

- DUE DATE part 1: **February 14th @ 10:00pm**For part 1, you must implement the following functions (and any helpers they need):`world-draw`, `move-spaceship`, `move-spaceship-bullets` and `move-invader-bullets`
- DUE DATE part 2: **February 21st @ 10:00pm**For part 2, you must implement the fully functioning game.

## Space Invaders

The purpose of this assignment is to design a **version** of the Space Invaders game. You can play the original game [here](#).

For your version of the game you will have to represent the following elements

- the spaceship
- four rows of invaders
- spaceship bullets
- invader bullets

The spaceship must be located at the bottom of the scene and should be allowed to move left and right only. The spaceship will also have the ability to fire bullets in order to hit invaders. Bullets are fired when the space bar is pressed by the user. Spaceship bullets move vertically (bottom to top) at a steady speed. When the spaceship reaches the edge of the scene then the spaceship stops moving until the user changes the spaceships direction and then the spaceship begins to move again.

Invaders are represented in rows, each row must have at least 9 invaders side by side. Invaders in the same row must have some space between them. Invaders do not move at all. At every clock tick a number of invaders get to fire bullets. The choice of which invader gets to fire on each tick is random. Invader bullets are the same as spaceship bullets in that they are the same size and move vertically. Invader bullets however move in the opposite direction as spaceship bullets, i.e., top to bottom.

When a spaceship bullet hits an invader, then the invader is destroyed. Destroyed invaders should no longer be visible on the canvas and should not be able to fire bullets. The spaceship bullet that hits an invader is also removed from the scene.

The game ends when either all invaders have been eliminated, or, when the spaceship has been hit by an invader bullet.

The version of the game you are going to design will be close to the original Space Invaders with some modifications/restrictions.

- 1.The space ship moves in a steady speed in a direction. The direction is either *left* or *right*. The ship changes direction when the user clicks on the left or right arrow key.
- 2.At any given point in time in the game there can only be at most 3 spaceship bullets in flight.
- 3.At any given point in time in the game there can only be at most 10 invader bullets in flight.

## Recommendations

You are free to use the below definitions/constants for your implementation.

```
(require 2htdp/image)
(require 2htdp/universe)

;; An Invader is a Posn
;; INTERP: represents the location of the invader

;; A Bullet is a Posn
;; INTERP: represents the location of a bullet

;; A Location is a Posn
;; INTERP: represents a location of a spaceship

;; A Direction is one of:
;; - 'left
;; - 'right
;; INTERP: represent direction of movement for the spaceship

(define-struct ship (dir loc))
;; A Ship is (make-ship Direction Location)
;; INTERP: represent the spaceship with its current direction
;;          and movement

;; A List of Invaders (LoI) is one of
;; - '()
;; - (cons Invader LoI)

;; A List of Bullets (LoB) is one of
;; - '()
;; - (cons Bullet LoB)

(define-struct world (ship invaders ship-bullets invader-bullets))
;; A World is (make-world Ship LoI LoB LoB)
;; INTERP: represent the ship, the current list of invaders, the inflight
spaceship bullets
;;          and the inflight invader bullets

(define WIDTH 500)
```

```
(define HEIGHT 500)

(define MAX-SHIP-BULLETS 3)

(define MAX-INVADER-BULLETS 15)

(define BACKGROUND (empty-scene WIDTH HEIGHT))

(define SPACESHIP-BULLET-IMAGE (circle 2 'solid 'black))

(define SHIP-WIDTH 25)

(define SHIP-HEIGHT 15)

(define SPACESHIP-IMAGE (rectangle SHIP-WIDTH SHIP-HEIGHT 'solid 'black))

(define INVADER-SIDE 20)

(define INVADER-IMAGE (square INVADER-SIDE 'solid 'red))

(define INVADER-BULLET-IMAGE (circle 2 'solid 'red))

(define SHIP-SPEED 10)

(define BULLET-SPEED 10)

(define SHIP-INIT (make-ship 'left (make-posn 250 480)))

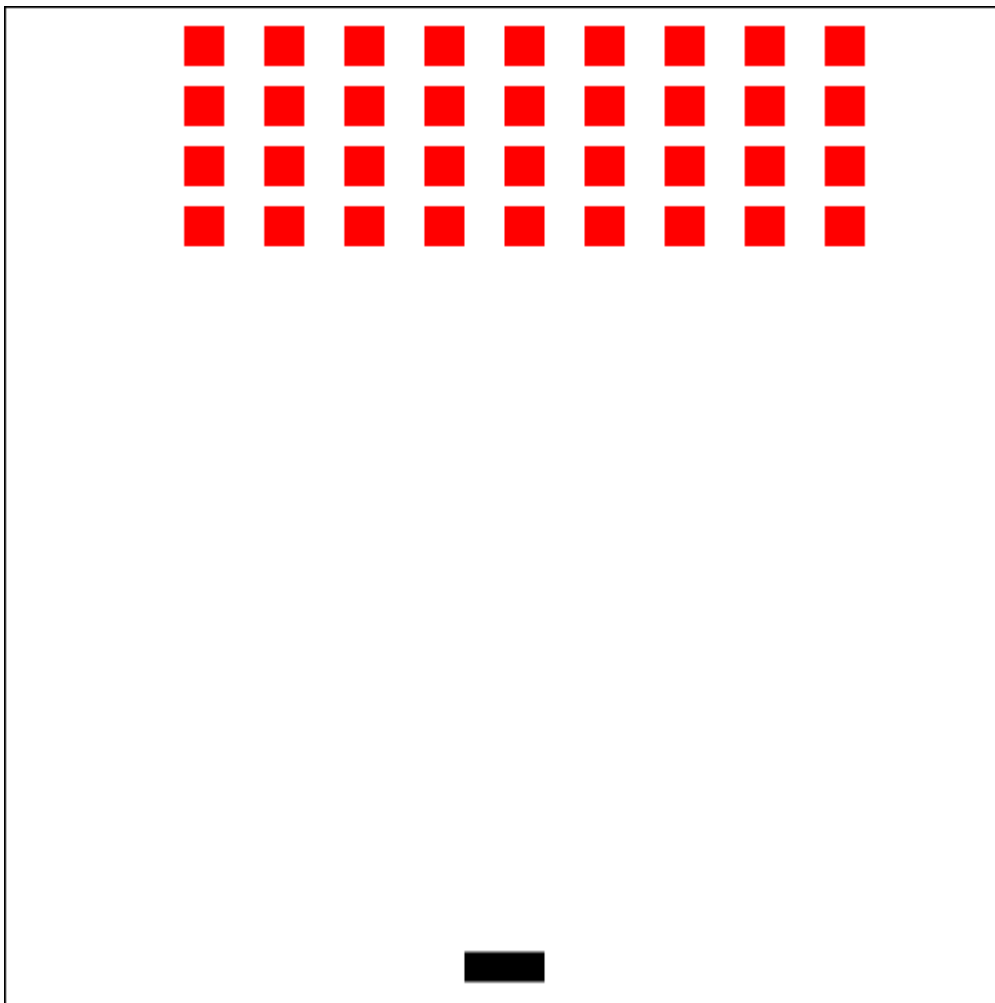
(define INVADERS-INIT
  (list (make-posn 100 20) (make-posn 140 20) (make-posn 180 20)
        (make-posn 220 20) (make-posn 260 20) (make-posn 300 20)
        (make-posn 340 20) (make-posn 380 20) (make-posn 420 20)
        (make-posn 100 50) (make-posn 140 50) (make-posn 180 50)
        (make-posn 220 50) (make-posn 260 50) (make-posn 300 50)
        (make-posn 340 50) (make-posn 380 50) (make-posn 420 50)
        (make-posn 100 80) (make-posn 140 80) (make-posn 180 80)
        (make-posn 220 80) (make-posn 260 80) (make-posn 300 80)
        (make-posn 340 80) (make-posn 380 80) (make-posn 420 80)
        (make-posn 100 110) (make-posn 140 110) (make-posn 180 110)
        (make-posn 220 110) (make-posn 260 110) (make-posn 300 110)
        (make-posn 340 110) (make-posn 380 110) (make-posn 420 110)))
```

```
(define WORLD-INIT (make-world SHIP-INIT INVADERS-INIT empty empty))
```

Start by designing the function

```
;; world-draw : World -> Image  
;; Draw the world on the canvas
```

This will allow you to take a **World** and draw it as an image. Ensure that you have a separate function to draw each component of the world. Here is an example image of the initial world.



You are **strongly** advised to also design the following functions

```
;; move-spaceship: Ship -> Ship  
;; move the ship in the appropriate direction  
  
;; move-spaceship-bullets : LoB -> LoB  
;; move each spaceship bullet in the list upwards by SPEED units
```

```
;; move-invader-bullets : LoB -> LoB
;; move each bullet in the list downwards by SPEED units

;; invaders-fire : LoB LoI -> LoB
;; fire from a random invader to hit the ship

;; remove-hits-and-out-of-bounds: World -> World
;; remove any invaders that have been hit by a spaceship bullet.
;; Remove any bullets that are out of bounds

;; ship-hit : Ship LoB -> Boolean
;; true if a bullet hit the ship, false otherwise
```

Given the above functions then the on-tick function should perform the following tasks

- 1.move the spaceship
- 2.move any spaceship bullets
- 3.fire any invaders bullets if you can
- 4.move any invader bullets
- 5.remove any invaders that were hit and any (invader or spaceship) bullets that are out of bounds

Here is another screen shot while playing the game

