­­

­Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Bootstrap:Reactive



­

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

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

|  |  |  |
| --- | --- | --- |
|  | **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:  **n**  **n**  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  2 \* 5  5 |
| \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **n**  **n**  2 \* 7  7  double  double  end  fun \_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_): |
| 2 \* n  n  double  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  end |
| # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Number  Number  triple  name domain range |
| examples:  **n**  **n**  9  9 \* 3  triple  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **n**  **n**  40 \* 3 |
| \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  triple  40  end  n  triple  fun \_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_): |
| n \* 3  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  end |

# Fast Functions!

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

**n**

**n**

**n**

**n**

n

55

6

plus1

n + 1

55 + 1

6 + 1

Number

Number

plus1

plus1

plus1

**n**

**n**

**n**

**n**

n

75 - 4

30

75

mystery

Number

Number

mystery

mystery

30 - 4

mystery

n - 4

|  |  |
| --- | --- |
| # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  name domain range | |
| examples:  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  end  fun \_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_): |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  end |
| # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  name domain range | |
| examples:  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  end  fun \_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_): |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  end |

# Fast Functions!

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

|  |  |
| --- | --- |
| # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Image  Number  red-spot  name domain range | |
| examples:  **radius**  **radius**  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  circle(20, “solid”, “red”)  20  red-spot |
| \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  circle(99, “solid”, “red”)  **radius**  99  red-spot  end  **radius**  fun \_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_):  radius  red-spot |
| circle(radius, “solid”, “red”)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  end |
| # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  name domain range | |
| examples:  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  end  fun \_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_): |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  end |

# 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:

Image

Number, String

double-radius

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -> \_\_\_\_\_\_\_\_\_\_\_\_

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

**radius**

examples:

**color**

50, “pink”

double-radius

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

the user types…

**radius**

**color**

circle(50 \* 2, “outline”, “pink”)

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

...which should become

**color**

**radius**

double-radius

918, “orange”

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

the user types…

**color**

**radius**

circle(918 \* 2, “outline”, “orange”)

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

...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!

double-radius

radius, color

fun \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) :

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:

Image

Number

double-width

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -> \_\_\_\_\_\_\_\_\_\_\_\_

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:

**length**

45

double-width

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

the user types…

**length**

**length**

rectangle(45, 45 \* 2, “solid”, “green”)

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

...which should become

**length**

8

double-width

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

the user types…

**length**

**length**

rectangle(8, 8 \* 2, “solid”, “green”)

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

...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!

length

double-width

fun \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) :

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:

JumperState

Number, Number

next-position

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -> \_\_\_\_\_\_\_\_\_\_\_\_

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:

**x**

**y**

30, 250

next-position

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

the user types…

**y**

jumper(30 + 5, 250 – 5)

**x**

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

...which should become

**y**

**x**

65, 800

next-position

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

the user types…

jumper(65 + 5, 800 – 5) – 5)

**y**

**x**

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

...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!

x , y

next-position

fun \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) :

jumper(x + 5, y – 5) – 5)

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

end

# Data Structure

# a CakeType is a **flavor, layers, & is-iceCream**

data **CakeType**:

flavor :: String,

| **cake**(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

layers :: Number,

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

is-iceCream :: Boolean

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

end

To make examples of this structure, I would write:

**cake1** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

cake(“Red Velvet”, 2, true)

**cake2** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

cake2.flavorr

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

cake2.layersr

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

cake2.is-iceCreamr

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

# 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

CakeType, CakeType

Boolean

taller-than

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_

Consumes 2 CakeTypes, produces true if the first CakeType has more layers than the second

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Give Examples

Write examples of your function in action

**a-cake2**

**a-cake1**

birthday-cake, chocolate-cake

taller-than

examples:

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

the user types…

**a-cake2**

**a-cake1**

birthday-cake.layers > chocolate-cake.layers

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

...which should become

**a-cake2**

**a-cake1**

strawberry-cake, pb-cake

taller-than

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

the user types…

**a-cake2**

**a-cake1**

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!

a-cake1, a-cake2

taller-than

fun \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) :

a-cake1.layers > a-cake2.layers

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

end

cake(“Vanilla”, 4, false)

# 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

CakeType, Number

Boolean

will-melt

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -> \_\_\_\_\_\_\_\_\_\_\_\_

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:

**temp**

**a-cake**

will-melt

cake3, 75

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

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

**temp**

**temp**

**a-cake**

**a-cake**

cake3.is-iceCream and (75 > 32)

cake4, 10

will-melt

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

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

**temp**

**a-cake**

cake4.is-iceCream and (10 > 32)

end

# Function

a-cake, temp

will-melt

fun \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) :

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

a-cake.is-iceCream and (temp > 32)

end

# Unit 3

# 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

ypos :: Number

xpos :: Number,

# a Sunset**State** is **the x and y location of a sunset**

data **SunsetState**:

| **sunset**(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

end

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

sunset(50, 270)

**sunsetA** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

sunset(200, 150)

**sunsetB** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

sunset(340, 100)

**sunsetC** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 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

SunsetState

# draw-state : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 Image

Consumes a SunsetState, and produces an image of a sunset

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Write an expression for each piece of your final image

rectangle(400, 300, “solid”, “light-blue”)

|  |  |
| --- | --- |
| Sun | circle(25, “solid”, “yellow”) |
| Ground | rectangle(400, 100, “solid” “brown”) |
| Sky |  |

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

a-sunset

draw-state

fun \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) :

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

put-image(rectangle(400, 100, “solid” “brown”), 200, 50,

put-image(circle(25, “solid”, “yellow”), a-sunset.xpos, a-sunset.ypos,

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

rectangle(400, 300, “solid”, “light-blue”)))

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

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

SunsetState

SunsetState

next-state-tick

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_

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

**a-sunset**

sunset(50, 330)

next-state-tick

examples:

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

the user types…

sunset(50 + 8, 330 - 4)

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

...which should become

**a-sunset**

sunset(240, 120)

next-state-tick

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

the user types…

sunset(240 + 8, 120 - 4)

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

...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!

a-sunset

next-state-tick

fun \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) :

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

cow-img :: Image

ypos :: Number,

xpos :: Number,

# a Cow**State** isthe x and y-position of the cow, and the cow’s image

data Cow**State**:

| cow(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

end

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

cow(100, 200, NORMAL-COW)

**CowA** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

cow(300, 300, FLYING-COW)

**CowB** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

cow(400, 200, NORMAL-COW)

**CowC** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 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

bird2x :: Number,

bird2y :: Number

bird1y :: Number,

bird1x :: Number,

persony :: Number,

# a **FallingState** isa person’s x-coordinate, and the x and y coordinates of 2 flying birds

data **FallingState**:

| fall(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

end

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

fall(600, 100, 700, 75, 250)

**FallingA** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

fall(450, 100, 650, 330, 100)

**FallingB** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

fall(300, 440, 97, 60, 270)

**FallingC** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 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

growth-rate :: Number

star-size :: Number,

ypos :: Number,

xpos :: Number,

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

data **StarState**:

| pstar(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

end

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

pstar(100, 400, 60, 2)

**PulsingStarA** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pstar(450, 200, 30, -2)

**PulsingStarB** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pstar(600, 130, 60, 2)

**PulsingStarC** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# 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

xpos :: Number

# a **DimmerState** isthe x-coordinate of a light dimmer switch

data **DimmerState**:

| **dimmer**(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

end

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

dimmer(60)

**LightDimmerA** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dimmer(30)

**LightDimmerB** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dimmer(0)

**LightDimmerC** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Unit 4

# 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

String

JumperState

location

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_

Consumes a JumperState and produces the location of the jumper: either “cliff”, “beach”, “water”, or “air”

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Give Examples

**location-name**

**a-jumper**

jumper(600, 240)

jumper(50, 75)

jumper(321, 200)

“cliff”

“water”

“beach”

“air”

jumper(100, 400)

location

location

location

location

examples:

\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

end

(worksheet continues next page)

## Function

location

a-jumper

(a-jumper.x <= 320) and (a-jumper.y <= 240)

fun \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) :

if \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

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

“beach”

“water”

(a-jumper.x > 320) and (a-jumper.y <= 240)

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

(a-jumper.x >= 300) and (a-jumper.y <= 380)

else if \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

“cliff”

else if \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

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

“air”

else: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

end

­end

# Piecewise Bug-Hunting

|  |  |  |
| --- | --- | --- |
|  | **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(200, 150)

sunset(50, 270)

**middle** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

sunset(340, 100)

**bottom-right** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 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

Image

SunsetState

draw-sun

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_

Consumes a SunsetState and produces a circle whose color (“yellow”, “orange”, or “red”) is based on the sunset’s y coordinate

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Give Examples

**sun-color**

circle(25, “solid”, “yellow”)

**a-sunset**

sunset(340, 100)

sunset(400, 0)

circle(25, “solid”, “orange”)

circle(25, “solid”, “red”)

circle(25, “solid”, “red”)

sunset(200, 150)

sunset(50, 270)

draw-sun

draw-sun

draw-sun

draw-sun

examples:

\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

end

(worksheet continues next page)

## Function

a-sunset

draw-sun

a-sunset.y >= 225

fun \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) :

if \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

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

a-sunset.y >= 150

circle(25, “solid”, “orange”)”)

circle(25, “solid”, “yellow”)

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

else if \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

circle(25, “solid”, “red”)”)

else: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

end

­end

# Unit 5

# 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. Make the green bars get smaller based on hunger and sleep levels.

## 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(50, 75)

pet(100, 100)

**MIDPET** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pet(0, 0)

**LOSEPET** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 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(50, 75)

pet(100, 100)

**MIDPET** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pet(0, 0)

**LOSEPET** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 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(50, 75)

pet(100, 100)

**MIDPET** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pet(0, 0)

**LOSEPET** = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 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:

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

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

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

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

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

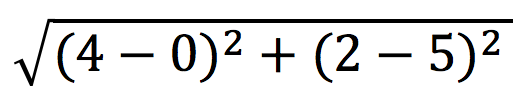
# Collision

# 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:



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

num-sqrt

+

num-sqr

num-sqr

-

-

4 0

2 5

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:*

Distance2 = (px - cx)2 + (py - cy)2

## Contract+Purpose Statement

Number, Number, Number, Number

Number

distance

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -> \_\_\_\_\_\_\_\_\_\_\_\_\_\_

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:

**cy**

**cx**

**py**

**px**

4, 2, 0, 5

distance

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_) is

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

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

**cy**

**cx**

**py**

**px**

80, 33, 6, 50

distance

\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_) is

num-sqrt(num-sqr(80 - 6) + num-sqr(33 - 50))

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

end

## Function

distance

px, py, cx, cy

fun \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) :

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

Number, Number, Number, Number

Boolean

is-collision

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ -> \_\_\_\_\_\_\_\_\_\_\_\_\_\_

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:

**py**

**cy**

**cx**

**px**

78, 30, 9, 2

is-collision

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

distance(78, 30, 9, 2) < 50

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

**py**

**cy**

**cx**

**px**

20, 40, 60, 80

is-collision

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) is

distance(20, 40, 60, 80) < 50

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

end

## Function

is-collision

px, py, cx, cy

fun \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) :

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

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

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

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

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

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

