

<u>mproving a World Program - A</u>	<u>,dd</u>	key
-------------------------------------	------------	-----

Course > 3a: How to Design Worlds > handler

> Questions 3-5

Questions 3-5

Question 3

1 point possible (graded)

Having finished implementing the key-handler, we decide we decide we want to add another feature to our program. Suppose we want the countdown to go immediately to zero if we click the mouse. We start by looking up on-mouse in the help desk, and then write a wish-list entry for handle-mouse.

In order, what are the four arguments to handle-mouse in our countdown program?

- The current Countdown, the x-position of the Countdown, the y-position of the Countdown and a MouseEvent
- The x-position of the mouse, the y-position of the mouse, a MouseEvent and the current Countdown
- Any WorldState, the width of MTS, the height of MTS and a MouseEvent
- The current Countdown, the x-position of the mouse, the y-position of the mouse and a MouseEvent

Explanation

Look in the help desk or the HtDW Recipe page for more information on on-mouse

Submit

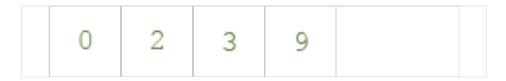
• Answers are displayed within the problem

Question

1 point possible (graded)

Drag the Countdown data to complete each check-expect for our handle-mouse function. Note, each value can be used more than once, and you don't need to use every one.

```
(check-expect (handle-mouse 2 0 3 "button-down") 0 )
(check-expect (handle-mouse 3 2 0 "button-up") 3 )
(check-expect (handle-mouse 9 2 3 "button-down") 0 )
(check-expect (handle-mouse 0 9 2 "drag") 0 )
```



Submit

Question 5

1 point possible (graded)

You came up with this design for handle-mouse:

(defi	(define (handle-mouse cd x y me)			
(cond [(mouse=? me "button-down") 0]))				
nat d	o we need to add to complete this function?			
A C	othing, this function is complete			
) A	dd the cond question			
	[(mouse=? me "button-up") 0]			
) A	dd the cond question			
	[(mouse=? me "button-up") cd]			
) A	dd the cond question			
	[else cd]			
•				
	ation			
e els	e case is necessary because MouseEvent is a large enumeration.			
Sub	mit			