创建一个主类

public class Main extends Sprite

在构造函数中监听舞台的初始化

if (stage) init();

else addEventListener(Event.ADDED\_TO\_STAGE, init);

在init函数中创建Context3D

创建成功会有一个回调事件，在事件中创建dGame3DSystem，创建一个游戏类Game，最后再创建一个timer用于游戏类的主循环

dGame3DSystem.CreateContext3D( stage , function( context3D:Object ):void

{

m\_pdGame3D = new dGame3DSystem( context3D );

m\_pdGame3D.SetScreenSize( stage.stageWidth , stage.stageHeight );

m\_pGame = new Game( m\_pdGame3D , stage );

var pTimer:dTimer = new dTimer();

pTimer.Create( 0 , 0 , function( p:dTimer , nRepeat:int ):void

{

if ( m\_pdGame3D )

{

EnterFrameMove();

}

} );

} );

在EnterFrameMove函数中调用Present函数

public function EnterFrameMove():void

{

m\_pGame.EnterFrameMove();

m\_pdGame3D.Present();

}

游戏类中可以初始化摄像机位置，创建场景，创建模型，创建人物。

关于摄像机的函数有

设置眼睛坐标

public function SetCameraEye( vEye:dVector3 ):void

设置目标点坐标

public function SetCameraLookat( vLookat:dVector3 ):void

设置上方向坐标

public function SetCameraUpVec( vUpVec:dVector3 ):void

获得眼睛坐标

public function GetCameraEye():dVector3

获得目标点坐标

public function GetCameraLookat():dVector3

获得上方向坐标

public function GetCameraUpVec():dVector3

获得摄像机方向

public function GetCameraDir():dVector3

获得摄像机方向与上方向的叉积

public function GetCameraCross():dVector3

设置最近裁剪面

public function SetCameraNearPlane( fNear:Number ):void

设置最远裁剪面

public function SetCameraFarPlane( fFar:Number ):void

获得最近裁剪面

public function GetCameraNearPlane():Number

获得最远裁剪面

public function GetCameraFarPlane():Number

设置摄像机眼睛绕目标点横向旋转角度

public function SetCameraRotationH( angle:Number ):void

设置摄像机眼睛绕目标点纵向旋转角度

public function SetCameraRotationV( angle:Number ):void

获得摄像机眼睛绕目标点横向旋转角度

public function GetCameraRotationH():Number

获得摄像机眼睛绕目标点纵向旋转角度

public function GetCameraRotationV():Number

设置摄像机眼睛与目标点的距离

public function SetCameraRotationRadio( length:Number ):void

获得摄像机眼睛与目标点的距离

public function GetCameraRotationRadio():Number

设置摄像机眼睛与目标点距离的最小,最大范围

public function SetCameraRotationRadioLimit( fNear:Number , fFar:Number ):void

设置摄像机向前移动(沿目标点方向)

public function SetCameraMoveForword( speed:Number ):void

设置摄像机向后移动

public function SetCameraMoveBack( speed:Number ):void

减少摄像机眼睛与目标点的距离

public function SetCameraMoveNear( speed:Number ):void

增加摄像机眼睛与目标点的距离

public function SetCameraMoveFar( speed:Number ):void

设置摄像机向左移动,(沿目标点方向与上方向的叉积方向)

public function SetCameraMoveLeft( speed:Number ):void

设置摄像机向右移动

public function SetCameraMoveRight( speed:Number ):void

设置摄像机目标点与眼睛的横向和纵向角度

public function SetCameraRotationLookat( angleH:Number , angleV:Number ):void

设置摄像机宽高比

public function SetCameraAspect( f:Number ):void

获得摄像机视矩阵

public function GetCameraView():dMatrix

获得摄像机投影矩阵

public function GetCameraProj():dMatrix

获得摄像机视投影相乘矩阵

public function GetCameraViewProj():dMatrix

设置摄像机正交投影宽高

public function SetCameraOrthoSize( width:Number , height:Number ):void

获得摄像机正交投影宽

public function GetCameraOrthoWidth():Number

获得摄像机正交投影高

public function GetCameraOrthoHeight():Number

设置摄像机透视类型,0为透视,1为正交

public function SetCameraPerspectiveType( nType:int ):void

获得摄像机透视类型

public function GetCameraPespectiveType():int

设置摄像机是否与地面碰撞

public function SetCameraCollection( bColl:Boolean ):void

获得摄像机是否与地面碰撞

public function GetCameraCollection():Boolean

场景相关函数有:

创建场景

public function CreateScene( width:int , height:int ):void

保存场景

public function SaveScene():ByteArray

读取场景

public function LoadSceneFromBin( data:ByteArray , onLoadComplate:Function , nFlag:int = 0 ):void

从文件中读取场景

public function LoadSceneFromFile( strFileName:String , onLoadComplate:Function , nFlag:int = 0 , onLoadProgress:Function = null ):void

获得场景文件名

public function GetSceneFileName():String

设置天空盒贴图文件

public function SetSkyboxTextureFileName( strFileName:String ):void

更新场景

public function UpdateScene( nFlag:int = 0 ):void

设置场景是否有碰撞

public function SetShowCanReach( bShow:Boolean ):void

创建模型

public function CreateMeshObj( strFileName:String ):int

创建特效

public function CreateEffectObj( strFileName:String , bPlayEndAutoDelete:Boolean = false ):int

创建人物

public function CreateCharacter( strFileName:String = null , pColorTransform:dColorTransform = null ):int

创建水面

public function CreateOcean( vPos:dVector3 ):int

考贝创建

public function CreateObjCopy( id:int ):int

删除物体

public function DeleteRenderObj( id:int ):void

设置物体是否显示

public function SetObjShow( id:int , bShow:int ):void

获得物体是否显示

public function GetObjShow( id:int ):int

设置物体坐标

public function SetObjPos( id:int , vPos:dVector3 ):void

获得物体坐标

public function GetObjPos( id:int ):dVector3

设置物体高度偏移

public function SetObjYOffset( id:int , fY:Number ):void

获得物体高度偏移

public function GetObjYOffset( id:int ):Number

设置物体缩放

public function SetObjSca( id:int , vSca:dVector3 ):void

获得物体缩放

public function GetObjSca( id:int ):dVector3

设置物体旋转

public function SetObjRot( id:int , vRot:dVector4 ):void

设置物体Y轴旋转

public function SetObjRotY( id:int , angle:Number ):void

获得物体旋转

public function GetObjRot( id:int ):dVector4

获得物体Y轴旋转

public function GetObjRotY( id:int ):Number

设置物体2维方向

public function SetObjDir2( id:int , x:Number , z:Number ):void

获得物体2维方向

public function GetObjDir2( id:int ):dVector2

获得物体类型

public function GetObjType( id:int ):int

获得物体文件名

public function GetObjFileName( id:int ):String

获得物体列表

public function GetObjList( nObjType:int ):Vector.<int>

获得物体绑定盒

public function GetObjBoundingBox( id:int ):dBoundingBox

获得物体AABB

public function GetObjAABB( id:int ):dBoundingBox

获得物体世界矩阵

public function GetObjWorldMatrix( id:int ):dMatrix

设置物体是否接收鼠标

public function SetObjHandleMouse( id:int , bHandle:Boolean ):void

获得物体是否接收鼠标

public function GetObjHandleMouse( id:int ):Boolean

给定点获得场景地形高度

public function GetSceneHeight( x:Number , z:Number ):Number

获得场景地形大小

public function GetSceneSizeX():int

获得场景地形大小

public function GetSceneSizeZ():int

重设场景大小

public function ResizeScene( newWidth:int , newHeight:int , onComplateFun:Function ):void

获得全局光照

public function GetGlobalLightDir():dVector3

设置全局光照

public function SetGlobalLightDir( vDir:dVector3 ):void

获得场景文件列表

public function GetSceneObjFileNameList():Vector.<String>

获得场景文件中物体列表

public function GetSceneStaticCharacter():Vector.<dSceneStaticCharacter>

场景与直线碰撞

public function CheckCollectionRay( vPos:dVector3 , vDir:dVector3 , vPosOut:dVector3 = null , nObjType:int = RENDEROBJ\_TYPE\_ALL ):int

场景与鼠标碰撞

public function CheckCollectionMousePt( x:int , y:int , nWindowWidth:int , nWindowHeight:int , vPosOut:dVector3 = null , nObjType:int = RENDEROBJ\_TYPE\_ALL ):int

设置是否显示地形

public function SetShowTerrain( bShow:Boolean ):void

获得是否显示地形

public function isShowTerrain():Boolean

角色相关函数有:

获得当前鼠标选中的角色

public function GetMousePassCharacterID():int

角色添加装备模型

public function CharacterAddPartMesh( id:int , strPartName:String , strFileName:String ,

strBoundingBonePartName:String = null , strBoundingBoneCharactorName:String = null , pColorTransform:dColorTransform = null ):void

角色添加动作

public function CharacterAddAnimationKey( id:int , strKeyName:String , strFileName:String , bCanMove:int = 1 , bAddToHorse:Boolean = false , onLoadComplate:Function = null ):void

设置角色动作属性

public function CharacterSetAnimationKeyDeclare( id:int , strKeyName:String , nStartTime:int , nLoopStartTime:int , nLoopEndTime:int ):void

设置物体在场景切换时不被删除

public function SetObjNoDelete( id:int , bNoDelete:Boolean ):void

设置人物坐骑

public function CharacterSetHorse( id:int , strHorseFileName:String , strHorseBoneName:String = "EQ-Horse" , strChatarctorBoneName:String = "EQ-Ride" ):void

播放人物动作

public function CharacterSetCurrentKey( id:int , strKeyName:String ):void

获得当前人物播放动作

public function CharacterGetCurrentKey( id:int ):String

获得人物动作最大播放时间

public function CharacterGetKeyMaxTime( id:int , strAniName:String ):int

设置人物移动和站立里的默认动作

public function CharacterSetRunAniName( id:int , strRunAniName:String , strStandAniName:String ):void

人物向给定方向移动

public function CharacterMoveDir( id:int , vDir:dVector3 ):void

人物向给定目标点移动

public function CharacterMoveTarget( id:int , vTarget:dVector3 , bSearchPath:Boolean = false , bCheckCollection:Boolean = false , moveEndFun:Function = null , fIgnoreLength:Number = 0.0 ):int

人物停止移动

public function CharacterStopMove( id:int ):void

设置人物移动速度

public function CharacterSetMoveSpeed( id:int , fSpeed:Number ):void

获得人物移动速度

public function CharacterGetMoveSpeed( id:int ):Number

获得人物是否正在移动中

public function CharacterIsRunning( id:int ):Boolean

设置人物名称

public function CharacterSetName( id:int , strName:String ):void

获得人物名称

public function CharacterGetName( id:int ):String

添加人物头上的数字

public function CharacterAddBNumber( id:int , list:Vector.<int> , nPlayType:int = BNUNBER\_PLAY\_JUMP ):void

添加特效

public function CharacterAddEffect( id:int , strEffectName:String , strHitEffectName:String , vStartPos:dVector3 , nBoneHero:int , strStartBoneName:String , nStartTime:int = 0 , fFlySpeed:Number = 10.0 ):void

设置人物2维方向

public function CharacterSetObjDir2( id:int , nTargetID:int ):void

获得人物骨骼坐标

public function CharacterGetBonePos( id:int , strBoneName:String ):dVector3