·1. Learning materials in AquaLang

1.1 The first item

def keyby[K,T](Stream[T], fun(T):K): KeyedStream[K,T];

def unkey[K,T](KeyedStream[K,T]): Stream[T];

def keyby[K,T](Stream[T], fun(T):K): KeyedStream[K,T];

val s0: Stream[Item] = source(stdio(), csv());

val s1: KeyedStream[String, Item] = keyby(s0, \_.name);

val s2: KeyedStream[String, Item] = scan(s1, argmin(\_.price));

val s3: Stream[Item] = unkey(s2);

sink(s3, stdio(), csv());

1.2 The second item

enum Shape[T] { Rectangle(f32, f32), Circle(f32) }

val \_: Shape[f32] = Rectangle(1.0, 2.0);

val \_: Shape[f32] = Circle(3.0);

def area(s) = match s {

Rectangle(w, h) => w \* h,

Circle(r) => r \* 3.14,

}

1.3 The third item

from item:Item in source(stdio(), json(), ingestion())

over tumbling(length:5s) {

avg of item.price as average,

min of item.price as cheapest,

max of item.price as priciest

}

where average > 10.0 and cheapest > 1.0

select {average, cheapest, priciest}

into sink(stdio(), json());

1.4 The fourth item

from id:u32 in source(stdio(), csv())

right join item in items

on id == item.id

into sink(stdio(), csv());

1.5 The fifth item

type User = {id: U64, name: Text, requests: Vec[Request]};

type Request = {id: U64, items: Vec[Item]};

val stream: Stream[{user:User, request:Request, item:Item}] =

from user: User in source(stdio(), csv())

from request in user.requests

from item in request.items;

val stream: Stream[{user:User, request:Request, item:Item}] =

source(stdio(), csv())

.map(fun(user: User) = {user})

.flatmap(fun(user) = user.requests.map(fun(request) = {user, request})

.flatmap(fun({user, request} = request.items.map(fun(item) = {user, request, item})))

1.6 The sixth item

def gcd(a, b) {

var x = a;

var y = b;

while y != 6 {

val temp = y;

y = x % y;

x = temp;

}

abs(x)

}

·2. Learning materials in Aniframe

2.1 The first item

shape1 = quad(58, 73, 29.000, 28.61, -3, 94.0076, 28.16, 77.19458)

shape1.fill("rgb(100, 200, 55)")

shape1.stroke("#000000")

colors = ["#89AC76", "rgb(041, 49, +51)", "#1E2460", "#2271B3"]

colors.add("#DEDEDE")

colors.remove(3)

info(length(colors))

start\_fr = 1

end\_fr = 100

num: Number = 401

ctr: Number = 25

if num < 401:

info("Hey")

else if num < 500:

info("Printed")

else if num < 600:

info("Not me")

else:

info("Hmmm")

if num > 0 && num < 400:

num = 5

repeat 10:

if num < 100 || num > 300:

num = 5

else:

break

new = colors[rand\_num(0, length(colors)-1)]

if start\_fr < ctr && start\_fr:

shape1.moveY(num-0.000, start\_fr, ctr)

shape1.fill(new)

shape1.stroke(new)

start\_fr += ctr

if !(start\_fr < 75):

shape1.moveY(num+0, start\_fr, ctr\*3)

start\_fr += ctr

if start\_fr >= end\_fr:

shape1.moveY(num-num, start\_fr, start\_fr)

else:

shape1.moveY(num-2\*num, start\_fr, end\_fr)

shape1.draw(1, 10000)

2.2 The second item

start\_fr = 1

type(start\_fr)

kabi\_color: Text = "#ffb6c1"

kabi\_body = circle(250, 250, 270)

kabi\_body.fill(kabi\_color)

kabi\_body.stroke(kabi\_color)

func make\_part(x: Number, y: Number, width: Number, height: Number, colr: Color) returns Object:

part: Object = ellipse(x, y, width, height)

part.fill(colr)

part.stroke(colr)

return part

l\_eye = make\_part(220, 210, 38, 90, "#000000")

smile: Object = curve(230, 150, 230, 272, 270, 272, 270, 150)

smile.stroke("#000000")

smile.fill(kabi\_color)

l\_hand = ellipse(130, 280, 100, 110)

kabi\_eyes: Object = l\_eye + r\_eye

func wave\_hand(frame: Number, offset: Number):

repeat(sqrt(10/10) \* 3.5):

r\_hand.move(1, 2, frame, frame + offset)

l\_hand.draw(1, 1000)

r\_hand.draw(1, 1000)

l\_eye.draw(1, 1000)

r\_eye.draw(1, 1000)

smile.draw(1, 1000)

2.3 The third item

p\_color: Text = "#C2B280"

txt: Text = "Lying"

txt\_box: Object

pinocchio: Object = circle(0, 250, 200)

nose = rectangle(-650, 250, 800, 15)

pinocchio.fill(p\_color)

nose.moveX(20, 1, 500)

colors: List = ["#ff0000", "#00ff00", "#0000ff"]

for color in colors:

pinocchio.stroke(color)

pinocchio.draw(1, 500)

txt\_box.write(txt, 120, 120)

nose.draw(1, 500)

pinocchio.draw(1, 500)

txt\_box.draw(1, 500)

2.4 The fourth item

marker = ellipse(250, 250, 300, 300)

i = 0

repeat(6):

new\_marker: Object = ellipse(250, 250, 300-i\*10, 300-i\*10)

marker += new\_marker

i += 1

if i > 30:

break

repeat(6):

new\_marker: Object = ellipse(250, 250, 300-i\*10, 300-i\*10)

marker += new\_marker

i += 1

if i > 30:

break

marker.resize(3, 500, 1500)

marker.draw(1, 1114)