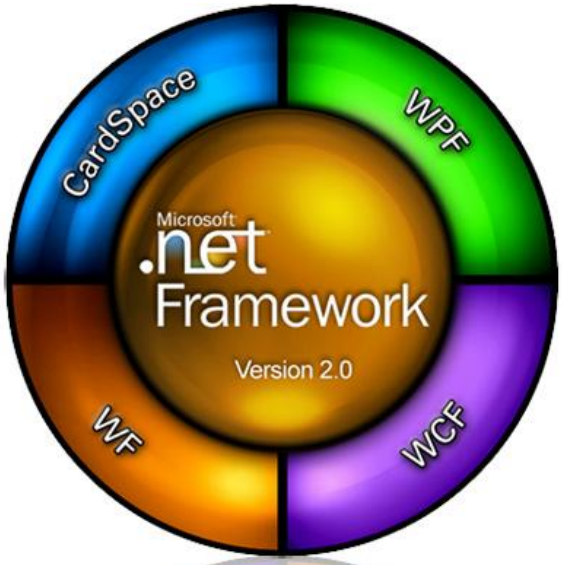

C# Programmierung

Eine Einführung in das .NET Framework

Mi

SDL.NET
Netzwerk
Lokalisierung
Bibliotheken
Registry
Setup-Projekt

Tag 8

Professionelle Entwicklung mit C# und .NET

Eigene Projekte durch Fremdbibliotheken ohne großen Aufwand erweitern und Projekte veröffentlichen

Plattformunabhängigkeit

- Nicht in der von z.B. Qt gewohnten Form
- Möglich durch **Mono** Projekt (bis .NET 2.0 *100%* implementiert, danach *ansatzweise*)
- Mit Visual Studio hat man beste Plattformunabhängigkeit zwischen MS Produkten: Windows, Windows Mobile, Windows Phone, Windows CE, X-Box (360) und weiteren
- **Durch Mono erreicht man eine Vielzahl von Geräten!**

SDL.NET

- Wrapper der in C geschriebenen Multimedia Bibliothek SDL – v.a. für Spiele sehr interessant
- In C# gibt es jeder Menge verfügbarer Game Engines z.B. Irrlicht, ExoEngine, Axiom, uvm.
- SDL ist auf 2D Spiele mit Sprites, Audio, Maus- und Tastatursteuerung ausgelegt
- **Demo** von SDL.NET in einem einfachen 2D Spiel

Mögliches Projekt

Beispiel

01 – Spiele mit SDL.NET

Das Netzwerk verwenden

- Sehr heikel, da man an der Uni große Sicherheitsbedenken hat
- Komplexe Materie – neben den Programmiertechnischen (v.a. asynchronen) Know-How muss man Netzwerkkenntnisse besitzen
- Sehr Fehleranfällig – Viele Jahre Erfahrung im Programmieren (von Socket oder TCP/IP) von Anwendungen notwendig

Beispiel

02 - Netzwerk

Bibliotheken

- Sind wichtig für effektive Entwicklung, **Modularisierung** – halten Code sauber und getrennt
- Unterschied: *Statische* und *Dynamische* Bibliotheken
- Erstellen einer Bibliothek in C# über **neues Projekt**
- Bibliotheken über *Reflection* auslesen
- Mögliche Anwendung: **Add-Ins**!

Lokalisierung

- Im Visual Studio einfache Möglichkeit eingebaut
- Lokalisierung von Datum oder Dezimalzeichen etc über CultureInfo für viele Sprachen implementiert
- Aktuelle Anwendung auf eine *Standardsprache* unabhängig vom Betriebsystem einstellen
- Lokalisieren (der Eigenschaften) von Steuer-elementen **im Designer**

Windows Registry

- Was ist die Windows Registry?
- Managed, OOP Zugriff über .NET mit C# möglich
- Klasse *Registry* gibt Zugriff über Registry Rootkey wie z.B. **LOCAL_MACHINE**, **CURRENT_USER**, …
- Durchführen von Aufgaben mit *RegistryKey*
- Sicherheitsanmerkungen (**Manifest**)

Setup Projekte

- Im Visual Studio als Projekt-Typ vorhanden
- Einfache Einstellungen wie z.B. *Desktop Verknüpfungen*, *Startmenüeinträge* und Erweiterungen wie die *Registrierung von Dateitypen* möglich
- Es müssen nur die Ausgaben von Projekten hinzugefügt werden (Ausgaben sind Binaries wie z.B. dll, exe)
- Die **Setup UI** kann angepasst werden

Beispiel

- Live – Schritt für Schritt
- Zunächst kleines (zusätzliches) Projekt erstellen (Bib.)
- Verwenden einer Klasse der Bibliothek
- Lokalisierung einbinden und ein Setup Projekt dazu erstellen

Fokus aufs Projekt

- Vorstellung der Projektideen – jemand noch **ohne** Idee?
- Wer ist noch ohne Projektpartner – hätte aber gerne einen / würde ein größeres Projekt machen?

➔ Vorstellung der Projektteams

Abschließende Fragen

- Morgen: **Fokus aufs Projekt** – ganzer Tag im CIP-Pool
- Freitag: *Release Candidate* in Projektpräsentation (nur kurz das Programm zeigen – 5 Min) vorstellen
- Deadline für *finale* Projektabgabe: 07.04.2012
- Mit Projektabgabe muss **kleine** (max ~2 Seiten) *Projektdokumentation* abgegeben werden, welche das Projekt (grob) beschreibt