Assignment 4 Specification

Nikou Kalbali

April 5, 2017

The purpose of this specification is to design and specify a modules which store the state of a Battleship Game. Reference: I used Professor smiths Assign3Part1 specification, his makefile, and my code from my previous assignment to do this Assignment.

Constants Module

Module

Constants

Uses

N/A

Syntax

Exported Constants

MAX_R = 10 *//dimension in the x-direction of the board area*
MAX_C = 10 *//dimension in the y-direction of the board area*
OVERLAP = 0 *//space not allowed between points and is therefore considered an overlap*

Exported Access Programs

none

Semantics

State Variables

none

State Invariant

none

Coordinate ADT Module

Template Module

CoordinateT

Uses

Constants

Syntax

Exported Types

CoordinateT = ?

Exported Access Programs

Routine name	In	Out	Exceptions
CoordinateT	stringl, real	CoordinateT	InvalidMovetException
letter		stringl	
number		real	
dist	CoordinateT	real	

Semantics

State Variables

letter: string
number: real

State Invariant

none

Assumptions

none

Access Routine Semantics

CoordinateT($row, column$):

- transition: $rc, cc := row, column$
- output: $out := self$
- exception $exc := ((\neg(0 \leq row \leq \text{Contants.MAX\_R}) \vee \neg(0 \leq column \leq \text{Constants.MAX\_C})) \Rightarrow \text{InvalidMovetException})$

letter():

- output: $out := rc$
- exception: none

number():

- output: $out := cc$
- exception: none

dist(c):

- output: $out := \sqrt{(self.rc - c.letter)^2 + (self.cc - c.number)^2}$
- exception: none

Board Module

Template Module

BoardT

Uses

CoordinateT

Syntax

Exported Types

BoardT = ?

Exported Access Programs

Routine name	**In**	**Out**	**Exceptions**
BoardT	CoordinateT, string, real	BoardT	OutOfBoundsException
pointOverlaping	CoordinateT	boolean	

Semantics

State Variables

lower_left: CoordinateT *//coordinate of lower left corner of ship*
width: real *//width of shipi*
height: real *//height of ship*

State Invariant

None

Assumptions

none once.

Access Routine Semantics

RegionT(c, w, h):

- transition: $\mathit{lower\_left}, \mathit{width}, \mathit{height} := c, w, h$
- output: $\mathit{out} := \mathit{self}$
- exception:

$$\begin{aligned}
&\mathit{exc} := \neg(w > 0 \wedge \\
&\quad h > 0 \wedge \\
&\quad (c.\text{letter}() + w) \leq \text{Constants.MAX\_R} \wedge \\
&\quad (c.\text{number}() + h) \leq \text{Constants.MAX\_C}) \Rightarrow \text{OutOfBoundsException}
\end{aligned}$$

pointOverlaping(c):

- output: $\mathit{out} := \exists(q : \text{CoordinateT} | q \in \text{Board} : c.\text{dist}(q) = \text{Constants.OVERLAP})$
- exception: none

Local Functions

Board: set of CoordinateT

$$\begin{aligned}
&\text{Board} \equiv \cup(q : \text{CoordinateT}| \\
&\quad \mathit{lower\_left}.\text{letter} \leq q.\text{letter} \leq (\mathit{lower\_left}.\text{letter} + \mathit{width}) \wedge \\
&\quad \mathit{lower\_left}.\text{number} \leq q.\text{number} \leq (\mathit{lower\_left}.\text{number} + \mathit{height}) : \{q\})
\end{aligned}$$

Shots Module

Shots Template Module

SHotsList(T)

Uses

N/A

Syntax

Exported Types

ShotsList(T) = ?

Exported Constants

none

Exported Access Programs

Routine name	In	Out	Exceptions
ShotsList		ShotsList	
add	T		
setval	T		
getval	T		

Semantics

State Variables

s: sequence of T

State Invariant

none

Assumptions

none

Access Routine Semantics

ShotsList():

- transition: $self.s :=<>$
- output: $out := self$
- exception: none

add(c):

- transition: $s := s[0-1]|| < c > ||s[0..|s|-1]$

setval(p):

- transition: $s[0] := p$

getval(i):

- output: $out := s[i]$