UI-Skribble Backflip

Flip Center

"Joints" to manipulate
rotation angles

End

Start

ClickyBeach: Class Diagram - Immoveable

Immoveable

position: Vector
skinColor. RGBColor
action: ACTION

+ constructor(_position: Vector
+ clicked(_hotspot: Vector): bo
+ update(): void
 # draw(): void

Tree

+ update(): void
# draw(): void

Human

headAngle: number
shoulder: number
elbow: number
pelvis: number
knee: number
leftHand: number
rightHand: number
leftFoot: number
rightFoot: number
flipAngle: number
flipCenter: Vector
surfPos: Vector
currentPos: Vector
walking: boolean

+ constructor(_position)
+ update(): void
# draw(): void
- backflip(): void
- surf(): void
- walk(_target: Vector): void

Ship

- chimneyOne: Vector
- chimneyTwo: Vector
- chimneyThree: Vector
- chimneys: Vector[]

+ constructor(_position)
+ update(): void
# draw(): void
- blowSteam(): void
- reset(): void

ClickyBeach: Activity Diagram - Human

constructor

_position: Vector

super(_position)

↓

generate a skinColor

↓

set joint values to default

↓

this.draw

◇

●

walk

_target: Vector

let head: Path2D = new Path2D
let skinColor: RGBColor = a random color
with r, b, g each between 0 and 255

↓

crc2.save()

↓

crc2.translate(this.currentPos.x,
this.currentPos.y)

↓

walk one frame at a
time until reaching
_target

↓

crc2.restore

↓

●

surf

●

let head: Path2D = new Path2D
let surfBoard: Path2D = new Path2D
let direction: number = either 1 or -1

↓

crc2.save()

↓

crc2.translate(this.surfPos.x,
this.surfPos.y)

↓

draw surfer

↓

change fill- and strokeStyle
to HSL(0, 0%, 74%)

↓

crc2.translate(40 * _direction, 0)

↓

draw surfboard

↓

crc2.restore

↓

●

draw

●

let head: Path2D = new Path2D
let skinColor: RGBColor = a random color
with r, b, g each between 0 and 255

↓

crc2.save()

↓

crc2.translate(_position.x, _position.y)

↓

draw human

↓

crc2.restore

↓

●

backflip

●

let head: Path2D = new Path2D
let skinColor: RGBColor = a random color
with r, b, g each between 0 and 255

↓

crc2.save()

↓

crc2.translate(_position.x, _position.y)

↓

do backflip motion
one frame at the time

↓

crc2.restore

↓

●

update

● → ◇

ACTION.REST → draw

ACTION.BACKFLIP → backflip → adjust backflip joints → this.action = ACTION.GO

ACTION.RETURN → walk(this.position) → this.action = ACTION.REST

ACTION.GO → walk(this.surfPos) → this.action = ACTION.SURF

ACTION.SURF → surf → this.action = ACTION.RETURN

◇ → ●

constructor

_position

super(_position)

this.chimneyOne = new Vector(212.5, -90);
this.chimneyTwo = new Vector(300, -90);
this.chimneyThree = new Vector(387.5, -90);
this.chimneys = [this.chimneyOne,
this.chimneyTwo, this.chimneyThree];

update

draw

blowSteam → draw → ⧖ → reset

blowSteam

let r1: number = randomInteger(5, 15);
let r2: number = randomInteger(20, 30);
let steamDensity: number = randomInteger(10, 15);
let gradient: CanvasGradient =
crc2.createRadialGradient(0, 0, r1, 0, 0, r2);
let particle: Path2D = new Path2D;

let clouds appear
from every chimney

draw

change fill and
strokeStyle to
HSL(11, 100%, 36%)

crc2.save

crc2.translate(_position.x, _position.y)

crc.beginPath()

draw vessel

crc2.closePath()

change fill and
strokeStyle to
HSL(11, 18%, 91%)

crc.beginPath()

draw cabin

crc2.closePath()

change fill and
strokeStyle to
HSL(11, 0%, 46%)

crc.beginPath()

draw chimneys

crc2.closePath()