

Homework 2

乘风破浪

加载相关的包

```
;require packages
#lang racket
(require test-engine/racket-tests)
(require 2htdp/image)
(require 2htdp/universe)
(require lang/posn)
```

宏定义船窗的颜色

```
:: Graphical Constants
(define Windowcolor (color 0 0 255))
```

导入相关的素材图片

```
;=====ORNAMENT=====
=====
;=====
=====
;ORNAMENT consist (Cloud Background Waves Flag)
;;=====Cloud=====
;Generate Cloud
(define (cloud n) (scale (* 0.05 n) (bitmap "Cloud.png")))
;;=====Background=====
;Generate Background
(define (bg.v1 n) (scale (* 0.05 n) (bitmap "sky.png")))
;;=====Wave=====
(define (Waves n) (scale (* 0.1 n) (bitmap "wave.png")))
;;=====Wave=====
(define (flag n) (flip-horizontal (scale (* 0.005 n) (bitmap "Flag.png"))))
;=====
=====
;=====ORNAMENT=====
=====
```

导入上节课绘制好的船只

```
;=====
=====
;=====SHIP=====
=====
;Number Color->Image
;Define parallelogram to use below
;Given:(Parallelogram 10 "red")
```

;Expected:Generate a parallelogram with side length 40,50(which $50^2=40^2+30^2$)

```
(define (Parallelogram n c)
  (overlay/xy (flip-vertical (flip-horizontal (right-triangle (* 4 n) (* 3 n) "solid" c)))
    (* 4 n) 0
    (right-triangle (* 4 n) (* 3 n) "solid" c)))
```

;Number Color->Image

;Create the Shipbody of the Ship

;Given:(Shipbody 10 "red")

;Expected:Generate a red Shipbody and 10 is to adjust the size of the picture

```
(define (shipbody n c)
  (overlay/xy
    (overlay/xy (overlay/xy
      (flip-horizontal (scale/xy 1 1.5
        (Parallelogram n c)))
      (* -7 n) (* 1.5 n)
      (rectangle (* 10 n) (* 3 n) "solid" c))
      (* -8 n) (* -1.5 n)
      (scale/xy 1.5 2
        (Parallelogram n c)))
    (* 6 n) 0
    (rotate -10 (triangle/sss (* 12 n) (* 5 n) (* 15 n) "solid" c))))
```

;Number Color->Image

;Define Right angle trapezoid to use below

;Given:(Trapezoid 10 "red" 1)

;Expected:Generate a red trapezoid,10 to adjust the size of the picture and 1 is to adjust the length of the rectangular part

```
(define (Trapezoid n c l)
  (overlay/xy
    (flip-horizontal (right-triangle (* 2 n) (* 3 n) "solid" c))
    (* 2 n) 0
    (rectangle (* (* 8 n) l) (* 3 n) "solid" c)))
```

;Number Color The length of the rectangular part->Image

;Define Cabin

;Given:(Cabin 10 "red" 1)

;Expected:Generate a red Cabin,10 to adjust the size of the picture and 1 is to adjust the length of the rectangular part

```
(define (Cabin n c l)
  (overlay/xy (Trapezoid (* n 0.3) Windowcolor 1)
    (* -1.1 n) (* -0.5 n)
    (overlay/xy (rectangle (* 2 n) (* 1 n) "solid" Windowcolor)
      (* -6 n) (* -0.5 n)
      (Trapezoid n c l))))
```

;Generate Ship with these three parts

```
(define (ship.v1 n)
  (overlay/xy (shipbody n (color 84 30 36))
    (* 9 n) (* -2.5 n)
    (overlay/xy
      (Cabin n (color 238 232 205) 1)
      (* 3 n) (* -2 n)
      (Cabin (* 0.8 n) (color 205 201 165) 0.8))))
```

;Generate Ship with flag

```
(define (ship.v2 n)
  (flip-horizontal(overlay/xy
    (flip-horizontal (flag n))
```

```

(* -16 n) (* 2 n)
(ship.v1 n))))
;Add Text
(define (ship.v3 n)
  (overlay/xy
    (text/font "Noah's Ark" (* 1.2 n) (color 33 119 184) "Gill Sans" 'swiss 'normal 'bold #f)
    (* -13 n) (* -5.5 n)
    (ship.v2 n)))
;=====SHIP=====
=====
;=====
=====

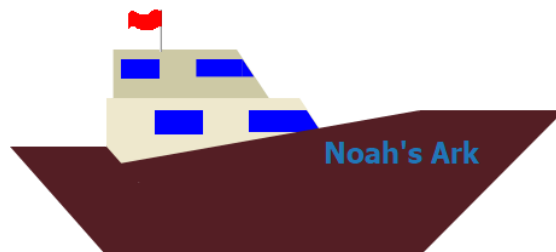
```

执行

```

(define SHIP (ship.v3 20))
SHIP
;Function flip-horizontal

```



在main函数之前的一些宏定义:

```

(define SHIP (ship.v3 20))
(define SHIP-WIDTH (image-width SHIP))
(define SHIP-HEIGHT (image-height SHIP))
(define WIDTH-OF-WORLD (image-width (bg.v1 8)))
(define HEIGHT-OF-WORLD (image-height (bg.v1 8)))
(define SHIP-CENTER (/ (image-width SHIP) 2))
(define Y-SHIP (- (- HEIGHT-OF-WORLD (/ (image-height SHIP) 2)) (/ (image-height SHIP) 4)))
(define bg (empty-scene WIDTH-OF-WORLD HEIGHT-OF-WORLD))
(define Cloud-WIDTH (image-width (cloud 5)))
(define Wave-WIDTH (image-width (Waves 3)))
(define Wave-HEIGHT (image-height (Waves 3)))

```

利用图片绘制一些组合体:

```

(define wave (beside (Waves 3) (Waves 3) (Waves 3) (Waves 3) (Waves 3)))

```

wave



导入背景(其中云朵的位置可以随意定义)

```

(define bg.v2
  (place-images

```

```
(list (text/font "Description:\n Key-D can accelerate the movement of the ship \n Mounse Click can
locate the ship" 24 "black" #f "decorative" "normal" "light" #f)
      (cloud 5)
      (cloud 5)
      (cloud 5)
      (cloud 5)
      wave)
(list (make-posn 300 50)
      (make-posn 150 (* 2 Cloud-WIDTH))
      (make-posn 650 (* 1.5 Cloud-WIDTH))
      (make-posn 1150 (* 2.5 Cloud-WIDTH))
      (make-posn 1650 (* 3.5 Cloud-WIDTH))
      (make-posn (/ WIDTH-OF-WORLD 2) (- HEIGHT-OF-WORLD (/ Wave-HEIGHT 2))))
(bg.v1 8)))
;may use loop to define more effectively
```



WorldState/Big-bang

Part 1[运行函数]

```
;; Data definitions

; WorldState is a Number
; interpretation the number of pixels between the left border and the SHIP

;; Functions

;; clock ticks
;; WorldState -> WorldState
;; the clock ticked; move the SHIP by 3 pixels

(check-expect (tock 20) 26)
(check-expect (tock (+ WIDTH-OF-WORLD (/ SHIP-WIDTH 2))) (- 0 (/ SHIP-WIDTH 2)))

(define (tock ws)
  (if (<= ws (+ WIDTH-OF-WORLD (/ SHIP-WIDTH 2))) (+ ws 6) (- 0 (/ SHIP-WIDTH 2))))
```

当ws大于画布边界+船长的一半时将ws初始化为-船长的一半(实现船循环)

其余时候船以6像素/次的速度向右移动

Part 2[键盘输入]

```
;; WorldState -> WorldState
;; Move with "w" "s"
;; according to the given world state
(define (move W a-key)
  (cond
    [(key=? a-key "d")
     (+ W 12)]
    [(key=? a-key "a")
     (- W 12)])))
```

当一直按下d键时速度增加为12像素/次的速度向右移动

当一直按下a键时速度增加为12像素/次的速度向左移动

Part 3[船的渲染]

```
;; WorldState -> Image
;; place the SHIP into the BACKGROUND scene,
;; according to the given world state
(define (render ws)
  (place-images
   (list wave
          (rotate (random 5 8) SHIP))
   (list (make-posn (- (/ WIDTH-OF-WORLD 2) 20) (- HEIGHT-OF-WORLD 50))
         (make-posn ws (+ Y-SHIP (random 20 30)))))
  bg.v2))
```

最重要的是体现出水中摇曳的方式:

- 旋转 5-8°之间的任意一角度
- 船位置放置在中间位置上随机+20-+30

Part4 [鼠标位置]

```
;; WorldState Number Number String -> WorldState
;; places the SHIP at the x-coordinate
;; if the given me is "button-down"

(check-expect (hyper 21 10 20 "enter") 21)
(check-expect (hyper 42 10 20 "button-down") 10)
(check-expect (hyper 42 10 20 "move") 42)

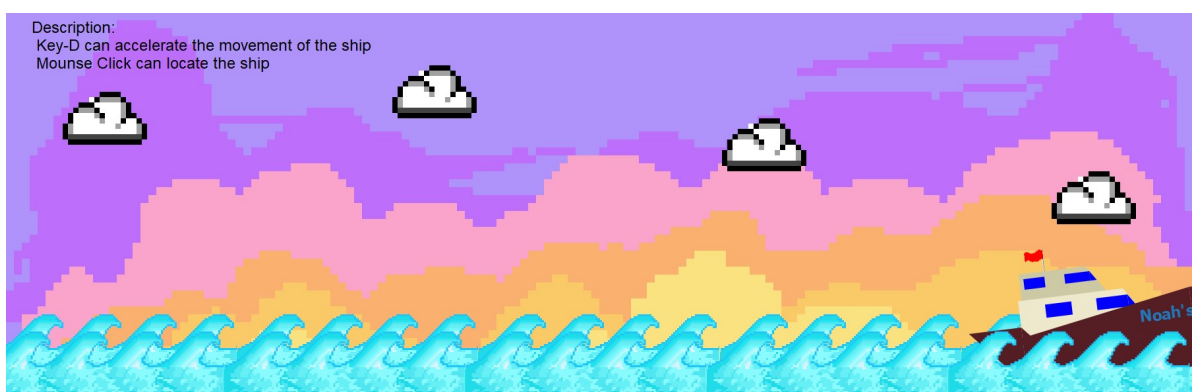
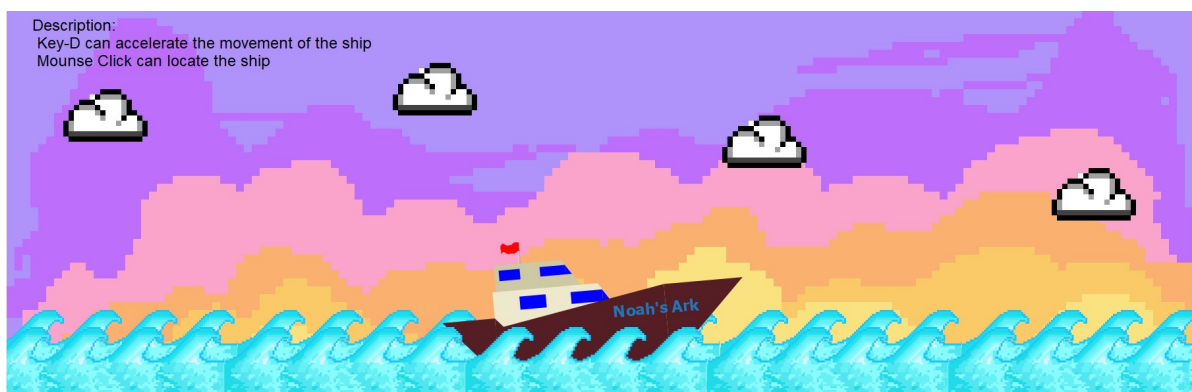
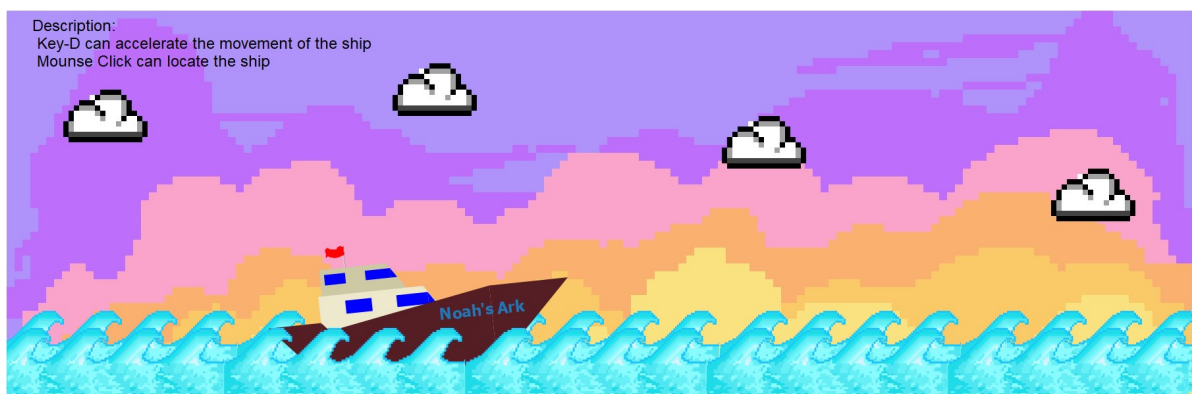
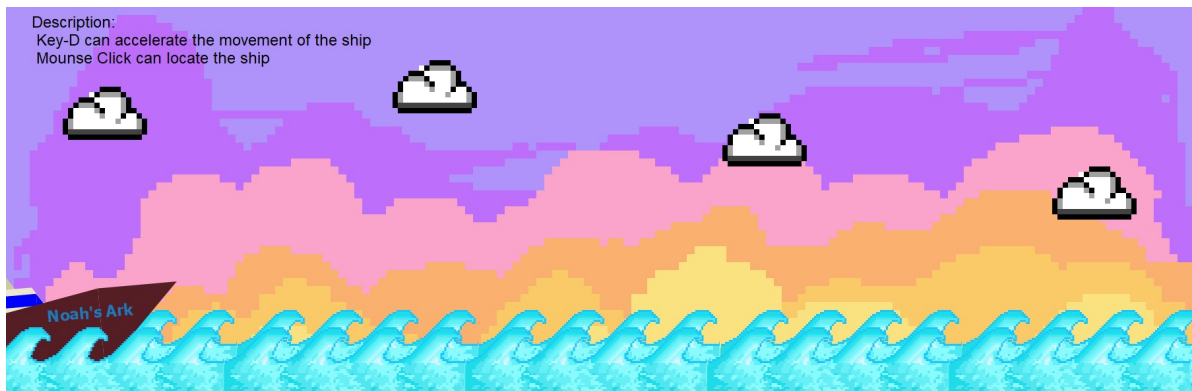
(define (hyper x-coordinate x-mouse y-mouse me)
  (cond
    [(string=? "button-down" me) x-mouse]
    [else x-coordinate]))
```

获取鼠标点击的x位置

main 函数

```
;; main : WorldState -> WorldState
;; launch the program from some initial state
;; run: (main 0)
(define (main ws)
  (big-bang ws
    [on-tick tock]
    [on-mouse hyper]
    [to-draw render]
    [on-key move]))
```

完成效果



Appendix

Code

```
;=====
;September 22
;Dormancy
;=====

;require packages
#lang racket
(require test-engine/racket-tests)
(require 2htdp/image)
(require 2htdp/universe)
(require lang/posn)
;; Graphical Constants
(define Windowcolor (color 0 0 255))

;=====ORNAMENT=====
=====
;=====
=====
;ORNAMENT consist (Cloud Bird Sun Waves)
;;=====Cloud=====
;Generate Cloud
(define (cloud n) (scale (* 0.05 n) (bitmap "Cloud.png")))
;;=====Background=====
;Generate Background
(define (bg.v1 n) (scale (* 0.05 n) (bitmap "sky.png")))
;;=====Wave=====
(define (Waves n) (scale (* 0.1 n) (bitmap "wave.png")))
;;=====Wave=====
(define (flag n) (flip-horizontal (scale (* 0.005 n) (bitmap "Flag.png"))))
;=====
=====
;=====ORNAMENT=====
=====

;=====
=====
;=====SHIP=====
=====

;Number Color->Image
;Define parallelogram to use below
;Given:(Parallelogram 10 "red")
;Expected:Generate a parallelogram with side length 40,50(which  $50^2=40^2+30^2$ )
(define (Parallelogram n c)
  (overlay/xy (flip-vertical (flip-horizontal (right-triangle (* 4 n) (* 3 n) "solid" c)))
    (* 4 n) 0
    (right-triangle (* 4 n) (* 3 n) "solid" c)))

;Number Color->Image
;Create the Shipbody of the Ship
;Given:(Shipbody 10 "red")
;Expected:Generate a red Shipbody and 10 is to adjust the size of the picture
(define (shipbody n c)
```

```

(overlay/xy
  (overlay/xy (overlay/xy
    (flip-horizontal (scale/xy 1 1.5
      (Parallelogram n c)))
    (* -7 n) (* 1.5 n)
    (rectangle (* 10 n) (* 3 n) "solid" c))
    (* -8 n) (* -1.5 n)
    (scale/xy 1.5 2
      (Parallelogram n c)))
  (* 6 n) 0
  (rotate -10 (triangle/sss (* 12 n) (* 5 n) (* 15 n) "solid" c))))

```

;Number Color->Image

;Define Right angle trapezoid to use below

;Given:(Trapezoid 10 "red" 1)

;Expected:Generate a red trapezoid,10 to adjust the size of the picture and 1 is to adjust the length of the rectangular part

```

(define (Trapezoid n c l)
  (overlay/xy
    (flip-horizontal (right-triangle (* 2 n) (* 3 n) "solid" c))
    (* 2 n) 0
    (rectangle (* (* 8 n) l) (* 3 n) "solid" c)))

```

;Number Color The length of the rectangular part->Image

;Define Cabin

;Given:(Cabin 10 "red" 1)

;Expected:Generate a red Cabin,10 to adjust the size of the picture and 1 is to adjust the length of the rectangular part

```

(define (Cabin n c l)
  (overlay/xy (Trapezoid (* n 0.3) Windowcolor 1)
    (* -1.1 n) (* -0.5 n)
    (overlay/xy (rectangle (* 2 n) (* 1 n) "solid" Windowcolor)
      (* -6 n) (* -0.5 n)
      (Trapezoid n c l))))

```

;Generate Ship with these three parts

```

(define (ship.v1 n)
  (overlay/xy (shipbody n (color 84 30 36))
    (* 9 n) (* -2.5 n)
    (overlay/xy
      (Cabin n (color 238 232 205) 1)
      (* 3 n) (* -2 n)
      (Cabin (* 0.8 n) (color 205 201 165) 0.8))))

```

;Generate Ship with flag

```

(define (ship.v2 n)
  (flip-horizontal(overlay/xy
    (flip-horizontal (flag n))
    (* -16 n) (* 2 n)
    (ship.v1 n))))

```

;Add Text

```

(define (ship.v3 n)
  (overlay/xy
    (text/font "Noah's Ark" (* 1.2 n) (color 33 119 184) "Gill Sans" 'swiss 'normal 'bold #f)
    (* -13 n) (* -5.5 n)
    (ship.v2 n)))

```

```

;=====SHIP=====
=====

```



```

;=====
=====

;=====
=====
;=====Main=====
=====

;; Graphical Constants
(define SHIP (ship.v3 20))
(define SHIP-WIDTH (image-width SHIP))
(define SHIP-HEIGHT (image-height SHIP))
(define WIDTH-OF-WORLD (image-width (bg.v1 8)))
(define HEIGHT-OF-WORLD (image-height (bg.v1 8)))
(define SHIP-CENTER (/ (image-width SHIP) 2))
(define Y-SHIP (- (- HEIGHT-OF-WORLD (/ (image-height SHIP) 2)) (/ (image-height SHIP) 4)))
(define bg (empty-scene WIDTH-OF-WORLD HEIGHT-OF-WORLD))
(define Cloud-WIDTH (image-width (cloud 5)))
(define Wave-WIDTH (image-width (Waves 3)))
(define Wave-HEIGHT (image-height (Waves 3)))
;;Define A Total wave to place(It's consist of five fundamental wave)
(define wave
  (beside (Waves 3)
    (Waves 3)
    (Waves 3)
    (Waves 3)))
;;Define The Background
;;Background->Background
(define bg.v2
  (place-images
    (list (text/font "Description:\n Key-D can accelerate the movement of the ship \n Mounse Click can
locate the ship" 24 "black" #f "decorative" "normal" "light" #f)
      (cloud 5)
      (cloud 5)
      (cloud 5)
      (cloud 5)
      wave)
    (list (make-posn 300 50)
      (make-posn 150 (* 2 Cloud-WIDTH))
      (make-posn 650 (* 1.5 Cloud-WIDTH))
      (make-posn 1150 (* 2.5 Cloud-WIDTH))
      (make-posn 1650 (* 3.5 Cloud-WIDTH))
      (make-posn (/ WIDTH-OF-WORLD 2) (- HEIGHT-OF-WORLD (/ Wave-HEIGHT 2))))
    (bg.v1 8)))
;; Data definitions

; WorldState is a Number
; interpretation the number of pixels between the left border and the SHIP

;; Functions

;; clock ticks
;; WorldState -> WorldState
;; the clock ticked; move the SHIP by 3 pixels

(check-expect (tock 20) 26)

```

```
(check-expect (tock (+ WIDTH-OF-WORLD (/ SHIP-WIDTH 2))) (- 0 (/ SHIP-WIDTH 2)))
```

```
(define (tock ws)
```

```
(if (<= ws (+ WIDTH-OF-WORLD (/ SHIP-WIDTH 2))) (+ ws 6) (- 0 (/ SHIP-WIDTH 2))))
```

```
;; WorldState -> WorldState
```

```
;; Move with "w" "s"
```

```
;; according to the given world state
```

```
(define (move W a-key)
```

```
(cond
```

```
[(key=? a-key "d")
```

```
(+ W 12)]
```

```
[(key=? a-key "a")
```

```
(- W 12)]))
```

```
;; WorldState -> Image
```

```
;; place the SHIP into the BACKGROUND scene,
```

```
;; according to the given world state
```

```
(define (render ws)
```

```
(place-images
```

```
(list wave
```

```
(rotate (random 5 8) SHIP))
```

```
(list (make-posn (- (/ WIDTH-OF-WORLD 2) 20) (- HEIGHT-OF-WORLD 50))
```

```
(make-posn ws (+ Y-SHIP (random 20 30))))))
```

```
bg.v2))
```

```
;; WorldState Number Number String -> WorldState
```

```
;; places the SHIP at the x-coordinate
```

```
;; if the given me is "button-down"
```

```
(check-expect (hyper 21 10 20 "enter") 21)
```

```
(check-expect (hyper 42 10 20 "button-down") 10)
```

```
(check-expect (hyper 42 10 20 "move") 42)
```

```
(define (hyper x-coordinate x-mouse y-mouse me)
```

```
(cond
```

```
[(string=? "button-down" me) x-mouse]
```

```
[else x-coordinate]))
```

```
;; main : WorldState -> WorldState
```

```
;; launch the program from some initial state
```

```
;; run: (main 0)
```

```
(define (main ws)
```

```
(big-bang ws
```

```
[on-tick tock]
```

```
[on-mouse hyper]
```

```
[to-draw render]
```

```
[on-key move]))
```

```
(main 0)
```