

УТВЕРЖДЕНО

Заведующий кафедрой «Управление
разработкой программного обеспечения»
_____ / Авдошин С.М./
« ____ » _____ 2011 г.

**ПРОГРАММА ВИЗУАЛИЗАЦИИ ОПЕРАЦИЙ НАД
КВАТЕРНИОНАМИ НА ПЛАТФОРМЕ WINDOWS PHONE 7**

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

Инв. № подп.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Руководитель работы

_____ / Гринкруг Е.М./
« ____ » _____ 2011 г.

Исполнитель: студент группы 171ПИ

_____ / Дубов М.С. /
« ____ » _____ 2011 г.

Национальный исследовательский университет – Высшая школа экономики
Факультет бизнес-информатики, отделение программной инженерии

УТВЕРЖДЕНО

**ПРОГРАММА ВИЗУАЛИЗАЦИИ ОПЕРАЦИЙ НАД
КВАТЕРНИОНАМИ НА ПЛАТФОРМЕ WINDOWS PHONE 7**

Пояснительная записка

Листов 23

Инв. № подп.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Содержание

1.	Введение.....	3
1.1.	Общие сведения о программе.....	3
1.2.	Основания для разработки.....	3
1.2.1.	Цель разработки.....	3
1.2.2.	Заказчик проекта.....	3
2.	Назначение разработки.....	3
2.1.	Назначение программы.....	3
2.2.	Область применения программы.....	3
3.	Технические характеристики.....	4
3.1.	Постановка задачи.....	4
3.2.	Используемые алгоритмы.....	4
3.3.	Метод организации входных и выходных данных.....	4
3.3.1.	Входные данные.....	4
3.3.2.	Выходные данные.....	4
3.4.	Состав технических и программных средств.....	5
3.4.1.	Технические средства.....	5
3.4.2.	Программные средства.....	5
4.	Ожидаемые технико-экономические показатели.....	5
4.1.	Библиотека DCL.Maths.....	5
4.2.	Библиотека DCL.Phone.Xna.....	5
4.3.	Приложение Planets.....	6
5.	Источники, используемые при разработке.....	6
6.	Приложение А. Описание и функциональное назначение классов и структур.....	7
6.1.	Библиотека DCL.Maths.....	7
6.2.	Библиотека DCL.Phone.Xna.....	7
6.3.	Приложение Planets.....	7
7.	Приложение Б. Описание и функциональное назначение методов, полей и свойств.....	8
7.1.	Библиотека DCL.Maths.....	8
7.1.1.	Структура Quaternion.....	8
7.1.2.	Структура Fraction.....	9
7.1.3.	Структура Angle.....	10
7.1.4.	Статический класс Common.....	11
7.2.	Библиотека DCL.Phone.Xna.....	12
7.2.1.	Абстрактный класс Shape.....	12
7.2.2.	Класс Ellipse – наследник Shape.....	14
7.2.3.	Класс Circle – наследник Ellipse.....	14
7.2.4.	Класс Ellipsoid – наследник Shape.....	15
7.2.5.	Класс Sphere – наследник Ellipse.....	15
7.2.6.	Класс Dot – наследник Shape.....	16
7.2.7.	Перечисление DrawMode.....	16
7.2.8.	Класс PivotGame – наследник Microsoft.Xna.Framework.Game.....	16
7.2.9.	Класс PivotGameItem.....	18
7.2.10.	Класс SelectionChangedEventArgs – наследник EventArgs.....	19
7.2.11.	Структура DrawingArea.....	20
7.2.12.	Перечисление ProjectionType.....	20
7.3.	Приложение Planets.....	20
7.3.1.	Класс Planets.....	20
7.3.2.	Класс FlatRingSector.....	22
7.3.2.	Класс FrameRateCounter.....	23

1. Введение

1.1. Общие сведения о программе

Наименование программы: программа визуализации операции над кватернионами на платформе Windows Phone 7.

Программный комплекс состоит из трех компонент: библиотек DCL.Maths и DCL.Phone.Xna, а также приложения Planets для платформы Windows Phone 7.

1.2. Основания для разработки

1.2.1. Цель разработки

Разработка осуществляется соответственно следующим документам:

- «Список тем курсовых работ студентов отделения программной инженерии факультета бизнес-информатики – 1 курс».
- Техническое задание

1.2.2. Заказчик проекта

Заказчиком проекта является ГУ-ВШЭ, отделение программной инженерии факультета бизнес-информатики, кафедра «Управление разработкой программного обеспечения».

2. Назначение разработки

2.1. Назначение программы

Программа Planets предназначена для визуализации операции поворота трехмерных моделей через кватернионы на платформе Windows Phone 7.

Библиотека DCL.Maths предоставляет средства для работы с кватернионами, обыкновенными дробями и градусными величинами, а также ряд общих математических функций.

Библиотека DCL.Phone.Xna предоставляет классы, отвечающие за построение и визуализацию трехмерных моделей, их преобразования и повороты посредством операций над кватернионами. Кроме того, библиотека содержит средства для построения пользовательских интерфейсов на платформе Windows Phone 7.

2.2. Область применения программы

- Визуализация поворота 3D-моделей с помощью кватернионов на примере естественных космических тел;
- Учебный процесс;
- Обе библиотеки могут быть применены при разработке других программных средств.

3. Технические характеристики

3.1. Постановка задачи

Программа визуализации операций над кватернионами должна осуществлять операцию поворота трехмерных моделей относительно заданной оси на заданный угол посредством использования кватернионов, а также осуществлять комбинацию нескольких заданных кватернионами поворотов. Программа должна давать пользователю устройства на платформе Windows Phone 7 возможность задавать ось и угол поворота путем использования различных манипуляций на экране устройства («жестов») [2].

Одной из подзадач является создание структуры данных для представления кватернионов, обладающей более широким функционалом, чем существующая структура *Microsoft.Xna.Framework.Quaternion*.

Трехмерные модели, используемые для визуализации операции поворота, должны строиться программно во время выполнения. Предполагается создание легко расширяемой библиотеки трехмерных моделей, пригодной для многократного использования.

3.2. Используемые алгоритмы

Поворот трехмерных моделей относительно заданной оси на заданный угол осуществляется поворотно алгоритмом, описанным в [1] и [3]. На основе данных об оси и угле поворота специальным образом вычисляется так называемый кватернион поворота; после этого каждая вершина, представленная трехмерным вектором, умножается справа на кватернион поворота (операция требует 12 умножений и 8 сложений); затем результат умножается слева на кватернион, обратный к кватерниону поворота (операция требует 16 умножений и 12 сложений).

Комбинирование нескольких поворотов осуществляется путем перемножения соответствующих кватернионов поворота в порядке, описанном в [1].

Построение трехмерных моделей фигур (таких как сфера, окружность) осуществляется параметрически на основе данных о типе фигуры и степени детализации модели. Время построения модели, время вычисления результатов операций поворота и перемещения, а также объем памяти для хранения модели находятся в квадратичной зависимости от параметра детализации.

3.3. Метод организации входных и выходных данных

3.3.1. Входные данные

Каждый объект, представляющий кватернион, задается четверкой вещественных чисел одинарной точности (типа *System.Single*).

При инициализации объекта, отвечающего за построение и вывод на экран трехмерной модели, на входе необходимо указать тип, местоположение модели, степень ее детализации, опционально – текстуру модели.

Программа определяет ось и угол поворота модели на основе анализа манипуляций пользователя на экране устройства. Направление движения пальца определяет ось поворота, в то время как длина расстояния, пройденного пальцем по экрану, определяет величину угла.

3.3.2. Выходные данные

Выходными данными являются изображения на экране устройства построенных программно трехмерных моделей в перспективной проекции. В изображениях моделей

отражены произведенные над ними преобразования (в том числе повороты посредством кватернионов).

3.4. Состав технических и программных средств

3.4.1. Технические средства

Основным средством распространения программного комплекса является интернет-магазин приложений Microsoft Marketplace for Mobile. На сегодняшний день это единственный способ распространения мобильных приложений для платформы Windows Phone 7. Альтернативным способом распространения программного комплекса является распространение на носителях типа CD-ROM, что продиктовано возможностью использования исходного кода в других программных продуктах.

3.4.2. Программные средства

Программный комплекс реализуется средствами среды .Net Framework ® с использованием библиотек, предоставляемых Windows Phone 7 Developer Tools.

Установка программного комплекса на устройство конечного пользователя осуществляется через Zune Software либо непосредственно на самом устройстве, либо на ПК под управлением ОС Windows 7 и подключенным к нему мобильным устройством.

4. Ожидаемые технико-экономические показатели

4.1. Библиотека *DCL.Maths*

Структура *DCL.Maths.Quaternion* предоставляет программисту весь основной функционал, необходимый при работе с кватернионами, а также реализует ряд дополнительных функций по сравнению с теми, что доступны в структуре *Microsoft.Xna.Framework.Quaternion* (например, округление текущего кватерниона до определенной точности).

Кроме того, в некоторых аспектах структура *DCL.Maths.Quaternion* более тесно интегрирована с другими членами пространства имен *Microsoft.Xna.Framework*, чем структура *Microsoft.Xna.Framework.Quaternion* (так, реализовано перемножение объектов типа *DCL.Maths.Quaternion* и *Microsoft.Xna.Framework.Vector3*, обеспечивающее выигрыш в скорости в сравнении с перемножением двух объектов типа *Quaternion*).

4.2. Библиотека *DCL.Phone.Xna*

Совокупность абстрактного класса *DCL.Phone.Xna.Shape* и производных от него классов предоставляют программисту простой и легко расширяемый инструмент программного построения трехмерных моделей различных фигур из графических примитивов. Возможность расширения библиотеки фигур наглядно продемонстрирована классом *RingSector* приложения *Planets*. Все классы, производные от *Shape*, тесно интегрированы с классами библиотеки *DCL.Maths*; в частности, в базовом классе реализованы операции поворота всей модели посредством кватернионов.

Класс *DCL.Phone.Xna.PivotGame* является расширением стандартного базового класса XNA-приложений *Microsoft.Xna.Framework.Game*, предоставляющим программисту возможность строить свое приложение на основе панорамного пользовательского интерфейса. Класс представляет собой менеджер экранов и имитирует в XNA-приложениях стандартный интерфейс *Pivot*, используемый во многих Silverlight-приложениях для Windows Phone (описан в [2] и [4]), стирая таким образом границу

между XNA- и Silverlight- приложениями в глазах пользователя. Свободных аналогов такому менеджеру экранов не существует.

4.3. Приложение Planets

Приложение Planets демонстрирует возможности библиотек *DCL.Maths* и *DCL.Phone.Xna*. Программа предназначена для публикации на Microsoft Marketplace for Mobile и последующего ее использования в учебном процессе (например, на уроках астрономии). Аналогов данной программы на Marketplace на сегодняшний день не существует.

5. Источники, используемые при разработке

- [1] Кантор И.Л., Солодовников А.С. Гиперкомплексные числа – М. : Наука, 1973. – 114 стр.
- [2] Petzold C. Programming Windows Phone 7 – Microsoft Press, 2010 – 997 стр.
- [3] Petzold C. 3D Programming for Windows – Microsoft Press, 2007 – 448 стр.
- [4] Microsoft Windows Phone 7 developer center: <http://create.msdn.com>

6. Приложение А. Описание и функциональное назначение классов и структур

6.1. Библиотека *DCL.Maths*

Имя класса/структуры	Описание
Quaternion	Структура, реализующая основные операции над кватернионами
Fraction	Структура, реализующая основные операции над дробями
Angle	Структура, реализующая основные операции над градусными величинами
Common	Статический класс, содержащий ряд общих математических методов и расширяющий функционал стандартных классов .Net Framework

6.2. Библиотека *DCL.Phone.Xna*

Имя класса/структуры	Описание
Shape	Абстрактный класс, предоставляющий основной функционал по отрисовке трехмерных моделей и их преобразованиям (в том числе поворотам через кватернионы)
Ellipse	Класс, наследующий <i>Shape</i> , отвечающий за построение эллипса
Circle	Класс, наследующий <i>Ellipse</i> , отвечающий за построение окружности
Ellipsoid	Класс, наследующий <i>Shape</i> , отвечающий за построение эллипсоида
Sphere	Класс, наследующий <i>Ellipsoid</i> , отвечающий за построение сферы
Dot	Класс, наследующий <i>Dot</i> , отвечающий за построение точки
PivotGame	Класс, наследующий <i>Game</i> , отвечающий за построение панорамного пользовательского интерфейса
PivotGameItem	Класс, отвечающий за представление отдельной страницы приложения
SelectionChangedEventArgs	Класс, представляющий аргументы для обработчика события <i>SelectionChanged</i>
DrawingArea	Структура, представляющая область отрисовки 3D-сцены

6.3. Приложение *Planets*

Имя класса/структуры	Описание
Planets	Граничный класс приложения, отвечающий за взаимодействие с пользователем
RingSector	Класс, наследующий <i>Shape</i> , отвечающий за построение кольцевого сектора
FrameRateCounter	Компонент тестирования, выводящий на экран текущее значение FPS

7. Приложение Б. Описание и функциональное назначение методов, полей и свойств

7.1. Библиотека *DCL.Maths*

7.1.1. Структура *Quaternion*

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
Re	public	Single		Вещественная часть
X	public	Single		Коэффициент при i
Y	public	Single		Коэффициент при j
Z	public	Single		Коэффициент при k
i	public static readonly	Quaternion		Мнимая единица i
j	public static readonly	Quaternion		Мнимая единица j
k	public static readonly	Quaternion		Мнимая единица k
Свойства				
VectorPart	public	Vector3		Векторная часть
Abs	public	Single		Модуль
Конструкторы				
Quaternion	public		Single	Конструктор приведения типов
Quaternion	public		Vector3	Конструктор приведения типов
Quaternion	public		Single, Single, Single, Single	Конструктор общего вида
Методы				
Conjugate	public	Quaternion		Сопряженный кватернион
Reciprocal	public	Quaternion		Обратный кватернион
Normalize	public	Quaternion		Нормализация
Round	public	Quaternion	Int32	Округление до определенного количества знаков после запятой
Approximate	public	Quaternion	Single	Округление до определенной точности
Rotation Quaternion	public static	Quaternion	Quaternion, Single	Вычисление кватерниона поворота для заданных оси и угла в радианах
Rotate	public static	Quaternion	Quaternion, Quaternion, Single	Поворот векторного кватерниона на заданные ось и угол в радианах
Rotate	public static	Quaternion	Quaternion, Quaternion, Quaternion, Single	Поворот векторного кватерниона на заданные ось (со смещением) и угол в радианах
Rotate	public static	Quaternion	Quaternion,	Комбинирование двух

Composition			Quaternion, Single Quaternion, Single	поворотов
Slerp	public static	Quaternion	Quaternion, Quaternion, Single	Сферическая интерполяция (SLERP) между двумя кватернионами
Lerp	public static	Quaternion	Quaternion, Quaternion, Single	Линейная интерполяция между двумя кватернионами
LSlerp	public static	Quaternion	Quaternion, Quaternion, Single	Приближенная сферическая интерполяция (получается из линейной по специальному алгоритму; процедура является гораздо менее ресурсоемкой, чем Slerp)
ToString	public override	String		Строковое представление кватерниона
ToString	public override	String	Int32	Строковое представление с определенной точностью
Equals	public override	Boolean	Object	Проверка на равенство
GetHashCode	public override	Int32		Хэш-код объекта
Приведение типов				
Quaternion	implicit	Quaternion	Single	
Quaternion	implicit	Quaternion	Vector3	
Vector3	explicit	Vector3	Quaternion	

7.1.2. Структура Fraction

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
Empty	public static readonly	Fraction		«Пустая» дробь
num	private	Int64		Числитель
den	private	UInt32		Знаменатель
Свойства				
Numerator	public	Int64		Числитель
Denominator	public	UInt32		Знаменатель
Конструкторы				
Fraction	public		Int64, UInt32	Конструктор общего вида
Методы				
Cancel	public	Fraction		Сокращение дроби
Reverse	public	Fraction		Переворачивание дроби
ReduceToDenominator	public	Fraction	UInt32	Приведение дроби к знаменателю
ToMixed	public	void	out Int64,	Получение из дроби

Number			out Fraction	смешанного числа
FromMixedNumber	public static	Fraction	Int64, Fraction	Создание дроби из смешанного числа
Parse	public static	Fraction	String	Создание дроби путем анализа строки
TryParse	public static	Boolean	String, out Fraction	Создание дроби путем анализа строки, не вызывающее исключений
IsEmpty	public static	Boolean	Fraction	Проверка дроби на равенство «пустой»
ToString	public override	String		Строковое представление дроби
Equals	public override	Boolean	Object	Проверка на равенство
GetHashCode	public override	Int32		Хэш-код объекта
CompareTo	public	Int32	Object	Реализация интерфейса IComparable
Приведение типов				
Fraction	implicit	Fraction	Int64	
Fraction	explicit	Fraction	Double	
Double	explicit	Double	Fraction	

7.1.3. Структура Angle

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
PI	public static readonly	Angle		Величина 180°
Eps	public static readonly	Angle		Величина 0°0'1''
val	internal	Fraction		Обыкновенная дробь, которая хранит значение и обертку над которой предоставляет класс Angle
Свойства				
Sign	public	Int32		Знак градусной величины
Degrees	public	UInt32		Градусы
Minutes	public	UInt32		Минуты
Seconds	public	UInt32		Секунды
Конструкторы				
Angle	public		UInt32, Int32	Конструктор общего вида
Angle	public		UInt32, UInt32, UInt32, Int32	Конструктор общего вида
Методы				
Parse	public static	Angle	String	Создание градусной величины путем анализа строки

TryParse	public static	Boolean	String, out Angle	Создание градусной величины путем анализа строки, не вызывающее исключений
Wrap	public static	Angle	Angle	Приведение угла к отрезку $[-180^\circ, 180^\circ]$
ToString	public override	String		Строковое представление градусной величины
ToString	public override	String	Boolean, Boolean	Строковое представление без вывода нулевых значений минут или секунд
Equals	public override	Boolean	Object	Проверка на равенство
GetHashCode	public override	Int32		Хэш-код объекта
CompareTo	public	Int32	Object	Реализация интерфейса IComparable
Приведение типов				
Double	explicit	Double	Angle	
Fraction	explicit	Fraction	Angle	
Angle	explicit	Angle	Double	
Angle	explicit	Angle	Fraction	

7.1.4. Статический класс Common

Имя	Модификаторы	Тип	Аргументы	Описание
Методы				
Round	public static	Double	Double, Int32	Округляет число до заданного числа знаков после запятой
Round	public static	Double	Double	Округляет число до целого по математическим правилам
Cbrt	public static	Double	Double	Вычисляет кубический корень из числа
Factorial	public static	Double	UInt32	Вычисляет факториал числа
DoubleFactorial	public static	Double	UInt32	Вычисляет двойной факториал числа
BinomialCoefficient	public static	Double	UInt32, UInt32	Вычисляет C_n^k
LCM	public static	UInt32	UInt32, UInt32	Вычисляет НОК двух чисел
GCD	public static	UInt32	UInt32, UInt32	Вычисляет НОД двух чисел
FractionalPart	public static	Double	Double	Возвращает дробную часть числа
FractionalPart	public static	Decimal	Decimal	Возвращает дробную часть числа
FractionalPartLength	public static	Int32	Double	Возвращает длину дробной части числа

Fractional PartLength	public static	Int32	Decimal	Возвращает длину дробной части числа
IsPrime	public static	Boolean	UInt32	Проверяет, является ли число простым
Swap	public static	void	ref Int32, ref Int32	Меняет местами значения двух переменных
Swap	public static	void	ref UInt32, ref UInt32	Меняет местами значения двух переменных

7.2. Библиотека *DCL.Phone.Xna*

7.2.1. Абстрактный класс *Shape*

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
current Vertices	protected	VertexPosition Normal Texture[]		Массив вершин модели (в том числе после преобразований)
startVertices	protected	VertexPosition Normal Texture[]		Массив вершин модели до всех преобразований
triangle Indices	protected	Int16[]		Массив индексов вершин модели для отрисовки ее по треугольникам
lineIndices	protected	Int16[]		Массив индексов вершин модели для отрисовки ее контура по линиям
startCenter	protected	Vector3		Центр фигуры до всех преобразований
qDefault Rotation	private	Quaternion		Кватернион стандартного поворота
qDefault RotationConj	private	Quaternion		Кватернион, сопряженный кватерниону стандартного поворота
Свойства				
Center	public	Vector3		Центр фигуры
Texture	public	Texture2D		Текстура модели
Graphics Device	public	Graphics Device		Графическое устройство, на которое отрисовывается модель
Default Rotation	protected	Quaternion		Стандартное вращение, заданное кватернионом
Конструкторы				
Shape	protected			Конструктор общего вида
Методы				
Reset	public	void		Возвращает модель к ее первоначальному состоянию

Rotate	public	void	Quaternion, Single	Поворачивает модель вокруг некоторой оси на некоторый угол
Rotate	public	void	Quaternion, Quaternion, Single	Поворачивает модель вокруг некоторой смещенной оси на некоторый угол
Rotate Composition	public	void	Quaternion, Single, Quaternion, Single	Осуществляет комбинирование двух поворотов модели
Rotate Composition	public	void	Quaternion, Single, Quaternion, Single, Quaternion, Single	Осуществляет комбинирование трех поворотов модели
RotateDefault	public	void		Осуществляет стандартный поворот модели
SetDefault Rotation	public	void	Quaternion, Single	Задает стандартный поворот модели
SetDefault Rotation	public	void	Quaternion, Single, Quaternion, Single	Задает стандартный поворот модели как комбинацию двух поворотов
SetDefault Rotation	public	void	Quaternion, Single, Quaternion, Single, Quaternion, Single	Задает стандартный поворот модели как комбинацию трех поворотов
Translate	public	void	Quaternion	Осуществляет перенос модели в пространстве на некоторый вектор
Scaling	public	void	Single	Осуществляет масштабирование модели относительно ее центра
Draw	public	void	BasicEffect, DrawMode	Рисует модель на графическом устройстве
DrawScene	public static	void	BasicEffect, DrawMode, params Shape[]	Осуществляет отрисовку нескольких моделей одновременно
DrawScene	public static	void	BasicEffect, DrawMode, Shape[], params Shape[]	Осуществляет отрисовку нескольких моделей одновременно

7.2.2. Класс *Ellipse* – наследник *Shape*

Имя	Модификаторы	Тип	Аргументы	Описание
Свойства				
Radius	public	Single		Базовый радиус эллипса
Axis	public	Vector3		Ось эллипса
Конструкторы				
Ellipse	public		Vector3, Single, Single, Single, Int32	Конструктор общего вида. Задаются центр модели, базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей.
Ellipse	public		Vector3, Single, Single, Single, Int32, Texture2D	Конструктор общего вида. Задаются центр модели, базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей, текстура модели.
Ellipse	public		Vector3, Single, Single, Single, Int32, Texture2D, Graphics Device	Конструктор общего вида. Задаются центр модели, базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей, текстура модели и стандартное графическое устройство.

7.2.3. Класс *Circle* – наследник *Ellipse*

Имя	Модификаторы	Тип	Аргументы	Описание
Конструкторы				
Circle	public		Vector3, Single, Int32	Конструктор общего вида. Задаются центр модели, степень детализации и радиус окружности.
Circle	public		Vector3, Single, Int32, Texture2D	Конструктор общего вида. Задаются центр модели, степень детализации, радиус окружности и текстура модели.
Circle	public		Vector3, Single, Int32, Texture2D, Graphics Device	Конструктор общего вида. Задаются центр модели, степень детализации и радиус окружности, текстура модели и стандартное графическое устройство.

				устройство.
--	--	--	--	-------------

7.2.4. Класс *Ellipsoid* – наследник *Shape*

Имя	Модификаторы	Тип	Аргументы	Описание
Свойства				
Radius	public	Single		Базовый радиус эллипсоида
Axis	public	Vector3		Ось эллипсоида
Конструкторы				
Ellipsoid	public		Vector3, Single, Single, Single, Single, Int32	Конструктор общего вида. Задаются центр модели, базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей.
Ellipsoid	public		Vector3, Single, Single, Single, Single, Int32, Texture2D	Конструктор общего вида. Задаются центр модели, базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей, текстура модели.
Ellipsoid	public		Vector3, Single, Single, Single, Single, Int32, Texture2D, Graphics Device	Конструктор общего вида. Задаются центр модели, базовый радиус, степень детализации, растяжение эллипса вдоль координатных осей, текстура модели и стандартное графическое устройство.

7.2.5. Класс *Sphere* – наследник *Ellipse*

Имя	Модификаторы	Тип	Аргументы	Описание
Конструкторы				
Sphere	public		Vector3, Single, Int32	Конструктор общего вида. Задаются центр модели, степень детализации и радиус сферы.
Sphere	public		Vector3, Single, Int32, Texture2D	Конструктор общего вида. Задаются центр модели, степень детализации, радиус сферы и текстура модели.
Sphere	public		Vector3, Single, Int32,	Конструктор общего вида. Задаются центр модели,

			Texture2D, Graphics Device	степень детализации и радиус сферы, текстура модели и стандартное графическое устройство.
--	--	--	----------------------------	---

7.2.6. Класс *Dot* – наследник *Shape*

Имя	Модификаторы	Тип	Аргументы	Описание
Конструкторы				
Dot	public		Vector3, Int32	Конструктор общего вида. Задаются положение точки и степень ее «жирности».
Dot	public		Vector3, Int32, Texture2D	Конструктор общего вида. Задаются положение точки, степень ее «жирности» и текстура.
Dot	public		Vector3, Int32, Texture2D, Graphics Device	Конструктор общего вида. Задаются положение точки, степень ее «жирности», текстура и стандартное графическое устройство.

7.2.7. Перечисление *DrawMode*

- Lines – рисуется контур модели;
- Solid – рисуется поверхность модели;
- SolidWithLines – рисуются как поверхность, так и контур модели;
- Textured – рисуется поверхность модели с натянутой на нее текстурой.

7.2.8. Класс *PivotGame* – наследник *Microsoft.Xna.Framework.Game*

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
HeaderFont	protected internal	SpriteFont		Шрифт заголовка страниц приложения
ContentFont	protected internal	SpriteFont		Шрифт текста содержимого приложения
TitleFont	protected internal	SpriteFont		Шрифт заголовка приложения
Selection Changed Event Handler	public	delegate void		Делегат обработчика события SelectionChanged
Selection Changed	public	event Selection Changed		Событие, возбуждаемое при смене страницы

		EventHandler		приложения
delta	private	Int32		Индикатор дистанции, приложенной пальцем при перелистывании страниц
deltaVector	private	Vector2		Векторное представление поля delta
selInd	private	Int32		Индекс выбранной страницы
cameraWidth	private	Single		Степень охвата сцены камерой
zoom	private	Single		Степень удаленности камеры от начала координат по оси Z
SelectedIndex SetFirstTime	private	Boolean		Определяет, меняется ли индекс страницы в первый раз
Switching TabsNow	private	Boolean		Определяет, меняется ли индекс страницы в данный момент
FontLoader	private	Content Manager		Загрузчик шрифтов
CallStack	private	List<Int32>		История переключения страниц
proj	private	ProjectionType		Тип проекции для 3D-сцены
Свойства				
Title	public	string		Заголовок приложения
Background Color	public	Color		Цвет фона страниц приложения
Foreground Color	public	Color		Цвет текста страниц приложения
Scene Background Color	public	Color		Цвет фона области, в которую отрисовывается трехмерная сцена
Scene Background Texture	public	Texture2D		Фоновая текстура области, в которую отрисовывается трехмерная сцена
DrawingArea	public	Rectangle		Область, в которую отрисовывается трехмерная сцена
SplashScreen Image	public	Texture2D		Картинка-заставка на время загрузки приложения
Graphics Device Manager	protected internal	Graphics Device Manager		Менеджер графического устройства
SpriteBatch	protected internal	SpriteBatch		Менеджер спрайтов, отвечающий за отрисовку текстур и текста

BasicEffect	protected	BasicEffect		Эффект, используемый при отрисовке сцены
Projection Type	protected	Projection		Тип проекции, используемой при отрисовке сцены
CameraScale	public	Single		Степень охвата сцены камерой
Zoom	public	Single		Степень удаленности камеры от начала координат по оси Z
SceneCenter Translation	protected	Vector2		Центр сцены
SelectedIndex	public	Int32		Номер текущей страницы приложения
<i>indexer</i>	public	PivotGame Item	Int32	Индикатор, возвращающий страницу приложения по ее номеру
ItemsCount	public	Int32		Количество страниц приложения
Changing Pages	public	Boolean		Определяет, происходит ли в данный момент смена страниц
Touches	protected	Touch Collection		Массив касаний экрана пользователем в каждый момент времени
Конструкторы				
PivotGame	public			Конструктор общего вида
Методы				
AddItem	public	void	PivotGame Item	Добавляет страницу в приложение
DrawString	public	void	SpriteFont, String, Vector2, Color	Отрисовывает строку на заданной позиции
DrawSprite	public	void	Texture2D, Vector2, Color	Отрисовывает текстуру на заданной позиции

7.2.9. Класс *PivotGameItem*

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
DrawingArea	public	DrawingArea		Область рисования сцены
DrawFrame	public	event EventHandler		Событие, возбуждаемое при перерисовке страницы (за исключением сцены)
DrawScene	public	event EventHandler		Событие, возбуждаемое при перерисовке сцены на странице
tinyTexture	private	RenderTarget		Текстура для рисования

		2D		чистого цвета
rectDrawingArea	private	Rectangle		Прямоугольник области рисования
rectBackgr1	private	Rectangle		1-й прямоугольник вне области рисования
rectBackgr2	private	Rectangle		2-й прямоугольник вне области рисования
rectBackgr3	private	Rectangle		3-й прямоугольник вне области рисования
rectBackgr4	private	Rectangle		4-й прямоугольник вне области рисования
pg	private	PivotGame		«Родитель», которому принадлежит страница
headerPosition	internal	Vector2		Позиция заголовка
titlePosition	internal	Vector2		Позиция названия приложения
bkgColor	private	Color		Цвет фона
frgColot	private	Color		Цвет текста
deactivatedColor	private	Color		Цвет заголовка невыбранной страницы
temp	private	Single		Вспомогательная переменная
ind	private	Int32		Вспомогательная переменная
Свойства				
Background Color	public	Color		Цвет фона страницы
Foreground Color	public	Color		Цвет текста страницы
Header	public	String		Заголовок страницы
Parent	public	PivotGame		Ссылка на объект класса приложения, в состав которого входит страницы
Конструкторы				
PivotGame Item	public			Конструктор общего вида

7.2.10. Класс *SelectionChangedEventArgs* – наследник *EventArgs*

Имя	Модификаторы	Тип	Аргументы	Описание
Свойства				
Previous Index	public	Int32		Индекс предыдущей страницы.
Конструкторы				
Selection Changed EventArgs	public		Int32	Конструктор общего вида. Задается индекс предыдущей страницы.

7.2.11. Структура *DrawingArea*

Имя	Модификаторы	Тип	Аргументы	Описание
Свойства				
X	public	Int32		Координата X верхнего левого угла области отрисовки сцены
Y	public	Int32		Координата Y верхнего левого угла области отрисовки сцены
Width	public	Int32		Ширина области отрисовки сцены
Height	public	Int32		Высота области отрисовки сцены
Background Color	public	Color		Цвет фона области отрисовки сцены
Background Texture	public	Texture2D		Фоновая текстура области отрисовки сцены

7.2.12. Перечисление *ProjectionType*

- Perspective – перспективная проекция;
- Orthographic – ортографическая проекция.

7.3. Приложение *Planets*

7.3.1. Класс *Planets*

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
Planet	private	Ellipsoid		Модель планеты
Moon	private	Ellipsoid		Модель основного спутника
Moon2	private	Ellipsoid		Модель 2-го спутника
Moon3	private	Ellipsoid		Модель 3-го спутника
Moon4	private	Ellipsoid		Модель 4-го спутника
Moon5	private	Ellipsoid		Модель 5-го спутника
Moon6	private	Ellipsoid		Модель 6-го спутника
Moon7	private	Ellipsoid		Модель 7-го спутника
SaturnRing	private	FlatRing Sector[]		Модели секторов, из которых состоит кольцо Сатурна
qAxis	private	Quaternion		Ось поворота
qNew	private	Quaternion		Ось поворота после изменения пользователем
Rotation Angle	private	Single		Угол поворота

touches Distance	private	Double		Расстояние между двумя касаниями экрана
randomAngle	private	Random		Генератор случайного угла для начальной инициализации
MoonAxis	private	Vector3		Ось спутника
watcher	private	Geo Coordinate Watcher		Объект, отвечающий за взаимодействие с GPS-приемником устройства
tLocation1	private	String		Строковое представление широты
tLocation2	private	String		Строковое представление долготы
gpsState	private	Int32		Индикатор состояния GPS-приемника
UseGPS	private	Boolean		Определяет, разрешено ли пользователем использование GPS
Settings Loaded	private	Boolean		Определяет, были ли загружены настройки
tinyFont	private	SpriteFont		Шрифт для мелкого текста
smallFont	private	SpriteFont		Шрифт для не крупного текста
IsTrial	private	Boolean		Определяет, какую версию приложения загрузил пользователь
Latitude	private	Single		Текущая широта
Longitude	private	Single		Текущая долгота
ico_	private	Texture2D		Иконки UI
pos_	private	Vector2		Позиции иконок UI
str_	private	String		Строки, используемые в UI
coord_	private	Vector2		Позиции строк, используемых в UI
t_	private	Texture2D		Текстуры моделей
OBLIQUITY	const	Single		Угол поворота моделей на каждом фрейме
Конструкторы				
Planets	public			Стандартный конструктор
Методы				
Initialize	protected override	void		Инициализация приложения
LoadContent	protected override	void		Загрузка ресурсов приложения, построение моделей планет
Update	protected override	void	GameTime	Обработка событий и логика программы на каждом фрейме
Draw	protected override	void	GameTime	Перерисовка экрана на каждом фрейме
Planets_	private	void	Object,	Обработчик события

Selection Changed			EventArgs	SelectionChanged
Reset	private	void		Сброс оси и угла поворота, восстановление исходного состояния моделей
Earth_Init	private	void		Построение моделей земли и луны
InitWatcher	private	void		запуск GPS-сенсора
Update Position Visualisation	private	void		Обновление строкового представления координат и позиции точки на модели земного шара
watcher_Status Changed	private	void	Object, GeoPosition StatusChanged EventArgs	Изменение состояния GPS-сенсора
watcher_Position Changed	private	void	Object, GeoPosition Changed EventArgs <Geo Coordinate>	Изменение данных о позиции, полученных GPS-сенсором
OnGPS Settings Closed	private	void	IAsyncResult	Закрытие диалогового окна настроек GPS
OnBuy Message Closed	private	void	IAsyncResult	Закрытие диалогового окна приобретения полной версии приложения
OnActivated	protected override	void	Object, EventArgs	Активация приложения
On Deactivated	protected override	void	Object, EventArgs	Деактивация приложения
LoadSettings	private	void		Загрузка настроек
SaveSettings	private	void		Сохранение настроек
LoadText Content	private	void		Загрузка текстовых данных
LoadIcons	private	void		Загрузка иконок UI

7.3.2. Класс *FlatRingSector*

Имя	Модификаторы	Тип	Аргументы	Описание
Свойства				
Radius	public	Single		Расстояние от центра до центральной точки кольцевого сектора
Width	public	Single		Ширина кольцевого сектора
Конструкторы				
FlatRing	public		Vector3,	Конструктор общего вида

Sector			Single, Single, Int32, Single	
FlatRing Sector	public		Vector3, Single, Single, Int32, Single, Texture2D	Конструктор общего вида
FlatRing Sector	public		Vector3, Single, Single, Int32, Single, Texture2D, Graphics Device	Конструктор общего вида

7.3.2. Класс *FrameRateCounter*

Имя	Модификаторы	Тип	Аргументы	Описание
Поля				
Content	private	Content Manager		Менеджер контента (загрузчик текстур и шрифтов)
spriteBatch	private	SpriteBatch		Менеджер отрисовки текста и спрайтов
spriteFont	private	SpriteFont		Шрифт компонента
frameRate	private	Int32		Текущее значение FPS
frame Counter	private	Int32		Вспомогательная переменная-счетчик
elapsedTime	private	TimeSpan		Пройденное время
Конструкторы				
FrameRate Counter	Game			Стандартный конструктор
Методы				
LoadContent	protected override	void		Загрузка шрифта
Unload Content	protected override	void		Выгрузка ресурсов из памяти
Update	protected override	void	GameTime	Обработка событий и логика компонента на каждом фрейме
Draw	protected override	void	GameTime	Перерисовка компонента на каждом фрейме