```
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 //Audio Variables stores the audio in the game
19 var music;
20 var bulletAudio;
21 var explosionAudio;
22
23 //Timer Variables stores information about the timer
24 var seconds; //Number of seconds game has been running
25 var timer;
26 var timerText;
27
28 //Misc Variables
29 var cursors; //Keyboard control
30 var gameOverText; //Game Over message
31 var restartButton; //Restart game button
32 var gameOver;
33
34 BasicGame.Game.prototype = {
35
36     create: function () {
37         //Specifying the physics game engine to ARCADE
38         this.physics.startSystem(Phaser.Physics.ARCADE);
39         //Adding the starfield, logo onto the screen
40         this.starfield = this.add.tileSprite(0, 0, 800, 600, 'starfield');
41         //Adding the ship onto the screen, set the physics and the
   boundarys
42         ship = this.add.sprite((this.world.width / 2), this.world.height -
    50, 'ship');
43         ship.anchor.setTo(0.5,0);
44         this.physics.enable(ship, Phaser.Physics.ARCADE);
45         ship.body.collideWorldBounds = true;
46
```

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

```
93          timerText = this.add.text(350, 16, 'Time: 0', {
94              font: '32px arial',
95              fill: '#fff'
96          });
97          //setup timer
98          timer = this.time.create(false);
99          seconds = 0;
100         timerText.text = 'Time: ' + seconds;
101
102         gameOverText = this.add.text(this.world.centerX, this.world.center
    Y-50, 'Game Over', {
103             font: '96px arial',
104             fill: '#fff',
105             align: 'center'
106         });
107         gameOverText.anchor.set(0.5);
108         //hides the gameState text
109         gameOverText.visible = false;
110         gameOver = false;
111
112         //Create a restart button and hide on screen
113         restartButton = this.add.button((this.world.width / 2),
    (this.world.height / 2)+50, 'startButton', this.restartGame);
114         restartButton.anchor.set(0.5);
115         restartButton.visible = false;
116
117         //Setting the keyboard to accept LEFT, RIGHT and SPACE input
118         this.input.keyboard.addKeyCapture([Phaser.Keyboard.LEFT, Phaser.Ke
    yboard.RIGHT, Phaser.Keyboard.SPACEBAR]);
119         cursors = this.input.keyboard.createCursorKeys();
120
121         //Load the audio into memory, starting music
122         bulletAudio = this.add.audio('bullet');
123         explosionAudio = this.add.audio('explosion');
124         music = this.add.audio('music', 1, true);
125         music.play('', 0, 1, true);
126
127         //Set a TimerEvent to occur every second and start the timer
128         timer.loop(1000, this.updateTimer, this);
129         timer.start();
130     },
131
132     update: function () {
133         //Scroll the background
134         this.starfield.tilePosition.y += 2;
135         //if lifeTotal is less than 1 or seconds = 60 or gameOver variable
    = true then execute 'truegameOver' function
136         if (lifeTotal < 1 || seconds == 60 || gameOver===true) {
```

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

```
182         }
183     },
184 
185     //function executed during playing the game to create a Life
186     createLife: function () {
187         //Generate random number between 0 and 500
188         var random = this.rnd.integerInRange(0, 500);
189         //if random number equals 0 then create a life in a random x
    position
190         if (random === 0) {
191             //Generating random position in the X Axis
192             var randomX = this.rnd.integerInRange(0, this.world.width - 15
    0);
193             //Creating a ufo from the the ufos group and setting physics
194             var life = lives.create(randomX, -50, 'life');
195             this.physics.enable(life, Phaser.Physics.ARCADE);
196             //Generating a random velocity
197             life.body.velocity.y = 150;
198         }
199     },
200 
201     //Generate bullet and position in the x axis, set the velocity and
    play the audio
202     fireBullet: function () {
203         if (this.time.now > nextFire && bullets.countDead() > 0) {
204             nextFire = this.time.now + fireRate;
205             var bullet = bullets.getFirstExists(false);
206             bullet.reset(ship.x, ship.y);
207             bullet.body.velocity.y = -400;
208             bulletAudio.play();
209         }
210     },
211 
212     //function executed during playing the game to check for collisions
213     collisionDetection: function () {
214         this.physics.arcade.overlap(ship, ufos, this.collideUfo, null, thi
    s);
215         this.physics.arcade.overlap(ship, lives, this.collectLife, null, t
    his);
216         this.physics.arcade.overlap(bullets, ufos, this.destroyUfo, null,
    this);
217     },
218 
219     //function executed if there is collision between player and ufo. UFO
    is destroyed, animation & sound, reduce lifeTotal
220     collideUfo: function (ship,ufo) {
221         explosionAudio.play();
222         ufo.kill();
```

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

```
267         music.stop();
268         gameOverText.visible = true;
269         restartButton.visible = true;
270         timer.stop();
271     },
272 
273     //Restart function, executed when restart button is pressed
274     restartGame: function () {
275         this.game.state.start('Game');
276     }
277 
278 };
```