

XNA

Alt Christian, Atzl Caroline,
Gintner Peter, Huemer Sandra

January 16, 2009

Inhaltsverzeichnis

- 1 Einführung
 - Allgemeines
 - Vergleich
- 2 XNA-Framework
 - Allgemeines
 - Schichten
- 3 Vor- und Nachteile
- 4 Erstellen einer Anwendung
 - Grundgerüst
 - Beispiel

Allgemeines zu XNA

- XNA ist ein Framework von Microsoft für die Entwicklung von Computerspielen.
- Schnellerer Entwicklungsprozess soll ermöglicht werden

Ziele der Entwicklung

- Die Entwicklungskosten einzelner Gametitel sollten gesenkt werden.
- Weniger Komplexität bei Handhabung der Hardware
- Kompatibilität von Tools und Middleware
- Fokus auf das Spiel an sich und nicht auf die Technik.

XNA Game Studio Express (GSE)

- Basiert auf Visual C# Express
- Ende August 2006 erste Beta-Version
- Besteht aus wenigen Komponenten
- Erlaubt das visuelle Verbinden von Objekten

XNA unterstützt

- 3D-Formate (*.x, *.fbx)
- 2D-Formate (*.jpg, *.bmp, *.dds, *.png, *.tga)
- Material File Formats (*.fx)
- Audio File Formats / XACT (*.xap)

Vergleich

Unterschied XNA und klassisches .NET-Framework

- XNA-Bibliotheken sind Managed Code
- NET-Framework ist die Grundlage auf dem PC
- Unterschiede liegen bei den vorhandenen Funktionsbibliotheken

Unterschied XNA Visual C#

- ausgetauscht wurden die Projekttypen
- 2 Projekttypen: einer zum Erzeugen einer XNA-Anwendung und ein Typ zum Entwickeln von Klassenbibliotheken
- mit XNA lassen sich visuell ansprechendere Ergebnisse schneller realisieren

Allgemeines zum XNA-Framework

- Das XNA Framework stellt eine Reihe verwalteter APIs zur Verfügung.
- XNA spielt die Rolle eines Mittlers zwischen der Anwendung und der Hardware.
- Das XNA Framework bietet zudem einen Rahmen, auf dem die Handlung eines Spiels aufgesetzt wird.

Schichten

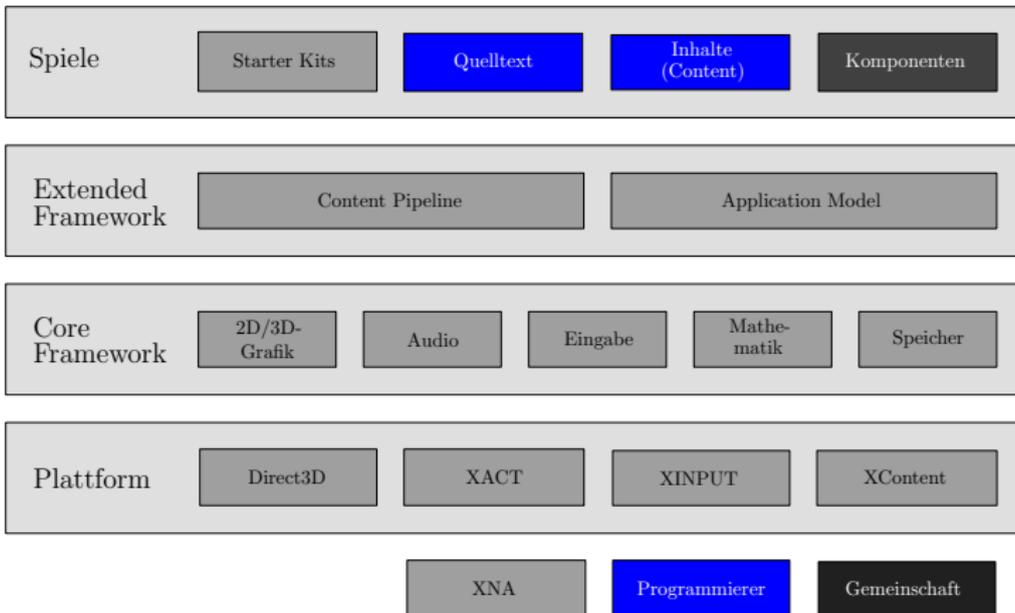
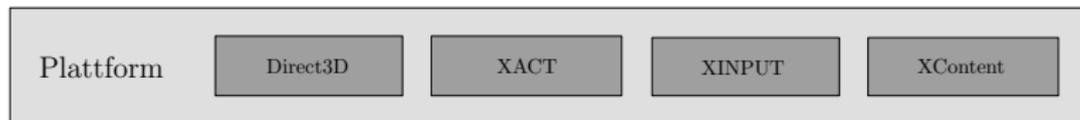


Figure: XNA Frameworkschichten [2]

Schicht: Plattform



- Die Plattform ist das Fundament des gesamten XNA Frameworks.
- Sie beinhaltet zahlreiche APIs.
- Direct3D dient zur Kommunikation mit der Grafikkarte.
- Man kann mit XINPUT auch die Signale eines Xbox 360 Gamepads abfragen.

Schicht: Core Framework



- Innerhalb dieser Schicht sind alle verwalteten Schnittstellen enthalten.
- Das Core Framework kümmert sich um das darstellen von Grafiken, das abspielen von Musik und das ansprechen von Tastatur, Maus, etc.

Core Framework: graphics

- basiert auf Direct3D 9 Schnittstelle
- keine fixed pipeline, sondern programmable pipelines
- Basic Effects

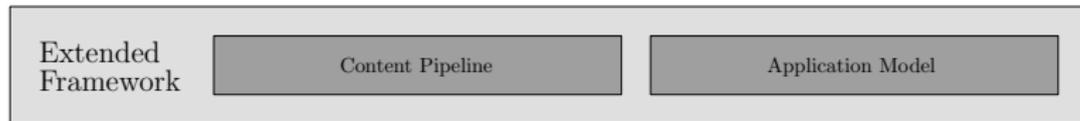
Core Framework: audio

- baut auf XACT auf
- Pakete werden mit XACT Tool erstellt
- ladet XACT Pakete und spricht Audio nur noch über den Namen an
- benötigt keine Initialisierungen oder Puffer

Core Framework: storage

- plattformunabhängige Speichermethode
- System.IO stellt dies schon zur Verfügung
- Xbox problematischer (Festplatte oder andere Speichergeräte)
- unter Windows emuliert

Schicht: Extended Framework



- Das erweiterte Framework baut auf den Standardfunktionalitäten auf und vereinfacht die Entwicklung von Computerspielen.
- Zwei wichtige Komponenten hören auf die Namen Application Model und Content Pipeline.

Application Model

- häufig auftretende Arbeiten werden vermieden
- bietet mitunter die Plattformunabhängigkeit
- Entwickler konzentriert sich nur noch auf die Programmierung

Application Model forts.

- erstellt unter Windows automatisch das Fenster für das Spiel
- implementiert die Game-Schleife
- initialisiert alle Geräte

Content Pipeline

- ist ein objektorientiertes Applikationsmodell
- bietet Dienste zur Verarbeitung von Spieleinhalten
- so eine Art Plug-in-System

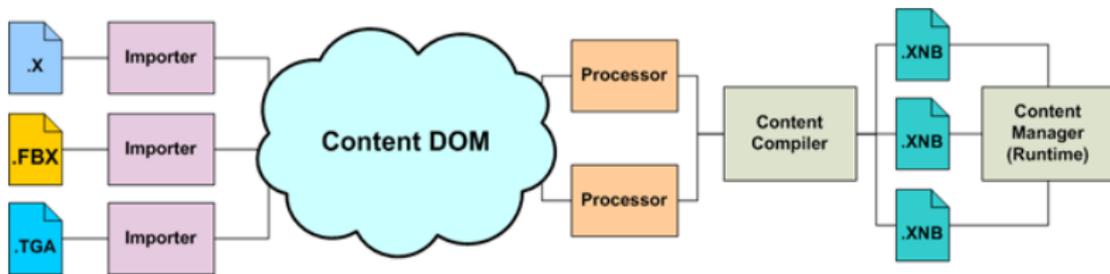
Content Pipeline

- Vorteile:
 - Modelle können in verschiedenen Formaten vorliegen
 - Abhängigkeit zwischen Modell und Spiele Engine wird gelockert
 - System wird einfach und leicht erweiterbar
 - Importer/Prozessor

Content Pipeline: Unterschied

- Importer:
 - Daten von der Festplatte lesen
 - erzeugt XNA Game Studio Content DOM
 - dieses wird dem Prozessor zur Verfügung gestellt
- Prozessor:
 - bezieht Daten aus einem Content DOM
 - erstellt typisiertes Objekt
 - stellt es dem Content Manager zur Verfügung

Content Pipeline: Importer / Prozessor



Schicht: Spiele



- Ein Spiel besteht aus Programmcode, Grafiken und sonstigen Ressourcen.
- Starter Kits beinhalten ein schon (nahezu) fertiges Spiel.
- Von Anfang an steht die eigentliche Programmlogik im Vordergrund.

Vor- und Nachteile

Vorteile

- Das Framework nimmt dem Spieleentwickler Arbeit ab und vereinfacht das Entwickeln von Computerspielen.
- Spiele laufen nicht nur auf einem Windows-PC, sondern auch auf der Xbox 360.
- Einfaches laden und speichern von Modellen, Texturen und Sounds.
- Das Framework nimmt dem Entwickler viele Routineaufgaben ab.

Vor- und Nachteile

Nachteile

- Es sind nur Vorlagen für das Visual C# Express vorhanden.
- Um für die Xbox 360 Entwickeln zu können, muss man bezahlen.
- Keine Unterstützung für 64-bit Systeme.
- DirectX10 wird noch nicht unterstützt.

Grundelemente der Game-Klasse

```
Class MyGame : Game {  
    protected override void Initialize(){  
        base.Initialize();  
    }  
    protected override void Update(GameTime gameTime){  
        //eigentliche Spiellogik  
    }  
    protected override void Draw(GameTime gameTime){  
        //Zeichenlogik  
        base.Draw(gameTime);  
    }  
    protected override void LoadContent(){ }  
    protected override void UnloadContent(){ }  
}
```

- Initialize(), Draw(), Update() müssen überschrieben werden
- Draw() und Update() werden periodisch aufgerufen
- LoadContent(), UnloadContent() sind optional

Beispiel

- Einfaches Model laden
- Model anzeigen
- Model rotieren

Membervariablen

```
public class Game1 : Microsoft.Xna.Framework.Game
{
    GraphicsDeviceManager g;
    ContentManager content;
    Model model;
    BasicEffect effect;
    float modelRotation = 0.0f;
```

- GraphicsDeviceManager
 - Konfiguration der Grafikkarte
- ContentManager
 - Laufzeitkomponente der ContentPipeline
 - Lädt ManagedObjects
 - Verwaltet Lebensdauer geladener Objekte
- Model
 - Speichert ein 3D-Model

Initialisierung

```
public Game1(){  
    g = new GraphicsDeviceManager( this );  
    content = new ContentManager( Services );  
}  
protected override void Initialize(){  
    g.PreferredBackBufferWidth = 600;  
    g.PreferredBackBufferHeight = 600;  
    g.ApplyChanges();  
    g.IsFullScreen = false;  
    Window.Title = "Tiger Rotation";  
    Window.AllowUserResizing = true;  
    base.Initialize();  
}
```

Model laden

```
protected override void LoadContent(){  
    model = content.Load<Model >("tiger");  
    effect = (BasicEffect)model.Meshes[0].Effects[0];  
    effect.View = Matrix.CreateLookAt( new Vector3(0f, 0f, 4f),  
        Vector3.Zero, Vector3.Up );  
    effect.Projection = Matrix.CreatePerspectiveFieldOfView(  
        MathHelper.Pi/4, 1f, 0.01f, 100000.0f );  
}
```

- Matrix.CreateLookAt

- cameraPosition
- cameraTarget
- cameraUpVector

- Matrix.CreatePerspectiveFieldOfView

- fieldOfView
- aspectRatio
- nearPlaneDistance
- float farPlaneDistance

Unload und Update

```
protected override void UnloadContent(){  
    content.Unload();  
}  
protected override void Update( gameTime gameTime ) {  
    base.Update(gameTime);  
}
```

Draw-Methode

```
protected override void Draw( gameTime gameTime ){  
    g.GraphicsDevice.Clear( Color.DarkBlue );  
    effect.World =Matrix.CreateRotationX( modelRotation ) *  
        Matrix.CreateRotationY( modelRotation ) *  
        Matrix.CreateRotationZ( modelRotation += 0.01f );  
    model.Meshes[0].Draw();  
}
```

GameTime

- Zeit seit letztem Update()
- RealTimeClock / FixedStepClock
- isRunningSlowly()
 - zu hoher Zeitverbrauch
 - Game-Klasse regelt FPS/UPS

Quellen

- [1] Primbs D., Schultz H.: Spiele selbst programmieren mit dem XNA Game Studio, c't 17/2008, Heise Verlag, Seite 126-130
- [2] Primbs D.: Spiel ohne Grenzen, iX 3/2007, Heise Verlag, Seite 128-131
- [3] Hochstatter C.: Neue Offenheit, c't 15/2007, Heise Verlag, seite 42
- [4] Miszalok V.: Course 3D_XNA: 3D-Computer Graphics with C# and XNA, http://www.miszalok.de/C_3D_XNA/Index_of_Course.htm
- [5] Walker M.: What is the XNA Framework, <http://blogs.msdn.com/xna/archive/2006/08/25/724607.aspx>
- [6] Konerow J.: Revolution in der Spiele-Welt, Oktober 2006, <http://entwickler.de/zonen/portale/psecom,id,101,online,961,p,0.html>
- [7] Eine Einfhrgung in XNA, Microsoft Corporation, <http://www.microsoft.com/germany/msdn/coding4fun/projekte/xna/einfuel>