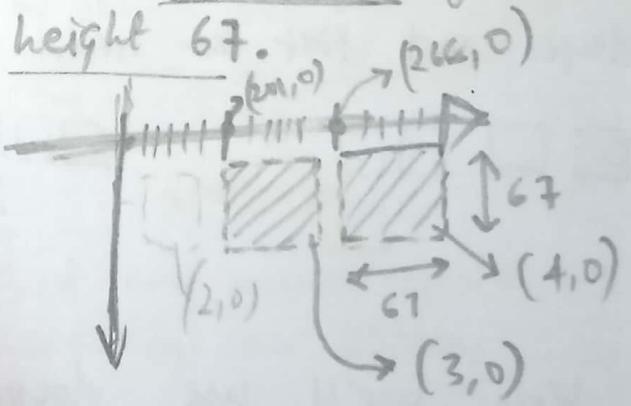
① Snake Game (oops)

In this game we'll use Array data structure.

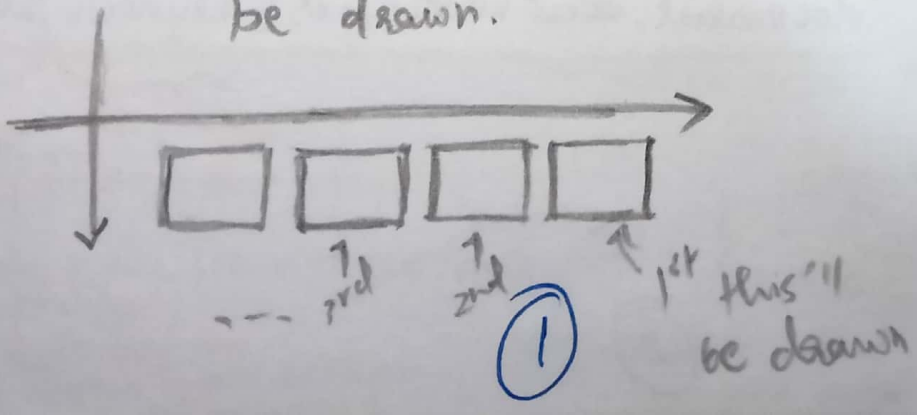
- How 1st Iteration of gameloop goes?

→ Sabse pehele init() fn call hua! init() me hee we've called 'createSnake()' fn which creates a dir. of (x,y) coordinates ie [(4,0), (3,0), (2,0), (1,0)]. Then we call gameloop() fn for 'n' times. what gameloop does is it calls draw() & update() fns. Draw() fn calls drawSnake() fn defined in init() fn's snake JSON obj. what it does is that it creates a rect. from (x * size, y * size) of size-2, size-2.
(w, h)

ex 1st it'll draw from (4*67, 0*67) = (268, 0) of width 67 & height 67.

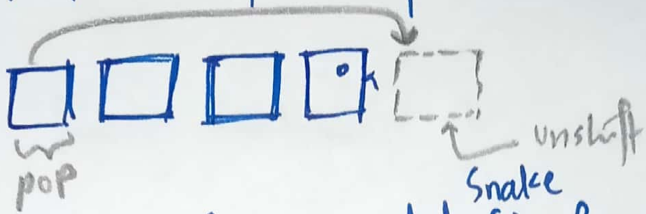


like this our snake will be drawn.



For adding Movement

what we'll do is, we'll remove last rect. of snake and add it ~~to~~ in front of snakes head.

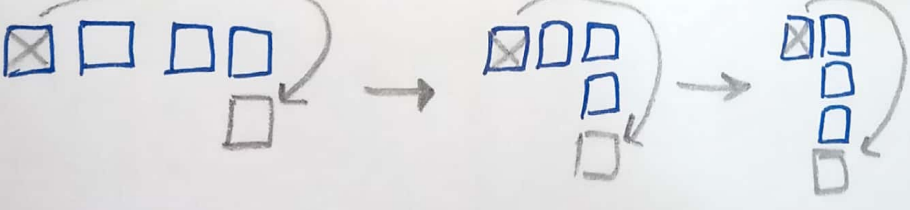


we'll do this using update() fn which we'll create in our snake JSON obj. To show OOPS cncpt we' call it in anothr fn called update(). Although we have created a new rect. in front of our snake but we haven't yet deleted the rect. which we ~~popd~~ popped or moved in front. To do so call ~~draw~~ add one new feature in draw() fn (to erase old frame) by writh pen.clearRect(0,0,W,H);

↳ clears everything drawn on canvas

Directn snake based on Arrow keys

what we'll do is when presd down arrow ↓, apn last rect. hata deye and first ke neeche laga deye.



like this we'll do for left and up also!

Here we'll use document.~~even~~ addEventListener('keydown', keyPressed)

whn prsd a key → trigr this fn (n this fn will implicitly receive an json objct which will've many details like key name etc.)

```
function keypressed(e){
    console.log(e.key);
}
```

↳ this tells/display prsd key

Scanned with CamScanner

```
<html>
  <head>
    <style>
      #mycanvas{
          background-image: url("Assets/bg.jpg");
          border: 20px solid green;
      }
    </style>
  </head>
  <body>
    <canvas id="mycanvas"></canvas>
    <script src="snake.js"></script>
</body>
</html>
```

snake.js

```
function init(){
    canvas = document.getElementById('mycanvas');
    w = H = canvas.width = canvas.height = 1000;
    pen = canvas.getContext('2d');
```

On which we draw

by which we draw diff objs on the canvas.

```
(cs) cell_size = 66;

    food = getrandomfood();
    game_over = false;
    score = 0;

    food_img = new Image();
    food_img.src = "Assets/apple.png";
```

food img which apears randomly on screen.

when true snake stops

Creat an img obj for food

```
trophy = new Image();
trophy.src = "Assets/trophy.png";
```
} to display score

```
snake = {
    init_len: 4,
    color: "blue",
    cells: [],
    direction: "right",

    createSnake: function () {
        for (var i = this.init_len; i > 0; i--) {
            this.cells.push ({x:i, y:0});
        }
    },

    drawSnake: function () {
        for (var i = 0; i < this.cells.length; i++) {
            pen.fillRect (this.cells[i].x * cs, this.cells[i].y * cs,
                          cs-2, cs-2);
        }
    },
```

```
updateSnake: function(){
    console.log("updating snake.");

    var headX = this.cells[0].x;
    var headY = this.cells[0].y;

    if (headX == food.x && headY == food.y){
        food = getrandomfood();
        score += 1;
    }
    else{
        this.cells.pop();
    }

    var nextx, nexty;
    if (snake.direction == 'right'){
        nextX = headX+1;
        nextY = headY;
    }
    else if (snake.direction == 'left'){
        nextX = headX-1;
        nextY = headY;
    }
    else if (snake.direction == 'down'){
        nextX = headx;
        nextY = headY+1;
    }
    else{
        nextX = headX;
        next Y = headY-1;
    }
}
```

If snake eats the food the it'll grow in size and food will apear elsewhere, else it'll keep movn in prev directn.

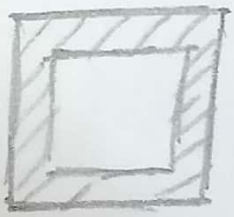
```
        this.cells.unshift({x: nextX, y: nextY});

        // Follow" logic prevents snake from going out of bounds
        var lastX = Math.round(WIDTH/CS);
        var lastY =     "       (HIGHT/CS);

        if (this.cells[0].x > lastX || this.cells[0].y > lastY ||
            this.cells[0].x < 0 || this.cells[0].y < 0){

            game-over = true;
        }
    } // updateSnake finished
} // Snake JSON object finished

snake.createSnake();

function keyPressed(e){
    if (e.key == 'ArrowRight') {
        snake.direction = 'right';
    }
    else if (e.key == 'ArrowLeft') {
        snake.direction = 'left';
    }
    else if (e.key == 'ArrowDown'){
        snake.direction = 'down';
    }
    else{
        snake.direction = 'up';
    }
}

document.addEventListener('keydown', keypressed);
} // Init() finished
```

Add^n Event Listener on doc. object.

```
draw(){
pen.clearRect(0,0,w,H); ← erase the old frame
snake.drawSnake();

pen.fillStyle = food.color;
pen.drawImage(food_img, food.x*cs, food.y*cs,
              cs, cs);
```
} To display the food obj/img

```
//to display the score on trophy
pen.drawImage(trophy, 18, 20, cs, cs);
pen.fillStyle = "black";
pen.font = "25px Roboto";
pen.fillText(score, 50, 50);
}
```
draw() finish

```
function update(){
    snake.updateSnake();
}

function getRandomFood(){
    var foodX = Math.round(Math.random()*(w-cs)/cs);
    var foodY =          "              H          ;

    var food = {
        x: foodX;
        y: foodY;
        color: "red";
    }
    return food;
}
```



We subtracted 'cs' taki apple shaded region me nhi aye. & divided it taki cs ke multiples me hee aye!

```
function gameloop() {
    if (game_over == true) {
        clearInterval(f);
        alert("Game over");
        return;
    }
    draw();
    update();
}

// main body
init();
var f = setInterval(gameloop, 100);
```