Programming Cookbook II

“The BlitzMax Years”

By

Nicholas Kingsley

Introduction

Welcome to the second part of my “Programming Cookbook” series.

This version of the book contains my BlitzMax code, almost all of which wont be in the first part of the series!

Hopefully you will find the code useful, and if not, then hope it will give you some fresh ideas!

Players Around A Planet

This was originally going to be the sequel to my “Humans On A Planet game”, but I never completed it.

```
SuperStrict

Import nicholas.manymouse
Import nicholas.SetDrawingCommands
Import bah.volumes
Import leadwerks.apptiming
Import sidesign.minib3d

Include "../Display/TDisplay.bmx"
Include "../Config/TConfig.bmx"
Include "../Spark/TSpark.bmx"
Include "TSetup.bmx"
Include "TPlayers.bmx"
Include "TPlanet.bmx"
Include "TTitle.bmx"
Include "TActivePlayers.bmx"
Incbin "../Display/title.PNG"

' Types

' Constants
Const SCORE_PAD:String                      =       "00000000000"
Const COLLECT_PAD:String                    =       "000000"
Const STAGE_PAD:String                      =       "000"
Const STAGE_PATH:String                     =       "Stages/"

Const GAMETYPE_SINGLECOMPUTER:Byte   =      1
Const GAMETYPE_NETWORK_SERVER:Byte   =      2
Const GAMETYPE_NETWORK_CLIENT:Byte   =      3

Const MAX_PLAYERS:Byte                      =       4

Const MESH_PLANET1:Byte                     =       0
Const MESH_PLANET2:Byte                     =       1
Const MESH_PLANET3:Byte                     =       2
Const IMAGE_PLAYER1:Byte                    =       3
Const IMAGE_PLAYER2:Byte                    =       4
Const IMAGE_PLAYER3:Byte                    =       5
Const IMAGE_PLAYER4:Byte                    =       6
Const IMAGE_PLAYER1FIRE:Byte         =      7
Const IMAGE_PLAYER2FIRE:Byte         =      8
Const IMAGE_PLAYER3FIRE:Byte         =      9
Const IMAGE_PLAYER4FIRE:Byte         =      10
Const IMAGE_GOODSHAPE:Byte                  =       11
Const IMAGE_BADSHAPE1:Byte                  =       12
Const IMAGE_BADSHAPE2:Byte                  =       13
Const IMAGE_BADSHAPE3:Byte                  =       14
Const IMAGE_BADSHAPE4:Byte                  =       15
Const IMAGE_BADSHAPE5:Byte                  =       16
Const IMAGE_BADSHAPE6:Byte                  =       17
Const IMAGE_INCREASEAMMO:Byte        =      18
Const IMAGE_INCREASEMOVEMENTSPEED:Byte=     19
Const IMAGE_INCREASESCORE:Byte              =       20
Const IMAGE_MEGABOMB:Byte                   =       21
Const IMAGE_TIMEFREEZE:Byte                 =       22
Const FONT_SCORE:Byte                       =       23
Const FONT_PLAYERSTATUS:Byte         =      24
Const FONT_GAMESTATUS:Byte                  =       25
Const FONT_ACTIVEPLAYER:Byte         =      26
Const TEXTURE_STARFIELD1:Byte        =      27
Const IMAGE_STARTGAME_STANDARD:Byte  =      28
Const IMAGE_STARTGAME_MOUSEOVER:Byte =      29
Const IMAGE_OPTIONS_STANDARD:Byte           =       30
Const IMAGE_OPTIONS_MOUSEOVER:Byte          =       31
Const IMAGE_INSTRUCTIONS_STANDARD:Byte              =       32
Const IMAGE_INSTRUCTIONS_MOUSEOVER:Byte             =       33
Const IMAGE_QUITGAME_STANDARD:Byte          =       34
Const IMAGE_QUITGAME_MOUSEOVER:Byte         =       35




' Globals
Global display:TDisplay                     =       Null
Global setup:TSetup                         =       Null
Global config:TConfig                       =       Null
Global planet:TPlanet                       =       Null
```

```
Global sparks:TSpark                        =       Null
Global activePlayers:TActivePlayers  =      Null
Global title:TTitle                         =       Null
Global players:TPlayers[MAX_PLAYERS]
Global loadData:TLoadData[MAX_FILES]
Global mouseControl:TControlPlayerInfo[]
Global joypadControl:TControlPlayerInfo[]
Global inUse:Byte[MAX_PLAYERS]              ' This contains the PLAYER_x value (or PLAYER_NONE) for
all active players
Global playerBackgroundColours:Int[]=[255,255,0,255,0,0,0,255,0,0,0,255]
Global playerForeColours:Int[]=[0,0,0,255,255,255,255,255,255,255,255,255]

' Local
Local screenWidth:Int
Local screenHeight:Int
Local numPlaying:Byte
Local gameType:Byte
Local loop:Byte

' Main Program
setup=TSetup.Create()
If setup<>Null
        If setup.setup(loadData)=False
                shutdownProgram(setup.returnErrorText(),loadData)
        EndIf
Else
        shutdownProgram("Unable to allocate memory for TSetup",loadData)
EndIf

While title.Do()=True
        screenWidth=display.returnDisplayWidth()
        screenHeight=display.returnDisplayHeight()
        activePlayers.initialise(screenWidth,screenHeight)
        If activePlayers.Do()=True
                numPlaying=activePlayers.returnNumPlaying()

                ' Setup the planet
                planet.initialise(screenWidth,screenHeight,numPlaying)

                gameType=GAMETYPE_SINGLECOMPUTER
                If setupPlayers(gameType,numPlaying,screenWidth,screenHeight,loadData)=True
                        If gameType=GAMETYPE_SINGLECOMPUTER
                                doSingleComputerGameLoop(numPlaying,loadData)
                        EndIf
                Else
                EndIf
        EndIf
EndWhile
End

'-------------------------------------------------------------------------------------------------------
-------------------------------------------------------------
' Functions
Function shutdownProgram(errorMessage:String,loadData:TLoadData[])
        clearAllStructures(loadData)

        If errorMessage<>""
                Notify errorMessage,True
        EndIf

        End
EndFunction

Function clearAllStructures(loadData:TLoadData[])
Local loop:Byte

        If setup<>Null
                setup.deleteSetupMiceJoypadArray()
                setup=Null
        EndIf

        If display<>Null
                display.loadRemoveFiles(False,loadData,Null)
                display.shutDownGraphics()
                display=Null
        EndIf

        config=Null
        planet=Null
        sparks=Null
        activePlayers=Null
        title=Null
        destroyPlayers()
EndFunction

Function
setupPlayers:Byte(gameType:Byte,numPlaying:Byte,screenWidth:Int,screenHeight:Int,loadData:TLoadData[
])
Local loop:Byte
Local index:Byte
Local y:Int
```

```
	For loop=0 To numPlaying-1
		index=inUse[loop]-PLAYER_1
		players[index]=TPlayers.Create(screenWidth,screenHeight,loadData,inUse[loop])
		If players[index]=Null
			Return False
		Else
			players[index].initialisePlayer()
		EndIf
	Next

	Return True
EndFunction

Function destroyPlayers()
Local loop:Byte

	For loop=0 To MAX_PLAYERS-1
		players[loop]=Null
	Next
EndFunction

Function findNextPlayer:Byte(currentPlayer:Byte,actionType:Byte)
Local newPlayer:Byte

	newPlayer=wrapB(currentPlayer+1,PLAYER_1,PLAYER_4)
	While newPlayer<>currentPlayer
		If players[newPlayer-PLAYER_1].returnIsAlive()=True
			Return True
		Else
			newPlayer=wrapB(currentPlayer+1,PLAYER_1,PLAYER_4)
		EndIf
	EndWhile

	Return False
EndFunction

Function doSingleComputerGameLoop:Byte(numPlaying:Byte,loadData:TLoadData[])
Local speed:Float
Local numJoypads:Byte
Local numMice:Byte
Local loop:Byte
Local index:Byte
Local score:Int
Local cont:Byte
Local numInPlay:Byte

	UpdateAppTime()
	speed=AppSpeed()

	numJoypads=display.returnNumJoypads()
	numMice=display.returnNumMice()
	cont=True

	While cont=True
		If KeyHit(KEY_ESCAPE)
			Return False
		EndIf

		Cls
		ResetCollisions
		ClearCollisions

		' Do any 3D stuff here
		planet.Display3D()
		For loop=0 To MAX_PLAYERS-1
			If inUse[loop]<>PLAYER_NONE
				index=inUse[loop]-PLAYER_1
				players[index].Display3D()
			EndIf
		Next

		UpdateWorld
		RenderWorld

		' Do all 2D stuff here
		BeginMax2D()

		'DrawImage loadData[IMAGE_STARFIELD1].image,0,0

		' Display the planet
		planet.Display2D()
		score=0
		numInPlay=0

		For loop=0 To MAX_PLAYERS-1
			If inUse[loop]<>PLAYER_NONE
				index=inUse[loop]-PLAYER_1
				players[index].Display2D()
				players[index].Process(speed)
				score:+players[index].returnScore()
```

```
                        If players[index].returnIsAlive()=True
                                numInPlay:+1
                        EndIf
                EndIf
        Next

        sparks.Display(speed*0.1)

        ' Display the accumaltive score
        SetImageFont loadData[FONT_SCORE].font
        SetDrawingCommands()
        DrawText Right$(SCORE_PAD+score,Len(SCORE_PAD)),0,0

        playerControlRoutine_Game(speed,numPlaying,numMice,numJoypads)

        EndMax2d()

        planet.Process(speed)
        If planet.returnIsAlive()=False Or numInPlay=0
                cont=False
        EndIf

        Flip

        UpdateAppTime()
        speed=AppSpeed()
    EndWhile

    planet.finish()
EndFunction

Function playerControlRoutine_Game(speed:Float,numPlaying:Byte,numMice:Byte,numJoypads:Byte)
Local loop:Byte
Local device:Int
Local eType:Int
Local item:Int
Local value:Int
Local whichPlayer:Byte

    ' Do mouse movement first, if there are any
    If numMice>0
        while ManyMouse_BMInterface_PollEvent(device,item,value,eType)
            MoveMouse 0,0
            whichPlayer=mouseControl[device].whichPlayer
    '       DrawText "WP:"+whichPlayer,0,0
            If whichPlayer<>PLAYER_NONE
                whichPlayer:-PLAYER_1

                Select eType
                    Case MANYMOUSE_EVENT_ABSMOTION

    players[whichPlayer].movePlayer(speed,item,Float(value),True)
                    Case MANYMOUSE_EVENT_RELMOTION

    players[whichPlayer].movePlayer(speed,item,Float(value),False)
                    Case    MANYMOUSE_EVENT_BUTTON
                                                                        If value=0
                                                                            Select item
                                                                                Case
0

                ' LMB

                    players[whichPlayer].fire()

'If     player[whichPlayer].checkToSeeIfShapeIsSelected()=True
'
                                        vortex.detonateAllShapes(whichPlayer)

                                'EndIf
                                                                                Case
1

' RMB

'vortex.detonateAllShapes(whichPlayer)

'player[whichPlayer].resetMines()
                                                                            EndSelect
                                                                        EndIf
                    EndSelect
                EndIf
        EndWhile
    EndIf

    If numJoypads>0
        For loop=0 To numJoypads-1
            whichPlayer=joypadControl[loop].whichPlayer
            If whichPlayer<>PLAYER_NONE
                whichPlayer:-PLAYER_1
```

```
		players[whichPlayer].movePlayer(speed,0,Float(JoyX(loop))*speed*1.75,False)

		players[whichPlayer].movePlayer(speed,1,Float(JoyY(loop))*speed*1.75,False)

                        If JoyHit(1,loop)=1
                        EndIf

                        If JoyHit(2,loop)=1
                        EndIf

                EndIf
            Next
        EndIf
EndFunction

#filesToLoad

DefData TYPE_3DGRAPHIC,                                     MESH_PATH+"Planet1.b3d"
DefData TYPE_3DGRAPHIC,                                     MESH_PATH+"Planet2.b3d"
DefData TYPE_3DGRAPHIC,                                     MESH_PATH+"Planet3.b3d"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"Player1.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"Player2.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"Player3.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"Player4.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"Player1Fire.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"Player2Fire.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"Player3Fire.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"Player4Fire.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"GoodShape.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"BadShape1.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"BadShape2.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"BadShape3.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"BadShape4.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"BadShape5.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"BadShape6.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"IncreaseAmmo.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"IncreaseMovementSpeed.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"IncreaseScore.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"MegaBomb.PNG"
DefData TYPE_2DGRAPHIC | TYPE_AUTOMIDHANDLE,IMAGE_PATH+"TimeFreeze.PNG"
DefData TYPE_FONT,                                                      FONT_PATH+"VDUB.ttf", 14
DefData TYPE_FONT,                                                      FONT_PATH+"PANNETJE.TTF",     8
DefData TYPE_FONT,                                                      FONT_PATH+"DooM.ttf", 14
DefData TYPE_FONT,                                                      FONT_PATH+"DBXLNUW_.TTF",32
DefData TYPE_TEXTURE,                                           TEXTURE_PATH+"starField1.png"
DefData TYPE_2DGRAPHIC,                                 IMAGE_PATH+"StartGame_Standard.PNG"
DefData TYPE_2DGRAPHIC,                                 IMAGE_PATH+"StartGame_MouseOver.PNG"
DefData TYPE_2DGRAPHIC,                                 IMAGE_PATH+"Options_Standard.PNG"
DefData TYPE_2DGRAPHIC,                                 IMAGE_PATH+"Options_MouseOver.PNG"
DefData TYPE_2DGRAPHIC,                                 IMAGE_PATH+"Instructions_Standard.PNG"
DefData TYPE_2DGRAPHIC,                                 IMAGE_PATH+"Instructions_MouseOver.PNG"
DefData TYPE_2DGRAPHIC,                                 IMAGE_PATH+"QuitGame_Standard.PNG"
DefData TYPE_2DGRAPHIC,                                 IMAGE_PATH+"QuitGame_MouseOver.PNG"


DefData TYPE_END

Type TPlayerArea
        Field x:Int
        Field y:Int
        Field width:Int
        Field height:Int
        Field state:Byte
        Field deviceType:Byte
        Field deviceIndex:Int
EndType

Type TActivePlayers
        Const STATE_WAITINGFORPLAYER:Byte    =       1
        Const STATE_PLAYERACTIVE:Byte        =       2

        Const NOT_INUSE:Int                          =       -1
        Const NOT_FOUND:Int                          =       -1

        Field screenWidth:Int
        Field screenHeight:Int

        Field loadData:TLoadData[]

        Field playerArea:TPlayerArea[MAX_PLAYERS]

        Field numPlaying:Byte

        Function Create:TActivePlayers(ld:TLoadData[])
        Local t:TActivePlayers
        Local loop:Byte

                t=New TActivePlayers
                If t<>Null
                        t.loadData=ld
                        For loop=0 To MAX_PLAYERS-1
```

```
                        t.playerArea[loop]=New TPlayerArea
                        If t.playerArea[loop]=Null
                                t=Null
                                Exit
                        EndIf
                Next
        EndIf

        Return t
EndFunction

Method initialise(sW:Int,sH:Int)
Local height:Int
Local loop:Byte

        screenWidth=sW
        screenHeight=sH

        ' Work out height for each area
        height=(screenHeight-36)/MAX_PLAYERS

        For loop=0 To MAX_PLAYERS-1
                inUse[loop]=PLAYER_NONE
                playerArea[loop].x=0
                playerArea[loop].y=((height+2)*loop)
                playerArea[loop].width=screenWidth
                playerArea[loop].height=height
                playerArea[loop].state=STATE_WAITINGFORPLAYER
                playerArea[loop].deviceType=CONTROL_NONE
                playerArea[loop].deviceIndex=NOT_INUSE
        Next

        Print "X1"
        setup.clearMiceAndJoypads()
        Print "X2"
        numPlaying=0
EndMethod

Method Display()
Local loop:Byte
Local text:String

        SetDrawingCommands()
        SetImageFont loadData[FONT_ACTIVEPLAYER].font

        For loop=0 To MAX_PLAYERS-1
                SetColor
playerBackgroundColours[(loop*3)],playerBackgroundColours[(loop*3)+1],playerBackgroundColours[(loop*
3)+2]
                DrawRect
playerArea[loop].x,playerArea[loop].y,playerArea[loop].width,playerArea[loop].height

                Select playerArea[loop].state
                        Case    STATE_WAITINGFORPLAYER
                                                                SetColor 0,0,0
                                                                text="Waiting For Player"
                        Case    STATE_PLAYERACTIVE
                                                                SetColor 0,0,0
                                                                text="Player has joined"
                EndSelect

                SetColor
playerForeColours[(loop*3)],playerForeColours[(loop*3)+1],playerForeColours[(loop*3)+2]
                DrawText text,((playerArea[loop].width-TextWidth(text)) Shr
1)+playerArea[loop].x,((playerArea[loop].height-TextHeight(text)) Shr 1)+playerArea[loop].y
        Next

        SetDrawingCommands()
        SetImageFont loadData[FONT_GAMESTATUS].font
        text="Playing : "+Chr$(48+numPlaying)+"/"+Chr$(48+MAX_PLAYERS)
        DrawText text,0,screenHeight-TextHeight(text)

        If numPlaying>0
                text="Press RETURN when ready to start"
                DrawText text,screenWidth-TextWidth(text),screenHeight-TextHeight(text)
        EndIf
EndMethod

Method findDevice:Int(deviceIndex:Byte,deviceType:Byte)
Local loop:Byte

        For loop=0 To MAX_PLAYERS-1
                If playerArea[loop].state=STATE_PLAYERACTIVE And
playerArea[loop].deviceIndex=deviceIndex And ..
                        playerArea[loop].deviceType=deviceType
                        Return loop
                EndIf
        Next

        Return NOT_FOUND
EndMethod
```

```
	Method findFreePlace:Int()
	Local loop:Byte

		For loop=0 To MAX_PLAYERS-1
			If playerArea[loop].state=STATE_WAITINGFORPLAYER
				Return loop
			EndIf
		Next

		Return NOT_FOUND
	EndMethod


	Method addPlayer(deviceIndex:Byte,deviceType:Byte)
	Local index:Int

		' Look to see if this device is already know
		index=findDevice(deviceIndex,deviceType)
		If index=NOT_FOUND
			index=findFreePlace()
			If index<>NOT_FOUND
				' Found a place, so store details
				playerArea[index].state=STATE_PLAYERACTIVE
				playerArea[index].deviceType=deviceType
				playerArea[index].deviceIndex=deviceIndex
				numPlaying:+1
			EndIf
		EndIf
	EndMethod

	Method removePlayer(deviceIndex:Byte,deviceType:Byte)
	Local index:Int

		index=findDevice(deviceIndex,deviceType)
		If index<>NOT_FOUND
			' Clear a place
			playerArea[index].state=STATE_WAITINGFORPLAYER
			playerArea[index].deviceType=CONTROL_NONE
			playerArea[index].deviceIndex=NOT_INUSE
			numPlaying:-1
		EndIf
	EndMethod

	Method returnNumPlaying:Byte()
		Return numPlaying
	EndMethod

	Method processControls()
	Local loop:Byte

		For loop=0 To MAX_PLAYERS-1
			If playerArea[loop].state=STATE_PLAYERACTIVE
				inUse[loop]=PLAYER_1+loop
				Select playerArea[loop].deviceType
					Case	CONTROL_MOUSE

	mouseControl[playerArea[loop].deviceIndex].whichPlayer=inUse[loop]
					Case	CONTROL_JOYPAD

	joypadControl[playerArea[loop].deviceIndex].whichPlayer=inUse[loop]
				EndSelect
			EndIf
		Next
	EndMethod

	Method detectControl()
	Local loop:Byte
	Local device:Int
	Local eType:Int
	Local item:Int
	Local value:Int

		While ManyMouse_BMInterface_PollEvent(device,item,value,eType)
			MoveMouse 0,0

			Select eType
				Case MANYMOUSE_EVENT_ABSMOTION
'players[whichPlayer].move
Shape(speed,item,Float(value),True)
				Case MANYMOUSE_EVENT_RELMOTION
'players[whichPlayer].move
Shape(speed,item,Float(value),False)
				Case	MANYMOUSE_EVENT_BUTTON
If value=1
	Select item
		Case  0

	  ' LMB

addPlayer(device,CONTROL_MOUSE)
```

```
                                                                                Case  1

' RMB

        removePlayer(device,CONTROL_MOUSE)
                                                                                        EndSelect
                                                                                EndIf
                        EndSelect
                EndWhile
        EndMethod

        Method Do:Byte()
                FlushManyMouse()
                FlushKeys()
                FlushMouse()

                Repeat
                        Cls

                        If KeyHit(KEY_ESCAPE)
                                Return False
                        Else
                                If KeyHit(KEY_RETURN)
                                        If numPlaying>0
                                                processControls()
                                                Return True
                                        EndIf
                                EndIf
                        EndIf

                        UpdateWorld
                        RenderWorld

                        BeginMax2D()
                        Display()

                        detectControl()
                        EndMax2D()

                        Flip
                Forever
        EndMethod
EndType

Type TShape
        Field image:TImage
        Field which:Byte
        Field xPos:Float
        Field yPos:Float
        Field moveAngle:Float
        Field moveSpeed:Float
        Field distance:Float
        Field animAngle:Float
        Field animSpeed:Float
        Field state:Byte
        Field timeToSplit:Float       ' Used with Enemy 3
        Field scale:Float                       ' Again used with enemy 3
        Field damage:Float
        Field isDead:Byte
EndType

Type TPlanet
        Const TEXT_AMOUNTTOCOLLECT:String    =       "AMOUNT TO COLLECT"
        Const TEXT_STAGE:String                      =       "STAGE"
        Const STAGE_COMPLETE:String                  =       "STAGE COMPLETE"
        Const AMMO_STANDARDAMOUNT:Int        =       16
        Const HEALTH_YSIZE:Int                       =       16
        Const MAX_HEALTH:Float                       =       100.0
        Const PLANET_RADIUS:Float                    =       82.0
        Const MAX_BADDIES:Byte                       =       6
        Const NEXTBADDIE_ATSTEP:Int                  =       4
        Const MAX_SHAPESTOCOLLECT:Int        =       255

        Const STATE_DONOTHING:Byte                   =       0
        Const STATE_ALLOWNEWSHAPE:Byte               =       1
        Const STATE_REDUCESPEED:Byte         =       2
        Const STATE_FASTSPEED:Byte                   =       3
        Const STATE_ALLOWNEWSHAPE2:Byte              =       4
        Const STATE_CHANGEMOVEANGLE:Byte     =       5

        Field screenWidth:Int
        Field screenHeight:Int
        Field halfSW:Int
        Field halfSH:Int
        Field loadData:TLoadData[]
        Field numPlaying:Byte
        Field MAX_PLANETSIZE:Float
        Field stageCompleteScale:Float
        Field stageCompleteAlpha:Float

        Field planetMesh:TMesh                       =       Null
```

```
Field skySphere:TMesh                       =       Null
Field camera:TCamera                        =       Null
Field light:TLight                                  =       Null

'Field planetImage:TImage                   =       Null

Field shapeList:TList                       =       Null

Field xPos:Float
Field yPos:Float
Field zPos:Float
Field xAngle:Float
Field yAngle:Float
Field zAngle:Float
Field skyYAngle:Float
Field isAlive:Byte

Field stage:Int
Field maxBaddie:Int

Field timeForShape:Float
Field resetTimeForShape:Float
Field numShapesToAdd:Int
Field freezeTime:Float
Field globalMoveSpeed:Float

Field goodShapesToCollect:Int

Field planetDamageX:Int
Field planetDamageY:Int
Field planetDamage:Float

Function Create:TPlanet(ld:TLoadData[])
Local t:TPlanet

        t=New TPlanet
        If t<>Null
                t.shapeList=CreateList()
                If t.shapeList<>Null
                        t.loadData=ld
                        t.planetMesh=CopyMesh(t.loadData[MESH_PLANET3].mesh)
                        HideEntity t.planetMesh

                        t.skySphere=CreateSphere(8)
                        EntityTexture t.skySphere,t.loadData[TEXTURE_STARFIELD1].texture
                        HideEntity t.skySphere
                        ScaleEntity t.skySphere,-12.0,-12.0,-12.0
                        display.MeshCentre(t.skySphere)

                        t.camera=CreateCamera()
                        HideEntity t.camera

                        t.light=CreateLight()
                        ' Check for errors later
                        't.planetImage=t.loadData[IMAGE_PLANET].image
                Else
                        t=Null
                EndIf
        EndIf

        Return t
EndFunction

Method finish()
        HideEntity skysphere
        HideEntity planetMesh
        HideEntity camera
EndMethod

Method initialise(sW:Int,sH:Int,nP:Byte)
        screenWidth=sW
        screenHeight=sH
        halfSW=screenWidth Shr 1
        halfSH=screenHeight Shr 1
        numPlaying=nP

        xPos=0.0
        yPos=0.0
        zPos=0.0

        isAlive=True

        stage=13
        globalMoveSpeed=1.0
        freezeTime=0.0

        MAX_PLANETSIZE=Float(Min(screenWidth,screenHeight))/4.0

        planetDamage=MAX_HEALTH

        planetDamageX=(screenWidth-Int(MAX_HEALTH)) Shr 1
```

```
		planetDamageY=screenHeight-8-HEALTH_YSIZE

		xAngle=0.0
		yAngle=0.0
		zAngle=0.0

		resetTimeForShape=300.0
		numShapesToAdd=numPlaying*8

		' Position the planet and camera
		CameraViewport camera,0,0,screenWidth,screenHeight
		PositionEntity planetMesh,0.0,0.0,0.0
		PositionEntity skySphere,0.0,0.0,0.0

		RotateEntity planetMesh,xAngle,yAngle,zAngle
		PositionEntity camera,0.0,0.0,-5.0

		ShowEntity planetMesh
		ShowEntity skySphere
		ShowEntity camera

		stageCompleteAlpha=0.0
		stageCompleteScale=0.0
		newStage()
	EndMethod

	Method newStage()
		stage:+1

		timeForShape=0.0

		goodShapesToCollect=numPlaying+2+((stage-1)*3)
		If goodShapesToCollect>=MAX_SHAPESTOCOLLECT
			goodShapesToCollect=MAX_SHAPESTOCOLLECT
		EndIf

		numShapesToAdd:+(numPlaying*3)
		If numShapesToAdd>=128
			numShapesToAdd=128
		EndIf

		maxBaddie=Byte(stage/NEXTBADDIE_ATSTEP)
		If maxBaddie<0
			maxBaddie=1
		Else
			If maxBaddie>MAX_BADDIES
				maxBaddie=MAX_BADDIES
			EndIf
		EndIf

		ClearList shapeList
	EndMethod

	Method Display3D()
	Local s:TShape

		PositionEntity skySphere,xPos,yPos,zPos
		RotateEntity skySphere,0.0,skyYAngle,0.0

		RotateEntity planetMesh,xAngle,yAngle,zAngle
		PositionEntity planetMesh,xPos,yPos,zPos
	EndMethod

	Method returnIsAlive:Byte()
		If isAlive=True
			Return True
		Else
			Return False
		EndIf
	EndMethod

	Method Display2D()
	Local s:TShape
	Local y:Int
	Local sx:Float
	Local sy:Float

		For s=EachIn shapeList
			SetDrawingCommands(1.0,s.animAngle,s.scale)
			DrawImage s.image,s.xPos,s.yPos
		Next

		SetDrawingCommands()
		SetImageFont loadData[FONT_GAMESTATUS].font
		y=screenHeight-TextHeight(COLLECT_PAD)
		DrawText Right$(COLLECT_PAD+goodShapesToCollect,Len(COLLECT_PAD)),screenWidth-
TextWidth(COLLECT_PAD),y
		DrawText TEXT_AMOUNTTOCOLLECT,screenWidth-TextWidth(TEXT_AMOUNTTOCOLLECT),y-
TextHeight(COLLECT_PAD)

		y=screenHeight-TextHeight(STAGE_PAD)
```

```
		DrawText Right$(STAGE_PAD+stage,Len(STAGE_PAD)),0,y
		DrawText TEXT_STAGE,0,y-TextHeight(STAGE_PAD)

		' Draw the health bar
		SetDrawingCommands()
		DrawRect planetDamageX-4,planetDamageY-4,MAX_HEALTH+8,HEALTH_YSIZE+8
		SetColor 0,0,0
		DrawRect planetDamageX-2,planetDamageY-2,MAX_HEALTH+4,HEALTH_YSIZE+4

		If planetDamage>=75.0
			SetColor 0,255,0
		Else
			If planetDamage>=50.0
				SetColor 255,255,0
			Else
				If planetDamage>=25.0
					SetColor 128,128,0
				Else
					SetColor 255,0,0
				EndIf
			EndIf
		EndIf

		DrawRect planetDamageX,planetDamageY,planetDamage,HEALTH_YSIZE
		DrawText planetDamage,200,0

		If stageCompleteAlpha>0.0
			SetDrawingCommands(stageCompleteAlpha,0.0,stageCompleteScale)
			SetImageFont loadData[FONT_SCORE].font
			sx=Float(TextWidth(STAGE_COMPLETE))*stageCompleteScale
			sy=Float(TextHeight(STAGE_COMPLETE))*stageCompleteScale
			DrawText STAGE_COMPLETE,(screenWidth-sx)/2.0,screenHeight-64
		EndIf
	EndMethod

	Method calcXYPos(image:TImage,pScale:Float,angle:Float,pxPos:Float Var,pyPos:Float Var)
	Local xSize:Float
	Local ySize:Float

		pxPos=(Cos(angle)*PLANET_RADIUS)+halfSW
		pyPos=(Sin(angle)*PLANET_RADIUS)+halfSH
	EndMethod

	Method updateAmountToCollect(amount:Int,xPos:Float,yPos:Float)
		goodShapesToCollect:+amount
		If goodShapesToCollect<=0
			stageCompleteAlpha=1.0
			stageCompleteScale=1.0
			newStage()
		Else
			If goodShapesToCollect>=MAX_SHAPESTOCOLLECT
				goodShapesToCollect=MAX_SHAPESTOCOLLECT
			EndIf
		EndIf

		If amount<0
			sparks.AddSpark(xPos,yPos,50,4,4,255,0,255,1.0,8.0,0.5,1.0,0.2,0.25)
		Else
			sparks.AddSpark(xPos,yPos,50,4,4,128,128,128,1.0,8.0,0.5,1.0,0.2,0.25)
		EndIf
	EndMethod

	Method updatePlanetDamage(amount:Float,hitPlayer:Byte,xPos:Float,yPos:Float)
		planetDamage:-amount
		If planetDamage<=0.0
			isAlive=False
		Else
			If planetDamage>=MAX_HEALTH
				planetDamage=MAX_HEALTH
			EndIf
		EndIf

		If hitPlayer=False
			sparks.AddSpark(xPos,yPos,50,4,4,64,64,128,1.0,8.0,0.5,1.0,0.2,0.25)
		Else
			sparks.AddSpark(xPos,yPos,50,4,4,128,64,32,1.0,8.0,0.5,1.0,0.2,0.25)
		EndIf
	EndMethod

	Method Process(speed:Float)
	Local s:TShape
	Local scaleDir:Int
	Local numColl:Int
	Local loop:Int
	Local entity:TEntity
	Local entityNum:Int
	Local playerIndex:Byte
	Local coll:Object[]
	Local player:TPlayers
	Local f:TFiring
```

```
		' Animate the planet
		yAngle:+speed*0.1
		skyYAngle:+speed*0.075

		If timeForShape<=0.0 Or CountList(shapeList)=0
			addShapes()
			timeForShape=resetTimeForShape
		EndIf

		If stageCompleteAlpha>0.0
			stageCompleteScale:+0.01*speed
			stageCompleteAlpha:-0.01*speed
		EndIf

		For s=EachIn shapeList
			If s.isDead=True Or s.distance<=PLANET_RADIUS
				If s.distance<=PLANET_RADIUS
					updatePlanetDamage((2.0*s.damage),False,s.xPos,s.yPos)
				EndIf

				ListRemove shapeList,s
			Else
				s.distance:-s.moveSpeed*speed*globalMoveSpeed
				s.animAngle=wrapF(s.animAngle+
(speed*s.animSpeed*globalMoveSpeed),0.0,359.0)

				s.xPos=(Cos(s.moveAngle)*s.distance)+Float(halfSW)
				s.yPos=(Sin(s.moveAngle)*s.distance)+Float(halfSH)

				' Now we check to see what this has collided with
				' First, check against player
				coll=CollideImage(s.image,s.xPos,s.yPos,0,COLLISION_LAYER_1,0)
				If coll<>Null
					If coll.Length>0
						s.isDead=True

						player=TPlayers(coll[0])
						If player<>Null
							Select s.which
								Case	IMAGE_GOODSHAPE

	player.changeScore(25)

	updateAmountToCollect(-1,s.xPos,s.yPos)
								Case	IMAGE_INCREASEAMMO

	player.changeAmmo(AMMO_STANDARDAMOUNT)
								Case	IMAGE_INCREASEMOVEMENTSPEED

	player.changeMoveSpeed(0.025)
								Case	IMAGE_INCREASESCORE

	player.changeScore(100)
								Case	IMAGE_MEGABOMB
' To
do
								Case	IMAGE_TIMEFREEZE
'
Time freeze

	freezeTime=500.0
								Case	IMAGE_BADSHAPE1

	player.killPlayer()
								Case
	IMAGE_BADSHAPE2,IMAGE_BADSHAPE3,IMAGE_BADSHAPE4,IMAGE_BADSHAPE5,IMAGE_BADSHAPE6

	updatePlanetDamage(s.damage,True,s.xPos,s.yPos)
							EndSelect
						EndIf
					EndIf

					coll=Null
				EndIf

				' Now we check again player firing
				If s.isDead=False
					coll=CollideImage(s.image,s.xPos,s.yPos,0,COLLISION_LAYER_2,0)
					If coll<>Null
						If coll.Length>0
							s.isDead=True

							f=TFiring(coll[0])
							If f<>Null
								f.isDead=True
								Select s.which
									Case	IMAGE_GOODSHAPE

	' Its bad!

	updateAmountToCollect(2,f.xPos,f.yPos)
```

```
        f.player.changeScore(-15)
                                                                Case    IMAGE_BADSHAPE1
        updateAmountToCollect(-4,f.xPos,f.yPos)
        f.player.changeScore(50)
                                                                Case
        IMAGE_BADSHAPE2,IMAGE_BADSHAPE3,IMAGE_BADSHAPE4,IMAGE_BADSHAPE5,IMAGE_BADSHAPE6
        f.player.changeScore((s.which-IMAGE_BADSHAPE1)*50)
        updateAmountToCollect(-2,f.xPos,f.yPos)
                                                        EndSelect
                                                EndIf
                                        EndIf
                                EndIf
                        EndIf
                        If s.scale<1.0
                                s.scale=Min(s.scale+(0.0075*speed),1.0)
                        EndIf
                        Select s.state
                                Case STATE_ALLOWNEWSHAPE,STATE_ALLOWNEWSHAPE2
                                                                        If s.timeToSplit<=0.0
                                                                                If
s.state=STATE_ALLOWNEWSHAPE
        addSplitShapes(s,-0.075,IMAGE_BADSHAPE3)
        addSplitShapes(s,0.075,IMAGE_BADSHAPE3)
        s.timeToSplit=Rnd(800.0,1000.0)
                                                                                Else
        addSplitShapes(s,-0.1,IMAGE_BADSHAPE1)
        addSplitShapes(s,0.0,IMAGE_GOODSHAPE)
        addSplitShapes(s,0.1,IMAGE_BADSHAPE1)
        s.timeToSplit=Rnd(100.0,Float(Max(screenWidth,screenHeight))/2.0)
                                                                                EndIf
                                                                        Else
                                                                                s.timeToSplit:-
speed
                                                                        EndIf
                                Case    STATE_REDUCESPEED
                                                                        If s.moveSpeed>0.0
                                                                                s.moveSpeed:-
0.0005*speed
                                                                                If s.moveSpeed<=0.0
        s.timeToSplit=Rnd(50.0,150.0)
                                                                                EndIf
                                                                        Else
                                                                                s.timeToSplit:-
speed
                                                                                If
s.timeToSplit<=0.0
        s.state=STATE_FASTSPEED
                                                                                EndIf
                                                                        EndIf
                                Case    STATE_FASTSPEED
                                                                        s.moveSpeed=2.0
                                                                        s.state=STATE_DONOTHING
                                Case    STATE_CHANGEMOVEANGLE
        s.moveAngle=wrapF(s.moveAngle+(s.timeToSplit*speed*globalMoveSpeed),0.0,359.0)
                        EndSelect
                EndIf
        Next
        If freezeTime<=0.0
                timeForShape:-speed
                If globalMoveSpeed<1.0
                        globalMoveSpeed:+speed*0.1
                        If globalMoveSpeed>=1.0
                                globalMoveSpeed=1.0
                        EndIf
                EndIf
        Else
                freezeTime:-speed
                If globalMoveSpeed>0.0
                        globalMoveSpeed:-speed*0.1
                        If globalMoveSpeed<=0.0
                                globalMoveSpeed=0.0
                        EndIf
                EndIf
```

```
	EndIf
EndMethod

Method addSplitShapes(s2:TShape=Null,sAngle:Float=0.0,uImage:Byte)
Local s:TShape

	s=New TShape
	If s<>Null
		s.which=uImage
		s.image=loadData[s.which].image
		s.moveAngle=s2.moveAngle
		s.which=s2.which
		s.xPos=s2.xPos
		s.yPos=s2.yPos
		s.moveSpeed=s2.moveSpeed-0.1
		s.animAngle=s2.animAngle
		s.animSpeed=s2.animSpeed
		s.damage=s2.damage*2.0
		s.distance=s2.distance

		If uImage=IMAGE_BADSHAPE3
			s.scale=0.0
			s.state=STATE_CHANGEMOVEANGLE
			s.timeToSplit=Rnd(0.1,0.4)*sAngle
		Else
			s.scale=1.0
			s.state=STATE_CHANGEMOVEANGLE
			s.timeToSplit=sAngle
		EndIf

		ListAddLast shapeList,s

		s=Null
	EndIf

EndMethod

Method addShapes()
Local loop:Int
Local which:Byte
Local s:TShape
Local r:Byte
Local stp:Float
Local angle:Float

	stp=360.0/numShapesToAdd
	angle=Rnd(0.0,stp*numShapesToAdd)

	For loop=1 To numShapesToAdd
		Select loop
			Case	1
					' Add a good image
					which=IMAGE_GOODSHAPE
			Case	2
					' Add a bad shape
					which=IMAGE_BADSHAPE1

			Default
					Select Rand(1,15)
						Case	1,2,3,4
								' Do nothing
								which=0
						Case	5,6,7
								' Do bonus

which=Rand(IMAGE_INCREASEAMMO,IMAGE_TIMEFREEZE)
						Default
								' Do a bad shape

which=Rand(IMAGE_BADSHAPE1,IMAGE_BADSHAPE1+maxBaddie)
					EndSelect
		EndSelect

		If which>0
			s=New TShape
			If s<>Null
				s.which=which
				s.moveAngle=angle 'Rnd(0.0,359.0)
				s.timeToSplit=0.0
				s.state=STATE_DONOTHING

				Select s.which
					Case	IMAGE_GOODSHAPE

s.moveSpeed=Rnd(0.3,0.5)

s.animAngle=Rnd(0.0,359.0)

s.animspeed=Rnd(0.1,0.3)
                                                                        s.damage=0
					Case IMAGE_BADSHAPE1
```

```
        s.moveSpeed=Rnd(0.1,0.2)
        s.animAngle=Rnd(0.0,359.0)
        s.animspeed=Rnd(0.05,0.075)
                                                                    s.damage=0.75
                                        Case IMAGE_BADSHAPE2
        s.moveSpeed=Rnd(0.075,0.1)
        s.animAngle=Rnd(0.0,359.0)
        s.animspeed=Rnd(0.05,0.075)
                                                                    s.damage=0.5
                                        Case IMAGE_BADSHAPE3
        s.moveSpeed=Rnd(0.2,0.4)
        s.animAngle=Rnd(0.0,359.0)
        s.animspeed=Rnd(0.3,0.5)
                                                                    s.damage=0.75
        s.state=STATE_ALLOWNEWSHAPE
        s.timeToSplit=Rnd(200.0,250.0)
                                        Case IMAGE_BADSHAPE4
        s.moveSpeed=0.45+Rnd(-0.15,0.15)
        s.timeToSplit=Rnd(60.0,80.0)
        s.animAngle=wrapF(s.moveAngle+90.0,0.0,359.0)
                                                                    s.animspeed=0.0
                                                                    s.damage=0.8
        s.state=STATE_REDUCESPEED
                                        Case IMAGE_BADSHAPE5
        s.moveSpeed=0.5+Rnd(-0.3,0.4)
        s.timeToSplit=Rnd(100.0,130.0)
        s.animAngle=wrapF(s.moveAngle+180.0,0.0,359.0)
                                                                    s.animspeed=0.0
                                                                    s.damage=0.05
        s.state=STATE_ALLOWNEWSHAPE2
                                        Case IMAGE_BADSHAPE6
        s.moveSpeed=Rnd(0.3,0.7)
                                                                    s.timeToSplit=Rnd(-
0.3,0.3)
        s.animAngle=Rnd(0.0,359.0)
        s.animspeed=Rnd(0.05,0.075)
                                                                    s.damage=0.5
        s.state=STATE_CHANGEMOVEANGLE
                                        Default
        s.moveSpeed=Rnd(0.05,0.6)
        s.animAngle=Rnd(0.0,359.0)
        s.animSpeed=Rnd(0.2,0.6)
                                                                    s.damage=0.0
                                EndSelect

                                s.scale=1.0
                                s.image=loadData[s.which].image
                                s.distance=Float(Min(screenWidth,screenHeight))+Rnd(0.0,16.0)
                                s.xPos=(Cos(s.moveAngle)*s.distance)+Float(halfSW)
                                s.yPos=(Sin(s.moveAngle)*s.distance)+Float(halfSH)

                                s.isDead=False

                                ListAddLast shapeList,s

                                s=Null
                                angle:+stp
                        EndIf
                EndIf
        Next
```