How to write a BreakOut Game: Part 1: Game Design and FrameWork Review

** If you have enjoyed these tutorials, you may want to check out my other contributions (link)

Table of Contents

Introduction

This series of tutorials will show you how to use the 2D Game FrameWork that I introduced in my Learning 2D Game Programming series to program a BreakOut Game.

If you have not done so, I suggests you at least go through the Basic Framework section (click here) to familiarise yourself with the concepts before proceeding on. You can of course ignore this advice and plunge straight along :)

Tutorial Index

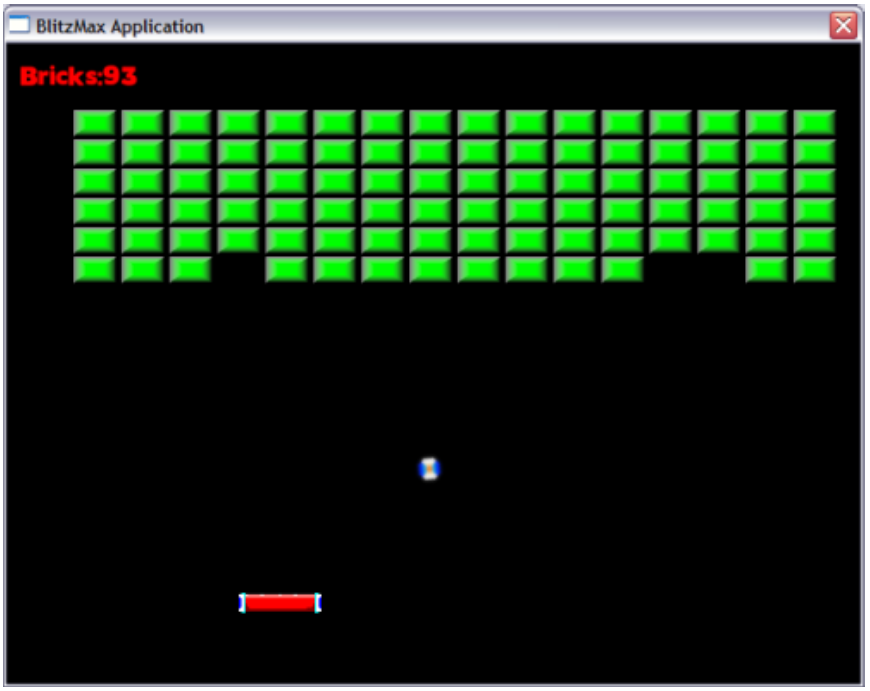
These series of tutorials will be divided into 2 sections
Section1: The basic Game elements. Adding the TBall, TPaddle and TBrick game objects
Section 2: Programming the Game States and enhancing the basic 2d Game Framework

The Game Design

As most of us are already familiar with the BreakOut game we do not need to go into too much detail on the design of the game. It is however good practice to have a complete picture of how your game is going to look like before embarking on the actual coding.

To pit it simply, a BreakOut type game consists of a player controlled paddle and a bouncing ball. The ball reflects off the paddle and the walls. When the ball hit any bricks (which are lined up on the graphics screen) the bricks are destroyed. The object is to remove the bricks whilst making sure that the ball remains in play.

Our game would look something like this image



Revisiting The Basic Game Framework

From what we have seen earlier, the Game Framework can be divided into several parts

- The initial Game Conditions setup
- The Main Loop
- The Game Objects (Types) Declarations
- The helper functions

The **Setting up Game Conditions** section consist of variable declarations, initiating the graphic screen, loading media e.g. images etc

The Main Loop section updates and draws all of the game objects contained within the GameObjectList. The backbuffer is cleared each time before the drawing starts and the Flip commands flips the backbuffer to the graphic screen at each refresh rate. So typically our game will be running at 60 FPS

```
' ----------------------MAIN LOOP-------------------------
  Repeat
    Cls
    For Local o:TGameObject=EachIn GameObjectList
       o.DrawSelf()
       o.UpdateSelf()
    Next
    Flip
  Until KeyDown(KEY_ESCAPE) Or AppTerminate()

End
```

The **Game Object Declarations** section consists of two parts. The TGameObject Type declaration where common attributes of all game objects such as display locations (X,Y), Speed, Image and Scaling factors are defined in the TGameObject Type. The generic DrawSelf behaviour (or method) is fully defined whilst the UpdateSelf method is declared as an abstract method thus forcing all descendant objects to have an UpdateSelf method.

To make a more robust Mother Object, we will be adding more attributes to this Type as we go through the tutorial.

```
' ---------------THE MOTHER OF ALL OBJECT, TYPE GAMEOBJECT-------------
Type TGameObject
```

```
    Field X:Int
    Field Y:Int
    Field XSpeed:Float=3
    Field YSpeed:Float=-3
    Field Image:TImage
    Field XScale:Float
    Field YScale:Float

    Method DrawSelf()
      SetScale XScale, YScale
      DrawImage Image,X,Y
    End Method

     Method UpdateSelf() Abstract

End Type
```

The other game objects and helper functions are going to change depending on the type of game we are writing. The generic helper function we have here, **CreateObject**, is the one that descendants of TGameObject can use to help create an instance of themselves. The created instances are always added to the GameObjectList

```
Function CreateObject(Obj:TGameObject,
Image:TImage,xstart:Int,ystart:Int,Scale:Float=1.0)

    Obj.X=xstart
    Obj.Y=ystart
    Obj.XScale=Scale
    Obj.YScale=Scale
    Obj.Image=Image

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

    ListAddLast GameObjectList, Obj

End Function
```

Summary

A fairly easy first part to this series. Having reviewed the Game Framework and with a fair idea of what we want to do, we will start in the next part with the first of 3 main game objects in this game, the bouncing ball (the other objects are the player controlled paddle and the bricks)

Goto the Next Tutorial