

```

1 package{
2
3 //import packages needed for the game
4 import flash.display.*;
5 import flash.events.*;
6 import flash.geom.*;
7 import flash.ui.*;
8
9 //3rd party packages
10 import Key;
11
12 //this class represents the flash root, so it must extend the movieclip type
13 public class gameMain extends MovieClip {
14
15     /*
16      constants used in the game go here
17     */
18     const gravity:Number = 1;
19     const friction:Number = .8;
20     const speed:Number = 5;
21     const jump:Number = 20;
22
23     /*
24      game variables
25     */
26     public var playerVelocity:Object;
27     public var onGround:Boolean;
28
29     /*
30      objects user in the game
31     */
32     public var player:MovieClip;
33     public var gameStage:MovieClip;
34
35     //add these two lines
36     public var bulletArray:Array;
37     public var enemyArray:Array;
38
39
40
41 //runs first
42 public function gameMain():void {
43
44     //initialize any game objects
45     bulletArray = new Array();
46     enemyArray = new Array();
47     for(var i = 0; i < 5; i++)
48     {
49         var tmpEnemy:nogoodMan = new nogoodMan();
50         tmpEnemy.x = i * 70;
51         this.addChild(tmpEnemy);
52         enemyArray.push(tmpEnemy);
53     }
54
55
56
57     //new player
58     player = new tinyMan();
59     player.x = 100;
60     player.y = 150;
61
62     //new stage
63     gameStage = new stagel();
64
65     //new player velocity
66     playerVelocity = new Object();
67     playerVelocity.x = 0;
68     playerVelocity.y = 0;
69
70     //misc
71     onGround = false;
72     Key.initialize( stage );
73
74     //add objects to the stage as needed
75     this.addChild( gameStage );
76     this.addChild( player );
77
78     //start game loop
79     this.addEventListener( 'enterFrame', onEnterFrame );
80 }
81
82 //the code within this function will run once every frame
83 public function onEnterFrame( e:Event ):void {
84
85     updateInput();
86     updateEnemies();
87     updateBullets();
88
89
90
91     //update velocity
92     playerVelocity.y += gravity;
93     playerVelocity.x *= friction;
94
95     //test to see if the player is on the ground
96     if( gameStage.hitTestPoint(player.x, player.y, true) && playerVelocity.y > 0 ) {
97         //if it has hit the ground, halt gravity
98         playerVelocity.y = 0;
99         //set on ground to true
100         onGround = true;
101     } else {
102         //hes not on the ground, let the game know this.
103         onGround = false;
104     }
105
106     //update position
107     player.x += playerVelocity.x;
108     player.y += playerVelocity.y;
109
110     this.x -= playerVelocity.x;
111     this.y -= playerVelocity.y;
112
113 }
114
115 //handle input
116 public function updateInput():void {
117
118     if( Key.isDown( Keyboard.LEFT ) ) {
119         playerVelocity.x = -speed;
120     }
121
122     if( Key.isDown( Keyboard.RIGHT ) ) {
123         playerVelocity.x = speed;
124     }
125
126     if( Key.isDown( Keyboard.UP ) ) {
127         if( onGround ) {
128             onGround = false;
129             playerVelocity.y = -jump;
130         }
131     }
132     if(Key.isDown(Keyboard.SPACE))
133     {
134         firebullet();
135     }
136 } //end updateInput
137
138 //function for firing a bullet using spacebar
139 public function firebullet():void
140 {
141     var tmpbullet:bullet = new bullet();
142     tmpbullet.velocity.x = 10;
143     tmpbullet.x = player.x;
144     tmpbullet.y = player.y - 55;
145     bulletArray.push(tmpbullet);
146     this.addChild(tmpbullet);
147 } //end firebullet
148
149 //function for updating enemies on the stage
150 public function updateEnemies():void {
151     for(var i in enemyArray)
152     {
153         enemyArray[i].update(player);
154         //update velocity
155         enemyArray[i].velocity.y += gravity;
156         enemyArray[i].velocity.x *= friction;
157
158         //test to see if the player is on the ground
159         if( gameStage.hitTestPoint(enemyArray[i].x, enemyArray[i].y, true) && enemyArray[i].velocity.y > 0 ) {
160             //if it has hit the ground, halt gravity
161             enemyArray[i].velocity.y = 0;
162             //set on ground to true
163             onGround = true;
164         } else {
165             //hes not on the ground, let the game know this.
166             onGround = false;
167         }
168
169         //update position
170         enemyArray[i].x += enemyArray[i].velocity.x;
171         enemyArray[i].y += enemyArray[i].velocity.y;
172     } //end for
173 } //end updateEnemies
174
175 //function for updating bullets
176 public function updateBullets():void {
177     for(var i in bulletArray)
178     {
179         bulletArray[i].x += bulletArray[i].velocity.x;
180         bulletArray[i].y += bulletArray[i].velocity.y;
181
182         for(var j in enemyArray)
183         {
184
185             if(bulletArray[i].hitTestObject(enemyArray[j]))
186             {
187                 //bullet hit an object - remove enemy and bullet
188                 this.removeChild(bulletArray[i]);
189                 this.removeChild(enemyArray[j]);
190                 bulletArray.splice(i,1);
191                 enemyArray.splice(j,1);
192             }
193
194         } //end inner for
195     } //end outer for
196 } //end updateBullets
197 } //end class
198 } //end package

```