

Lesson 1

Circles Competit

Time: 5 minutes

	Math	Circle of Evaluation	Scheme Code
Round 1	$(1 + 2) - (3 * 7)$		$(- (+ 1 2) (* 3 7))$
Round 2	$3 - (1 + 2)$		$(- 3 (+ 1 2))$
Round 3	$3 - (1 + (5 * 6))$		$(- 3 (+ 1 (* 5 6)))$
Round 4	$(1 + (5 * 6)) - 3$		$(- (+ 1 (* 5 6)) 3)$

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

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

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

(define (double n) (* 2 n))

; triple : # -> #

name domain range

(EXAMPLE (triple 7) (* 3 7))

(EXAMPLE (triple 9) (* 3 9))

(define (triple n) (* 3 n))

; plus1 : # -> #

(EXAMPLE (plus1 4) (+ 1 4))

(EXAMPLE (plus1 13) (+ 1 13))

(define (plus1 n) (+ 1 n))

; mystery : # -> #

(EXAMPLE (mystery 5) (- 7 4))

(EXAMPLE (mystery 8) (- 7 8))

(define (mystery x) (- 7 x))

Lesson 2

DESIGN RECIPE: SAMPLE

State the problem:

Make a circle (spot) of size 100, with the provided color

Contract+Purpose Statement

Every contract has three parts:

; spot100 : String -> Image
name Domain Range

; Makes a size 100 circle in a color
What does the function do?

Give Examples

On the computer, write an example of your function in action, using EXAMPLE.

(EXAMPLE (spot100 “green”)(circle 100 “solid” “green”))
the user types... ...which should become

(EXAMPLE (spot100 “blue”)(circle 100 “solid” “blue”))
the user types... ...which should become

color

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!

(define (spot100 color) (circle 100 “solid” color))
function name variable names ...and the computer does this

DESIGN RECIPE: DOUBLE-RADIUS

State the problem:

Write a function, `double-radius`, which takes in a radius and color, and makes an outline circle of the same color, but double the size

Contract+Purpose Statement

Every contract has three parts:

`; double-radius _____ : Number String _____ -> Image _____`
name Domain Range

`; Doubles the radius, and draws a circle of that new size and color`
What does the function do?

Give Examples

On the computer, write an example of your function in action, using EXAMPLE.

EXAMPLE `(double-radius 6 "blue") (circle (* 2 6) "outline" "blue")`
the user types... ..which should become

(EXAMPLE `(double-radius 10 "red") (circle (* 2 10) "outline" "red")`)
the user types... ..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!

`(define (double-radius radius color) (circle (* 2 radius) "outline" color))`

DESIGN RECIPE: DOUBLE-WIDTH

State the problem:

Write a function, `double-width`, which takes in a height and a color, and makes a solid rectangle, where the width is twice the height

Contract+Purpose Statement

Every contract has three parts:

```

; double-width : Number String _____ -> Image
    name                                Domain                        Range

```

; draws a rectangle where the width is twice the height, of the given color
 What does the function do?

Give Examples

On the computer, write an example of your function in action, using EXAMPLE.

```

(EXAMPLE (double-width 3 "green") (rectangle 3 (* 2 3) "solid" "green"))
        the user types...                    ...which should become

```

```

(EXAMPLE (double-width 7 "yellow") (rectangle 7 (* 2 7) "solid" "yellow"))
        the user types...                    ...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!

```

(define (double-width h color) (rectangle h (* 2 h) "solid" color))

```


DESIGN RECIPE: PAINT-JOB

State the problem:

Write a function called paint-job that takes an auto and changes its color

Contract+Purpose Statement

; paint-job _____ : auto string _____ -> auto _____
name Domain Range

; takes in an auto and a color, and returns an auto with that color

What does the function do?

Give Examples

(EXAMPLE (paint-job _____ car1 "green" _____))
(make-auto (auto-model car1) _____)
_____ (auto-hp car1) _____
_____ (auto-rims car1) _____
_____ "green" _____
_____ (auto-value car1) _____))

(EXAMPLE (paint-job _____ car2 "black" _____))
(make-auto (auto-model car2) _____)
_____ (auto-hp car2) _____
_____ (auto-rims car2) _____
_____ "black" _____
_____ (auto-value car2) _____))

Function

(define (paint-job _____ auto color _____))
(make-auto (auto-model auto) _____)
_____ (auto-hp auto) _____
_____ (auto-rims auto) _____
_____ color _____
_____ (auto-value auto) _____))

DESIGN RECIPE: TURBO-CHARGE

State the problem:

Write a function called turbo-charge that takes an auto and adds 20 hp

Contract+Purpose Statement

```
; _turbo-charge_____ : _auto_____ -> _auto_____
   name                               Domain                               Range
; takes in an auto and returns the same auto, with twenty more hp_____
   What does the function do?
```

Give Examples

(EXAMPLE (_turbo-charge__ car1_____))

```
(make-auto _(auto-model car1)_____
            _(+ 20 (auto-hp car1))_____
            _(auto-rims car1)_____
            _(auto-color car1)_____
            _(auto-value car1) )_____
```

(EXAMPLE (_turbo-charge__ car2_____))

```
(make-auto _(auto-model car2)_____
            _(+ 20 (auto-hp car2))_____
            _(auto-rims car2)_____
            _(auto-color car2)_____
            _(auto-value car2) )_____
```

Function

```
(define (_turbo-charge__ _auto_____)
```

```
(make-auto _(auto-model auto)_____
            _(+ 20 (auto-hp auto))_____
            _(auto-rims auto)_____
            _(auto-color auto)_____
            _(auto-value auto) )_____
```

Lesson 3

DESIGN RECIPE: PIMP

State the problem:

+100 hp, red, +10000 value, 30" rims

Contract+Purpose Statement

; __pimp__ : __auto__ -> __auto__
name Domain Range

; takes in an auto and makes it red, 30" rims, +100 hp and +10000 value
What does the function do?

Give Examples

(EXAMPLE (__pimp__ car1))

_(make-auto (auto-model car1))
(+ 100 (auto-hp car1))
30
"red"
(+ 10000 (auto-value car1)))

(EXAMPLE (__pimp__ car2))

_(make-auto (auto-model car2))
(+ 100 (auto-hp car2))
30
"red"
(+ 10000 (auto-value car2)))

Function

(define (__pimp__ __auto__))

_(make-auto (auto-model auto))
(+ 100 (auto-hp auto))
30
"red"
(+ 10000 (auto-value auto)))

DEFINE-STRUCT

Autos:

; an auto is a (make-auto string number number string number)

```
(define-struct auto (model_____
                    hp_____
                    rims_____
                    color_____
                    value_____))
```

; a party is a string string number

```
(define-struct party (location_____
                    theme_____
                    guests_____))
```

DESIGN RECIPE: RSVP

State the problem:

Add 1 to the # of guests

Contract+Purpose Statement

; RSVP _____ : party _____ -> party _____
name Domain Range

; Add 1 to the number of guests in the party _____
What does the function do?

Give Examples

(EXAMPLE (RSVP _____ Halloween _____))
_____(make-party (party-location Halloween) _____)
_____(party-theme Halloween) _____
_____(+ 1 (party-guests Halloween))) _____)

(EXAMPLE (RSVP _____ Summer _____))
_____(make-party (party-location Summer) _____)
_____(party-theme Summer) _____
_____(+ 1 (party-guests Summer))) _____)

Function

(define (RSVP _____ party _____))
_____(make-party (party-location party) _____)
_____(party-theme party) _____
_____(+ 1 (party-guests party))) _____)

DESIGN RECIPE: RELOCATE

State the problem:

Write a function called `relocate` that takes in a location and moves the party there

Contract+Purpose Statement

```
; _relocate_____ : _____ party string_____ -> _____ party_____
      name                Domain                Range
; _____ Moves a party to a new location_____
      What does the function do?
```

Give Examples

On the computer, write an example of your function in action, using EXAMPLE.

```
(EXAMPLE ( _relocate_____ "Halloween" "home"_____ )
  _____ (make-party "home"_____ )
  _____ (party-theme Halloween)_____ )
  _____ (party-guests Halloween)_____ )
```

```
(EXAMPLE ( _relocate_____ "Summer" "backyard"_____ )
  _____ (make-party "backyard"_____ )
  _____ (party-theme Summer)_____ )
  _____ (party-guests Summer)_____ )
```

Function

```
(define ( _relocate_____ _ party place_____ )
  _____ (make-party place_____ )
  _____ (party-theme party)_____ )
  _____ (party-guests party)_____ )
```

Dissecting a Demo: Ninja World

What changes?

_____the x coordinate of the dog _____

Ninja World:

; a world is a __number_____

(define-struct world (_dogX_____))

My constructor function is:

1) (How do you make a world?)__make-world_____

What is its contract? ;make-world : number → world_____

My accessor function is:

2) (How do you get the dogX out of the world?)

_____world-dogX_____

What is its contract? __ ; world-dogX : world → number

DESIGN RECIPE: UPDATE-WORLD (NINJA WORLD)

State the problem:

Write a function called `update-world`, which adds 10 to the `dogX`

Contract+Purpose Statement

```
; update-world : world _____ -> world _____  
   name           Domain           Range  
; adds 10 to a world _____  
   What does the function do?
```

Give Examples

On the computer, write an example of your function in action, using EXAMPLE.

```
(EXAMPLE (update-world START _____)  
         (make-world (+ 10 (world-dogX START))) _____)
```

```
(EXAMPLE (update-world NEXT _____)  
         (make-world (+ 10 (world-dogX NEXT))) _____)
```

Function

```
(define (update-world w _____)  
       (make-world (+ 10 (world-dogX w))) _____)
```

Lesson 4

Review: define-struct

Last week we talked about a function that created new structs. For the structs below, what function would you use for each of the following?

```
; an auto is a String _Number_ Number _String_ Number  
(define-struct auto (model hp rims color value))
```

Make an auto? `__make-auto`_____

Get the model out of an auto? `__auto-model`_____

Get the hp out of an auto? `__auto-hp`_____

```
; a party is a _String_ String Number  
(define-struct team (location theme guests))
```

Make a party? `__make-party`_____

Get the location out of the party? `__party-location`_____

Get the theme out of the party? `__party-theme`_____

Get the guests out of the party? `__party-guests`_____

```
; a world is a Number  
(define-struct world (dogX))
```

What function would you use to:

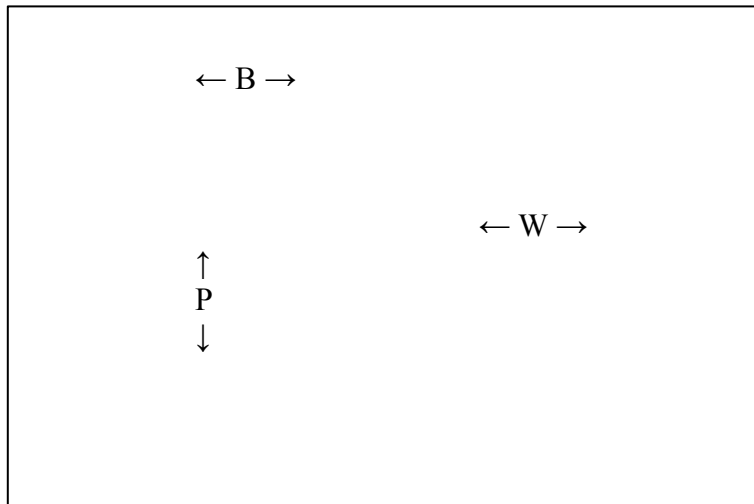
Make a world? `__make-world`_____

Get the dogX out of the world? `__world-dogX`_____

GAME DESIGN

“Start Simple, Get Complex”

Draw a rough sketch of your game in action



What images will you need for your game?

Background	cave
player	archer
danger	bats
projectile	arrow
danger	wumpus

List everything that has changed, and the datatype you will use to represent it

Changed (position? score? color? costume?)	Datatype (number? string? image? boolean?)
Wumpus status	String
archer y	Number
Wumpus x	Number
Bat x	Number
Arrow x	Number

; a world is a _string number number number number

```
(define-struct world ( __wstatus  
 __archerY  
 __wumpusX  
 __batX  
 __arrowX))
```

My constructor function is...

; make-world : _string number number number number → World

My accessor functions are...

; world-wstatus : world → string

; world-archerY : world → number

; world-wumpusX : world → number

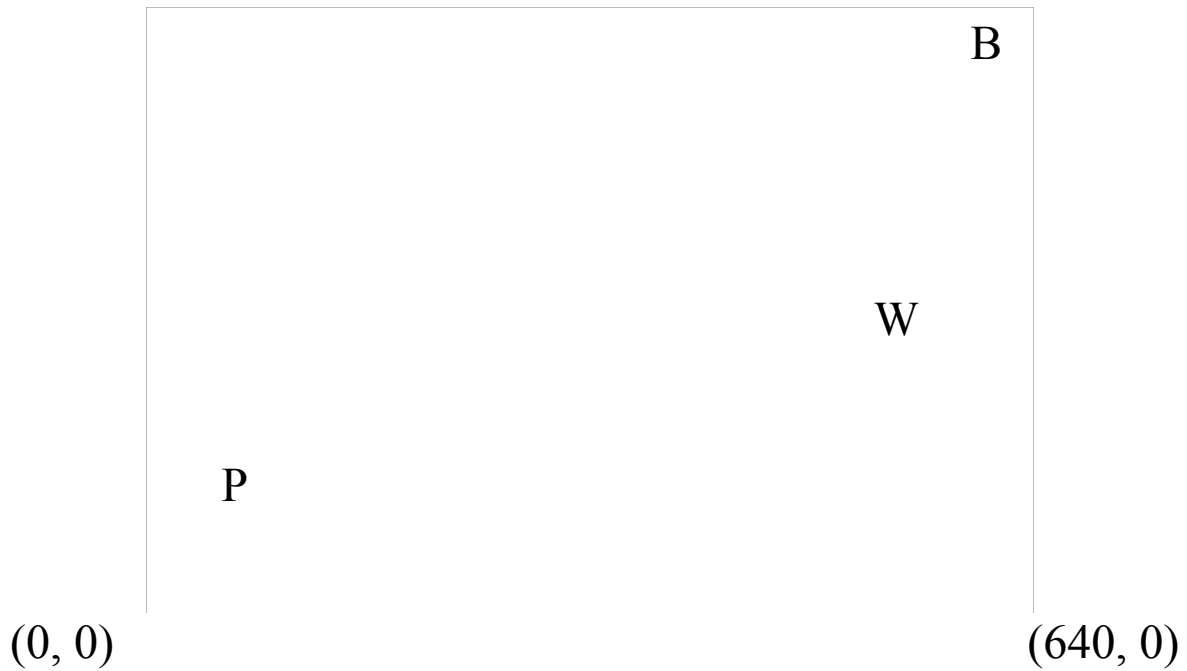
; world-batX : world → number

; world-arrowX : world → number

(0, 480)

START

(640,480)



At the start of my game, this is where everything is:

Object (top to bottom of stack)	Position (x, y)
Bats	(620, 440)
archer	(50, 40)
arrow	(800, 25)
wumpus	(600, 210)
Background	

(define START(make-world _ "asleep" _____

___ 40 _____

___ 210 _____

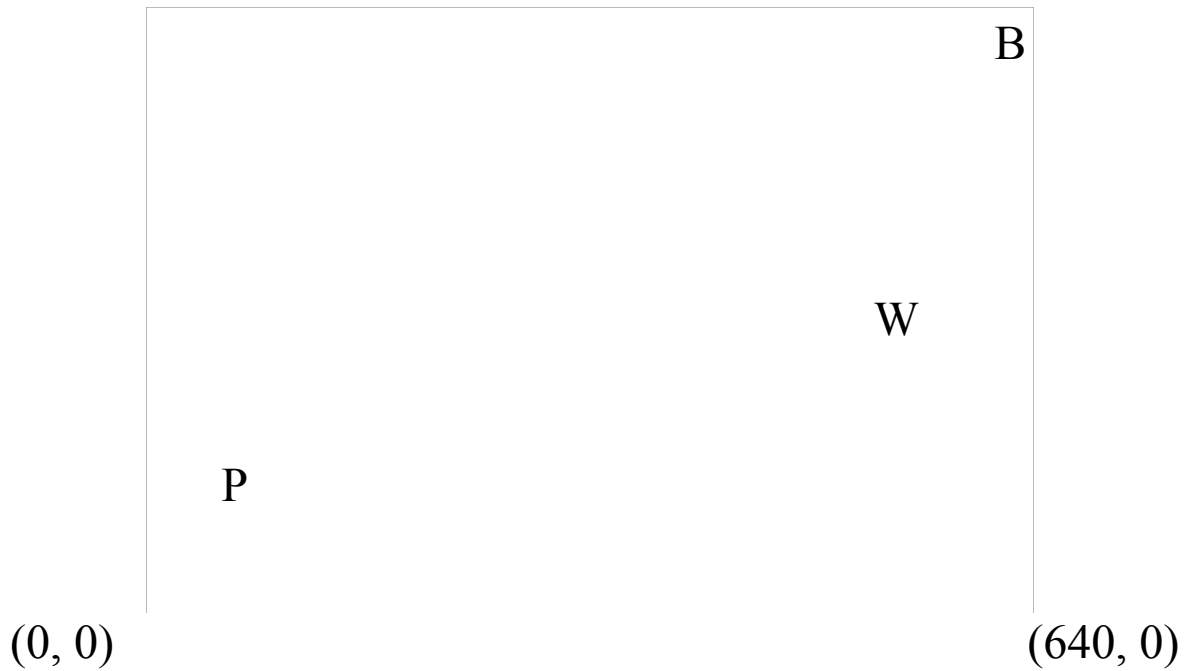
___ 620 _____

___ 800 _____))

(0, 480)

NEXT

(640,480)



A split second later, this is where everything is:

Object (top to bottom of stack)	Position (x, y)
Bats	(580, 440)
archer	(50, 40)
arrow	(820, 25)
wumpus	(600, 210)
Background	

(define NEXT (make-world _ "asleep" _____

___ 40 _____

___ 210 _____

___ 580 _____

___ 820 _____))

DRAW-WORLD

Contract

```
; _draw-world____ : __world_____ -> __image_____
```

Using put-image

```
(define (_draw-world____ _w_____)
```

```
(put-image _FLOCK_____
```

```
  _(world-batX w) 440_____
```

```
(put-image __PLAYER_____
```

```
  __50 (world-archerY w)_____
```

```
(put-image __ARROW_____
```

```
  (world-arrowX w) (+ 5 (world-archerY w))_____
```

```
(put-image _DANGER_____
```

```
  _(world-wumpusX w) 210_____
```

```
BACKGROUND _)))))_____
```

DESIGN RECIPE: UPDATE-WORLD

State the problem (What changes?):

Add 20 to arrowX, subtract 40 from batX

Contract+Purpose Statement

; update-world : world -> world
name Domain Range

; Add 20 to arrowX, subtract 40 from batX

Give Examples

(EXAMPLE (update-world START))

(make-world (world-wstatus START))

(world-archerY START)

(world-wumpusX START)

(- (world-batX START) 40)

(+ 20 (world-arrowX START)))

(EXAMPLE (update-world NEXT))

(make-world (world-wstatus NEXT))

(world-archerY NEXT)

(world-wumpusX NEXT)

(- (world-batX NEXT) 20)

(+ 40 (world-arrowX NEXT)))

Function

(define (update-world w))

(make-world (world-wstatus w))

(world-archerY w)

(world-wumpusX w)

(- (world-batX w) 40)

(+ 40 (world-arrowX w)))

DESIGN RECIPE

State the Problem

For each keypress in the Ninja World game, show how (keypress START <key>) should change your world.

Contract+Purpose Statement

; keypress _____ : world string -> world _____
name Domain Ranges

Give Examples

(EXAMPLE (keypress START "up" _____))

(make-world _(world-dogX w) _____
_(world-rubyX w) _____
_(+ 10 (world-catY w)) _____
_____))

(EXAMPLE (keypress START "down" _____))

(make-world _(world-dogX w) _____
_(world-rubyX w) _____
_(- (world-catY w) 10) _____
_____))

```

(define (_keypress _____ _w key _____)
  (cond
    [(_string=? key "up" _____)
     ____
     (make-world (world-dogX w) _____
                 ____
                 (world-rubyX w) _____
                 ____
                 (+ (world-catY w) 10))) _____]
    [(_string=? key "down" _____)
     ____
     (make-world (world-dogX w) _____
                 ____
                 (world-rubyX w) _____
                 ____
                 (- (world-catY w) 10))) _____]
    [_____]
  ))

```

DESIGN RECIPE

State the Problem

For each keypress in your game, show how (keypress START <key>) should change your world.

Contract+Purpose Statement

;keypress _____ : world string _____ -> world
name Domain Range

Give Examples

(EXAMPLE (keypress START _"up"_____))

(make-world ___(world-wstatus w)_____
_(+ (world-archery w) 10)_____
_(world-wumpusX w)_____
_(world-batX w)_____
_(world-arrowX w)_____))

(EXAMPLE (keypress START _"down"_____))

(make-world ___(world-wstatus w)_____
_(- (world-archery w) 10)_____
_(world-wumpusX w)_____
_(world-batX w)_____
_(world-arrowX w)_____))

```
(EXAMPLE (keypress START _____)
  (make-world _____
    _____
    _____
    _____
    _____))
```

```
(define (_keypress _____ w key _____)
  (cond
    [(_____ string=? key "up" _____)
     (make-world _____(world-wstatus w)_____
       _____(+ (world-archerY w) 10)_____
       _____(world-wumpusX w)_____
       _____(world-batX w)_____
       _____(world-arrowX w)_____)]
    [(_____ string=? key "down" _____)
     (make-world _____(world-wstatus w)_____
       _____(- (world-archerY w) 10)_____
       _____(world-wumpusX w)_____
       _____(world-batX w)_____
       _____(world-arrowX w)_____)]
    [(_____ string=? key " " _____)
     (make-world _____"awake"_____
       _____(world-archerY w)_____
       _____(world-wumpusX w)_____
       _____(world-batX w)_____
       _____50_____)]
```


Extended update-world:

```
; off-right?: number -> boolean  
name domain range
```

```
(EXAMPLE (off-right? 800) (> 800 640))
```

```
(EXAMPLE (off-right? 150) (> 150 640))
```

```
(define (off-right? x) (> x 640))
```

```
; off-left?: number -> boolean
```

```
(EXAMPLE (off-left? -100) (< -100 0))
```

```
(EXAMPLE (off-left? 500) (< 500 0))
```

```
(define (off-left? x) (< x 0))
```

```
; _____: _____ -> _____
```

```
(EXAMPLE (_____ _____) _____)
```

```
(EXAMPLE (_____ _____) _____)
```

```
(define (_____ _____) _____)
```

```
; _____: _____ -> _____
```

```
(EXAMPLE (_____ _____) _____)
```

```
(EXAMPLE (_____ _____) _____)
```

```
(define (_____ _____) _____)
```

TEST	RESULT
(off-left? (world-batX w))	<pre>(make-world _(world-wstatus w)_____ _(world-archerY w)_____ _(world-wumpusX w)_____ _800_____ __ (world-arrowX w)_____)</pre>
(off-left? (world-wumpusX w))	<pre>(make-world _(world-wstatus w)_____ _(world-archerY w)_____ _800_____ _(world-batX w)_____ __ (world-arrowX w)_____)</pre>
	<pre>(make-world _____ _____ _____ _____ _____)</pre>
	<pre>(make-world _____ _____ _____ _____ _____)</pre>

Lesson 8

Design Recipe: line-length

Write a function called line-length, which takes in two numbers and returns the difference between them. It should always subtract the smaller number from the bigger one.

Contract+Purpose Statement

Every contract has three parts:

; line-length : number number -> number
name Domain Range

Give Examples

(EXAMPLE (line-length 2 7) (- 7 2))

(EXAMPLE (line-length 7 2) (- 7 2))

Function Header

Write the Function Header, giving variable names to all your input values that change.

(define (line-length a b)
function name variable names

(cond

[(> a b)

(- a b)]

[else

(- b a)]

))

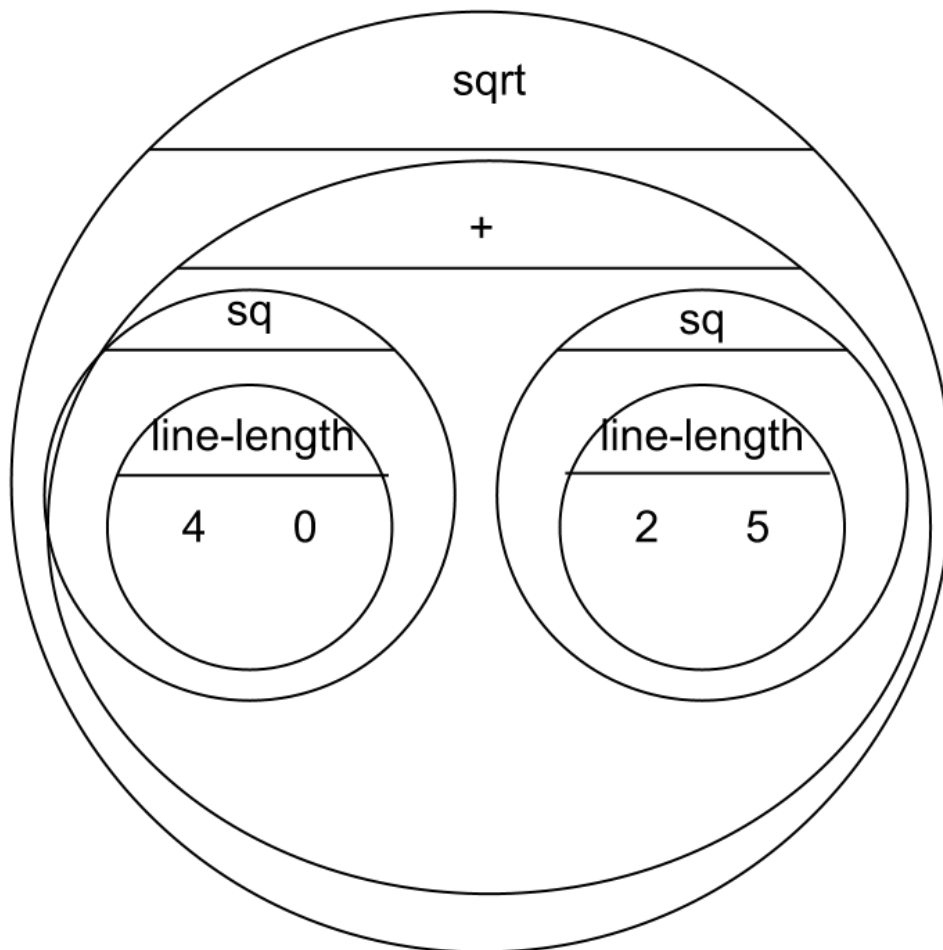
Distance:

The Player is at (4, 2) and the Target is at (0, 5).

Distance takes in the player-x, player-y, character-x and character-y.

Use the formula below to fill in the EXAMPLE:

$$\sqrt{(\text{line-length } 4 \ 0)^2 + (\text{line-length } 2 \ 5)^2}$$



Convert it into Racket code:

(EXAMPLE (`_distance` `_` `_4 2 0 5` `_`))

(`sqrt (+ (sq (line-length 4 0)) (sq (line-length 2 5)))`) `_`)

Design Recipe: distance

Write a function `distance`, which takes FOUR inputs:

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

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

$$\text{Distance} = ((\text{line-length } px \text{ } cx)^2 + (\text{line-length } py \text{ } cy)^2)$$

Contract+Purpose Statement

`distance` : number number number number -> number

name Domain Range

What does the function do?

Give Examples

(EXAMPLE (`distance 4 2 0 5`) _____)

(`sqrt (+ (sq (line-length 4 0)) (sq (line-length 2 5)))`) _____)

(EXAMPLE (`distance 7 8 9 1`) _____)

(`sqrt (+ (sq (line-length 7 9)) (sq (line-length 8 1)))`) _____)

Function Header

(define (`distance` `px py cx cy`) _____)

function name variable names

____(`sqrt (+ (sq (line-length px cx)) (sq (line-length py cy)))`) _____)

DESIGN RECIPE: COLLIDE?

Write a function `collide?`, 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 **75 pixels** of the coordinates of the other character. Otherwise, `false`.

Contract+Purpose Statement

```
; __collide?__ : __number number number number__ -> __boolean__
   name                Domain                Range
;
   What does the function do?
```

Give Examples

(EXAMPLE (`__collide?__ 1 3 234 91`_____))

____(< (distance 1 3 234 91) 75)_____)

(EXAMPLE (`__collide?__ 543 25 24 431`_____))

____(< (distance 543 25 24 431) 75)_____)

Function Header

(define (`__collide?__` _____ `__px py cx cy`_____))

____(< (distance `px py cx cy`) 75)_____)

TEST	RESULT
(collide? (world-wumpusX w) 210 50 (world-archerY w))	(make-world (world-wstatus w) _____ 0 _____ 1000 _____ (world-batX w) _____ (world-arrowX w) _____)
(collide? (world-batX w) 440 50 (world-archerY w))	(make-world (world-wstatus w) _____ 0 _____ (world-wumpusX w) _____ 1500 _____ (world-arrowX w) _____)
(collide? (world-wumpusX w) 210 (world-arrowX w) (+ 25 (world-archerY w)))	(make-world "asleep" _____ (world-archerY w) _____ 800 _____ (world-batX w) _____ 900) _____)
	(make-world _____ _____ _____ _____ _____)

Supplemental

***The Teacher Game* file contains a fully working game up to this point. As you can see, while the game works it is not fully fleshed out. Additional code, which could be completed in a couple more hours of small group work, is implemented in *Teacher Game 2*.**

Changes include the following:

Adding a “Game Over” screen

Implementing the wumpus status: awake/asleep/dead

Writing a second collide function, with decreased range

Providing fine resolution player mvmt