**AGAWANG SULOK**

-Ancheta, Ma. Frecelyn M.

- Caasi, Jeah

**Index.html**

<!DOCTYPE HTML>

<html>

<head>

<title>Agawan</title>

<style type="text/css">

body{

padding:0px;

margin:0px;

}

</style>

<script src="js/phaser.min.js"></script>

<script src="js/Boot.js"></script>

<script src="js/Preloader.js"></script>

<script src="js/gameplay.js"></script>

</head>

<body>

<div id = "game"></div>

<div id = "orientation"></div>

<script type="text/javascript">

window.onload = function(){

var game = new Phaser.Game(820, 700, Phaser.AUTO, 'Agawan');

var width = 820;

var height = 700;

game.state.add('Boot',Agawan.Boot);

game.state.add('Preloader',Agawan.Preloader);

game.state.add('gameplay',Agawan.gameplay);

game.state.start('Boot');

};

</script>

</body>

</html>

**Gameplay.js**

Agawan.gameplay = function(game){

};

var left=false;

var right=false;

var up= false;

var down=false;

var speed = 50;

var playerSpeed = 200;

Agawan.gameplay.prototype = {

create: function(game){

this.stage.backgroundColor = '6B8E23';

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

this.spot = this.add.physicsGroup();

spot1 = new SpotPlatform(this.game,game.world.width-30, game.world.height-670, 'spots', this.spot);

spot1.body.setSize(25,25);

spot2 = new SpotPlatform(this.game,game.world.width-30, game.world.height-350, 'spots', this.spot);

spot2.body.setSize(25,25);

spot3 = new SpotPlatform(this.game,game.world.width-30, game.world.height-70, 'spots', this.spot);

spot3.body.setSize(25,25);

spot4 = new SpotPlatform(this.game,game.world.width-790, game.world.height-670, 'spots', this.spot);

spot4.body.setSize(25,25);

spot5 = new SpotPlatform(this.game,game.world.width-790, game.world.height-350, 'spots', this.spot);

spot5.body.setSize(25,25);

spot6 = new SpotPlatform(this.game,game.world.width-790, game.world.height-70, 'spots', this.spot);

spot6.body.setSize(25,25);

spot7 = new SpotPlatform(this.game,game.world.centerX, game.world.height-670, 'spots', this.spot);

spot7.body.setSize(25,25);

spot8 = new SpotPlatform(this.game,game.world.centerX, game.world.height-70, 'spots', this.spot);

spot8.body.setSize(25,25);

this.enemy = this.add.physicsGroup();

enemy1 = this.enemy.create(game.world.width-70, game.world.height-670, 'enemy',this.enemy);

game.physics.arcade.enable(enemy1);

enemy1.body.fixedRotation = true;

enemy1.body.setSize(27,40);

enemy2 = this.enemy.create(game.world.width-70, game.world.height-350, 'enemy',this.enemy);

game.physics.arcade.enable(enemy2);

enemy2.body.fixedRotation = true;

enemy2.body.setSize(27,40);

enemy3 = this.enemy.create(game.world.width-70, game.world.height-70, 'enemy',this.enemy);

game.physics.arcade.enable(enemy3);

enemy3.body.fixedRotation = true;

enemy3.body.setSize(27,40);

enemy4 = this.enemy.create(game.world.width-760, game.world.height-670, 'enemy',this.enemy);

game.physics.arcade.enable(enemy4);

enemy4.body.fixedRotation = true;

enemy4.body.setSize(27,40);

enemy5 = this.enemy.create(game.world.width-760, game.world.height-350, 'enemy',this.enemy);

game.physics.arcade.enable(enemy5);

enemy5.body.fixedRotation = true;

enemy5.body.setSize(27,40);

enemy6 = this.enemy.create(game.world.width-760, game.world.height-70, 'enemy',this.enemy);

game.physics.arcade.enable(enemy6);

enemy6.body.fixedRotation = true;

enemy6.body.setSize(27,40);

enemy7 = this.enemy.create(game.world.centerX, game.world.height-640, 'enemy',this.enemy);

game.physics.arcade.enable(enemy7);

enemy7.body.fixedRotation = true;

enemy7.body.setSize(27,40);

enemy8 = this.enemy.create(game.world.centerX, game.world.height-70, 'enemy',this.enemy);

game.physics.arcade.enable(enemy8);

enemy8.body.fixedRotation = true;

enemy8.body.setSize(27,40);

player = game.add.sprite(game.world.centerX, game.world.centerY, 'girl');

game.physics.arcade.enable(player);

player.body.allowGravity = false;

player.body.isCircle = false; // collision circle

player.body.setSize (40,64,0,0);

player.body.fixedRotation=true; // do not rotate on collision

player.body.collideWorldBounds = true;

game.camera.follow(player);

buttonup = game.add.button(game.world.width- 750, game.world.height-200, 'buttonup', null, this, 1, 0, 1, 0);

buttonup.fixedToCamera = true;

buttonup.alpha = 0.5;

buttonup.events.onInputOver.add(function(){up=true;});

buttonup.events.onInputOut.add(function(){up=false;});

buttonup.events.onInputDown.add(function(){up=true;});

buttonup.events.onInputUp.add(function(){up=false;});

buttondown = game.add.button(game.world.width-750, game.world.height-100, 'buttondown', null, this, 0, 1, 0, 1);

buttondown.fixedToCamera = true;

buttondown.alpha = 0.5

buttondown.events.onInputOver.add(function(){down=true;});

buttondown.events.onInputOut.add(function(){down=false;});

buttondown.events.onInputDown.add(function(){down=true;});

buttondown.events.onInputUp.add(function(){down=false;});

buttonright = game.add.button(game.world.width-150, game.world.height-138, 'buttonright', null, this, 1, 0, 1, 0);

buttonright.fixedToCamera = true;

buttonright.alpha = 0.5;

buttonright.events.onInputOver.add(function(){right=true;});

buttonright.events.onInputOut.add(function(){right=false;});

buttonright.events.onInputDown.add(function(){right=true;});

buttonright.events.onInputUp.add(function(){right=false;});

buttonleft = game.add.button(game.world.width-250, game.world.height-138, 'buttonleft', null, this, 0, 1, 0, 1);

buttonleft.fixedToCamera = true;

buttonleft.alpha = 0.5;

buttonleft.events.onInputOver.add(function(){left=true;});

buttonleft.events.onInputOut.add(function(){left=false;});

buttonleft.events.onInputDown.add(function(){left=true;});

buttonleft.events.onInputUp.add(function(){left=false;});

},

die: function(){

player.kill();

this.game.paused = true;

// add proper informational text

this.\_fontStyle = { font: "80px Arial", fill: "#0000", stroke: "#600", strokeThickness: 5, align: "center" };

var pausedText = this.add.text(220, 250, " Game Over \nTap to retry", this.\_fontStyle);

// set event listener for the user's click/tap the screen

this.input.onDown.add(function(){

// remove the pause text

pausedText.destroy();

this.state.restart();

// unpause the game

this.game.paused = false;

}, this);

},

win: function(){

this.game.paused = true;

// add proper informational text

this.\_fontStyle = { font: "80px Arial", fill: "#0000", stroke: "#600", strokeThickness: 5, align: "center" };

var pausedText = this.add.text(240, 250, " You win \nTap to retry", this.\_fontStyle);

// set event listener for the user's click/tap the screen

this.input.onDown.add(function(){

// remove the pause text

pausedText.destroy();

this.state.restart();

// unpause the game

this.game.paused = false;

}, this);

},

follow1: function(){

if (player.body.x < enemy1.body.x)

{

enemy1.body.velocity.x = speed \* -3;

}

else

{

enemy1.body.velocity.x = speed;

}

if (player.body.y < enemy1.body.y)

{

enemy1.body.velocity.y = speed \* -3;

}

else

{

enemy1.body.velocity.y = speed;

}

//-----------------------------------

if (player.body.x < enemy2.body.x)

{

enemy2.body.velocity.x = speed \* -3;

}

else

{

enemy2.body.velocity.x = speed;

}

if (player.body.y < enemy2.body.y)

{

enemy2.body.velocity.y = speed \* -3;

}

else

{

enemy2.body.velocity.y = speed;

}

//------------------------------------

if (player.body.x < enemy3.body.x)

{

enemy3.body.velocity.x = speed \* -3;

}

else

{

enemy3.body.velocity.x = speed;

}

if (player.body.y < enemy3.body.y)

{

enemy3.body.velocity.y = speed \* -3;

}

else

{

enemy3.body.velocity.y = speed;

}

},

follow1\_1: function(){

if (spot1.body.x < enemy1.body.x)

{

enemy1.body.velocity.x = speed \* -3;

}

else

{

enemy1.body.velocity.x = speed;

}

if (spot1.body.y < enemy1.body.y)

{

enemy1.body.velocity.y = speed \* -3;

}

else

{

enemy1.body.velocity.y = speed;

}

//-------------------------------------

if (spot3.body.x < enemy3.body.x)

{

enemy3.body.velocity.x = speed \* -3;

}

else

{

enemy3.body.velocity.x = speed;

}

if (spot3.body.y < enemy3.body.y)

{

enemy3.body.velocity.y = speed \* -3;

}

else

{

enemy3.body.velocity.y = speed;

}

},

follow2: function(){

if (player.body.x < enemy4.body.x)

{

enemy4.body.velocity.x = speed \* -3;

}

else

{

enemy4.body.velocity.x = speed;

}

if (player.body.y < enemy4.body.y)

{

enemy4.body.velocity.y = speed \* -3;

}

else

{

enemy4.body.velocity.y = speed;

}

//-----------------------------------

if (player.body.x < enemy5.body.x)

{

enemy5.body.velocity.x = speed \* -3;

}

else

{

enemy5.body.velocity.x = speed;

}

if (player.body.y < enemy5.body.y)

{

enemy5.body.velocity.y = speed \* -3;

}

else

{

enemy5.body.velocity.y = speed;

}

//------------------------------------

if (player.body.x < enemy6.body.x)

{

enemy6.body.velocity.x = speed \* -3;

}

else

{

enemy6.body.velocity.x = speed;

}

if (player.body.y < enemy6.body.y)

{

enemy6.body.velocity.y = speed \* -3;

}

else

{

enemy6.body.velocity.y = speed;

}

},

follow2\_1: function(){

if (spot4.body.x < enemy4.body.x)

{

enemy4.body.velocity.x = speed \* -3;

}

else

{

enemy4.body.velocity.x = speed;

}

if (spot4.body.y < enemy4.body.y)

{

enemy4.body.velocity.y = speed \* -3;

}

else

{

enemy4.body.velocity.y = speed;

}

//-----------------------------------

if (spot6.body.x < enemy6.body.x)

{

enemy6.body.velocity.x = speed \* -3;

}

else

{

enemy6.body.velocity.x = speed;

}

if (spot6.body.y < enemy6.body.y)

{

enemy6.body.velocity.y = speed \* -3;

}

else

{

enemy6.body.velocity.y = speed;

}

},

follow3: function(){

if (player.body.x < enemy7.body.x)

{

enemy7.body.velocity.x = speed \* -3;

}

else

{

enemy7.body.velocity.x = speed;

}

if (player.body.y < enemy7.body.y)

{

enemy7.body.velocity.y = speed \* -3;

}

else

{

enemy7.body.velocity.y = speed;

}

},

follow3\_1:function(){

if (spot7.body.x < enemy7.body.x)

{

enemy7.body.velocity.x = speed \* -3;

}

else

{

enemy7.body.velocity.x = speed;

}

if (spot7.body.y < enemy7.body.y)

{

enemy7.body.velocity.y = speed \* -3;

}

else

{

enemy7.body.velocity.y = speed;

}

},

follow4: function(){

if (player.body.x < enemy8.body.x)

{

enemy8.body.velocity.x = speed \* -3;

}

else

{

enemy8.body.velocity.x = speed;

}

if (player.body.y < enemy8.body.y)

{

enemy8.body.velocity.y = speed \* -3;

}

else

{

enemy8.body.velocity.y = speed;

}

},

follow4\_1:function(){

if (spot8.body.x < enemy8.body.x)

{

enemy8.body.velocity.x = speed \* -3;

}

else

{

enemy8.body.velocity.x = speed;

}

if (spot8.body.y < enemy8.body.y)

{

enemy8.body.velocity.y = speed \* -3;

}

else

{

enemy8.body.velocity.y = speed;

}

},

update: function(game){

enemy1.body.velocity.x = 0;

enemy2.body.velocity.x = 0;

enemy3.body.velocity.x = 0;

enemy4.body.velocity.x = 0;

enemy5.body.velocity.x = 0;

enemy6.body.velocity.x = 0;

enemy7.body.velocity.x = 0;

enemy8.body.velocity.x = 0;

enemy1.body.velocity.y = 0;

enemy2.body.velocity.y = 0;

enemy3.body.velocity.y = 0;

enemy4.body.velocity.y = 0;

enemy5.body.velocity.y = 0;

enemy6.body.velocity.y = 0;

enemy7.body.velocity.y = 0;

enemy8.body.velocity.y = 0;

game.physics.arcade.collide(player,this.enemy,null,this.die,this);

game.physics.arcade.collide(player,this.spot,null,this.win,this);

game.physics.arcade.collide(this.enemy,this.spot);

game.physics.arcade.collide(this.enemy);

//game.physics.arcade.overlap(box1,this.spot)

if(player.body.x > 520){

this.follow1();

}

else{

this.follow1\_1();

}

if(player.body.x < 300){

this.follow2();

}

else{

this.follow2\_1();

}

if(player.body.y < 300){

this.follow3();

}

else{

this.follow3\_1();

}

if(player.body.y > 400){

this.follow4();

}

else{

this.follow4\_1();

}

if (left)

{

player.body.velocity.x =-playerSpeed;

}

else if (right)

{

player.body.velocity.x = playerSpeed;

}

else if (up)

{

player.body.velocity.y = -playerSpeed;

}

else if (down) {

player.body.velocity.y = playerSpeed;

}

else{

player.body.velocity.x = 0;

player.body.velocity.y = 0;

}

},

};

SpotPlatform = function (game, x, y, key, group) {

Phaser.Sprite.call(this, game, x, y,'spots');

game.physics.arcade.enable(this);

this.anchor.x = 0.5;

this.body.immovable = true;

group.add(this);

};

SpotPlatform.prototype = Object.create(Phaser.Sprite.prototype);

SpotPlatform.prototype.constructor = SpotPlatform;

**Preloader.js**

Agawan.Preloader = function(game){

};

Agawan.Preloader.prototype = {

preload: function(game){

//player

game.load.image('enemy', 'img/phaser-dude.png',27,40);

game.load.image('girl', 'img/girl.png',40,64);

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

game.load.image('spots', 'img/bilog.png',25,25);

game.load.image('wall1', 'img/wall1.png',820,5);

game.load.image('wall2', 'img/wall2.png',5,700);

//game.load.spritesheet('enemy','img/droid.png',55,55);

game.load.spritesheet('buttonleft','img/bleft.png',80,80);

game.load.spritesheet('buttonright','img/bright.png',80,80);

game.load.spritesheet('buttonup','img/bup.png',80,80);

game.load.spritesheet('buttondown','img/bdown.png',80,80);

},

create: function(game){

this.\_fontStyle = { font: "80px Arial", fill: "#FFCCFF", stroke: "#000", strokeThickness: 5, align: "center" };

var pausedText = this.add.text(130, 100, "Agawang Sulok", this.\_fontStyle);

this.stage.backgroundColor = '6B8E23';

sbutton = game.add.button(game.world.centerX - 150,game.world.height - 300,'sbutton',this.startGame,this);

sbutton.width = 300; sbutton.height = 80;

},

startGame:function(game){

this.state.start('gameplay');

},

};

**Boot.js**

var Agawan = {};

Agawan.Boot = function(game){

};

Agawan.Boot.prototype = {

init: function(){

this.input.maxPointers = 1;

this.stage.disableVisibilityChange = true;

},

preload: function(game){

game.load.audio('music', 'img/music.mp3');

},

create: function(game){

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

music.loop = true;

music.play();

Agawan.orientated = true;

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

this.scale.width = this.game.width;

this.scale.height = this.game.height;

this.scale.pageAlignHorizontally = true;

this.scale.pageVertically = true;

this.scale.setScreenSize = true

this.state.start('Preloader');

}

};