



## FLAPPY BIRD GAME WITH PURE JAVASCRIPT

\* PRESENTED BY:

ABDELRAHMAN MOHAMED EHAB MOHAMED

\* PRESENTED TO:

ENG./ MARIHAN



# HTML Code

index.html X JS game.js


index.html > html > body

```
1  <!DOCTYPE html>
2  <html lang="en">
3  > <head> ...
27 </head>
28 <body>
29
30 <!-- *****Html Code***** -->
31 <canvas id="bird" width="320" height="480"></canvas>
32
33 <!-- *****Script***** -->
34 <script src="game.js"></script>
35 </body>
36 </html>
```

# Java Script Code

## Draw to The Canvas :

```
const cvs = document.getElementById( "mycanvas" );
const ctx = cvs.getContext("2d");
let frames = 0;

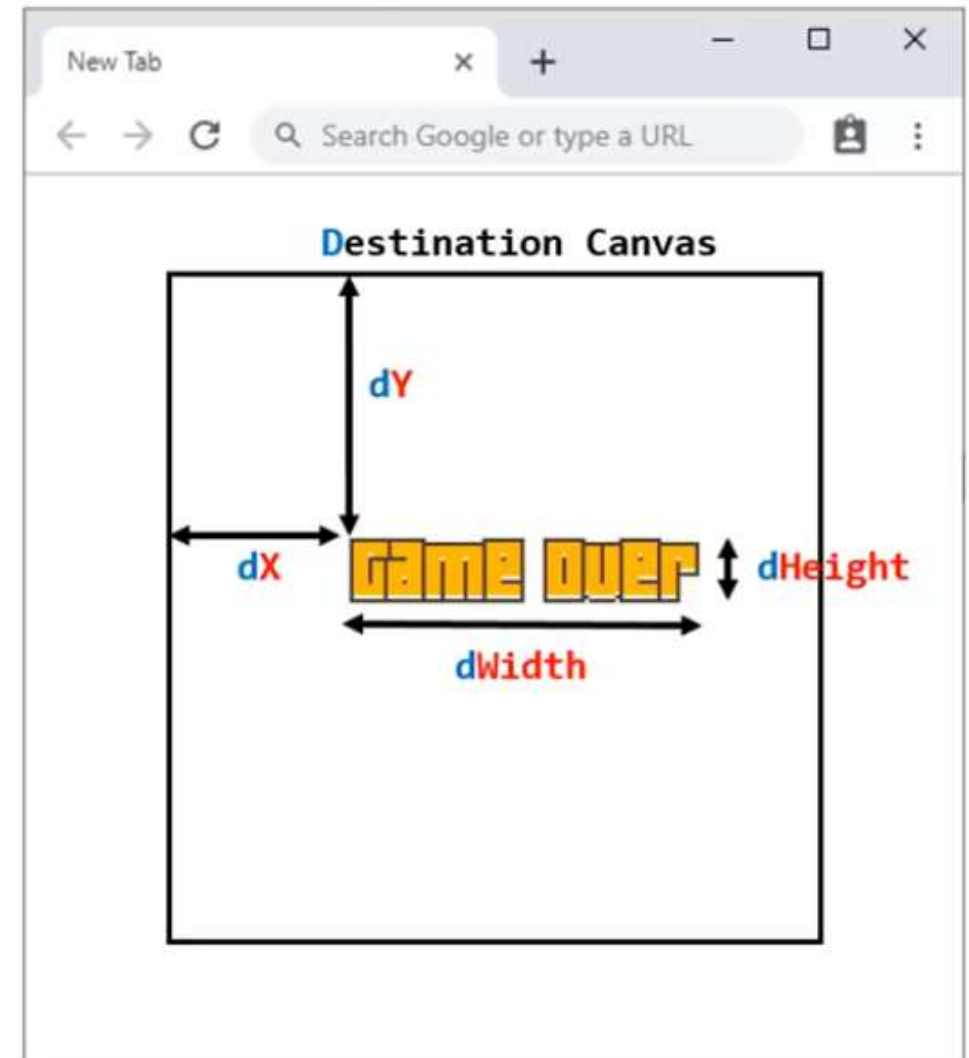
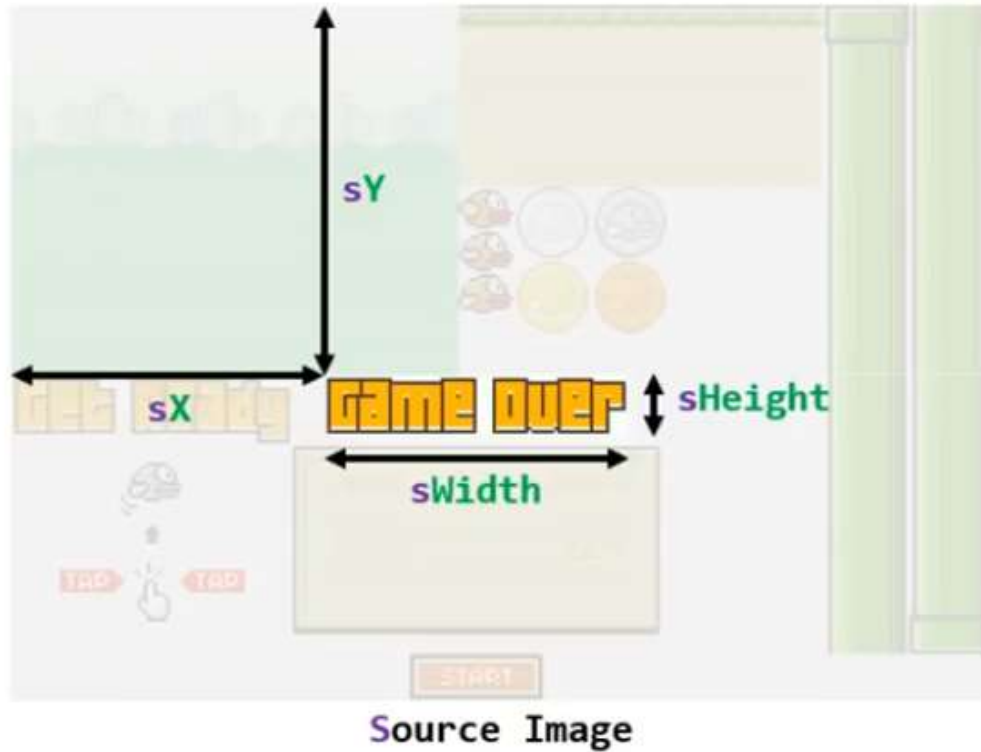
function draw(){
    ctx.fillStyle = "#70c5ce"; 
    ctx.fillRect( 0, 0, cvs.width, cvs.height );
}

function loop(){
    update();
    draw();
    frames++;
    requestAnimationFrame(loop);
}

loop();
```

# Draw image to The Canvas :

## Java Script Code



```
const sprite = new Image();  
sprite.src = "img/sprite.png";
```

```
ctx.drawImage(sprite, sX, sY, sWidth, sHeight, dX, dY, dWidth, dHeight);
```

# Draw image to The Canvas :

## Java Script Code

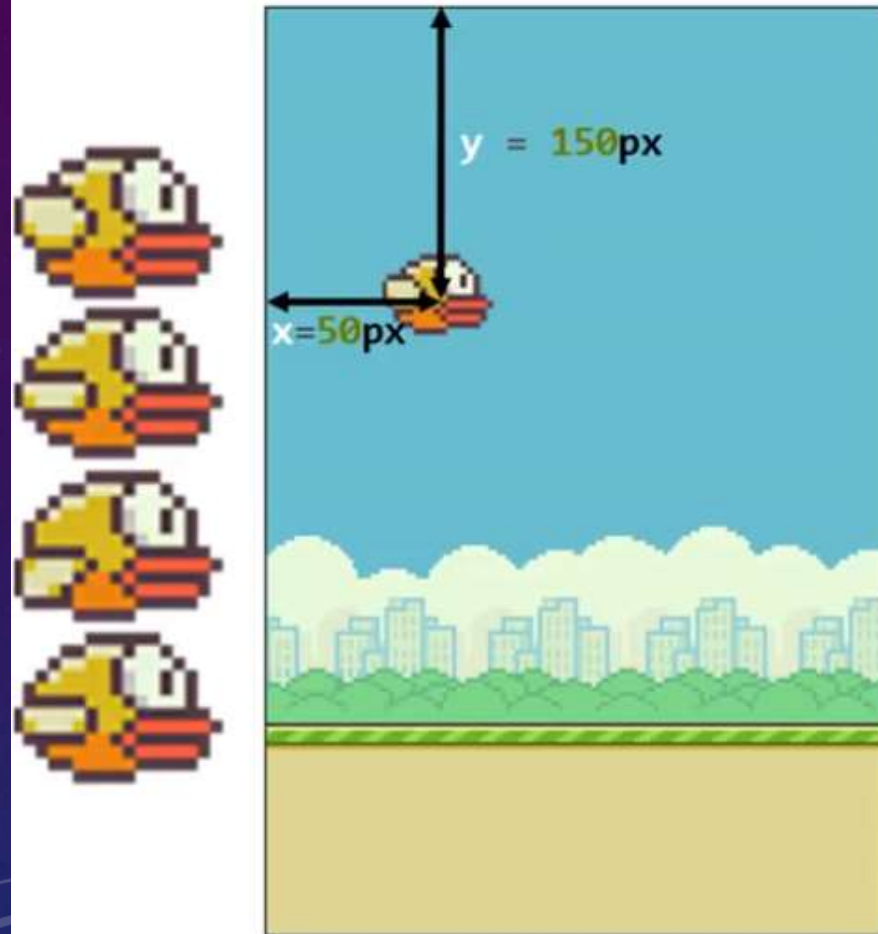


```
const name = {  
  sX : 276,  
  sY : 112,  
  w : 34,  
  h : 26,  
  x : 0,  
  y : 0,  
  
  draw : function(){  
    ctx.drawImage(sprite, this.sX, this.sY, this.w,  
      this.h, this.x, this.y, this.w, this.h);  
  }  
}  
  
function draw(){  
  name.draw();  
}
```



# Draw The Bird:

# Java Script Code

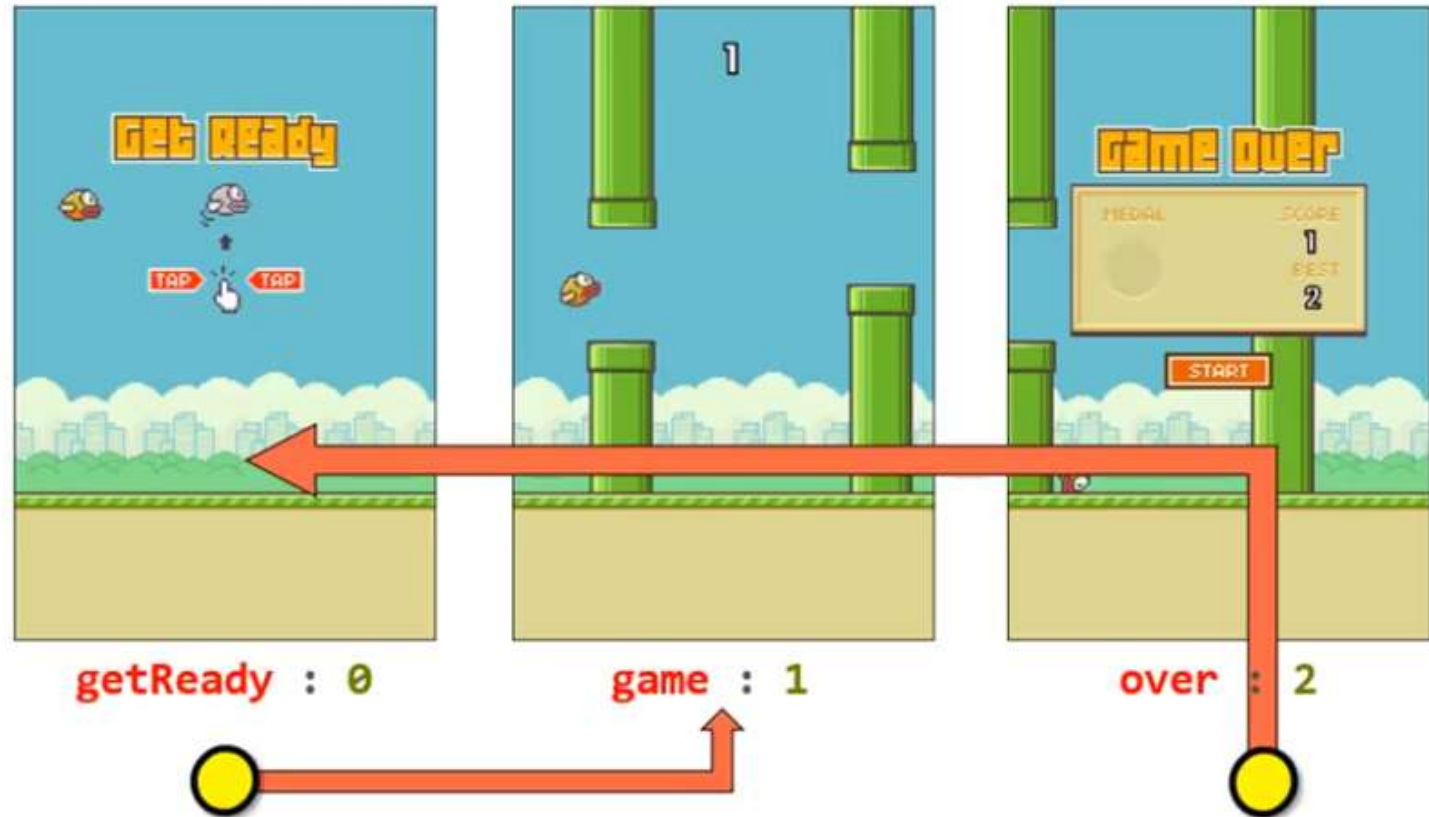


```
const bird = {  
  animation : [  
    { sX : 276, sY : 112 }, this.animation[0]  
    { sX : 276, sY : 139 }, this.animation[1]  
    { sX : 276, sY : 164 }, this.animation[2]  
    { sX : 276, sY : 139 } this.animation[3]  
  ],  
  x : 50,  
  y : 150,  
  w : 34,  
  h : 26,  
  frame : 0,  
  draw : function(){  
    let bird = this.animation[this.frame];  
    ctx.drawImage(sprite, bird.sX, bird.sY,  
      this.w, this.h, this.x - this.w/2,  
      this.y - this.h/2, this.w, this.h);  
  }  
}
```

# Game State:

## Java Script Code

```
const state = {  
  current : 0,  
  getReady : 0,  
  game : 1,  
  over : 2  
}
```



```
cvs.addEventListener("click", function(evt){  
  switch (state.current){  
    case state.getReady:  
      state.current = state.game;  
      break;  
    case state.game:  
      bird.flap();  
      break;  
    case state.over:  
      state.current = state.getReady;  
      break;  
  }  
})
```

# Java Script Code

## Handle The States:

```
const getReady = {
  sX : 0,
  sY : 228,
  w : 173,
  h : 152,
  x : cvs.width/2 - 173/2,
  y : 80,

  draw : function(){
    if(state.current == state.getReady){
      ctx.drawImage(sprite, this.sX,
        this.sY, this.w, this.h, this.x,
        this.y, this.w, this.h);
    }
  }
}
```

```
const gameOver = {
  sX : 175,
  sY : 228,
  w : 225,
  h : 202,
  x : cvs.width/2 - 225/2,
  y : 90,

  draw : function(){
    ctx.drawImage(sprite, this.sX,
      this.sY, this.w, this.h, this.x,
      this.y, this.w, this.h);
  }
}
```



# Animation of the bird:

# Java Script Code

```
const bird = {  
  frame : 0,  
  draw : function(){  
    let bird = this.animation[this.frame];  
    ctx.drawImage(sprite, bird.sX, bird.sY,  
      this.w, this.h, this.x, this.y, this.w,  
      this.h);  
  }  
  update : function(){  
  
    this.frame += frames % 5 == 0 ? 1 : 0;  
  
  }  
}
```

```
function loop(){  
  frames++;  
  requestAnimationFrame(loop);  
}  
  
frames = 0;    this.frame: 0,    frames % 5 == 0  
frames = 1;    this.frame= 1,    this.frame += 1;  
frames = 2;                                frames % 5 != 0  
frames = 3;  
frames = 4;  
frames = 5;    this.frame: 2,    frames % 5 == 0  
    . frames % 5 != 0  
    .  
frames = 10;    this.frame: 3,    frames % 5 == 0  
    . frames % 5 != 0  
    .  
frames = 15;    this.frame: 4,    frames % 5 == 0  
    . frames % 5 != 0  
    .  
frames = 20;                                frames % 5 == 0
```

# Animation of the bird: Java Script Code

```
const bird = {  
  frame : 0,  
  draw : function(){  
    let bird = this.animation[this.frame];  
    ctx.drawImage(sprite, bird.sX, bird.sY,  
      this.w, this.h, this.x, this.y, this.w,  
      this.h);  
  }  
  update : function(){  
    this.period = state.current == state.getReady ? 10 : 5;  
    this.frame += frames % this.period == 0 ? 1 : 0;  
    this.frame = this.frame % this.animation.length;  
  }  
}
```



```
function loop(){  
  frames++;  
  requestAnimationFrame(loop);  
}
```

`this.animation.length` => 4

`this.frame = 0`      `0 % 4 = 0`

`this.frame = 1`      `1 % 4 = 1`

`this.frame = 2`      `2 % 4 = 2`

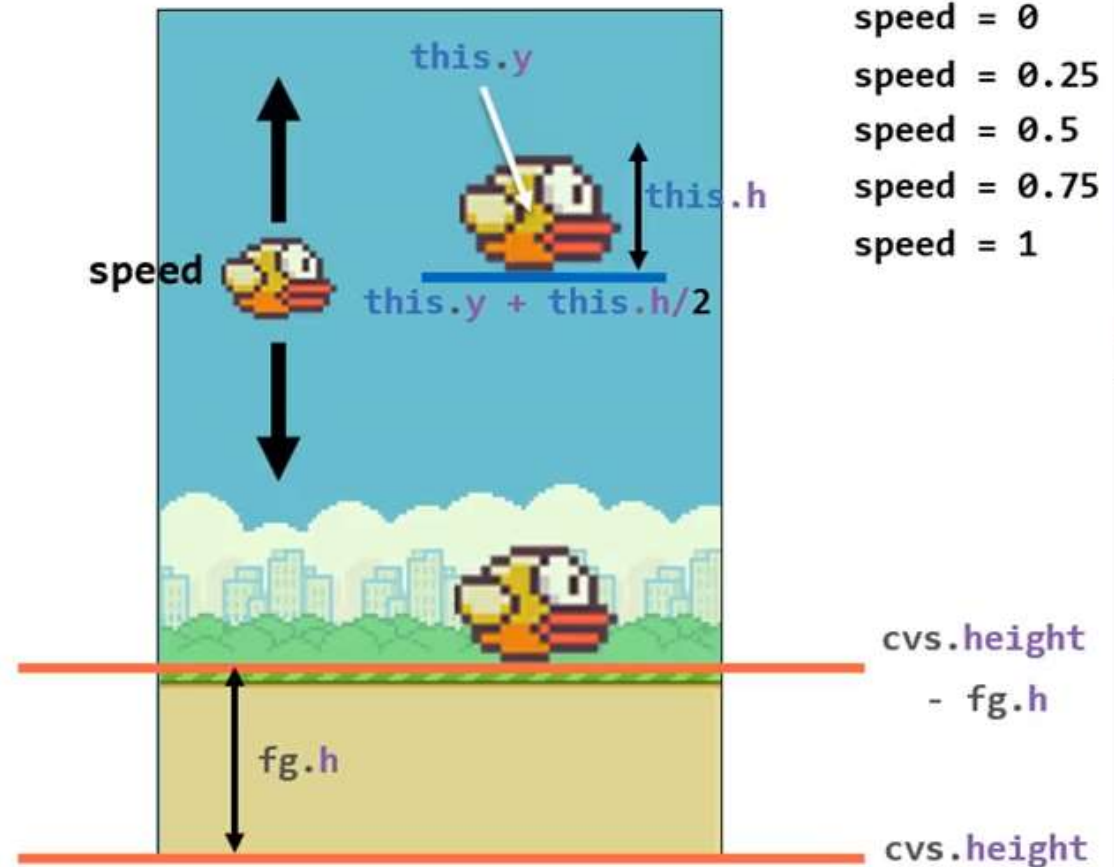
`this.frame = 3`      `3 % 4 = 3`

`this.frame = 4`      `4 % 4 = 0`

# Java Script Code

## Control the gravity and flap:

```
const bird = {
  speed : 0,
  gravity : 0.25,
  jump : 4.6,
  update : function(){
    if(state.current == state.getReady){
      this.y = 150;
    }else{
      this.speed += this.gravity;
      this.y += this.speed;
      if(this.y + this.h/2 >= cvs.height-fg.h){
        this.y = cvs.height - fg.h - this.h/2;
        if(state.current == state.game){
          state.current = state.over;
        }
      }
    }
  },
  flap : function(){
    this.speed = - this.jump;
  }
}
```



# Java Script Code

## Bird Rotation:

```
const DEGREE = Math.PI/180;

const bird = {
  rotation : 0,

  draw : function(){
    let bird = this.animation[this.frame];
    ctx.save();

    ctx.translate(this.x, this.y);

    ctx.rotate(this.rotation);

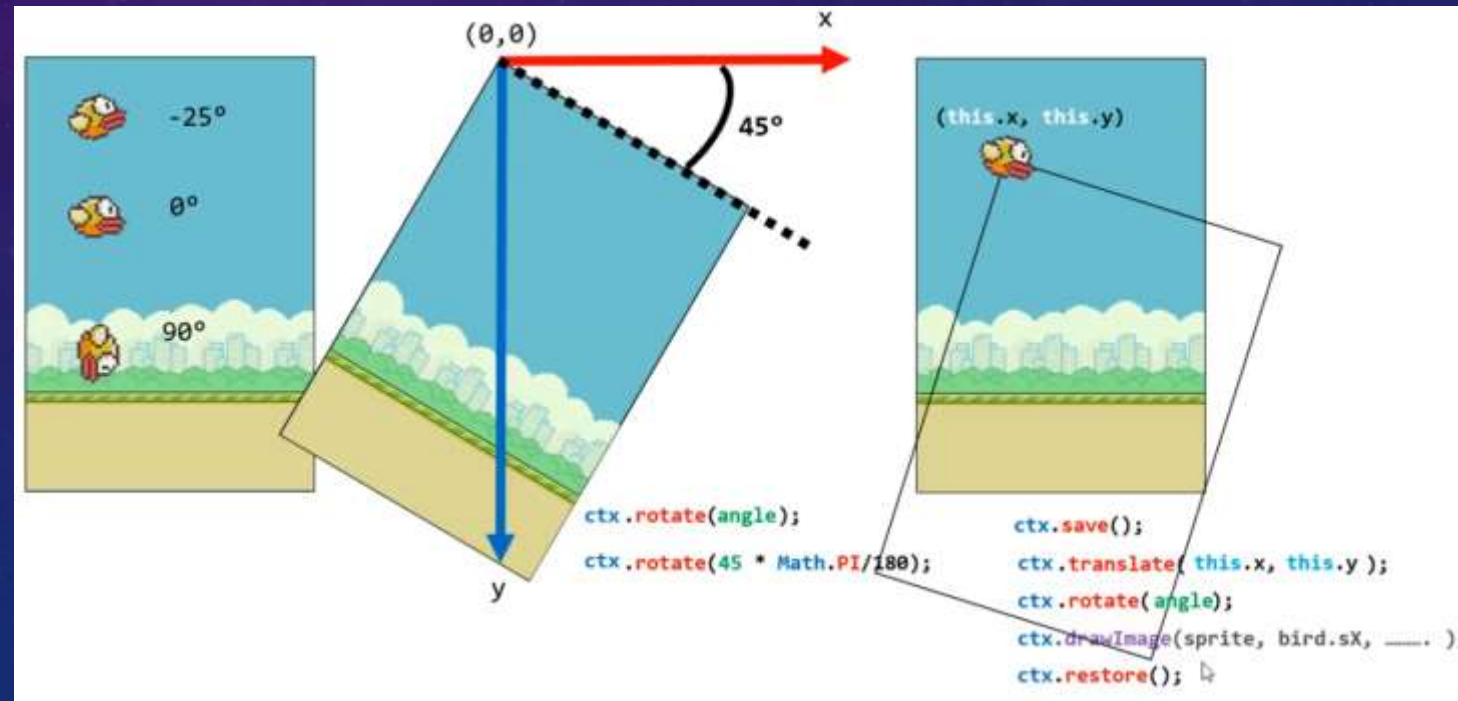
    ctx.drawImage(sprite, bird.sX, bird.sY, this.w,
      this.h, -this.w/2, -this.h/2,
      this.w, this.h);

    ctx.restore();
  },
}
```

$$180^\circ \longrightarrow \pi$$
$$45^\circ \longrightarrow x$$

$$x = 45^\circ * \frac{\pi}{180^\circ}$$

$$45 * \text{DEGREE}$$





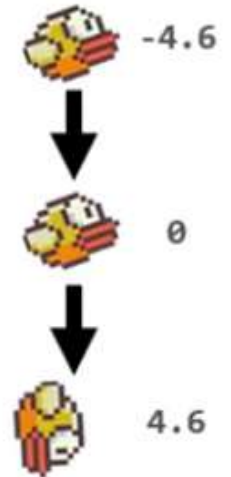
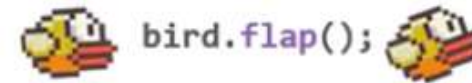
# Bird Update:

# Java Script Code

```
update : function(){
  this.frequency = state.current == state.getReady ? 10 : 5;
  this.frame += frames % this.frequency == 0 ? 1 : 0;
  this.frame = this.frame % this.animation.length;
  if(state.current === state.getReady){
    this.y = 150;
    this.rotation = 0 * DEGREE;
  }else{
    this.speed += this.gravity;
    this.y += this.speed;

    if(this.y + this.h/2 >= cvs.height-fg.h){
      this.y = cvs.height - fg.h- this.h/2;
      if(state.current == state.game){
        state.current = state.over;
      }
    }
    if(this.speed >= this.jump) {
      this.rotation = 90 * DEGREE;
    }else{
      this.rotation = - 25 * DEGREE;
    }
  }
}
```

`this.speed += this.gravity;`



`this.speed >= this.jump`

# Move ForGround:

## Java Script Code

```
const fg = {  
  sX : 276,  
  sY : 0,  
  w : 224,  
  h : 112,  
  x : 0,  
  dx : 2,  
  
  update : function(){  
    if(state.current == state.game){  
      this.x = (this.x - this.dx) % (this.w / 2);  
    }  
  }  
}
```

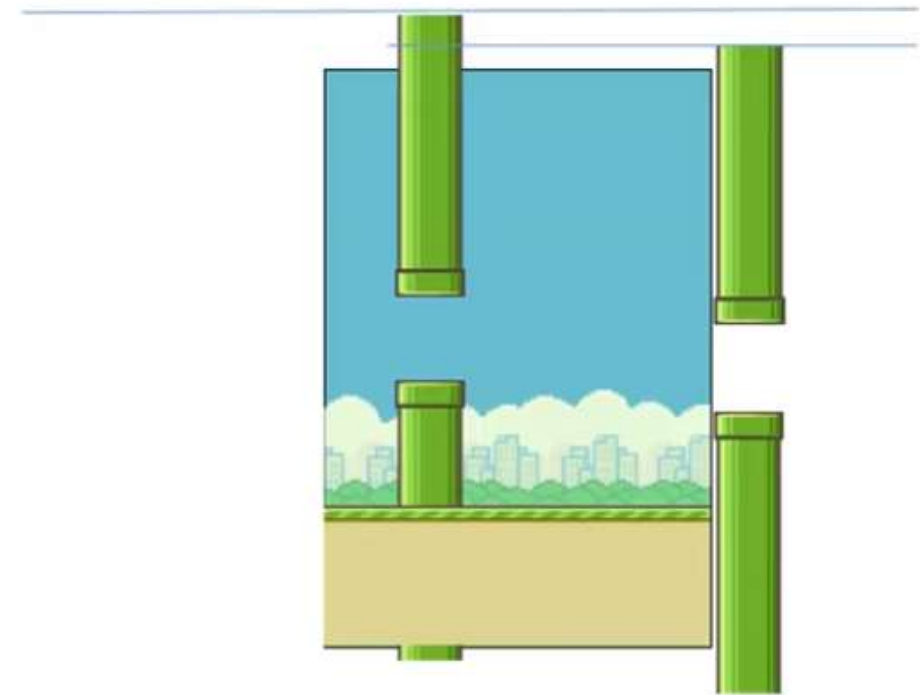
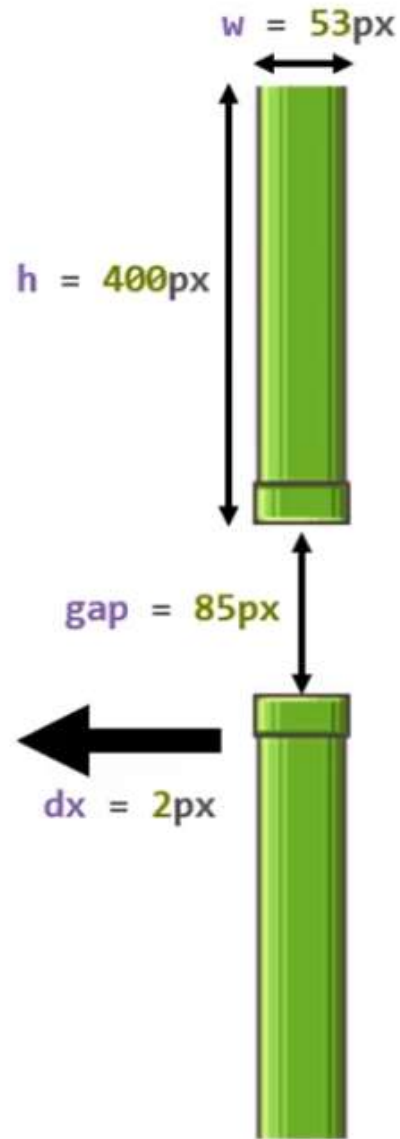


```
(this.x - this.dx) % (this.w / 2)  
( 0 - 2 ) % ( 224 / 2 )  
( 0 - 2 ) % 112 => -2 % 112 => -2  
( -2 - 2 ) % 112 => -4 % 112 => -4  
( -4 - 2 ) % 112 => -6 % 112 => -6  
( -6 - 2 ) % 112 => -8 % 112 => -8  
.  
.  
.  
(-100 - 2) % 112 => -110 % 112 => -110  
(-110 - 2) % 112 => -112 % 112 => 0
```

# Create The Pipe:

## Java Script Code

```
const pipes = {  
  bottom : {  
    sX : 502,  
    sY : 0  
  },  
  top : {  
    sX : 553,  
    sY : 0  
  },  
  w : 53,  
  h : 400,  
  gap : 85,  
  dx : 2,  
  position: [],  
  maxYPos : -150,  
  update: function(){}  
  draw: function(){}  
}
```

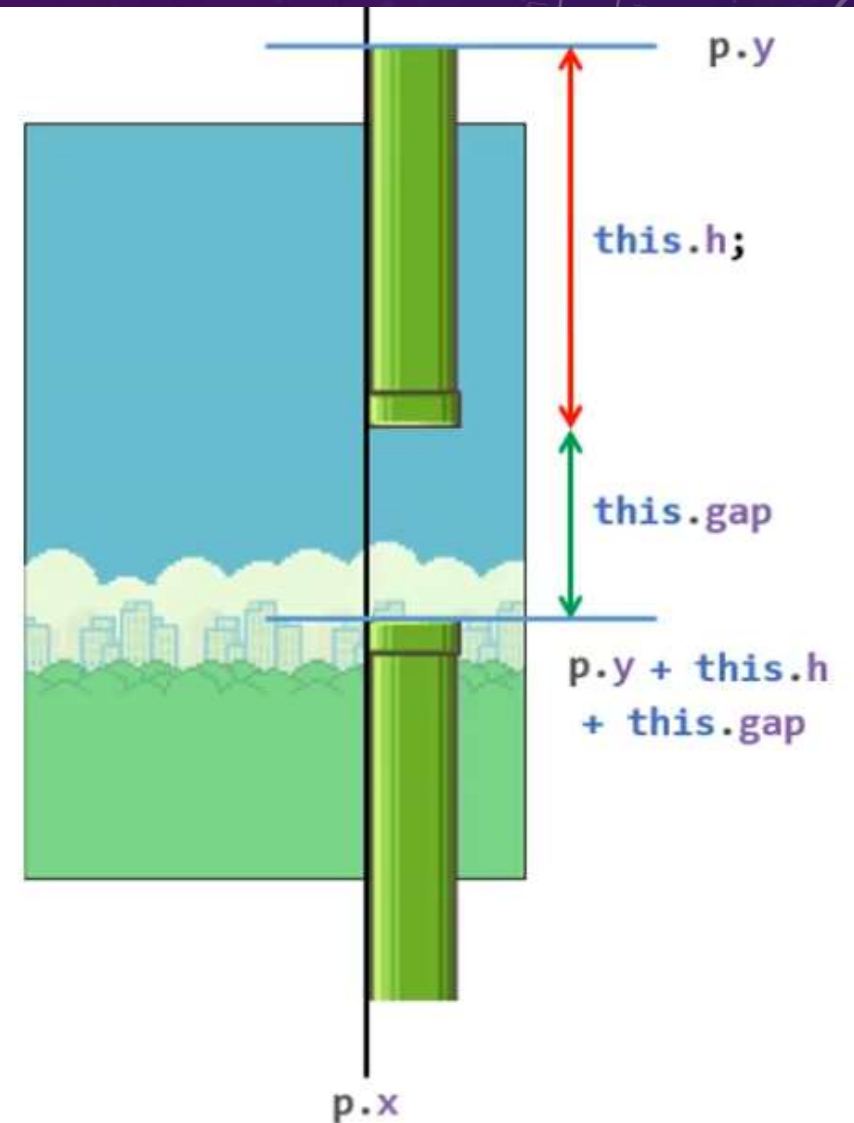


```
if(state.current !== state.game) return;  
if(frames%100 == 0){  
  this.position.push(  
    {  
      x : cvs.width,  
      y : this.maxYPos * ( Math.random() + 1 )  
    }  
  );  
  -150 * ( 0 + 1 ) = - 150  
  -150 * ( 1 + 1 ) = - 300  
}
```

# Draw The pipes :

# Java Script Code

```
draw: function(){  
  for(let i = 0; i < this.position.length; i++){  
    let p = this.position[i];  
    let topYPos = p.y;  
    let bottomYPos = p.y + this.gap + this.h;  
  
    ctx.drawImage(sprite, this.top.sX, this.top.sY,  
      this.w, this.h, p.x, topYPos, this.w, this.h);  
  
    ctx.drawImage(sprite, this.bottom.sX, this.bottom.sY,  
      this.w, this.h, p.x, bottomYPos, this.w, this.h);  
  }  
},
```





# move the pipes :

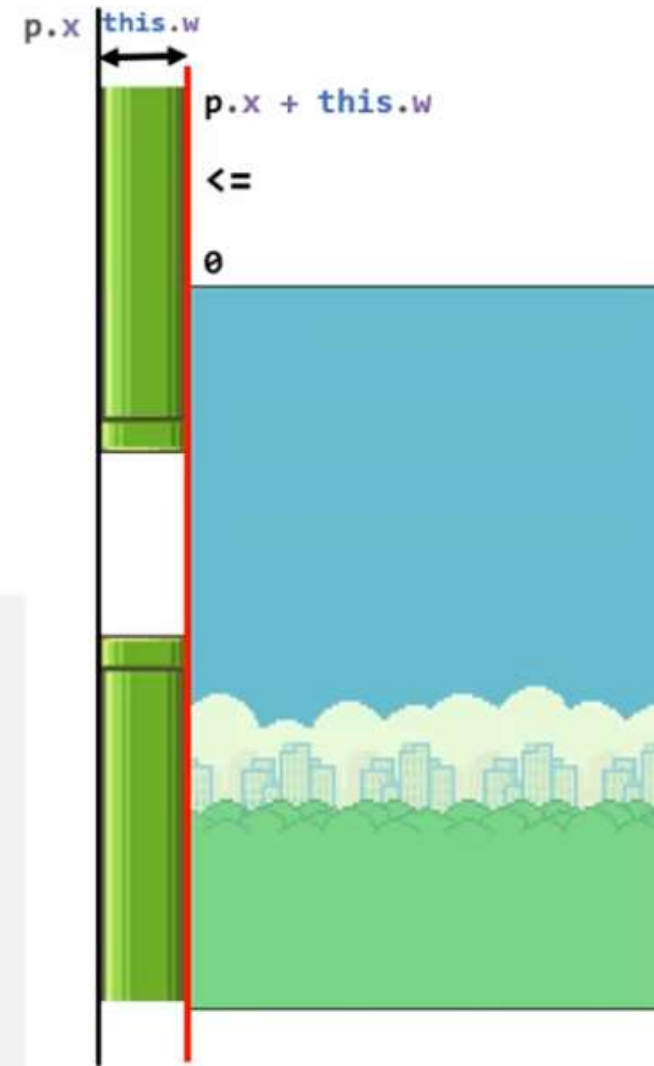
# Java Script Code

```
update: function(){
  if(frames%100 == 0){
    this.position.push(
      {
        x : cvs.width,
        y : this.maxYPos * ( Math.random() + 1 )
      }
    );
  }

  for(let i = 0; i < this.position.length; i++){
    let p = this.position[i];
    p.x -= this.dx;

    if(p.x + this.w <= 0){
      this.position.shift();
    }
  }
}
```

```
list = [0,1,2,3,4]
list.shift();
list = [1,2,3,4]
list.shift();
list = [2,3,4]
```



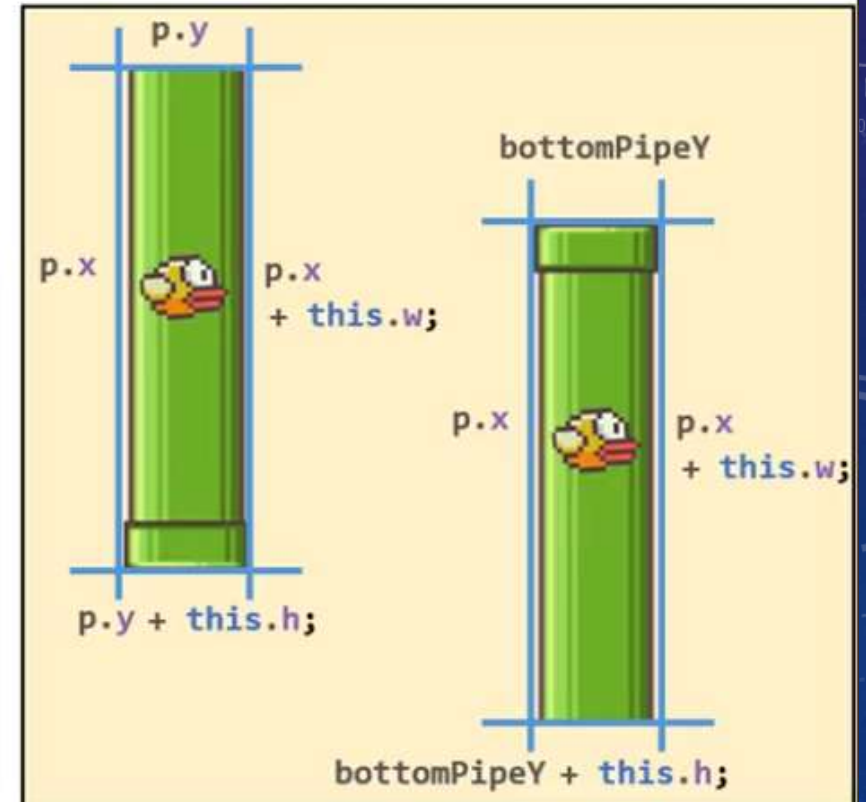
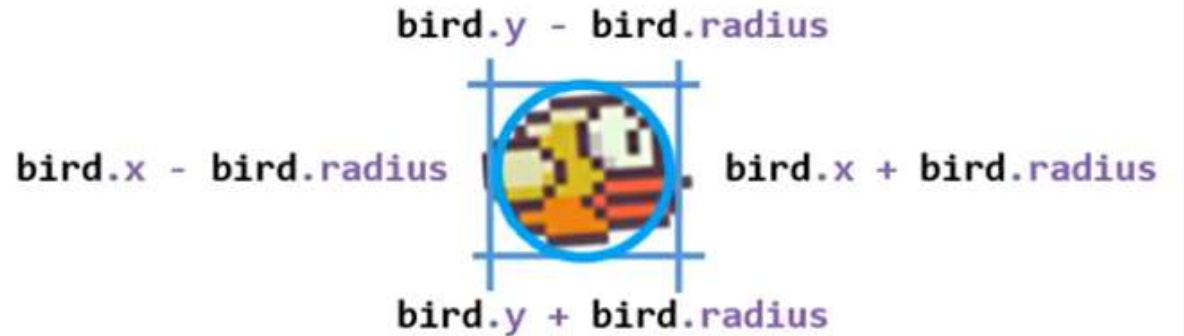
# Collision detection:

# Java Script Code

```
const bird = {  
  radius : 12,  
}
```



```
const pipes = {  
  update: function(){  
    for(let i = 0; i < this.position.length; i++){  
      let p = this.position[i];  
      let bottomPipeY = p.y + this.gap + this.h;  
      if(bird.x + bird.radius > p.x && bird.x - bird.radius < p.x + this.w  
        && bird.y + bird.radius > p.y && bird.y - bird.radius < p.y + this.h  
      ){  
        state.current = state.over;  
      }  
      if(bird.x + bird.radius > p.x && bird.x - bird.radius < p.x + this.w  
        && bird.y + bird.radius > bottomPipeY  
        && bird.y - bird.radius < bottomPipeY + this.h  
      ){  
        state.current = state.over;  
      }  
    }  
  }  
}
```



Next 

## SCORE:

## Java Script Code

```
const score= {  
  best : 0,      localStorage.getItem("best") || 0  
  value : 0,  
  draw : function(){  
    ctx.fillStyle = "#FFF";  
    ctx.strokeStyle = "#000";  
  
    if(state.current == state.game){  
      ctx.lineWidth = 2;  
      ctx.font = "35px Teko";  
      ctx.fillText(this.value, cvs.width/2, 50);  
      ctx.strokeText(this.value, cvs.width/2, 50);  
    }else if(state.current == state.over){  
      ctx.font = "25px Teko";  
      ctx.fillText(this.value, 225, 186);  
      ctx.strokeText(this.value, 225, 186);  
      ctx.fillText(this.best, 225, 228);  
      ctx.strokeText(this.best, 225, 228);  
    }  
  }  
}
```



```
score.best = Math.max( score.value, score.best );  
  
localStorage.setItem("best", score.best)  
  
localStorage.getItem("best")
```



## Handle Click Event:

# Java Script Code

```
canvas.addEventListener("click", function(evt){
  switch (state.current){
    case state.getReady:
      state.current = state.game;
      break;

    case state.game:
      bird.flap();
      break;

    case state.over:
      let rect = canvas.getBoundingClientRect();
      let clickX = evt.clientX - rect.left;
      let clickY = evt.clientY - rect.top;
      if(clickX >= startBtn.x && clickX <= startBtn.x + startBtn.w
        && clickY >= startBtn.y && clickY <= startBtn.y + startBtn.h
      ){
        bird.speedReset();
        pipes.reset();
        score.reset();
        state.current = state.getReady;
      }
      break;
  }
});
```



```
const startBtn = {
  x : 120,
  y : 263,
  w : 83,
  h : 29
}
```

```
pipes.reset();      pipes.position = [];

bird.speedReset();   bird.speed = 0;

score.reset();       score.value = 0;
```

Next 