

```

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 var gameOverText; //Game Over message
26 var restartButton; //Restart game button
27 var gameOver;
28
29 BasicGame.Game.prototype = {
30
31     create: function () {
32         //Specifying the physics game engine to ARCADE
33         this.physics.startSystem(Phaser.Physics.ARCADE);
34         //Adding the starfield, logo onto the screen
35         this.starfield = this.add.tileSprite(0, 0, 800, 600, 'starfield');
36         //Adding the ship onto the screen, set the physics and the
        boundarys
37         ship = this.add.sprite((this.world.width / 2), this.world.height -
        50, 'ship');
38         ship.anchor.setTo(0.5,0);
39         this.physics.enable(ship, Phaser.Physics.ARCADE);
40         ship.body.collideWorldBounds = true;
41
42         //Creating Groups
43         //Create the ufos group, set the physics and the boundarys
44         ufos = this.add.group();
45         this.physics.enable(ufos, Phaser.Physics.ARCADE);
46

```

```

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

```

```

93     timer = this.time.create(false);
94     seconds = 0;
95     timerText.text = 'Time: ' + seconds;
96
97     gameOverText = this.add.text(this.world.centerX, this.world.center
Y-50, 'Game Over', {
98         font: '96px arial',
99         fill: '#fff',
100        align: 'center'
101    });
102    gameOverText.anchor.set(0.5);
103    //hides the gameState text
104    gameOverText.visible = false;
105    gameOver = false;
106
107    //Create a restart button and hide on screen
108    restartButton = this.add.button((this.world.width / 2),
(this.world.height / 2)+50, 'startButton', this.restartGame);
109    restartButton.anchor.set(0.5);
110    restartButton.visible = false;
111
112    //Setting the keyboard to accept LEFT, RIGHT and SPACE input
113    this.input.keyboard.addKeyCapture([Phaser.Keyboard.LEFT, Phaser.Ke
yboard.RIGHT, Phaser.Keyboard.SPACEBAR]);
114    cursors = this.input.keyboard.createCursorKeys();
115
116    //Set a TimerEvent to occur every second and start the timer
117    timer.loop(1000, this.updateTimer, this);
118    timer.start();
119    },
120
121    update: function () {
122        //Scroll the background
123        this.starfield.tilePosition.y += 2;
124        //if lifeTotal is less than 1 or seconds = 60 or gameOver variable
= true then execute 'truegameOver' function
125        if (lifeTotal < 1 || seconds == 60 || gameOver===true) {
126            this.gameOver();
127        }
128        //else execute
'createUfo','createLife','moveShip','collisionDetection' function
129        else {
130            this.createUfo();
131            this.createLife();
132            this.moveShip();
133            this.collisionDetection();
134        }
135    },

```

```

136
137 //moves ship and fires bullet from keyboard controls
138 moveShip: function () {
139     //if left arrow key pressed move players ship left
140     if (cursors.left.isDown) {
141         // Move to the left
142         ship.body.velocity.x = -200;
143     }
144     //if right arrow key pressed move players ship right
145     else if (cursors.right.isDown) {
146         ship.body.velocity.x = 200;
147     }
148     //else stop ship
149     else {
150         ship.body.velocity.x = 0;
151     }
152     //if space bar is pressed execute the 'fireBullet' function
153     if (this.input.keyboard.isDown(Phaser.Keyboard.SPACEBAR)) {
154         this.fireBullet();
155     }
156 },
157
158 //function executed during playing the game to create a UFO
159 createUfo: function () {
160     //Generate random number between 0 and 20
161     var random = this.rnd.integerInRange(0, 20);
162     //if random number equals 0 then create a ufo in a random x
position and random y velocity
163     if (random === 0) {
164         //Generating random position in the X Axis
165         var randomX = this.rnd.integerInRange(0, this.world.width - 15
0);
166         //Creating a ufo from the the ufos group and setting physics
167         var ufo = ufos.create(randomX, -50, 'ufo');
168         this.physics.enable(ufo, Phaser.Physics.ARCADE);
169         //Generating a random velocity
170         ufo.body.velocity.y = this.rnd.integerInRange(200, 300);
171     }
172 },
173
174 //function executed during playing the game to create a Life
175 createLife: function () {
176     //Generate random number between 0 and 500
177     var random = this.rnd.integerInRange(0, 500);
178     //if random number equals 0 then create a life in a random x
position
179     if (random === 0) {
180         //Generating random position in the X Axis

```

```

181         var randomX = this.rnd.integerInRange(0, this.world.width - 15
0);
182         //Creating a ufo from the the ufos group and setting physics
183         var life = lives.create(randomX, -50, 'life');
184         this.physics.enable(life, Phaser.Physics.ARCADE);
185         //Generating a random velocity
186         life.body.velocity.y = 150;
187     }
188 },
189
190     //Generate bullet and position in the x axis, set the velocity and
play the audio
191     fireBullet: function () {
192         if (this.time.now > nextFire && bullets.countDead() > 0) {
193             nextFire = this.time.now + fireRate;
194             var bullet = bullets.getFirstExists(false);
195             bullet.reset(ship.x, ship.y);
196             bullet.body.velocity.y = -400;
197         }
198     },
199
200     //function executed during playing the game to check for collisions
201     collisionDetection: function () {
202         this.physics.arcade.overlap(ship, ufos, this.collideUfo, null, thi
s);
203         this.physics.arcade.overlap(ship, lives, this.collectLife, null, t
his);
204         this.physics.arcade.overlap(bullets, ufos, this.destroyUfo, null,
this);
205     },
206
207     //function executed if there is collision between player and ufo. UFO
is destroyed, animation & sound, reduce lifeTotal
208     collideUfo: function (ship,ufo) {
209         ufo.kill();
210         var animation = this.add.sprite(ufo.body.x, ufo.body.y, 'kaboom');
211         animation.animations.add('explode');
212         animation.animations.play('explode', 30, false, true);
213         lifeTotal--;
214         lifeTotalText.text = 'Lives: ' + lifeTotal;
215     },
216
217     //function executed if there is collision between ufo and bullet. UFO
is destroyed, animation & sound, increase score
218     destroyUfo: function (bullet, ufo) {
219         ufo.kill();
220         bullet.kill();
221         var animation = this.add.sprite(ufo.body.x, ufo.body.y, 'kaboom');

```

```

222     animation.animations.add('explode');
223     animation.animations.play('explode', 30, false, true);
224     score += 100;
225     scoreText.text = 'Score: ' + score;
226 },
227
228     //function executed if there is collision between player and life.
    Life is destroyed, animation & sound, increase lifeTotal
229     collectLife: function (ship, life) {
230         life.kill();
231         lifeTotal++;
232         lifeTotalText.text = 'Lives: ' + lifeTotal;
233         var animation = this.add.sprite(life.body.x, life.body.y, 'lifeAni
    mation');
234         animation.animations.add('lifeAnimation');
235         animation.animations.play('lifeAnimation', 30, false, true);
236     },
237
238     //Updates timer and outputs to the screen
239     updateTimer: function () {
240         seconds++;
241         timerText.text = 'Time: ' + seconds;
242     },
243
244     //function is executed when the game ends. Stops Ship, Kills all
    objects, stops timer, Display Restart Button
245     gameOver: function () {
246         ship.body.velocity.x = 0;
247         ship.body.x = (this.world.width/2)-(ship.body.width/2);
248         ufos.callAll('kill');
249         lives.callAll('kill');
250         bullets.callAll('kill');
251         gameOverText.visible = true;
252         restartButton.visible = true;
253         timer.stop();
254     },
255
256     //Restart function, executed when restart button is pressed
257     restartGame: function () {
258         this.game.state.start('Game');
259     }
260
261 };

```