

- Building the snake game
 - Modules required
 - Main function (create snake object & calling the step function for n times)
 - Snake init function(input : width, height, length,)
 - snake generate
 - food generate
 - Gui start and initialize with snake and food
 - global variable : done (indicates if the game ended in a step), width,height (of the playground), snake list, food_loc (food location)
 - done = false
 - Snake generate function (input : length)
 - global variable snake = snake list(length)
 - Food generate function
 - global variable food_loc = random(within width,height) not exists in snake
 - Step function (input : directions)
 - if done is false : game continues otherwise return
 - move snake in the direction
 - verify if the snake survives
 - verify if food eaten
 - render gui
 - Move snake function (input : directions)
 - add new head position in the snake list
 - pop out the tail position in the snake list
 - Snake survive function
 - if snake head goes out of bounds of width/height of playground or overlapps with its own body then snake dies (done=true) otherwise survives
 - Food eaten function
 - if snake head == food_loc : food generate else pop snake tail #allowing snake to grow
- Building a NN model
 - Parameters
 - obstacle on the front (Y/N)
 - obstacle on the left (Y/N)
 - obstacle on the right (Y/N)
 - angle between snake head direction and apple direction (-180 to 180) normalize(-1 to 1)
 - direction to be taken (-1 = left, 0 = forward, 1 = right)
 - Decision (-1 = snake dies, 0 = wrong direction, 1 = right direction)[if score increased or distance from food decreased then right otherwise wrong direction]
 - Data creation
 - get snake direction (head – (head[-1]))
 - front = head + snake direction, left = ?

dir = [1,0] left = [0,1] right = [0,-1]

dir = [0,1] left = [-1,0] right = [1,0]

dir = [-1,0] left = [0,-1] right = [0,1]

dir = [0,-1] left = [1,0] right = [-1,0]

So,

right = opposite direction of left
= -left ,

left = direction vector rotated by 90 degree anti clockwise
= $(Dx \cdot \cos 90 - Dy \cdot \sin 90, Dx \cdot \sin 90 + Dy \cdot \cos 90)$
= $(-Dy, Dx)$

- obstacle(point) : check point is within the environment bounds and not on the snake body
- apple direction = apple – snake head
- angle between snake direction and apple direction
 - $\cos(\text{angle}) = \text{dot product} / (\text{prod of length of both vectors})$
 - $\sin(\text{angle}) = \text{cross product} / (\text{prod of length of both vectors})$
 - $\tan(\text{angle}) = \text{cross product} / \text{dot product}$
 - $\text{angle} = \text{atan}(\text{cross product} / \text{dot product})$
- direction to be taken = take a step in random direction and see if you survive