

```

1  BasicGame.Game = function (game) {};
2
3  //Graphical Object
4  var ship;
5  var ufos; //Group of Enemy UFOs which drop from the top of the screen
6  var lives; //Group of Lives which are collected
7
8  var bullets; //Bullets which your spaceship fires
9  var fireRate = 100; // Rate at which bullets are fired
10 var nextFire = 0;
11
12 //Score & Life Objects
13 var score; //Players Score
14 var lifeTotal; //Players total number of lives
15 var scoreText; //Text which is used to display the score
16 var lifeTotalText; //Text which is used to display the number of lives
17
18 //Timer Variables stores information about the timer
19 var seconds; //Number of seconds game has been running
20 var timer;
21 var timerText;
22
23 //Misc Variables
24 var cursors; //Keyboard control
25
26 BasicGame.Game.prototype = {
27
28     create: function () {
29         //Specifying the physics game engine to ARCADE
30         this.physics.startSystem(Phaser.Physics.ARCADE);
31         //Adding the starfield, logo onto the screen
32         this.starfield = this.add.tileSprite(0, 0, 800, 600, 'starfield');
33         //Adding the ship onto the screen, set the physics and the
34         //boundarys
35         ship = this.add.sprite((this.world.width / 2), this.world.height -
36         50, 'ship');
37         ship.anchor.setTo(0.5,0);
38         this.physics.enable(ship, Phaser.Physics.ARCADE);
39         ship.body.collideWorldBounds = true;
40
41         //Creating Groups
42         //Create the ufos group, set the physics and the boundarys
43         ufos = this.add.group();
44         this.physics.enable(ufos, Phaser.Physics.ARCADE);
45
46         ufos.setAll('outOfBoundsKill', true);
47         ufos.setAll('checkWorldBounds', true);
48         ufos.setAll('anchor.x', 0.5);

```

```

47     ufos.setAll('anchor.y', 0.5);
48
49     //Create the lives group, set the physics and the boundaries
50     lives = this.add.group();
51     this.physics.enable(lives, Phaser.Physics.ARCADE);
52
53     lives.setAll('outOfBoundsKill', true);
54     lives.setAll('checkWorldBounds', true);
55     lives.setAll('anchor.x', 0.5);
56     lives.setAll('anchor.y', 0.5);
57
58     //Create the bullets group, set the physics, multiples and
    boundaries
59     bullets = this.add.group();
60     bullets.enableBody = true;
61     bullets.physicsBodyType = Phaser.Physics.ARCADE;
62     bullets.createMultiple(30, 'bullet', 0, false);
63     bullets.setAll('anchor.x', 0.5);
64     bullets.setAll('anchor.y', 0.5);
65     bullets.setAll('outOfBoundsKill', true);
66     bullets.setAll('checkWorldBounds', true);
67
68     //Setting up and adding the Score, Life and Timer to the Screen
69     scoreText = this.add.text(16, 16, 'Score: 0', {
70         font: '32px arial',
71         fill: '#fff'
72     });
73     //sets the score to 0 and output to the screen
74     score = 0;
75     scoreText.text = "Score: " + score;
76
77     lifeTotalText = this.add.text(this.world.width - 150, 16, 'Lives:
    3', {
78         font: '32px arial',
79         fill: '#fff'
80     });
81     //sets the lifeTotal to 3 and output to the screen
82     lifeTotal = 3;
83     lifeTotalText.text = 'Lives: ' + lifeTotal;
84
85     timerText = this.add.text(350, 16, 'Time: 0', {
86         font: '32px arial',
87         fill: '#fff'
88     });
89     //setup timer
90     timer = this.time.create(false);
91     seconds = 0;
92     timerText.text = 'Time: ' + seconds;

```

```

93
94     //Setting the keyboard to accept LEFT, RIGHT and SPACE input
95     this.input.keyboard.addKeyCapture([Phaser.Keyboard.LEFT, Phaser.Keyboard.RIGHT, Phaser.Keyboard.SPACEBAR]);
96     cursors = this.input.keyboard.createCursorKeys();
97
98     //Set a TimerEvent to occur every second and start the timer
99     timer.loop(1000, this.updateTimer, this);
100    timer.start();
101    },
102
103    update: function () {
104        //execute 'createUfo', 'createLife', 'moveShip', 'collisionDetection'
105        function
106        this.createUfo();
107        this.createLife();
108        this.moveShip();
109        this.collisionDetection();
110    },
111
112    //moves ship and fires bullet from keyboard controls
113    moveShip: function () {
114        //if left arrow key pressed move players ship left
115        if (cursors.left.isDown) {
116            // Move to the left
117            ship.body.velocity.x = -200;
118        }
119        //if right arrow key pressed move players ship right
120        else if (cursors.right.isDown) {
121            ship.body.velocity.x = 200;
122        }
123        //else stop ship
124        else {
125            ship.body.velocity.x = 0;
126        }
127        //if space bar is pressed execute the 'fireBullet' function
128        if (this.input.keyboard.isDown(Phaser.Keyboard.SPACEBAR)) {
129            this.fireBullet();
130        }
131    },
132
133    //function executed during playing the game to create a UFO
134    createUfo: function () {
135        //Generate random number between 0 and 20
136        var random = this.rnd.integerInRange(0, 20);
137        //if random number equals 0 then create a ufo in a random x
138        position and random y velocity
139        if (random === 0) {

```

```

138 //Generating random position in the X Axis
139 var randomX = this.rnd.integerInRange(0, this.world.width - 15
0);
140 //Creating a ufo from the the ufos group and setting physics
141 var ufo = ufos.create(randomX, -50, 'ufo');
142 this.physics.enable(ufo, Phaser.Physics.ARCADE);
143 //Generating a random velocity
144 ufo.body.velocity.y = this.rnd.integerInRange(200, 300);
145 }
146 },
147
148 //function executed during playing the game to create a Life
149 createLife: function () {
150 //Generate random number between 0 and 500
151 var random = this.rnd.integerInRange(0, 500);
152 //if random number equals 0 then create a life in a random x
position
153 if (random === 0) {
154 //Generating random position in the X Axis
155 var randomX = this.rnd.integerInRange(0, this.world.width - 15
0);
156 //Creating a ufo from the the ufos group and setting physics
157 var life = lives.create(randomX, -50, 'life');
158 this.physics.enable(life, Phaser.Physics.ARCADE);
159 //Generating a random velocity
160 life.body.velocity.y = 150;
161 }
162 },
163
164 //Generate bullet and position in the x axis, set the velocity and
play the audio
165 fireBullet: function () {
166 if (this.time.now > nextFire && bullets.countDead() > 0) {
167 nextFire = this.time.now + fireRate;
168 var bullet = bullets.getFirstExists(false);
169 bullet.reset(ship.x, ship.y);
170 bullet.body.velocity.y = -400;
171 }
172 },
173
174 //function executed during playing the game to check for collisions
175 collisionDetection: function () {
176 this.physics.arcade.overlap(ship, ufos, this.collideUfo, null, thi
s);
177 this.physics.arcade.overlap(ship, lives, this.collectLife, null, t
his);
178 this.physics.arcade.overlap(bullets, ufos, this.destroyUfo, null,
this);

```

```

179     },
180
181     //function executed if there is collision between player and ufo. UFO
is destroyed, animation & sound, reduce lifeTotal
182     collideUfo: function (ship,ufo) {
183         ufo.kill();
184         lifeTotal--;
185         lifeTotalText.text = 'Lives: ' + lifeTotal;
186     },
187
188     //function executed if there is collision between ufo and bullet. UFO
is destroyed, animation & sound, increase score
189     destroyUfo: function (bullet, ufo) {
190         ufo.kill();
191         bullet.kill();
192         score += 100;
193         scoreText.text = 'Score: ' + score;
194     },
195
196     //function executed if there is collision between player and life.
Life is destroyed, animation & sound, increase lifeTotal
197     collectLife: function (ship, life) {
198         life.kill();
199         lifeTotal++;
200         lifeTotalText.text = 'Lives: ' + lifeTotal;
201     },
202
203     //Updates timer and outputs to the screen
204     updateTimer: function () {
205         seconds++;
206         timerText.text = 'Time: ' + seconds;
207     }
208
209 };

```