**index.html**

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<title>Araw lilim</title>

<script type="text/javascript" src="js/phaser.min.js"></script>

<script type="text/javascript" src="js/boot.js"></script>

<script type="text/javascript" src="js/preload.js"></script>

<script type="text/javascript" src="js/menu.js"></script>

<script type="text/javascript" src="js/play.js"></script>

<script type="text/javascript" src="js/game.js"></script>

<style type="text/css">

body {

margin: 0;

}

</style>

</head>

<body>

</body>

</html>

**boot.js**

var w = 800, h = 600;

var game = new Phaser.Game(w, h, Phaser.CANVAS, '');

var player, enemy1, enemy2, enemy3, keyboard, startButton, restartButton, playing;

var button, platform1,platform2,platform3,platform4,platform5;

var bounds = 10000;

var btn, score;

var bestText, timeText, scoreText;

var gameOverText, bestScoreText;

var diamond, shineDiamond, shineDiamonds;

game.state.add("bootGame", bootGame);

game.state.add("preloadGame", preloadGame);

game.state.add("menuGame", menuGame);

game.state.add("playGame", playGame);

game.state.start("bootGame");

**preload.js**

preloadGame = {

preload:function() {

game.load.image("bg","img/bg.png");

game.load.image("Menu","img/menu.png");

game.load.image("diamond","img/diamond.png");

game.load.spritesheet('buttonplay', 'img/play-btn.png', 120, 50);

game.load.spritesheet("player","img/player.png",30,30);

game.load.spritesheet("enemy1","img/enemy1.png",40,40);

game.load.spritesheet("enemy2","img/enemy2.png",40,40);

game.load.spritesheet("enemy3","img/enemy3.png",40,40);

game.load.image("platform1","img/plat1.png");

game.load.image("platform2","img/plat2.png");

game.load.image("platform3","img/plat3.png");

game.load.image("platform4","img/plat4.png");

game.load.image("platform5","img/plat5.png");

game.load.image("buttonleft","img/btn-left.png",50,50);

game.load.image("buttonright","img/btn-right.png",50,50);

game.load.image("buttonup","img/btn-up.png",50,50);

game.load.image("buttondown","img/btn-down.png",50,50);

game.load.image('About1','img/about1.png');

game.load.image('About2','img/about2.png');

game.load.image('Ins1','img/instruction1.png');

game.load.image('Ins2','img/instruction2.png');

game.load.spritesheet("menu2","img/backtomenu.png");

game.load.image('Go','img/go.png');

game.load.image('ArawLilim','img/arawlilim.png');

game.load.image("restart","img/restart.png");

game.load.audio('bgMusic1', 'audio/menubgmusic.mp3');

game.load.image('pause','img/pause.png');

},

create:function(){

game.state.start("playGame");

game.state.start("menuGame");

},

}

**boot.js**

bootGame = {

create:function() {

game.physics.startSystem(Phaser.Physics.ARCADE);

game.stage.backgroundColor = "#ccc";

keyboard = game.input.keyboard.createCursorKeys();

game.state.start("preloadGame");

}

}

**menu.js**

menuGame = {

create:function(){

menumusic = game.add.audio("bgMusic1");

menumusic.loop = true;

menumusic.play();

game.scale.scaleMode = Phaser.ScaleManager.EXACT\_FIT;

game.scale.forceLandscape = true;

game.scale.pageAlignHorizontally = true;

game.scale.pageAlignVertically = true;

game.scale.setScreenSize =true;

bg = game.add.image(1,-90,"Menu");

bg.scale.y = 2;

start = game.add.image(205,120,"ArawLilim");

start.scale.y = 1.6;

start.scale.x = 1.4;

startButton = game.add.button(game.width/2.3,game.height/1.9, "buttonplay",this.buttonPlay);

aboutText = game.add.button(310,430,"About1",this.about);

aboutText.anchor.set(0.6);

aboutText.scale.set(1);

instruc = game.add.button(500,430,"Ins1",this.ins);

instruc.anchor.set(0.6);

instruc.scale.set(1);

},

about: function(){

about=game.add.image(0,0,"About2");

about.scale.set(3.2);

restartButton=game.add.button(350,30,"menu2",restartB,this);

function restartB() {

restartButton.destroy();

game.state.start("menuGame");

}

},

ins: function(){

about=game.add.image(0,0,"Ins2");

about.scale.set(3.2);

restartButton=game.add.button(350,550,"menu2",restartB,this);

function restartB() {

restartButton.destroy();

game.state.start("menuGame");

}

},

update:function(){

// if(keyboard.up.isDown){

// game.state.start("playGame");

},

buttonPlay:function(){

game.state.start("playGame");

menumusic.stop();

},

}

**play.js**

playGame = {

create:function(){

game.scale.scaleMode = Phaser.ScaleManager.EXACT\_FIT;

game.scale.forceLandscape = true;

game.scale.pageAlignHorizontally = true;

game.scale.pageAlignVertically = true;

game.scale.setScreenSize =true;

platform1 = game.add.sprite(100,100,"platform1");

platform2 = game.add.sprite(100,450,"platform2");

platform3 = game.add.sprite(600,100,"platform3");

platform4 = game.add.sprite(350,280,"platform4");

platform5 = game.add.sprite(600,450,"platform5");

platform1.scale.x = 1;

platform2.scale.x = 1;

platform3.scale.x = 1;

platform4.scale.x = 1;

platform5.scale.x = 1;

button = game.add.button(330,450,'buttonleft',pushRight);

button = game.add.button(420,453,'buttonright',pushLeft);

button = game.add.button(375,410,'buttonup',pushUp);

button = game.add.button(373,495,'buttondown',pushDown);

player = game.add.sprite(350,550,"player");

enemy1 = game.add.sprite(100,200,"enemy1");

enemy2 = game.add.sprite(500,200,"enemy2");

enemy3 = game.add.sprite(300,350,"enemy3");

game.physics.arcade.enable(player);

game.physics.arcade.enable(enemy1);

game.physics.arcade.enable(enemy2);

game.physics.arcade.enable(enemy3);

game.physics.arcade.enable(platform1);

game.physics.arcade.enable(platform2);

game.physics.arcade.enable(platform3);

game.physics.arcade.enable(platform4);

game.physics.arcade.enable(platform5);

platform1.body.immovable = true;

platform1.collideWorldBoundsWorldBounds = true;

platform2.body.immovable = true;

platform2.collideWorldBoundsWorldBounds = true;

platform3.body.immovable = true;

platform3.collideWorldBoundsWorldBounds = true;

platform4.body.immovable = true;

platform4.collideWorldBoundsWorldBounds = true;

platform5.body.immovable = true;

platform5.collideWorldBoundsWorldBounds = true;

game.physics.enable(enemy1, enemy2, enemy3, Phaser.Physics.ARCADE);

player.body.collideWorldBounds = true;

enemy1.body.collideWorldBounds = true;

enemy2.body.collideWorldBounds = true;

enemy3.body.collideWorldBounds = true;

enemy1.body.bounce.set(1);

enemy2.body.bounce.set(1);

enemy3.body.bounce.set(1);

enemy1.checkWorldBounds = true;

enemy2.checkWorldBounds = true;

enemy3.checkWorldBounds = true;

enemy1.events.onOutOfBounds.add(theEnemy1Drop, this);

enemy2.events.onOutOfBounds.add(theEnemy1Drop, this);

enemy3.events.onOutOfBounds.add(theEnemy1Drop, this);

shineDiamond = game.add.group();

shineDiamond.enableBody = true;

createDiamonds(300);

menumusic = game.add.audio('menumusic');

scoreText = game.add.text(w-150,10,"Score: 0",{fill:"black"});

bestText = game.add.text(w-150,40,"Best: "+getScore() ,{fill:"black"});

gameOverText = game.add.text((w/2)-100,10,"");

startButton = game.add.button(390,300, 'Go', buttonplay);

startButton.anchor.set(0.5);

},

update:function(){

game.physics.arcade.collide(enemy1, platform1, enemy1Hitplatform1);

game.physics.arcade.collide(enemy1, platform2, enemy1Hitplatform2);

game.physics.arcade.collide(enemy1, platform3, enemy1Hitplatform3);

game.physics.arcade.collide(enemy1, platform4, enemy1Hitplatform4);

game.physics.arcade.collide(enemy1, platform5, enemy1Hitplatform5);

game.physics.arcade.collide(enemy2, platform1, enemy2Hitplatform1);

game.physics.arcade.collide(enemy2, platform2, enemy2Hitplatform2);

game.physics.arcade.collide(enemy2, platform3, enemy2Hitplatform3);

game.physics.arcade.collide(enemy2, platform4, enemy2Hitplatform4);

game.physics.arcade.collide(enemy2, platform5, enemy2Hitplatform5);

game.physics.arcade.collide(enemy3, platform1, enemy3Hitplatform1);

game.physics.arcade.collide(enemy3, platform2, enemy3Hitplatform2);

game.physics.arcade.collide(enemy3, platform3, enemy3Hitplatform3);

game.physics.arcade.collide(enemy3, platform4, enemy3Hitplatform4);

game.physics.arcade.collide(enemy3, platform5, enemy3Hitplatform5);

game.physics.arcade.overlap(player,enemy1, collectEnemy);

game.physics.arcade.overlap(player,enemy2, collectEnemy);

game.physics.arcade.overlap(player,enemy3, collectEnemy);

game.physics.arcade.overlap(player,shineDiamond,collectDiamonds);

}

},

function createDiamonds(time){

setInterval(function(){

diamonds = diamond.create(Math.random()\*w,-100,"diamond");

diamonds.body.gravity.y = 1000;

var scale = Math.random();

diamonds.scale.y = scale\*2;

diamonds.scale.x = scale\*2;

},time);

}

function pushRight(){

player.animations.play('walk-right');

player.animations.stop();

player.body.velocity.x = -150;

}

function pushLeft() {

player.body.velocity.x = 150;

player.animations.play('walk-left');

}

function pushUp() {

player.body.velocity.y = -150;

player.animations.play('walk-up');

player.animations.stop();

}

function pushDown() {

player.body.velocity.y = 150;

player.animations.play('walk-down');

}

var score = 0;

function collectDiamonds(player,diamond){

score = score + 5;

diamond.kill();

if(getScore()<=score){

saveScore(score);

bestText.text = "Best: "+score;

console.log("x");

}

else{

console.log("x");

}

scoreText.text = "Score: "+score;

}

function createDiamonds(time){

setInterval(function(){

var shineDiamonds = shineDiamond.create(Math.random()\*bounds,-1,'diamond');

shineDiamonds.body.gravity.y = 100;

shineDiamonds.body.bounce.y = 0.1;

},time)

}

function saveScore(Score){

localStorage.setItem("gameScore",Score);

}

function getScore(){

return (localStorage.getItem("gameScore") == null || localStorage.getItem("gameScore") == "")?0:localStorage.getItem("gameScore");

}

function audioLoop(time){

setInterval(function(){

bg.play();

}, time);

}

function collectEnemy(player,enemy1,enemy2,enemy3){

enemy1.kill();

game.\_paused = true;

gameOverText.text = "Game Over.\nHi: "+getScore()+"\nScores: "+score;

restartButton = game.add.button(350,280,"restart",restartB);

function restartB(){

window.location.href=window.location.href;

}

}

// function collectEnemy(player,enemy2){

// enemy2.kill();

// game.\_paused = true;

// gameOverText.text = "Game Over.\nHi: "+getScore()+"\nScores: "+score;

// restartButton = game.add.button(350,280,"restart",restartB);

// function restartB(){

// window.location.href=window.location.href;

// }

// }

// function collectEnemy(player,enemy3){

// enemy3.kill();

// game.\_paused = true;

// gameOverText.text = "Game Over.\nHi: "+getScore()+"\nScores: "+score;

// restartButton = game.add.button(350,280,"restart",restartB);

// function restartB(){

// window.location.href=window.location.href;

// }

// }

function enemy1Hitplatform1(enemy1, platform1) {

enemy1.animations.play('enemy1');

enemy1.body.velocity.x = -1\*5\*(platform1.x-enemy1.x);

}

function enemy1Hitplatform2(enemy1, platform2) {

enemy1.animations.play('enemy1');

enemy1.body.velocity.x = -1\*5\*(platform2.x-enemy1.x);

}

function enemy1Hitplatform3(enemy1, platform3) {

enemy1.animations.play('enemy1');

enemy1.body.velocity.x = -1\*5\*(platform3.x-enemy1.x);

}

function enemy1Hitplatform4(enemy1, platform4) {

enemy1.animations.play('enemy1');

enemy1.body.velocity.x = -1\*5\*(platform4.x-enemy1.x);

}

function enemy1Hitplatform5(enemy1, platform5) {

enemy1.animations.play('enemy1');

enemy1.body.velocity.x = -1\*5\*(platform5.x-enemy1.x);

}

function enemy2Hitplatform1(enemy2, platform1) {

enemy2.animations.play('enemy2');

enemy2.body.velocity.x = -1\*5\*(platform1.x-enemy2.x);

}

function enemy2Hitplatform2(enemy2, platform2) {

enemy2.animations.play('enemy2');

enemy2.body.velocity.x = -1\*5\*(platform2.x-enemy2.x);

}

function enemy2Hitplatform3(enemy2, platform3) {

enemy2.animations.play('enemy2');

enemy2.body.velocity.x = -1\*5\*(platform3.x-enemy2.x);

}

function enemy2Hitplatform4(enemy2, platform4) {

enemy2.animations.play('enemy2');

enemy2.body.velocity.x = -1\*5\*(platform4.x-enemy2.x);

}

function enemy2Hitplatform5(enemy2, platform5) {

enemy2.animations.play('enemy2');

enemy2.body.velocity.x = -1\*5\*(platform5.x-enemy2.x);

}

function enemy3Hitplatform1(enemy3, platform1) {

enemy3.animations.play('enemy3');

enemy3.body.velocity.x = -1\*5\*(platform1.x-enemy3.x);

}

function enemy3Hitplatform2(enemy3, platform2) {

enemy3.animations.play('enemy3');

enemy3.body.velocity.x = -1\*5\*(platform2.x-enemy3.x);

}

function enemy3Hitplatform3(enemy3, platform3) {

enemy3.animations.play('enemy3');

enemy3.body.velocity.x = -1\*5\*(platform3.x-enemy3.x);

}

function enemy3Hitplatform4(enemy3, platform4) {

enemy3.animations.play('enemy3');

enemy3.body.velocity.x = -1\*5\*(platform4.x-enemy3.x);

}

function enemy3Hitplatform5(enemy3, platform5) {

enemy3.animations.play('enemy3');

enemy3.body.velocity.x = -1\*5\*(platform5.x-enemy3.x);

}

function buttonplay() {

startButton.destroy();

enemy1.body.velocity.set(150, -150);

enemy2.body.velocity.set(150, -150);

enemy3.body.velocity.set(150, -150);

playing = true;

}

function theEnemy1Drop() {

lives--;

if(lives) {

livesText.setText('Lives: '+lives);

lifeLostText.visible = true;

enemy1.reset(game.world.width\*0.5, game.world.height-25);

platform1.reset(game.world.width\*0.5, game.world.height-5);

platform2.reset(game.world.width\*0.5, game.world.height-5);

platform3.reset(game.world.width\*0.5, game.world.height-5);

platform4.reset(game.world.width\*0.5, game.world.height-5);

platform5.reset(game.world.width\*0.5, game.world.height-5);

game.input.onDown.addOnce(function(){

lifeLostText.visible = false;

enemy1.body.velocity.set(150, -150);

}, this);

}

}