

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

Департамент программной инженерии

Контрольное домашнее задание
по дисциплине
«Программирование»

Тема работы: Обработка данных из файла

Выполнил студент группы БПИ 151

Абрамов А.М.

Преподаватель: Подбельский Вадим Валериевич

Содержание

1	Титульная страница	1
2	Пространства имен	2
2.1	Пространство имен WinFormAnimation2D	2
3	Классы	2
3.1	Класс ActionState	2
3.1.1	Подробное описание	2
3.2	Класс CameraDevice	2
3.2.1	Подробное описание	2
3.3	Класс DrawConfig	2
3.3.1	Подробное описание	2
3.3.2	Данные класса	2
3.4	Класс Entity	3
3.4.1	Подробное описание	3
3.4.2	Методы	3
3.5	Класс Geometry	3
3.5.1	Подробное описание	3
3.6	Интерфейс ITransformState	3
3.6.1	Подробное описание	3
3.7	Класс MeshDraw	3
3.7.1	Подробное описание	3
3.7.2	Конструктор(ы)	3
3.7.3	Методы	4
3.8	Класс MouseState	4
3.8.1	Подробное описание	4
3.9	Класс Renderer	4
3.9.1	Подробное описание	4
3.9.2	Методы	4
	Алфавитный указатель	5

1 Титульная страница

This program is a demo of skeletal animation in C#. Feed it collada files with animation and bone data. In blender spread the mesh on X-Y plane (so Z is normal vector) Also make it in the X>0, Y>0 quadrant. When importing a collada file remember to modify it by hand. You should change Z_UP to Y_UP. (I think this is only necessary while we work with Grpahics object, when we go to opengl this should fix itself)

2 Пространства имен

2.1 Пространство имен WinFormAnimation2D

Классы

- class ActionState
- class CameraDevice
- class DrawConfig
- class Entity
- class Geometry
- interface ITransformState
- class MeshDraw
- class MouseState
- class Renderer

3 Классы

3.1 Класс ActionState

Базовые классы:BaseForEventDriven.

3.1.1 Подробное описание

This class knows what argumets to pass to NodeInterpolator

3.2 Класс CameraDevice

Открытые члены

- Vector3 ConvertScreen2WorldCoordinates (Point screen_coords)
- Matrix4 MatrixToOpenGL ()

3.2.1 Подробное описание

Maintains camera abstraction.

3.3 Класс DrawConfig

Открытые атрибуты

- bool EnableTexture2D = false

3.3.1 Подробное описание

This class will be passed to the Entity (стр. 3)'s GetSettings() so it make the scene look best.

3.3.2 Данные класса

3.3.2.1 bool EnableTexture2D = false

OpenGL settings here is a template: public bool _enable = false;

3.4 Класс Entity

Открытые члены

- void RecursiveCalculateVertexTransform (Node nd, Matrix4x4 current)
- void RenderModel (DrawConfig settings)

3.4.1 Подробное описание

Represents the currently loaded object. We will have lots of these.

3.4.2 Методы

3.4.2.1 void RecursiveCalculateVertexTransform (Node nd, Matrix4x4 current)

bind tells the original delta in global coord, so we can find current delta

3.5 Класс Geometry

Открытые члены

- Geometry (IList< Mesh > scene_meshes, Node nd, BoneNode armature)

3.5.1 Подробное описание

Stores info on extra geometry of the entity.

3.6 Интерфейс ITransformState

Производные классы: CameraDrawing2D и CameraFreeFly3D.

3.6.1 Подробное описание

Implement this when class allows local matrix transforms. (Entity (стр. 3), Camera)

3.7 Класс MeshDraw

Открытые члены

- MeshDraw (Mesh mesh, IList< Material > materials)
- void RenderVBO ()

3.7.1 Подробное описание

Mesh rendering using VBOs.

Based on http://www.opentk.com/files/T08_VBO.cs

3.7.2 Конструктор(ы)

3.7.2.1 MeshDraw (Mesh mesh, IList< Material > materials)

Uploads the data to the GPU.

3.7.3 Методы

3.7.3.1 void RenderVBO ()

Render mesh from GPU memory. The pipeline is restored afterwards.

3.8 Класс MouseState

Открытые члены

- void RecordMouseClicked (MouseEventArgs e)
- void RecordMouseMove (MouseEventArgs e)

Открытые атрибуты

- Point ClickPos
- Point CurrentPos
- readonly int HorizHysteresis = 4
- OpenTK.Vector3 InnerWorldClickPos
- OpenTK.Vector3 InnerWorldPos

3.8.1 Подробное описание

Simple class to store mouse status data. Monitor mouse status (delta, position, click_position, etc.)

3.9 Класс Renderer

Открытые члены

- void DrawAxis3D ()

3.9.1 Подробное описание

Class to control openGL settings and do the actual drawing. All openGL calls will be here.

3.9.2 Методы

3.9.2.1 void DrawAxis3D ()

Important points to remember: Set normals. Must be clock wise vertex draw order The x-axis is accross the screen, so the Z-axis triangle must have component along X: +1 since look at looks towards the center, we need to offset it a bit to see the Z axis.

Предметный указатель

ActionState, 2

CameraDevice, 2

DrawAxis3D

 WinFormAnimation2D::Renderer, 4

DrawConfig, 2

EnableTexture2D

 WinFormAnimation2D::DrawConfig, 2

Entity, 3

Geometry, 3

ITransformState, 3

MeshDraw, 3

 WinFormAnimation2D::MeshDraw, 3

MouseState, 4

RecursiveCalculateVertexTransform

 WinFormAnimation2D::Entity, 3

RenderVBO

 WinFormAnimation2D::MeshDraw, 4

Renderer, 4

WinFormAnimation2D::DrawConfig

 EnableTexture2D, 2

WinFormAnimation2D::Entity

 RecursiveCalculateVertexTransform, 3

WinFormAnimation2D::MeshDraw

 MeshDraw, 3

 RenderVBO, 4

WinFormAnimation2D::Renderer

 DrawAxis3D, 4

WinFormAnimation2D, 2