

Table of Contents

1. [Introduction](#)
2. [The New Game Program](#)
3. [The Missile Type](#)
4. [Modifying the TSpaceShip Type](#)
5. [Summary](#)
6. [Blitzmax commands introduced in this tutorial](#)

Introduction

In this final part of our introduction series, we will be adding the capability to check for collisions between our fired missiles and the AlienShip.

So we know already upfront that only the AlienShip needs to take care of this collision checking. So we need

- A Collision Checking behaviour (method) within our TAlienShip
- We need a way to indicate that the collision has occurred. Since we do not wish to destroy anyone (at this stage), the AlienShip will simply change color to red when hit by the missile
- We have to modify our parent object TGameObject to set the normal drawing color to white.

OK, let's get started

The New Game Program

I'm presenting you the new additions below. Go ahead and build and run the program so you can see what it does. The highlighted text are our new changes:-

```
' -----SETUP GAME CONDITIONS-----  
Global GameObjectList:TList=CreateList()  
Graphics 640,480,0  
  
Local URL:String="http://www.2dgamecreators.com/tutorials  
/gameprogramming/basic/"  
Local Player:TSpaceShip =  
TSpaceShip.Create(URL+"/blobship_1-1.png",320,420)  
Local Alien:TAlienShip =  
TAlienShip.Create(URL+"/cartoonufo_1-1.png",320,0)  
  
' -----MAIN LOOP-----  
Repeat  
  Cls  
  For o:TGameObject=EachIn GameObjectList  
    o.DrawSelf()  
    o.UpdateState()  
  Next  
  Flip
```

```
Until KeyDown(KEY_ESCAPE) Or AppTerminate()  
End
```

```
' -----TYPES, ATTRIBUTES AND BEHAVIOURS-----  
Type TGameObject
```

```
    Field X:Int = 320  
    Field Y:Int = 420  
    Field Speed:Int=3  
    Field Image:TImage
```

```
    Method DrawSelf()  
        SetColor 255,255,255  
        DrawImage Image,X,Y  
    End Method
```

```
    Method UpdateState() Abstract
```

```
End Type
```

```
Type TSpaceShip Extends TGameObject
```

```
    Field MissileDelay:Int
```

```
    Function Create:TSpaceShip(File:String,xstart:Int,ystart:Int)  
        Local Ship:TSpaceShip=New TSpaceShip  
        Ship.X=xstart  
        Ship.Y=ystart  
        Ship.Image=LoadImage(LoadBank(File))
```

```
    If Ship.Image=NULL  
        Print "Not able to load image file. Program aborting"  
    End  
EndIf
```

```
    ListAddLast GameObjectList, Ship  
    Return Ship  
End Function
```

```
    Method UpdateState()
```

```
        If KeyDown(KEY_LEFT)  
            X :- Speed  
        EndIf  
        If KeyDown(KEY_RIGHT)  
            X :+ Speed  
        EndIf  
        If KeyDown(KEY_SPACE)  
            FireMissile()  
        EndIf  
        MissileDelay :- 1
```

```
        If X<0 Then X=0  
        If X>620 Then X=620
```

```
    End Method
```

```

Method FireMissile()
    If MissileDelay<0
        Local FileName:String="D:\Program Files\BlitzMax\samples\firepaint
\bullet.png"
        Local Missile:TMissile=TMissile.Create(FileName, X+23,Y)
        MissileDelay=10
    End If
End Method

```

```

End Type

```

```

Type TAlienShip Extends TGameObject

```

```

    Field Explosion:Int

```

```

Function Create:TAlienShip(File:String,xstart:Int,ystart:Int)
    Local Ship:TAlienShip=New TAlienShip
    Ship.X=xstart
    Ship.Y=ystart
    Ship.Image=LoadImage(LoadBank(File))

```

```

    If Ship.Image=NULL
        Print "Not able to load image file. Program aborting"
    End
EndIf

```

```

    ListAddLast GameObjectList, Ship
    Return Ship
End Function

```

```

Method UpdateState()
    X :- Speed
    If X<-ImageWidth(Image) Then X=620
    CheckCollision()
    If explosion>0 explosion :- 1

```

```

End Method

```

```

Method CheckCollision()

```

```

    For Local g:TGameObject=EachIn GameObjectList
        If g<>Self Then
            If ImagesCollide(self.Image,X,Y,0,g.Image,g.X,g.Y,0)
                ListRemove(GameObjectList,g)
                Self.Expllosion :+ 10
            EndIf
        EndIf
    Next

```

```

End Method

```

```

Method DrawSelf()

```

```

    If explosion>0

```

```

        SetColor 255-explosion,0,0
    Else
        SetColor 255,255,255
    EndIf
    DrawImage Image,X,Y
End Method

```

End Type

Type TMissile Extends TGameObject

```

Function Create:TMissile(File:String,xstart:Int,ystart:Int)
    Local Ship:TMissile=New TMissile
    Ship.X=xstart
    Ship.Y=ystart
    Ship.Image=LoadImage(LoadBank(File))

    If Ship.Image=NULL
        Print "Not able to load image file. Program aborting"
    End
EndIf

    ListAddLast GameObjectList, Ship
    Return Ship
End Function

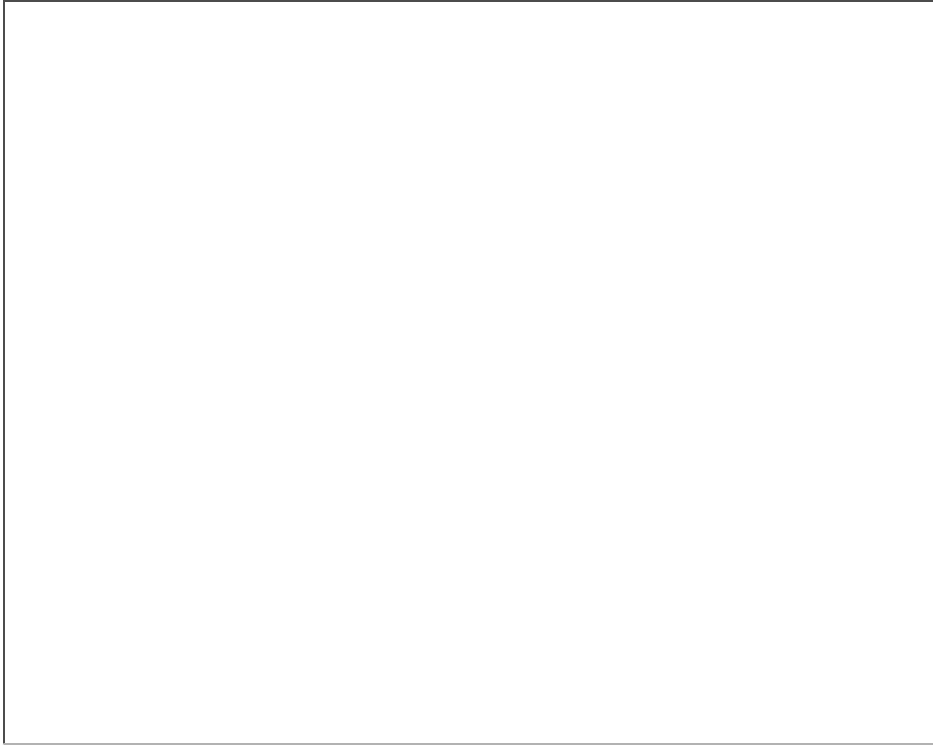
Method UpdateState()
    Y :- Speed
    If Y<0 Then ListRemove(GameObjectList,Self)
End Method

```

End Type

⁃.

Now we can see the AlienShip glowing red whenever its hit by our missiles.



The Modified TAlienShip Type

We need to make two changes to the UpdateState Method, one to always check for collision. The other to reduce our explosion variable so that the red glow will be slowly reduced in intensity..

```
Method UpdateState()  
  X := Speed  
  If X<-ImageWidth(Image) Then X=620  
  CheckCollision()  
  If explosion>0 explosion := 1  
  
End Method
```

The CheckCollision Method iterate through each object in the GameObjectList and checks whether a collision has occurred or not using the built-in BlitzMax **ImagesCollide** function.

When a collision is detected, we use **ListRemove** to remove the bullet from the GameObjectList and increase our explosion variable by 10 points (to keep our AlienShip glowing).

```
Method CheckCollision()  
  
  For Local g:TGameObject=EachIn GameObjectList  
    If g<>Self Then  
      If ImagesCollide(self.Image,X,Y,0,g.Image,g.X,g.Y,0)  
        ListRemove(GameObjectList,g)  
        Self.Explosion :=+ 10  
      EndIf  
    EndIf  
  Next  
  
End Method
```

Finally we have to create a new DrawSelf Method specific to the TAlienShip type as it now behaves differently to the parent Method. This technique is called Method Override, ie we are overriding the parent method with our own.

The Method simply checks the number in the Explosion variable if its more than zero we'll set the drawing color to red tint. Otherwise the drawing color is set to white.

```
Method DrawSelf()  
    If explosion>0  
        SetColor 255-explosion,0,0  
    Else  
        SetColor 255,255,255  
    EndIf  
    DrawImage Image,X,Y  
End Method
```

Summary

There we have it, a very rudimentary "game" built using BlitzMax. It's not very exciting at the moment but the whole idea now is to start learning the necessary concepts of 2D programming where we can then use the framework we have built over the last few tutorials to design better, more complex and fun games.

Blitzmax commands introduced in this tutorial

Only one new function introduced in this tutorial:-

Function ImagesCollide(image1:TImage,x1,y1,frame1,image2:TImage,x2,y2,frame2)

Tests if two images collide

Back to the [index](#), [Previous](#) Tutorial