**Public variables:**

var gameRun = true / false;

var shipXpos = number;

var shipYpos = number;

var shipXspeed = number;

var shipYspeed = number;

var shipColor = colour;

const SHIP\_WIDTH = number;

const SHIP\_HEIGHT = number;

const LEFT\_KEY = number;

const RIGHT\_KEY = number;

const UP\_KEY = number;

const DOWN\_KEY = number;

const SPACE\_KEY = number;

var healthWidth = number;

const HEALTH\_HEIGHT = number;

var leftKeyPressed = true / false;

var rightKeyPressed = true / false;

var upKeyPressed = true / false;

var downKeyPressed = true / false;

var spaceKeyPressed = true / false;

var enemies = [ ];

var enemyTotal = number;

var setUp = true;

var bullets = [ ];

var difficulty = ' ';

var playerImg = new Image();

playerImg.src = image source;

var enemyImg = new Image();

enemyImg.src = image source;

var player = new Ship(playerImg, shipXpos, shipYpos, SHIP\_WIDTH, SHIP\_HEIGHT, shipColor, shipXspeed, shipYspeed);

**Private variables:**

var eWidth = Math.floor(Math.random() \* (Number - Number) + Number);

var eHeight = Math.floor(Math.random() \* (Number - Number) + Number);

var eXpos = Math.floor(Math.random() \* ((canvas.width - eWidth) - Number) + Number);

var eYpos = Number;

var eXspeed = Math.floor(Math.random() \* (Number - Number) + Number);

var eYspeed = Math.floor(Math.random() \* (Number - Number) + Number);

const BULLET\_WIDTH = Number;

const BULLET\_HEIGHT = Number;

var bulletXpos = player.x + player.w / Number - BULLET\_WIDTH / Number;

var bulletYpos = player.y - BULLET\_HEIGHT;

var bulletYspeed = Number;

var color = colour;

var self = this;

var collided = true / false;

**Pseudocode:**

Mainloop {

if game is running {

Color rect game canvas

Color rect health bar

Color text heatlh

While set up is true {

While difficulty is not equal to easy or hard {

Ask, “What difficulty do you want”

Set difficulty input to lowercase

}

Run Setting up function

Put Set up back to false

}

If there is more than 0 enemy {

Enemies (array) .forEach(function(enemy, index) {

Draw enemy img func

Enemy move func

})

}

If there is more than 0 bullets {

Bullets (array) .forEach(function(bullet, index) {

Draw bullet func

Bullet move func

If the bullet is out of bounds or has hit an enemy {

Delete bullets [index]

}

})

Fully clear bullets from array instead of leaving them as an undefined placeholder

}

If there are no enemies remaining {

Game run = false

}

Draw player ship img func

Player ship movement func

Player collision func

}

else {

Color rect black for lose screen

If there is less than 1 enemies remaining {

Color text, “You win, restart game”

}

else {

Color text, “you lose, restart game”

}

}

} End of Main loop

func Key pressed event {

if key code == left key {

Left Key Pressed = true

}

if key code == right key {

Right Key Pressed = true

}

if key code == Up key {

Up Key Pressed = true

}

if key code == Down key{

Down Key Pressed = true

}

if key code == Space key {

Space Key Pressed = true

}

}

function Key pressed event {

if key code == Left key) {

Left Key Pressed= false

}

if key code == Right key) {

Right Key Pressed= false

}

if key code == Up key) {

Up Key Pressed= false

}

if key code == Down key) {

Down Key Pressed= false

}

if key code == Space key) {

Space Key Pressed= false

Make bullet function

}

function Make enemies {

var enemy Width = Math.floor (Math.random() \* (Max - Min) + Min)

var enemy Height = Math.floor(Math.random() \* (Max - Min) + Min)

var enemy X position = Math.floor ( Math.random() \* ((Canvas width - Width) - 0) + 0)

var enemy Y position = 0

var enemy X speed = Math.floor ( Math.random () \* (Max - Min) + Min)

var enemy Y speed = Math.floor ( Math.random () \* (Max - Min) + Min)

var e = new Enemy ( Image , enemy X position, enemy Y position, enemy Width, enemy Height, enemy X speed, enemy Yspeed)

Enemies (array) .push (e)

if the difficulty is equal to easy) {

Set the enemy total to 4

}

if the difficulty is equal to hard{

Set the enemy total to 8

}

}

Class Enemy {

Constructor (image src, x, y, width, height, xSpeed, ySpeed) {

this.src = src

this.x = x

this.y = y

this.w = w

this.h = h

this.ySpeed = ySpeed

this.xSpeed = xSpeed

}

Enemy move func {

this.y += this.ySpeed;

this.x += this.xSpeed;

if this.x is greater than the canvas width - this.w or this.x is less than 0 {

this.xSpeed \*= -1

}

if this.y is greater than the canvas height {

this.y = 0 - this.h

Health bar width – number;

if health bar width is less than or equal to 0 {

Game Run = false

}

}

}

Draw enemy image {

Canvas Context. Draw Image (this.src, this.x, this.y, this.w, this.h)

}

}

Class Bullet {

Constructor (x, y, width, height, ySpeed, colour) {

this.x = x

this.y = y

this.w = w

this.h = h

this.ySpeed = ySpeed

this.c = c

}

Draw bullet func {

Canvas Context. fillStyle = this.c

Canvas Context. fillRect (this.x, this.y, this.w, this.h)

}

Bullet move func {

this.y -= this.ySpeed;

}

Out of bounds func {

Return, this.y - this.h is less than 0 or this.x - this.w is less than 0 or this.x is greater than canvas width or this.y is greater than canvas.height;

}

Has hit item (Item) {

return this.x + this.w >= item.x and

this.x <= item.x + item.w and

this.y >= item.y and

this.y <= item.y + item.h

}

Has hit (enemy) {

return This has hit item (enemy)

}

Has collided func {

var self = this

var collided = false

Enemies. For each func (enemy, i) {

if (self. Has hit enemy (enemy)) {

Delete enemies[ i ]

Collided = true

}

})

Fully clear bullets from array instead of leaving them as an undefined placeholder

Return collided

}

}

Class Ship {

Constructor (image src, x, y, width, height, colour, xSpeed, ySpeed) {

this.x = x;

this.y = y;

this.w = w;

this.h = h;

this.c = c;

this.ySpeed = ySpeed;

this.xSpeed = xSpeed;

this.src = src;

}

Draw Ship func {

Canvas Context .fill Style = this.c

Canvas Context. Fill Rect (this.x, this.y, this.w, this.h)

}

Draw ship image {

Canvas Context. Draw Image (this.src, this.x, this.y, this.w, this.h)

}

Ship Movement func {

if (left key pressed and this.x is greater than 0) {

this.x -= this.xSpeed;

}

if (right key pressed and this.x is less than canvas width - this.w) {

this.x += this.xSpeed;

}

if (up key pressed and this.y is greater than 0) {

this.y -= this.ySpeed;

}

if (down key pressed and this.y is less than canvas height - this.h) {

this.y += this.ySpeed;

}

}