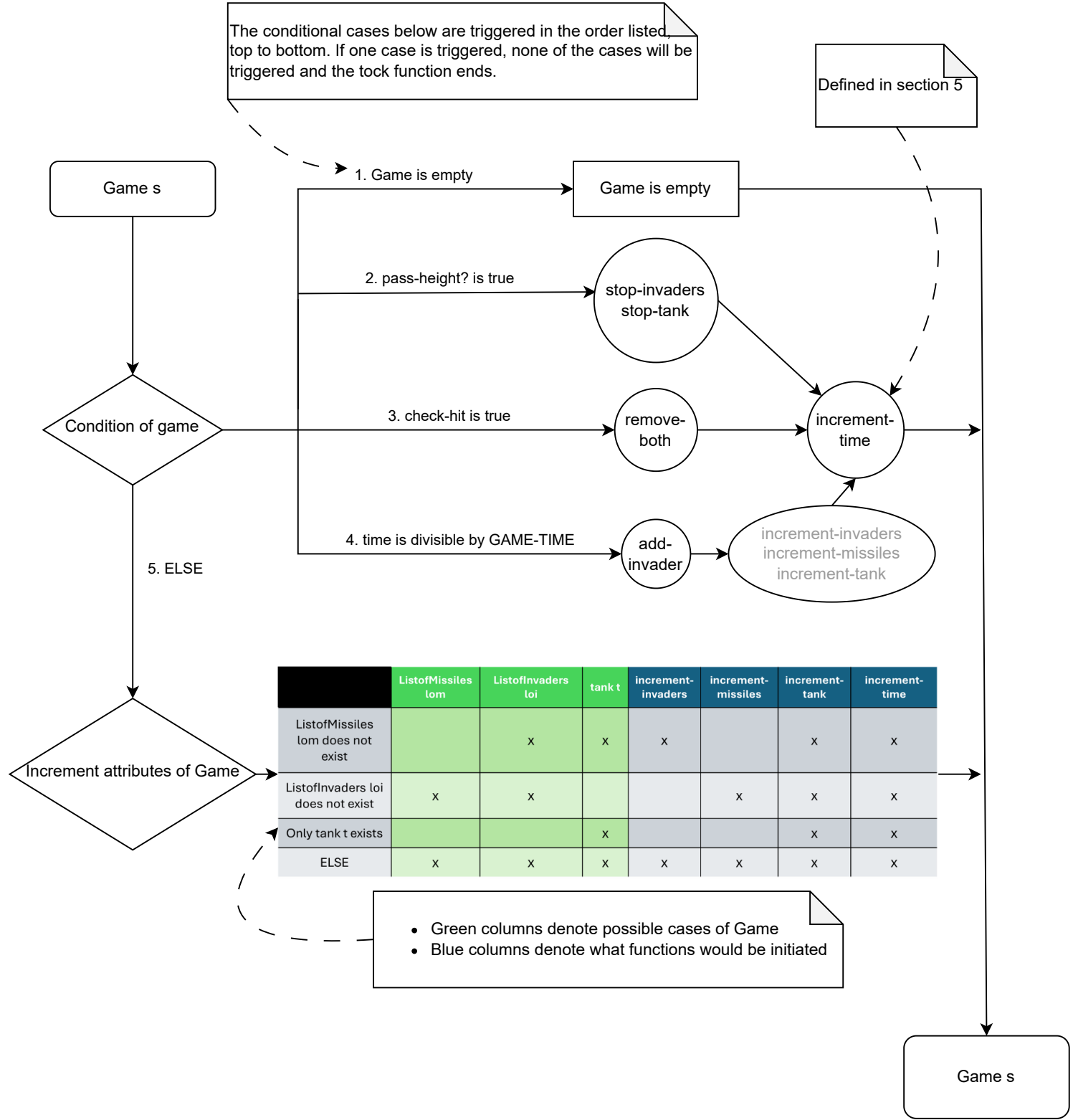**TOCK FUNCTION
(MAIN BODY)**
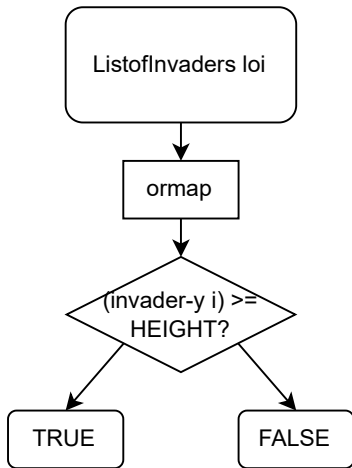
The conditional cases below are triggered in the order listed, top to bottom. If one case is triggered, none of the cases will be triggered and the tock function ends.
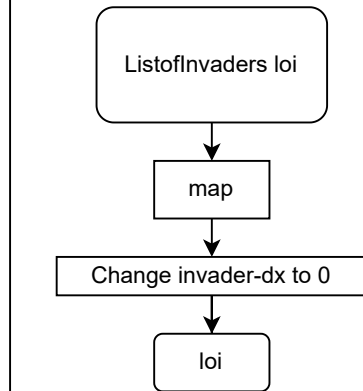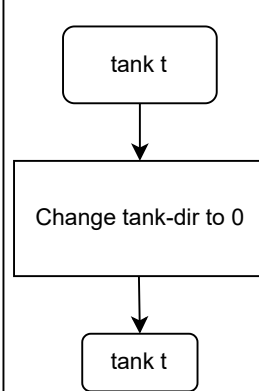
Defined in section 5

Game s

1. Game is empty

Game is empty

Condition of game

2. pass-height? is true

stop-invaders
stop-tank

3. check-hit is true

remove-both

increment-time

4. time is divisible by GAME-TIME

add-invader

increment-invaders
increment-missiles
increment-tank

5. ELSE

Increment attributes of Game

	ListofMissiles lom	ListofInvaders loi	tank t	increment-invaders	increment-missiles	increment-tank	increment-time
ListofMissiles lom does not exist		x	x	x		x	x
ListofInvaders loi does not exist	x	x			x	x	x
Only tank t exists			x			x	x
ELSE	x	x	x	x	x	x	x

- Green columns denote possible cases of Game
- Blue columns denote what functions would be initiated

Game s

CONDITIONAL CASE 2: pass-height? is true

pass-height?
(ListofInvaders -> Boolean)

ListofInvaders loi

↓

ormap

↓

(invader-y i) >= HEIGHT?

TRUE   FALSE

stop-invaders
(ListofInvaders -> ListofInvaders)

ListofInvaders loi

↓

map

↓

Change invader-dx to 0

↓

loi

stop-tank
(Tank -> Tank)

tank t

↓

Change tank-dir to 0
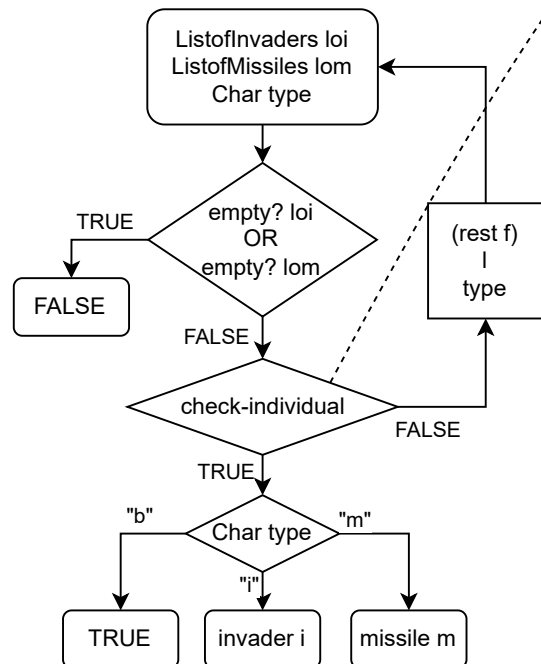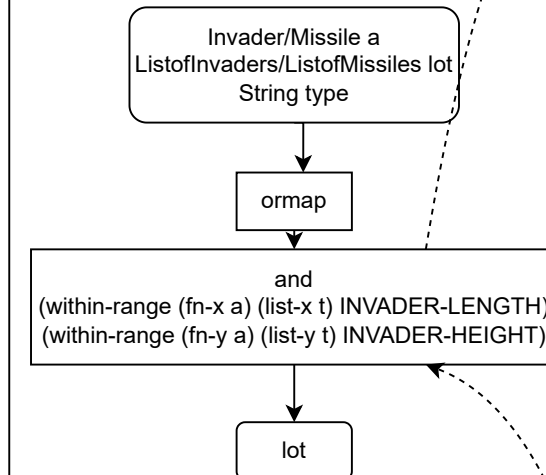
↓

tank t

CONDITIONAL CASE 3: check-hit is true

check-hit
(ListofInvaders ListofMissiles Char -> Boolean/Invader)

ListofInvaders loi
ListofMissiles lom
Char type

↓

empty? loi
OR
empty? lom

TRUE → FALSE

FALSE

↓

check-individual

(rest f)
l
type

FALSE

TRUE

↓

"b"   Char type   "m"
       "i"

TRUE   invader i   missile m

	"b"	"i"	"m"
f (element)	loi	loi	lom
l (list)	lom	lom	loi
t (type)	"invader"	"invader"	"missile"

check-individual
(Invader/Missile ListofInvaders/ListofMissiles String -> Boolean)

Invader/Missile a
ListofInvaders/ListofMissiles lot
String type

↓

ormap

↓

and
(within-range (fn-x a) (list-x t) INVADER-LENGTH)
(within-range (fn-y a) (list-y t) INVADER-HEIGHT)

↓

lot

	a (element)		t (list)	
	fn-x	fn-y	list-x (t)	list-y (t)
"invader"	invader-x	invader-y	missile-x	missile-y
"missile"	missile-x	missile-y	invader-x	invader-y

t is an element of lot, as defined by the anonymized function.

within-range

a
t
Integer f

↓

and
(f/2 - a) >= t
(f/2 + a) <= t

TRUE   FALSE

f is a constant integer.
f is a defined length
of the invader body.

remove-both
(Char ListofInvaders ListofMissiles -> ListofInvaders/ListofMissiles)

remove invade/missile from list

check-hit

invader/missile

Char s
ListofInvaders loi
ListofMissiles lom

type? s

"m"

lom

"i"

loi

filter

not (equal to
invader/missile sharing
same coordinates)

ListofInvaders/ListofMissiles

**CONDITIONAL CASE 4:
time is divisible GAME-TIME**

add-invader
(ListofInvaders Tank -> ListofInvaders)

Tank t
ListofInvaders loi

append new invader into loi

create new invader at
random x-position and
opposite direction of tank

append new invader
into loi

loi
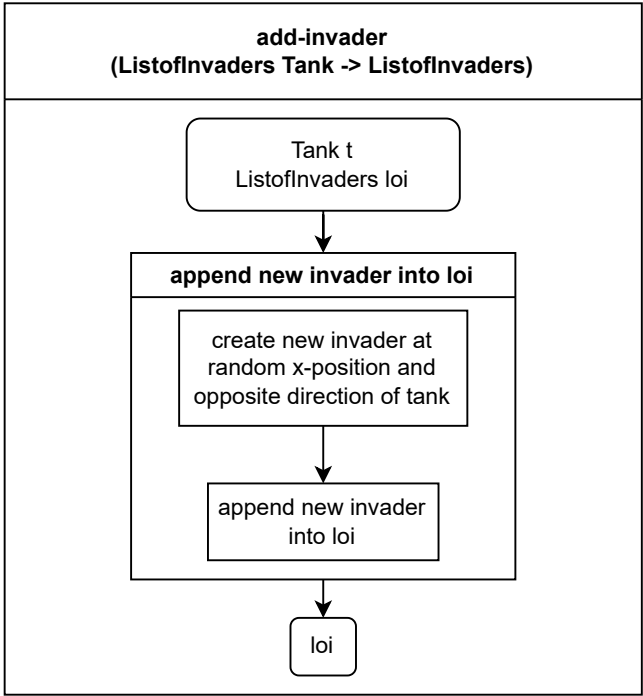
CONDITIONAL CASE 5: ELSE

increment-invader
(ListofInvaders -> ListofInvaders)

ListofInvaders loi

empty? loi
- TRUE → empty
- FALSE

(rest loi)

invader-x <= 0?
- TRUE → Change invader-x to 1 and reverse direction
- FALSE

invader-x >= WIDTH?
- TRUE → Change invader-x to WIDTH - 1 and reverse direction
- FALSE → Increment invader-x & y by invader-dx and INVADER-Y-SPEED
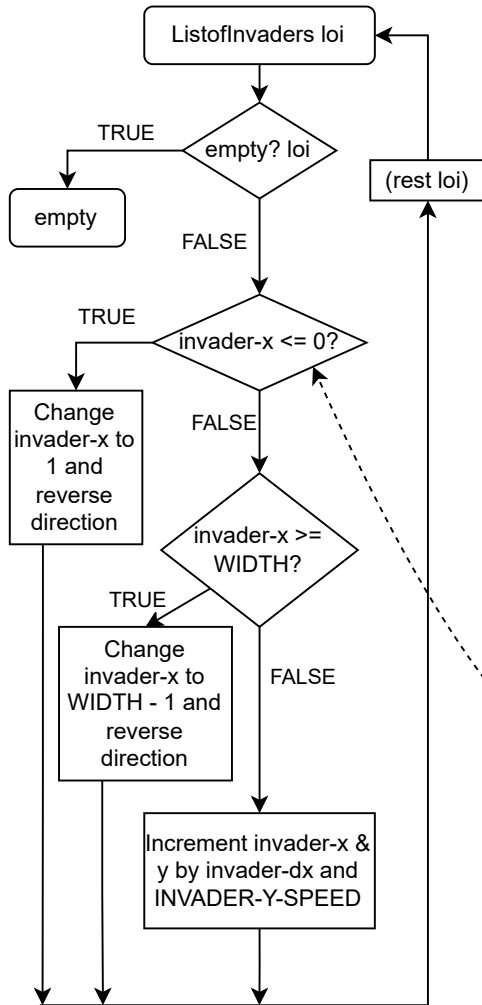
increment-tank
(Tank -> Tank)

Tank t

t-x >= WIDTH
- TRUE → Change t-x to WIDTH - 1
- FALSE

t-x <= 0?
- TRUE → Change t-x to 0
- FALSE → Move t in t-dir

t
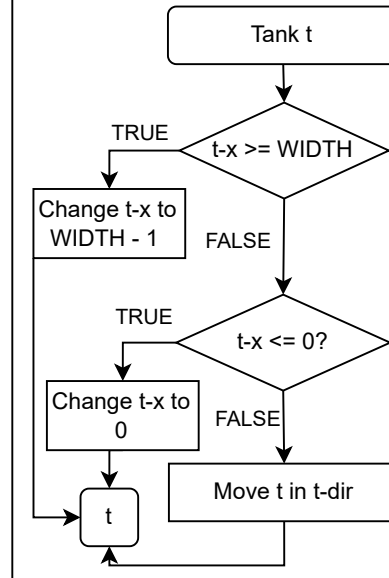
increment-time
(Integer -> Integer)

Integer t

Increase t by TICK

t
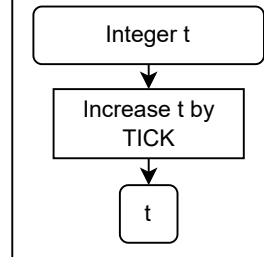
increment-missile
(ListofMissiles -> ListofMissiles)

ListofMissiles lom

Increment all missiles

map

increment missile-y value

filter

not (missile-y < 0)

lom
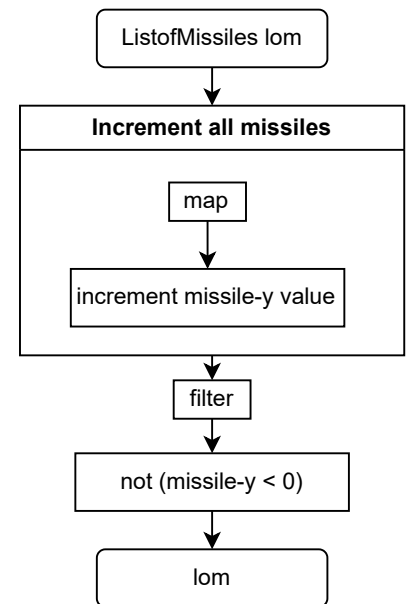
Refers first element of their respective lists.

For increment-missiles, the first element is skipped when calling (rest lom), so said referred element is not included in the return list.