



FBK

Flexibler Browserspielkern

Eine Übersicht

Martin Brenn

- Grundgerüst für die Entwicklung von webbasierten Spielen (Browsergames)
- Plugin-basiert
 - Komplette Spielmechanik ist als Plugin abgelegt
 - Interface basierend um Implementationen auszutauschen (Beispielsweise *IMessageSystem*)
- Konsolenbasierte Anwendung
 - Ausführung als Dienst ist geplant
- OpenSource (AGPL-Lizenz)
 - <http://bitbucket.org/mbrenn>

1. Download (<http://depon.net/downloads/fbk.zip>)
2. Entpacken
3. FBK.exe starten
4. Kennwort für Administrator eingeben

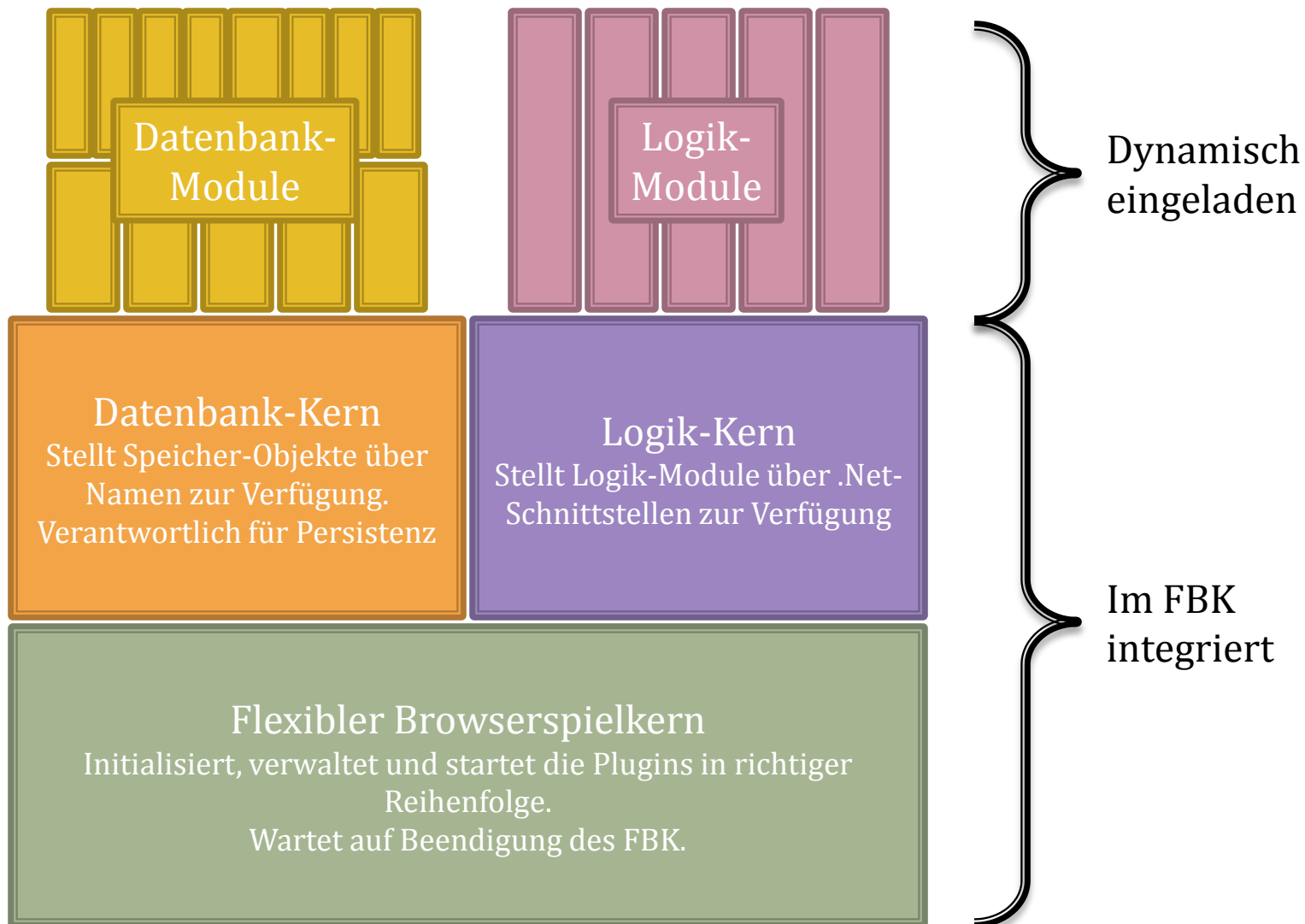
**Keine weitere
Einrichtung
erforderlich!**



- Ich selbst
 - Eigene Browser Spiele und technologisches Interesse
- Interessierte Personen
 - Anfänger
- Intranet-Browsergames
 - Vereine/Clans
 - Arbeit
- Prototypen

- Genutzt von den eigenen Projekten:
 - <http://www.anotherspacegame.de/>
 - Klon bekannter Weltraumbrowserspiele
 - Kontinuierliche Zeit
 - <http://www.jungfrauenspiel.de/>
 - Taktisches Browserspiel im Team
 - Diskrete Zeit (Ticks)
- Die gleiche Engine ermöglicht zwei vollkommen unterschiedliche Spiele.
- Nur wenige Spezialklassen für das Jungfrauenspiel benötigt.

- Sprache: C#
- Framework: .Net 3.5 SP 1
 - Nutzung von LINQ und XLINQ
- Ein Browsergame, ein Prozess:
 - Webserver: Nutzung des HttpListeners (http.sys)
 - Datenbank: Integriert (Speicherhaltung mit regelmäßiger Speicherung)
 - Logik: Thread
- Visual Studio 2008 inkl. MSBuild
Community Tasks



- Definition der Reihenfolge über Abhängigkeiten
- Registrierung an Ereignisse
 - AfterConfigurationLoaded
 - AfterDataStorageLoaded
 - AfterLogicCreated, AfterLogicStarted
 - ...
 - BeforeShutdown
- Erzeugung von
 - Datenbankcontainern
 - Logikmodulen



Standard-Plugins

Der FBK unterstützt zwei Zeitmodelle:

- Echtzeit: Ereignis bei bestimmter Uhrzeit
 - Einmalig, mehrfach oder ewig
 - Für getickte Spiele
- Spielzeit
 - Läuft nur weiter, wenn FBK gestartet ist
 - Beliebiger Faktor zur Echtzeit
 - Spiele mit kontinuierlicher Zeit

Spielebenen (Standardplugin)

- Spiel
 - Karte (isometrisch oder ‚Galaxien‘)
 - Felder/Planeten
 - Spieler
 - Städte/Planeten
 - Gebäude
 - Forschungen
 - Einheiten
 - Truppen (Zusammengefasste oder bewegte Einheiten)

- Integriertes Nachrichtensystem
 - Benachrichtigung per E-Mail
 - Nachrichten der Administration an alle Spieler
- Allianzsystem
 - Spieler können in eine oder mehrere Allianzen eintreten
 - Einladungen/Bewerbungen
 - Rechtesystem
 - Allianzseite
 - Nachrichten an Allianzmitglieder

- Kampfsystem nach Art ‚Napoleon‘
 - Einheiten stehen sich in Reihe gegenüber und schießen sich gegenseitig ab (Übliches Verfahren in Browserspielen)
- Kämpfe mit Millionen Einheiten in Sekunden berechnet
- Automatische Erstellung von Kampfnachrichten
- Einfluss von Einheitentyp und Forschungen
- Kampfsimulator inklusive

- Vereinheitlichte Schnittstelle zu Social Networks
- SocialNetwork-Provider
 - Facebook (vorhanden)
 - Zooml (vorhanden)
 - Twitter (geplant)
- Genutzt für Single-SignOn-Lösungen

- Sehr einfaches CMS zur Einbindung von freien Inhalten
- Konfiguriert über Xml
- Inhalte werden über das Admin-Interface festgelegt

- Plünderungen nach Kämpfen
- Trümmerfelder
- Energiemanagement (Deaktivierung von Gebäuden bei Energiemangel)
- Spielerpunkte
- Neulingsschutz
- Handelsplattform
 - Auktionen (Rückwärts- und Vorwärts)
- ...



Webserver

- Trennung von Logik und Ansicht
- Nutzung des http.sys-Systemtreibers
 - Gemeinsame Nutzung des Port 80 (oder anderen Ports) mit anderen FBK-Instanzen oder Internet Information Server (IIS)
- Integrierter XSRF-Schutz über Cookies, Tokens und Url
 - Seitenweise abschaltbar
- Timeout für jeden Webrequest mit Hilfe des ThreadWatchers

- Für jeden Seitentyp ist eine eigene Klasse implementiert:
 - Übersichtsseite
 - Gebäudeansicht
 - Forschungen
 - Kampfsimulator ...
- Durch Trennung von Logik und Ansicht ist eine schnelle Implementierung weiterer Seiten möglich.
- Durch Nutzung von .Net sogar in IronPython/IronRuby

- Master/Slave-Templates
 - Gemeinsames Master-Template in das die eigentlichen Seiten eingebettet werden.
 - Unterschiedliche Master-Templates möglich
- Ausdrucksparser
 - Mathematische Operationen ($a+b*c$)
 - Text-Operationen („Hallo“ + name)
 - Objekt-Referenzierungen (Player.Playername)
 - Methoden-Aufrufe (Town.GetLevel(„metallmine“))
 - Nutzung von .Net-Reflections bei unbekanntem Typen

- Template-Engine
 - Variablenzuweisungen
 - IF
 - FOREACH
 - Für jede .Net-Klasse, die *IEnumerable* implementiert
 - WebControls
 - Tabellen, Countdowns, Eingabefeldern, Knöpfen

- Erweiterung des DCSS-Standards
 - <http://depon.net/downloads/dynamiccss.pdf>
- Automatische Erstellung von CSS-Dateien
- Nutzung von Variablen innerhalb des CSS

```
@dcssdefine {  
    background: blue;  
    foreground: #00FFFF;  
}  
body {  
    background-color: [background];  
    color: [foreground];  
}
```



Technische Informationen

- Zentraler Sammelpunkt für alle Datencontainer (*DataStorage*)
- Nur in Datencontainer dürfen Daten persistieren
- Wird ein Container benötigt, so wird ein Handler zum Datencontainer erstellt.
 - Nur innerhalb eines Threads verwendbar
 - Am Ende der Verarbeitung zu zerstören
- Echte Datenbankverbindungen möglich

- Zweistufig aufgebaut:
 1. **Datencontainer** (*IDataContainerHandler*), dieser enthält mehrere
 2. **Datenpools** (*IDataHandler*)
- Der Datencontainer kann genutzt werden um eine echte Datenbankverbindung aufzubauen.
- Solang dieser nicht zerstört per Dispose wird, ist diese aufrecht zu erhalten.

- Logikkern
 - Erzeugt LogicSessions, die den Zugriff auf die Logik erzeugen.
- LogicSession
 - Wird erzeugt, wenn Zugriff auf die Logik benötigt wird.
 - Stellt Zugriff auf Datencontainern zur Verfügung und vernichtet die Handler, wenn die LogicSession selbst beendet wird.

- LogicState
 - Beinhaltet Daten, die während der Lebenszeit eines Prozess gespeichert werden sollen.
 - Konfigurationen
 - Persistente Daten haben in Datencontainern verwaltet zu werden.
- LogicHandler
 - Für eine LogicSession gültiger Zugriff auf die Logik-Objekte
 - Innerhalb des LogicHandlers können Datenbankverbindungen aufgebaut werden.