

OpenAnno

Dusmania 10, 2008

- 1 Einführung
 - Zusammenfassung
 - Lizenz
 - Geschichte
 - Milestone 2008.0
- 2 Technik
 - Zusammenfassung
 - Architektur
 - FIFE
- 3 Zukunft
 - Milestone 2008.1
- 4 Vorspielen

Ein Anno-ähnliches Spiel



- Ziel: Erschaffung eines quelloffenen, Anno-ähnlichen Spiels

Ein Anno-ähnliches Spiel



- Ziel: Erschaffung eines quelloffenen, Anno-ähnlichen Spiels
- Kein Anno-Klon

Ein Anno-ähnliches Spiel



- Ziel: Erschaffung eines quelloffenen, Anno-ähnlichen Spiels
- Kein Anno-Klon
- Noch in der frühen Entwicklungsphase

Ein Anno-ähnliches Spiel



- Ziel: Erschaffung eines quelloffenen, Anno-ähnlichen Spiels
- Kein Anno-Klon
- Noch in der frühen Entwicklungsphase
- Helfer sind immer willkommen!

GPL / Creative Commons

- Code: GPLv2

GPL / Creative Commons

- Code: GPLv2
- Grafik/Musik: Creative Commons CC-BY-SA 3.0

Python: Der 4. Versuch

- 2005: C-Version

Python: Der 4. Versuch

- 2005: C-Version
- 2006: C++-Version mit neuem Team

Python: Der 4. Versuch

- 2005: C-Version
- 2006: C++-Version mit neuem Team
- Ende 2007: Experimente mit 3D

Python: Der 4. Versuch

- 2005: C-Version
- 2006: C++-Version mit neuem Team
- Ende 2007: Experimente mit 3D
- Anfang 2008: Beschluss zum Wechsel auf Python und die FIFE-Engine

Python: Der 4. Versuch

- 2005: C-Version
- 2006: C++-Version mit neuem Team
- Ende 2007: Experimente mit 3D
- Anfang 2008: Beschluss zum Wechsel auf Python und die FIFE-Engine
- März 2008: Commit #100

Python: Der 4. Versuch

- 2005: C-Version
- 2006: C++-Version mit neuem Team
- Ende 2007: Experimente mit 3D
- Anfang 2008: Beschluss zum Wechsel auf Python und die FIFE-Engine
- März 2008: Commit #100
- Juni 2008: Commit #500

Python: Der 4. Versuch

- 2005: C-Version
- 2006: C++-Version mit neuem Team
- Ende 2007: Experimente mit 3D
- Anfang 2008: Beschluss zum Wechsel auf Python und die FIFE-Engine
- März 2008: Commit #100
- Juni 2008: Commit #500
- Juli 2008: interner Milestone 2008.0 fertig

Python: Der 4. Versuch

- 2005: C-Version
- 2006: C++-Version mit neuem Team
- Ende 2007: Experimente mit 3D
- Anfang 2008: Beschluss zum Wechsel auf Python und die FIFE-Engine
- März 2008: Commit #100
- Juni 2008: Commit #500
- Juli 2008: interner Milestone 2008.0 fertig
- Juli 2008: Commit #1000

Was funktioniert:

- Kartendarstellung

Was funktioniert:

- Kartendarstellung
- Bewegen von Einheiten (momentan nur Schiffe)

Was funktioniert:

- Kartendarstellung
- Bewegen von Einheiten (momentan nur Schiffe)
- Bauen eines Kontors, Holzfällers und Schafhirten

Was funktioniert:

- Kartendarstellung
- Bewegen von Einheiten (momentan nur Schiffe)
- Bauen eines Kontors, Holzfällers und Schafhirten
- Grasende Schafe

Was funktioniert:

- Kartendarstellung
- Bewegen von Einheiten (momentan nur Schiffe)
- Bauen eines Kontors, Holzfällers und Schafhirten
- Grasende Schafe
- Holzfäller fällt Bäume

Was funktioniert:

- Kartendarstellung
- Bewegen von Einheiten (momentan nur Schiffe)
- Bauen eines Kontors, Holzfällers und Schafhirten
- Grasende Schafe
- Holzfäller fällt Bäume
- Bäume werden ins Kontor transportiert

Was funktioniert:

- Kartendarstellung
- Bewegen von Einheiten (momentan nur Schiffe)
- Bauen eines Kontors, Holzfällers und Schafhirten
- Grasende Schafe
- Holzfäller fällt Bäume
- Bäume werden ins Kontor transportiert
- Anpassung des FIFE-Karteneditors an unser Format

FIFE, Python und SQLite

- Basis: FIFE (<http://www.fifengine.de/>), eine offene Spiele-Engine in C++

FIFE, Python und SQLite

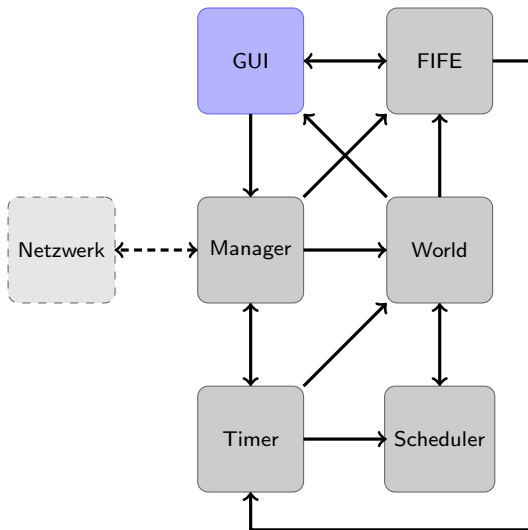
- Basis: FIFE (<http://www.fifengine.de/>), eine offene Spiele-Engine in C++
- Enger Kontakt zu den Entwicklern

FIFE, Python und SQLite

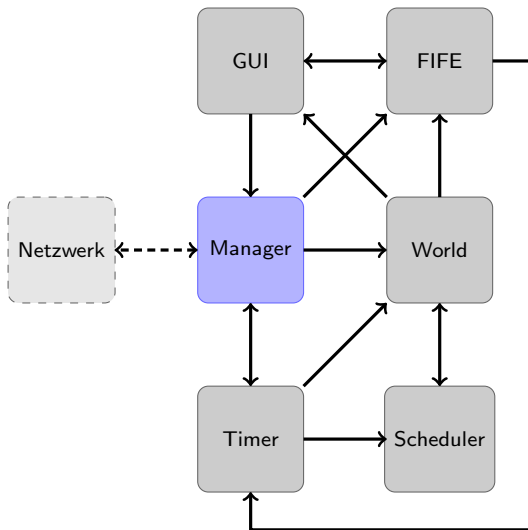
- Basis: FIFE (<http://www.fifengine.de/>), eine offene Spiele-Engine in C++
- Enger Kontakt zu den Entwicklern
- Implementierung des Spiels selbst in Python

FIFE, Python und SQLite

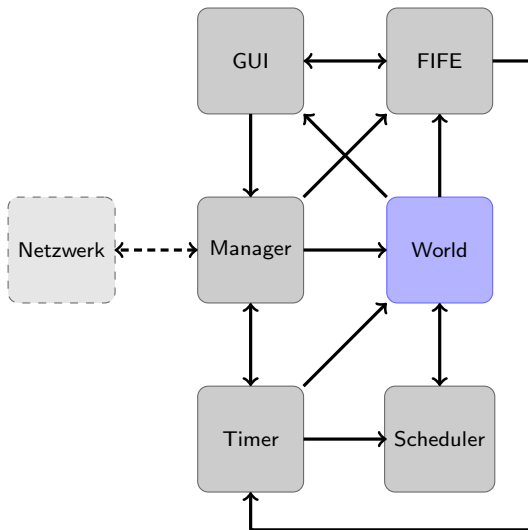
- Basis: FIFE (<http://www.fifengine.de/>), eine offene Spiele-Engine in C++
- Enger Kontakt zu den Entwicklern
- Implementierung des Spiels selbst in Python
- SQLite als Datenspeicher



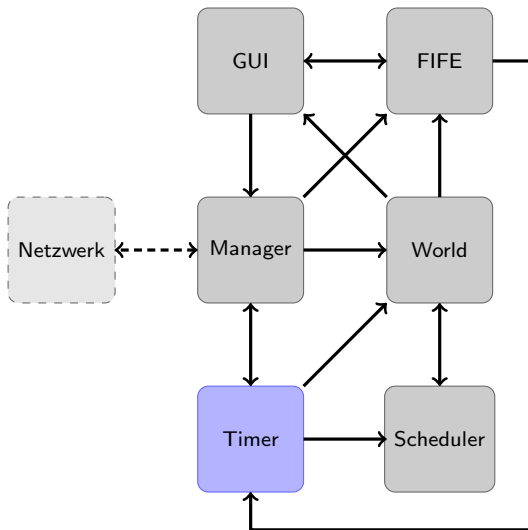
- GUI: Kommuniziert viel mit FIFE, gibt Befehle an den Manager und fragt World-Objekte nach Informationen, darf aber keine Aktionen direkt auf ihnen ausführen



- **Manager (Singleplayer):**
Bekommt Befehle von der GUI und leitet sie an die World-Objekte und FIFE weiter
- **Manager (Multiplayer):**
 - Bekommt Befehle von der GUI und schickt sie an die Mitspieler
 - Führt Befehle der Mitspieler aus



- World: Sammlung von Objekten, die auf allen mitspielenden OpenAnno-Instanzen synchronisiert sind



- Timer: Bekommt regelmäßig von FIFE die Kontrolle übergeben

Über FIFE

- Enstanden im September 2005 als »Flexible Isometric Fallout-like Engine«

Über FIFE

- Enstanden im September 2005 als »Flexible Isometric Fallout-like Engine«
- Von Grund auf neu geschrieben; kein Fork

Über FIFE

- Enstanden im September 2005 als »Flexible Isometric Fallout-like Engine«
- Von Grund auf neu geschrieben; kein Fork
- Ursprünglich als Engine für isometrische Rollenspiele wie Fallout gedacht

Über FIFE

- Enstanden im September 2005 als »Flexible Isometric Fallout-like Engine«
- Von Grund auf neu geschrieben; kein Fork
- Ursprünglich als Engine für isometrische Rollenspiele wie Fallout gedacht
 - Fallout-Dateiformate wurden benutzt, um Beispielinhalte zu haben, ohne diese selbst erstellen zu müssen

Über FIFE

- Enstanden im September 2005 als »Flexible Isometric Fallout-like Engine«
- Von Grund auf neu geschrieben; kein Fork
- Ursprünglich als Engine für isometrische Rollenspiele wie Fallout gedacht
 - Fallout-Dateiformate wurden benutzt, um Beispielinhalte zu haben, ohne diese selbst erstellen zu müssen
 - **Damit aber auch Abhängigkeiten von Fallout**

Über FIFE

- Enstanden im September 2005 als »Flexible Isometric Fallout-like Engine«
- Von Grund auf neu geschrieben; kein Fork
- Ursprünglich als Engine für isometrische Rollenspiele wie Fallout gedacht
 - Fallout-Dateiformate wurden benutzt, um Beispielinhalte zu haben, ohne diese selbst erstellen zu müssen
 - Damit aber auch Abhängigkeiten von Fallout
- Ab Sommer 2006 Entfernung von den ursprünglichen Falloutwurzeln und Beschränkungen; Konzeption von eigenen, von Fallout unabhängigen Dateiformaten

Über FIFE

- Enstanden im September 2005 als »Flexible Isometric Fallout-like Engine«
- Von Grund auf neu geschrieben; kein Fork
- Ursprünglich als Engine für isometrische Rollenspiele wie Fallout gedacht
 - Fallout-Dateiformate wurden benutzt, um Beispielinhalte zu haben, ohne diese selbst erstellen zu müssen
 - Damit aber auch Abhängigkeiten von Fallout
- Ab Sommer 2006 Entfernung von den ursprünglichen Falloutwurzeln und Beschränkungen; Konzeption von eigenen, von Fallout unabhängigen Dateiformaten
- 2.5D-Engine; vermischt 2D- und 3D-Techniken

Über FIFE

- Enstanden im September 2005 als »Flexible Isometric Fallout-like Engine«
- Von Grund auf neu geschrieben; kein Fork
- Ursprünglich als Engine für isometrische Rollenspiele wie Fallout gedacht
 - Fallout-Dateiformate wurden benutzt, um Beispielinhalte zu haben, ohne diese selbst erstellen zu müssen
 - Damit aber auch Abhängigkeiten von Fallout
- Ab Sommer 2006 Entfernung von den ursprünglichen Falloutwurzeln und Beschränkungen; Konzeption von eigenen, von Fallout unabhängigen Dateiformaten
- 2.5D-Engine; vermischt 2D- und 3D-Techniken
- Geschrieben in C++ und Python: FIFE-Nutzer können beide Sprachen flexibel für ihre Spiele einsetzen

Über FIFE

- Enstanden im September 2005 als »Flexible Isometric Fallout-like Engine«
- Von Grund auf neu geschrieben; kein Fork
- Ursprünglich als Engine für isometrische Rollenspiele wie Fallout gedacht
 - Fallout-Dateiformate wurden benutzt, um Beispielinhalte zu haben, ohne diese selbst erstellen zu müssen
 - Damit aber auch Abhängigkeiten von Fallout
- Ab Sommer 2006 Entfernung von den ursprünglichen Falloutwurzeln und Beschränkungen; Konzeption von eigenen, von Fallout unabhängigen Dateiformaten
- 2.5D-Engine; vermischt 2D- und 3D-Techniken
- Geschrieben in C++ und Python: FIFE-Nutzer können beide Sprachen flexibel für ihre Spiele einsetzen
- **Lizenz: seit Milestone 2008.1 LGPL, vorher GPL**

Derzeitiger Stand

- Mission Statement: »FIFE ist ein flexibles Framework zur Erstellung von isometrischen Spielen, welche auf verschiedenen Plattformen laufen (mindestens Linux, Mac, Win32 angestrebt).«

Derzeitiger Stand

- Mission Statement: »FIFE ist ein flexibles Framework zur Erstellung von isometrischen Spielen, welche auf verschiedenen Plattformen laufen (mindestens Linux, Mac, Win32 angestrebt).«
- Unterstützung der Vogelperspektive neben Isometrie und Ähnlichem

Derzeitiger Stand

- Mission Statement: »FIFE ist ein flexibles Framework zur Erstellung von isometrischen Spielen, welche auf verschiedenen Plattformen laufen (mindestens Linux, Mac, Win32 angestrebt).«
- Unterstützung der Vogelperspektive neben Isometrie und Ähnlichem
- Falloutwurzeln nur noch rudimentär erkennbar; Engine jetzt wesentlich flexibler

Derzeitiger Stand

- Mission Statement: »FIFE ist ein flexibles Framework zur Erstellung von isometrischen Spielen, welche auf verschiedenen Plattformen laufen (mindestens Linux, Mac, Win32 angestrebt).«
- Unterstützung der Vogelperspektive neben Isometrie und Ähnlichem
- Falloutwurzeln nur noch rudimentär erkennbar; Engine jetzt wesentlich flexibler
- **Fallout-Dateiformate zum Großteil nicht mehr unterstützt**

Derzeitiger Stand

- Mission Statement: »FIFE ist ein flexibles Framework zur Erstellung von isometrischen Spielen, welche auf verschiedenen Plattformen laufen (mindestens Linux, Mac, Win32 angestrebt).«
- Unterstützung der Vogelperspektive neben Isometrie und Ähnlichem
- Falloutwurzeln nur noch rudimentär erkennbar; Engine jetzt wesentlich flexibler
- Fallout-Dateiformate zum Großteil nicht mehr unterstützt
- Editor für eigene Karten grundlegend funktionsfähig, jedoch noch in Entwicklung

Derzeitiger Stand

- Mission Statement: »FIFE ist ein flexibles Framework zur Erstellung von isometrischen Spielen, welche auf verschiedenen Plattformen laufen (mindestens Linux, Mac, Win32 angestrebt).«
- Unterstützung der Vogelperspektive neben Isometrie und Ähnlichem
- Falloutwurzeln nur noch rudimentär erkennbar; Engine jetzt wesentlich flexibler
- Fallout-Dateiformate zum Großteil nicht mehr unterstützt
- Editor für eigene Karten grundlegend funktionsfähig, jedoch noch in Entwicklung
- 3D-View-Matrix erlaubt spezielle Effekte: Zoom (nur OpenGL), Rotation, Neigung

Zukunft der Engine

- Umbenennung des FIFE-Akronyms zur Verdeutlichung der Loslösung von den Falloutwurzeln

Zukunft der Engine

- Umbenennung des FIFE-Akronyms zur Verdeutlichung der Loslösung von den Falloutwurzeln
- Netzwerkunterstützung für Multiplayer- und Videosupport

Zukunft der Engine

- Umbenennung des FIFE-Akronyms zur Verdeutlichung der Loslösung von den Falloutwurzeln
- Netzwerkunterstützung für Multiplayer- und Videosupport
- Verbessertes Demospiel, um die Fähigkeiten der Engine einem breiteren Publikum zu präsentieren

Zukunft der Engine

- Umbenennung des FIFE-Akronyms zur Verdeutlichung der Loslösung von den Falloutwurzeln
- Netzwerkunterstützung für Multiplayer- und Videosupport
- Verbessertes Demospiel, um die Fähigkeiten der Engine einem breiteren Publikum zu präsentieren
- Mehr Entwickler und Nutzer für FIFE gewinnen, um eine aktive Community aufzubauen (Vorbild: große 3D-Engines wie Ogre)

Was hoffentlich bald funktioniert:

- Grundlegende Multiplayer-Unterstützung

Was hoffentlich bald funktioniert:

- Grundlegende Multiplayer-Unterstützung
- Mehr funktionierende Gebäude

Was hoffentlich bald funktioniert:

- Grundlegende Multiplayer-Unterstützung
- Mehr funktionierende Gebäude
- Grafiken!

Was hoffentlich bald funktioniert:

- Grundlegende Multiplayer-Unterstützung
- Mehr funktionierende Gebäude
- Grafiken!
- Pathfinding

Vorspielen!