```
import flash.filters.BevelFilter;
import flash.filters.BlurFilter;
import flash.filters.GlowFilter;
//import is the key word to link to a class file enabling us to create instances of them.

//Create an array
var bevelArray:Array = new Array;
//Store the BevelFilter Array in the BevelArray variable index of [0]
bevelArray[0] = new BevelFilter();
//make the MovieClip filters properties hold the array of values
spaceShip_mc.filters = bevelArray;

//If someone wants to use the mouse? Do you want them to?
spaceShip_mc.onPress = function()
{
    this.startDrag(false);
}
spaceShip_mc.onRelease = function()
{
    stopDrag();
}
spaceShip_mc.onReleaseOutside = function()
{
    stopDrag();
}

//To listen for key strokes typed in we will add a Key Listener
Key.addListener(this);

//When any "Key" is pressed down Do...
function onKeyDown():Void
{
    //Get the key being pressed and Do something for the ones you are looking for.
    if(Key.getCode() == Key.RIGHT)
    {
        //Concatenating is add to an abject. by asking an Array to .concat is asking
        //to add the values of another to the end of its values.
        //the MovieClip Filters array is receiving the Bevel + Blur
        spaceShip_mc.filters = bevelArray.concat(new BlurFilter(15));//The number (15) is the thickness of the blur or pixels wide
        //On Enter Frame happens everytime a frame is played at the timeline frame rate
        spaceShip_mc.onEnterFrame = function()
        {
            //if the horizontal value is Greater than the width
            if(this._x > 650)
            {
                //reset the value to the left
                this._x = - 20;
            }
            else
            {
                //Add vaule to the Horizontal X value.
                this._x += 20;
            }
        }
    }
    if(Key.getCode() == Key.LEFT)
    {
        spaceShip_mc.filters = bevelArray.concat(new BlurFilter(15));
        spaceShip_mc.onEnterFrame = function()
        {
            if(this._x < -20)
            {
                this._x = 650;
            }
            else
            {
                this._x -= 20;
            }
        }
    }
    if(Key.getCode() == Key.UP)
    {
        spaceShip_mc.filters = bevelArray.concat(new BlurFilter(15));
        spaceShip_mc.onEnterFrame = function()
        {
```

```
76          if(this._y < -10)
77          {
78              this._y = 500;
79          }
80          else
81          {
82              this._y -= 20;
83          }
84      }
85  }
86  if (Key.getCode() == Key.DOWN)
87  {
88      spaceShip_mc.filters = bevelArray.concat(new BlurFilter(15));
89      spaceShip_mc.onEnterFrame = function()
90      {
91          if(this._y > 500)
92          {
93              this._y = -20;
94          }
95          else
96          {
97              this._y += 20;
98          }
99      }
100 }
101 }
102 //End of onKeyDown
103
104 //OnKeyUP listens for when the key is released
105 function onKeyUp():Void
106 {
107     //reset the ship filters to just the blur again
108     spaceShip_mc.filters = bevelArray;
109     //delete the onEnterFrame event to stop it from calling itself.
110     delete spaceShip_mc.onEnterFrame;
111 }
112 //End of funtion onKeyUp
113
114 //To count the times the ship has been hit
115 var hitNumber:Number = 1;
116
117 //SetInterval is calling the addPlanet function every .4th of a second
118 //This interval time determine how many planets you will have to dodge.
119 //Change 400 to another value and watch the difference
120
121 var level:Number = 0;
122 var planetCount:Number = 0;
123 var makePlanet:Number;
124 addPlanet();
125
126 //This dynamically creates planets that fly down
127 function addPlanet():Void
128 {
129     //Get the Next Highest Depth available in the flash player
130     var depth:Number = getNextHighestDepth();
131     //create a MovieClip that recieves the value of a movieclip loaded/attached and
132     //give it a unique instance name value by adding the unique depth to the end of the instance name
133     var clipName:MovieClip = _root.attachMovie("planets","planet_mc" + depth,depth);
134     //Create a random horizontal X value to drop from.
135     clipName._x = Math.random() * 600  + 40;
136     //Give the vertical Y value a negative value so it is above the screen before it starts.
137     clipName._y = -20;
138
139     //Count the planets made
140     planetCount++;
141
142     //When creating a movieclip dynamically you can specify any events you would like to have for it
143     //If you load a thumbnail photo in a gallery dynamically, you could specify the onRelease for example to
144     //load a large photo pertaining to it.
145     //Here we are creating an onEnterFrame to move the planet, check for a collisioin, and remove it when it is done
146     clipName.onEnterFrame = function()
147     {
148         if(this._y > 900 )
149         {
150             //clipName is the caller of this function so we may use this meaning "clipName"
```

```
        //and remove itself when it goes off the screen
        this.removeMovieClip();
    }
    else
    {
        //Add a value to the vertical animation here
        //NOTE: this is one key factor to your planets speed. The frame rate of your timeline is
        //first because it calls the onEnterFrame every frame, the setInterval of this addPlanet function will affect the amount on the screen
        //but the Vertical Y value also moves your objects by an increment. The bigger the number
        //the faster it will reach its vertical max off the screen
        if (level == 1)
        {
            this._y += 20;
        }
        else if(level == 2)
        {
            this._y += 25;
        }
        else if(level == 3)
        {
            this._y += 30;
        }
        //TO CREATE more difficult levels you could start small and increase its increments
    }

    //The key word hitTest refers to 2 movie clips boundaires touching
    //Because we are creating so many movies and there is only one ship, We will ask the movieclip
    //onEnterFrame if it is touching the space ship and not ask the spaceship if it is touching the one of thousands
    // of planets.
    if(clipName.hitTest(spaceShip_mc))
    {
        //Concatenate or add a GlowFilter filter to the current filters and assign them to the spaceship

        spaceShip_mc.filters = spaceShip_mc.filters.concat(new GlowFilter(glowColor(),20,20,20));

        //move the spaceship back to show inpact
        spaceShip_mc._y += 1;
        //setTimeout is just like an interval except it only happens once. So in a certain amount of time we specify.
        //run a function
        var my_timedProcess:Number = setTimeout(resetFilters, 100);
        //clipName is the instance of this movie clip that is being created form the linkage name of
        //our library symbol. We have an animation on that Movieclip time line that is stopped in frame 1.
        //Here once we know it has run into the ship we can make it play the movieclip timeline to explode.
        clipName.play();

        //Create a display to show how many times your ship has been hit.
        //the hit number is actually updated in the last frame of the planet to ensure that the onEnterFrame event
        //does not add a number every frame.
        hits_txt.text = "Hits: " + hitNumber;
    }
    planets_txt.text = "Planets Passed: " + planetCount;
}


if( planetCount >= 300)
    {
        gotoAndPlay("won",1)
    }
else if( planetCount >= 100 && planetCount < 300)
    {
        switch (planetCount)
        {
            case 100:
                clearInterval(makePlanet);
                makePlanet = setInterval(addPlanet,300);
                break;
        }
        level = 3;
        if( planetCount < 120)
        {
            level_txt.text = "Level 3";
        }
    }
    else if(planetCount >39 && planetCount < 100)
    {
```

```
226
227             switch (planetCount)
228             {
229                 case 40:
230                     clearInterval(makePlanet);
231                     makePlanet = setInterval(addPlanet,300);
232                     break;
233             }
234             level = 2;
235             if( planetCount < 50)
236             {
237                 level_txt.text = "Level 2";
238             }
239         }
240         else if (planetCount < 40)
241         {
242             switch (planetCount)
243             {
244                 case 1:
245                     clearInterval(makePlanet);
246                     makePlanet = setInterval(addPlanet,500);
247                     break;
248             }
249
250             level = 1;
251
252             if( planetCount < 10)
253             {
254                 level_txt.text = "Level 1";
255             }
256         }
257 }
258
259 function glowColor():Number
260 {
261         if(hitNumber < 5)
262             {
263                 return 0xFFFF00;
264             }
265             else if(hitNumber > 4 && hitNumber < 8)
266             {
267                 return 0xFF9900;
268             }
269             else if (hitNumber > 7 && hitNumber < 11)
270             {
271                 return 0xFF0000;
272             }
273             else if (hitNumber > 10)
274             {
275                 gotoAndPlay("lost",1)
276             }
277
278 }
279
280 //To reset the filters of your space ship after it has had a glow added
281 function resetFilters():Void
282 {
283     //Create an Array to hold the values of the current filters held in the space ship movieclip
284     var filtersArray:Array = spaceShip_mc.filters;
285     //Next ask if the last variable in this array is an instance of the Glow filter
286     //if it is...
287     if(filtersArray[filtersArray.length -1] instanceof GlowFilter)
288     {
289         //use the .pop() method to remove the last index variable from an array
290         filtersArray.pop();
291     }
292     //reassign the the movieclip filters to the new value without the Glow.
293     spaceShip_mc.filters = filtersArray;
294 }
295
296
297
298
```