DOWNLOADS

Spiele entwickeln und Daten einlesen

Mit diesen Bibliotheken können Sie Spiele programmieren und der BinaryMapper hilft bei der Deserialisierung.

Soll ein Dokument auf der Festplatte gespeichert werden, so muss es serialisiert werden. Aus den Strukturen, die im Speicher gehalten werden, wird also eine Abfolge von Daten erzeugt.

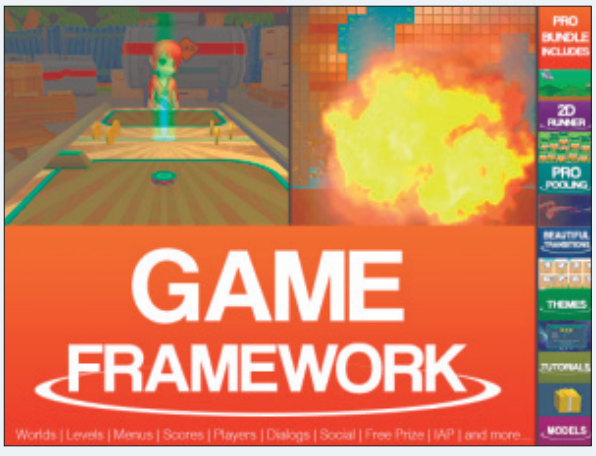
Das ist von der Programmierung her meistens wenig aufwendig. Denn das Programm weiß sehr gut um die Datenformate, die es vorhält, und kann sie dementsprechend einfach auf Platte bannen.

Aber das dicke Ende kommt noch. Sollen die Daten nämlich von der Platte wieder in den Speicher eingelesen werden, sodass das Programm mit ihnen weiterarbeiten kann, wird es knifflig.

Beim Deserialisieren gilt es die Datei zu parsen und die einzelnen Inhalte in die jeweiligen Strukturen zu überführen. Glücklich der, der eine Bibliothek zur Verfügung hat, die das für ihn vollführt.

Bibliotheken für die Spieleentwicklung

Ein Spiel kann in der Herstellung viele Millionen Dollar verschlingen. Doch es geht auch viel günstiger. Mit den Bibliotheken und Tools, die wir im Schwerpunkt zusammengestellt haben, können Sie loslegen. Mit dabei: zentrale Spiele-Engines und Spielerverwaltung.



Das .NET-Betriebssystem Cosmos

Es muss ja nicht immer Windows sein. Nein, auch nicht Linux oder macOS. Mit Cosmos können Sie selbst Betriebssysteme entwickeln, und das auch noch komplett in .NET. Komplett bedeutet hier vom Treiber bis hin zur Anwendung. Dabei lässt sich viel über Betriebssysteme lernen.

BinaryMapper ist eine solche Bibliothek. Sie deserialisiert binäre Daten. So holt folgender Code Daten aus der binären Datei *dnp.bin* in die Struktur *DNPFILE_HEADER_STRUCT*:

```
public class DNPFILE_HEADER_STRUCT
{
  public uint Signature;
  public uint Version;
  public uint SomeOffset;
  public uint NumberOfEntries;
  [ArraySize(nameof(NumberOfEntries))]
  public DNPFILE_STREAM_STRUCT[] MyStructures;
}

public class DNPFILE_STREAM_STRUCT
{
  public uint Flags;
}

var stream = File.OpenRead("dnp.bin");
var streamBinaryMapper = new StreamBinaryMapper();
var header = streamBinaryMapper
  .ReadObject< DNPFILE_HEADER_STRUCT >(stream);
```

BinaryMapper ist für .NET Core 2.0 geschrieben und ist deshalb auch plattformübergreifend einsetzbar. ■

Schwerpunkt

Game Framework
Game Framework für Unity ist ein freies, Community-unterstütztes Framework, dessen Ziel es ist, die Features zu bieten, die die meisten Spiele brauchen. Und das in flexibler Weise und mit einem Minimum an Codierung.

Forge
Forge ist eine Sammlung von Libraries, die entwickelt wurden, um das Schreiben eines Spiels einfacher zu machen. Auf gewisse Weise ist es eine Game Engine. Die Kernbibliothek in Forge ist Forge.Entities, die eine neue ECS-Implementierung (Entity Component System) hat.

C64 Studio
C64 Studio ist eine .NET-basierte IDE. Das Programm ist ausgerichtet auf die Spieleentwicklung. Der interne Assembler verwendet die ACME-Syntax.

behaviac
behaviac ist ein Framework der Spiele-AI-Entwicklung, und es kann auch als schnelles Design-Tool für Spiele-Prototypen genutzt werden.

Entity-Engine
Ein C# Entity Component System Framework zur Verwendung in der Spieleentwicklung. Es ist noch in der Entwicklung. Ein Entity Component System (ECS) ist ein architektonisches Designmuster, das dynamisches Erstellen von Spielobjekten durch die Verwendung von Komponenten erlaubt.

GameInc
Ein Spieleentwicklungsstudio-Management-Game, erstellt mit Unity sowie C# 6 und inspiriert durch Spiele wie Game Dev Story und Game Dev Tycoon.

Ultraviolet
Ultraviolet ist ein plattformübergreifendes .NET-Spieleentwicklungs-Framework, geschrieben in C# und veröffentlicht unter der MIT-Lizenz. Es ist stark inspiriert durch Microsofts XNA Framework. Geschrieben ist die aktuelle Implementierung auf SDL2 und OpenGL, aber ihr Design macht das Re-Implementieren unter Verwendung anderer Technologien einfach.

3D-Parkour-Game
Dieses 3D-Parkour-Spiel wurde über die Unity-Engine entwickelt. Der Entwicklungsprozess enthält die Konstruktion der Spielszenen, der Mauseingabemechanismen sowie der Aktionskontrolle der Charaktere und die Interaktionen zwischen den Charakteren.

Gorgon
Eine modulare Sammlung von Libraries, die für grafische Spieleentwicklung nützlich sind. Gorgon verwendet Direct 3D.

Aktuelle Downloads

Cosmos
Cosmos ist ein Construction Kit für Betriebssysteme. Erstellen Sie Ihr eigenes Betriebssystem unter Verwendung verwalteter Sprachen wie C# oder VB.NET.

ImageMagick
ImageMagick ist eine leistungsstarke Bildbearbeitungsbibliothek und unterstützt über 100 Dateiformate – Subformate nicht eingeschlossen. Mit Magick.NET können Sie ImageMagick in Ihrer C#/VB.NET/.NET-Core-Applikation verwenden, ohne ImageMagick auf Ihrem Rechner installieren zu müssen.

Blazor
Blazor ist ein .NET Web Framework. Sie schreiben Blazor-Apps unter Verwendung von C#/Razor und HTML. Dank WebAssembly läuft die App in jedem Browser.

RepoDb
Eine dynamische ORM .NET Library, die zum Erstellen von Entity-basierten Repository-Klassen während des Zugriffs auf Daten von der Datenbank verwendet wird.

Deipax
Eine schnelle, anpassbare Bibliothek, die sehr einfach Klone von Objekten erzeugt.

VoidMain
VoidMain ist ein Framework zum Erstellen von Befehlszeilenanwendungen, inspiriert von ASP.NET Core. Fast jeder Teil des Frameworks kann erweitert oder ersetzt werden.

ASPNetCoreUnitTesting
Ausführen von Unit Tests auf Diensten, APIs auf N-Layer-Architekturen unter Einsatz von ASP.NET Core, xUnit, Moq et cetera.

FluentMigrator
Fluent Migrator ist ein Migrationsframework für .NET, ähnlich wie Ruby on Rails Migrations. Es bietet eine strukturierte Möglichkeit, das Datenbankschema zu ändern.

XyLang
XyLang ist eine einfachere und freundlichere .NET-Programmiersprache.

ULIS.NET
.NET-Port der Universal Language Intelligence Service JavaScript Library, die eine universelle Sprachunterstützung über das Bing-Übersetzungs-API bietet.

Juhta.NET
Juhta.NET ist ein Open-Source-App-Framework, erstellt mit .NET Core. Juhta.NET soll eine modulare Annäherung für die Anwendungsentwicklung bieten.