class Board

def self.build\_stacks(n)

@board = Array.new(n) { Array.new() }

end

def initialize(n, max\_height)

raise "rows and cols must be >= 4" if n < 4 || max\_height < 4

@max\_height = max\_height

@stacks = Board.build\_stacks(n)

end

def max\_height

@max\_height

end

# attr\_reader :max\_height

def add(token, stack\_idx)

if @stacks[stack\_idx].length < @max\_height

@stacks[stack\_idx] << token

return true

else

false

end

end

def vertical\_winner?(token)

# @stacks.each do |stack|

# stack.each do |piece|

# if stack.length != @max\_height || piece != token

# break

# end

# return true

# end

# end

# false

@stacks.each do |stack|

return true if stack.length == @max\_height && stack.all? { |piece| piece == token }

end

false

end

def horizontal\_winner?(token)

(0...@stacks[0].length).each do |i|

count = 0

(0...@stacks.length).each do |j|

count += 1 if @stacks[j][i] == token

end

return true if count == @stacks.length

end

false

end

def winner?(token)

self.vertical\_winner?(token) || self.horizontal\_winner?(token)

end

# This Board#print method is given for free and does not need to be modified

# It is used to make your game playable.

def print

@stacks.each { |stack| p stack }

end

end



[6:10](https://app-academy.slack.com/archives/C03EGEH0YSG/p1656119421322159)

class Player

attr\_reader :token

def initialize(token)

@token = token

end

# This Player#get\_stack\_index method is given for free and does not need to be modified

# It is used to make your game playable.

def get\_stack\_index

p "player " + @token + ", enter a valid stack index"

gets.chomp.to\_i

end

end

[6:10](https://app-academy.slack.com/archives/C03EGEH0YSG/p1656119443891569)

require\_relative 'board'

require\_relative 'player'

class ConnectTacToe

def initialize(n, max\_height)

@board = Board.new(n, max\_height)

@player\_1 = Player.new('y')

@player\_2 = Player.new('b')

@current\_player = @player\_1

end

def switch\_players!

if @current\_player == @player\_1

@current\_player = @player\_2

else

@current\_player = @player\_1

end

end

def play\_turn

idx = @current\_player.get\_stack\_index

@board.add(@current\_player.token, idx)

self.switch\_players!

end

# This ConnectTacToe#play method is given for free and does not need to be modified

# It is used to make your game playable.

def play

until @board.winner?(@player\_1.token) || @board.winner?(@player\_2.token)

@board.print

self.play\_turn

p "--------------------"

end

self.switch\_players!

@board.print

p @current\_player.token + " has won!"

end

end

[6:11](https://app-academy.slack.com/archives/C03EGEH0YSG/p1656119466099589)

def win\_diagonal?(mark)

left\_to\_right = (0...@grid.length).all? do |i|

pos = [i, i]

self[pos] == mark

end

right\_to\_left = (0...@grid.length).all? do |i|

row = i

col = @grid.length - 1 - i

pos = [row, col]

self[pos] == mark

end

left\_to\_right || right\_to\_left

end