AllCycle
(self)

Cycles through turning all lights on then off

Example call
northController.AllCycle()

Turn all lights off

AllCycle

Turn both lights to green

Wait self.INTERVAL

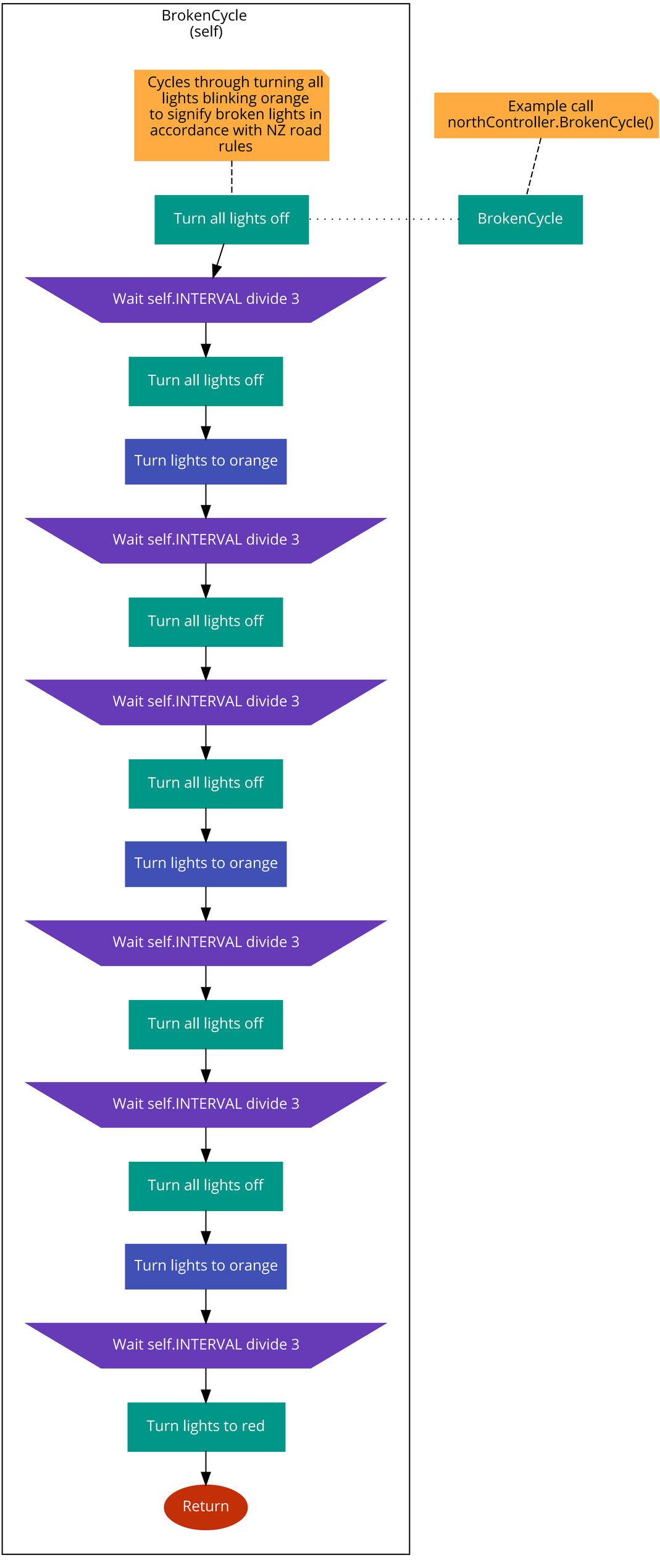
Turn all lights off

Turn both lights to orange

Wait self.INTERVAL

Turn lights to red

Return


BrokenCycle
(self)
Cycles through turning all lights blinking orange to signify broken lights in accordance with NZ road rules
Example call northController.BrokenCycle()
Turn all lights off
BrokenCycle
Wait self.INTERVAL divide 3
Turn all lights off
Turn lights to orange
Wait self.INTERVAL divide 3
Turn all lights off
Wait self.INTERVAL divide 3
Turn all lights off
Turn lights to orange
Wait self.INTERVAL divide 3
Turn all lights off
Wait self.INTERVAL divide 3
Turn all lights off
Turn lights to orange
Wait self.INTERVAL divide 3
Turn lights to red
Return

IncrementAllWaits

Is used to add 1 to the wait times for every light

Example function call

northController.IncrementRightWait()

IncrementAllWaits

northController.IncrementOtherWait()

eastController.IncrementRightWait()

eastController.IncrementOtherWait()

southController.IncrementRightWait()

southController.IncrementOtherWait()

westController.IncrementRightWait()

westController.IncrementOtherWait()

Return

OtherCycle
(self)

Cycles through turning all other lights on then off
Leaving right lights off

Turn all lights off

Turn right light to red
Turn other light to green

Wait self.INTERVAL

Turn all lights off

Turn right light to red
Turn other light to orange

Wait self.INTERVAL

Turn lights to red

Return

Example call
northController.OtherCycle()

OtherCycle

ResetAllWaits

Is used to reset the wait times for every light

Example function call

northController.ResetRightWait()

ResetAllWaits

northController.ResetOtherWait()

eastController.ResetRightWait()

eastController.ResetRightWait()

southController.ResetRightWait()

southController.ResetOtherWait()

westController.ResetRightWait()

westController.ResetOtherWait()

Return

RightCycle
(self)

Cycles through turning all
right lights on then off

Example call
northController.RightCycle()

Turn all lights off

RightCycle

Turn right light to green
Turn other light to red

Wait self.INTERVAL

Turn all lights off

Turn right light to orange
Turn other light to red

Wait self.INTERVAL

Turn lights to red

Return

RunDual
(controllerOne, controllerTwo)

We use threading to turn on two seperate light controllers at once

set a thread worker to controllerOne.Cycle

set a thread worker to controllerTwo.Cycle

start both thread workers

Call join() to wait for them to finish

Reset the wait time for controllerOne

Reset the wait time for controllerTwo

Return

northController, southController

An example function call

RunDual

sensor
(controller, lane)

This controls the removal of cars from the releant lane of traffic

Example function call

sensor

northController, northLane

lights are broken

True

False

No cars go

More cars in the lane

True

Both lights are green or orange

True

False

Right turning light is green or orange

Remove next car in lane

True

False

False

Other light is green or orange

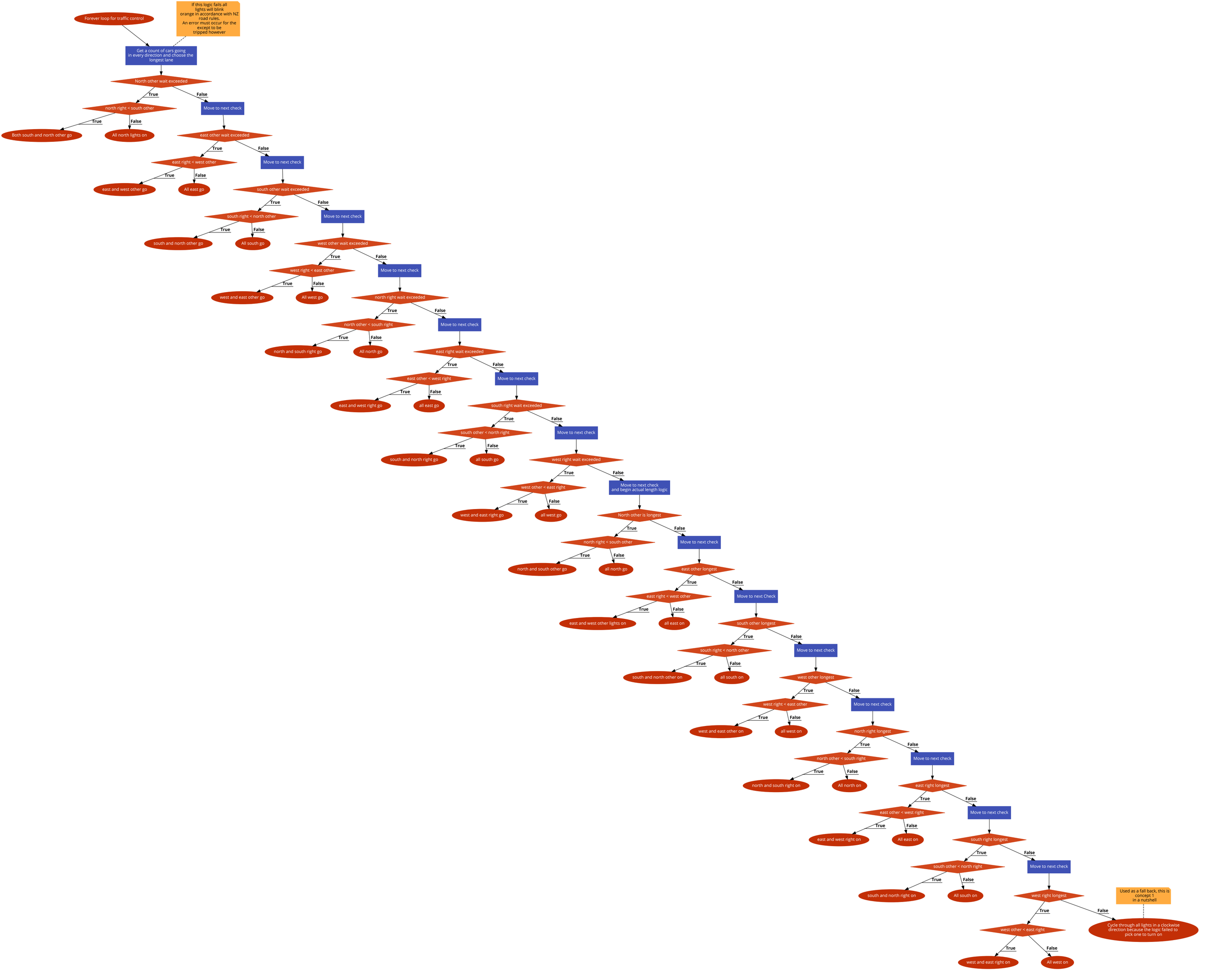
Remove next car going right in said lane

True

Remove next car going straight or left in said lane

False

Return


Forever loop for traffic control
if this logic fails all lights will blink orange in accordance with NZ road rules. An error must occur for the except to be tripped however
Get a count of cars going in every direction and choose the longest lane
North other wait exceeded
True
False
north right < south other
Move to next check
True
False
Both south and north other go
All north lights on
east other wait exceeded
True
False
east right < west other
Move to next check
True
False
east and west other go
All east go
south other wait exceeded
True
False
south right < north other
Move to next check
True
False
south and north other go
All south go
west other wait exceeded
True
False
west right < east other
Move to next check
True
False
west and east other go
All west go
north right wait exceeded
True
False
north other < south right
Move to next check
True
False
north and south right go
All north go
east right wait exceeded
True
False
east other < west right
Move to next check
True
False
east and west right go
all east go
south right wait exceeded
True
False
south other < north right
Move to next check
True
False
south and north right go
all south go
west right wait exceeded
True
False
west other < east right
Move to next check and begin actual length logic
True
False
west and east right go
all west go
North other is longest
True
False
north right < south other
Move to next check
True
False
north and south other go
all north go
east other longest
True
False
east right < west other
Move to next Check
True
False
east and west other lights on
all east on
south other longest
True
False
south right < north other
Move to next check
True
False
south and north other on
all south on
west other longest
True
False
west right < east other
Move to next check
True
False
west and east other on
all west on
north right longest
True
False
north other < south right
Move to next check
True
False
north and south right on
All north on
east right longest
True
False
east other < west right
Move to next check
True
False
east and west right on
All east on
south right longest
True
False
south other < north right
Move to next check
True
False
south and north right on
All south on
west right longest
True
False
west other < east right
Used as a fall back, this is concept 1 in a nutshell
True
False
west and east right on
All west on
Cycle through all lights in a clockwise direction because the logic failed to pick one to turn on