consumes the coordinates of 2 characters: px, py, cx, and cy, produces
# true if the distance between them is less than 50 pixels
```

Give Examples

Write examples of your function in action

```
examples:
    is-collision (78, 30, 9, 2) is
                 distance(78, 30, 9, 2) < 50
    is-collision (20, 40, 60, 80) is
                 distance(20, 40, 60, 80) < 50
end
```

Function

```
fun is-collision(px, py, cx, cy):
    distance(px, py, cx, cy) < 50
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:
    _____(_____)        is
          the user types…

    _____
                ...which should become

    _____(_____)        is
          the user types…

    _____
                ...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:
    _____(_____)        is
```

the user types…

_____

…which should become

```
    _____(_____)        is
```

the user types…

_____

…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

Sketch A Sketch B Sketch C

What things are changing?

Thing	Describe how it changes

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)

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

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

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

Sketch A Sketch B Sketch C

What things are changing?

Thing	Describe how it changes

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)

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

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

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

Sketch A Sketch B Sketch C

What things are changing?

Thing	**Describe how it changes**

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)

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

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

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

Sketch A	Sketch B	Sketch C

What things are changing?

Thing	Describe how it changes

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)

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

Component	***When is there work to be done?***	**To-Do**	**Done**
Data Structure	*If any new field(s) were added, changed or removed*	☐	☐
draw-state	*If something is displayed in a new way or position*	☐	☐
next-state-tick	*If the Data Structure changed, or the animation happens automatically*	☐	☐
next-state-key	*If the Data Structure changed, or a keypress triggers the animation*	☐	☐
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?

Thing	Describe how it changes

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

Field name (dangerX, score, playerIMG…)	**Datatype** (Number, String, Image, Boolean…)

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

Component	***When is there work to be done?***	**To-Do**	**Done**
Data Structure	*If any new field(s) were added, changed or removed*	☐	☐
draw-state	*If something is displayed in a new way or position*	☐	☐
next-state-tick	*If the Data Structure changed, or the animation happens automatically*	☐	☐
next-state-key	*If the Data Structure changed, or a keypress triggers the animation*	☐	☐
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

_____

_____

_____

_____

_____

Contracts

Name	Domain	Range	example
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	

Contracts

Name	Domain	Range	example
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	
#	:	→	