```
BasicGame.Game = function (game) {};

//Graphical Object
var ship;
var ufos; //Group of Enemy UFOs which drop from the top of the screen
var lives; //Group of Lives which are collected

var bullets; //Bullets which your spaceship fires
var fireRate = 100; // Rate at which bullets are fired
var nextFire = 0;

//Score & Life Objects
var score; //Players Score
var lifeTotal; //Players total number of lives
var scoreText; //Text which is used to display the score
var lifeTotalText; //Text which is used to display the number of lives

//Timer Variables stores information about the timer
var seconds; //Number of seconds game has been running
var timer;
var timerText;

//Misc Variables
var cursors; //Keyboard control
var gameOverText; //Game Over message
var restartButton; //Restart game button
var gameOver;

BasicGame.Game.prototype = {

    create: function () {
        //Specifying the physics game engine to ARCADE
        this.physics.startSystem(Phaser.Physics.ARCADE);
        //Adding the starfield, logo onto the screen
        this.starfield = this.add.tileSprite(0, 0, 800, 600, 'starfield');
        //Adding the ship onto the screen, set the physics and the
boundarys
        ship = this.add.sprite((this.world.width / 2), this.world.height -
 50, 'ship');
        ship.anchor.setTo(0.5,0);
        this.physics.enable(ship, Phaser.Physics.ARCADE);
        ship.body.collideWorldBounds = true;

        //Creating Groups
        //Create the ufos group, set the physics and the boundarys
        ufos = this.add.group();
        this.physics.enable(ufos, Phaser.Physics.ARCADE);

```

```
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 boundarys
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
    boundarys
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         //if lifeTotal is less than 1 or seconds = 60 or gameOver variable
    = true then execute 'truegameOver' function
123         if (lifeTotal < 1 || seconds == 60 || gameOver===true) {
124             this.gameOver();
125         }
126         //else execute
    'createUfo','createLife','moveShip','collisionDetection' function
127         else {
128             this.createUfo();
129             this.createLife();
130             this.moveShip();
131             this.collisionDetection();
132         }
133     },
134
135     //moves ship and fires bullet from keyboard controls
```

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

```
180             //Creating a ufo from the the ufos group and setting physics
181             var life = lives.create(randomX, -50, 'life');
182             this.physics.enable(life, Phaser.Physics.ARCADE);
183             //Generating a random velocity
184             life.body.velocity.y = 150;
185         }
186     },
187
188     //Generate bullet and position in the x axis, set the velocity and
    play the audio
189     fireBullet: function () {
190         if (this.time.now > nextFire && bullets.countDead() > 0) {
191             nextFire = this.time.now + fireRate;
192             var bullet = bullets.getFirstExists(false);
193             bullet.reset(ship.x, ship.y);
194             bullet.body.velocity.y = -400;
195         }
196     },
197
198     //function executed during playing the game to check for collisions
199     collisionDetection: function () {
200         this.physics.arcade.overlap(ship, ufos, this.collideUfo, null, thi
    s);
201         this.physics.arcade.overlap(ship, lives, this.collectLife, null, t
    his);
202         this.physics.arcade.overlap(bullets, ufos, this.destroyUfo, null,
    this);
203     },
204
205     //function executed if there is collision between player and ufo. UFO
    is destroyed, animation & sound, reduce lifeTotal
206     collideUfo: function (ship,ufo) {
207         ufo.kill();
208         lifeTotal--;
209         lifeTotalText.text = 'Lives: ' + lifeTotal;
210     },
211
212     //function executed if there is collision between ufo and bullet. UFO
    is destroyed, animation & sound, increase score
213     destroyUfo: function (bullet, ufo) {
214         ufo.kill();
215         bullet.kill();
216         score += 100;
217         scoreText.text = 'Score: ' + score;
218     },
219
220     //function executed if there is collision between player and life.
    Life is destroyed, animation & sound, increase lifeTotal
```

```
221     collectLife: function (ship, life) {
222         life.kill();
223         lifeTotal++;
224         lifeTotalText.text = 'Lives: ' + lifeTotal;
225     },
226 
227     //Updates timer and outputs to the screen
228     updateTimer: function () {
229         seconds++;
230         timerText.text = 'Time: ' + seconds;
231     },
232 
233     //function is executed when the game ends. Stops Ship, Kills all
    objects, stops timer, Display Restart Button
234     gameOver: function () {
235         ship.body.velocity.x = 0;
236         ship.body.x = (this.world.width/2)-(ship.body.width/2);
237         ufos.callAll('kill');
238         lives.callAll('kill');
239         bullets.callAll('kill');
240         gameOverText.visible = true;
241         restartButton.visible = true;
242         timer.stop();
243     },
244 
245     //Restart function, executed when restart button is pressed
246     restartGame: function () {
247         this.game.state.start('Game');
248     }
249 
250 };
```