

CELLmicrocosmos 3.1

Projekt im WS 07/08

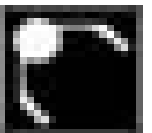
Teilnehmer:

Christina Ander
Florian Heißenberg
Jörn Winnebald
Lukas Jelonek
Philipp Unruh
Regina Bisdorf



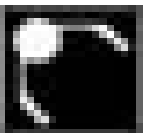
Motivation

- Visualisierung von Zellen
 - in der Medizin
 - zu Schulungszwecken
- Visualisierung von
 - der Struktur der Zelle
 - Stoffwechselwegen
 - Reaktionen der Zelle auf die Umwelt
- Derzeit wird für jede Visualisierung ein Grafiker benötigt



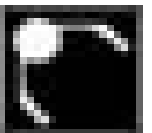
Anforderungen

- Erweiterbarkeit
 - eigene Zellmodelle importieren
- Austauschmöglichkeit
 - Projekte speichern/laden
- stabile Softwarearchitektur
 - Erweiterbarkeit um neue Features



Rückblick

- CELLmicrocosmos 3.0 – Zelleditor
- Erstellen/Speichern/Laden von Zellen möglich
- Schwachstellen in CM 3.0
 - Softwarearchitektur nicht leicht erweiterbar
 - Platzierungsalgorithmus nicht sehr robust
 - GUI nicht modular
 - Speichern/Laden nicht voll funktionsfähig



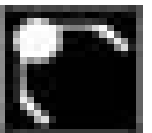
Aufgaben für CELLmicrocosmos 3.1

- Restrukturierung/Refactoring des bisherigen Codes
- Neugestaltung/Portierung der GUI auf Infonode
- Stereo View
- Neue 3D-Modelle für Zellstrukturen
- Verbesserung des Platzierungsalgorithmus
- Speichern und Laden der Projekte



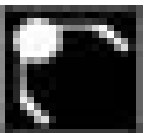
Infonode

- personalisiertes Layout der GUI
- Fensterkomponenten können
 - geschlossen und wiederhergestellt werden
 - minimiert, maximiert, vom Hauptfenster abgelöst und wieder eingefügt werden
 - in Tabs angeordnet werden
- einfache Bedienung durch Drag and Drop

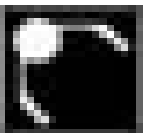
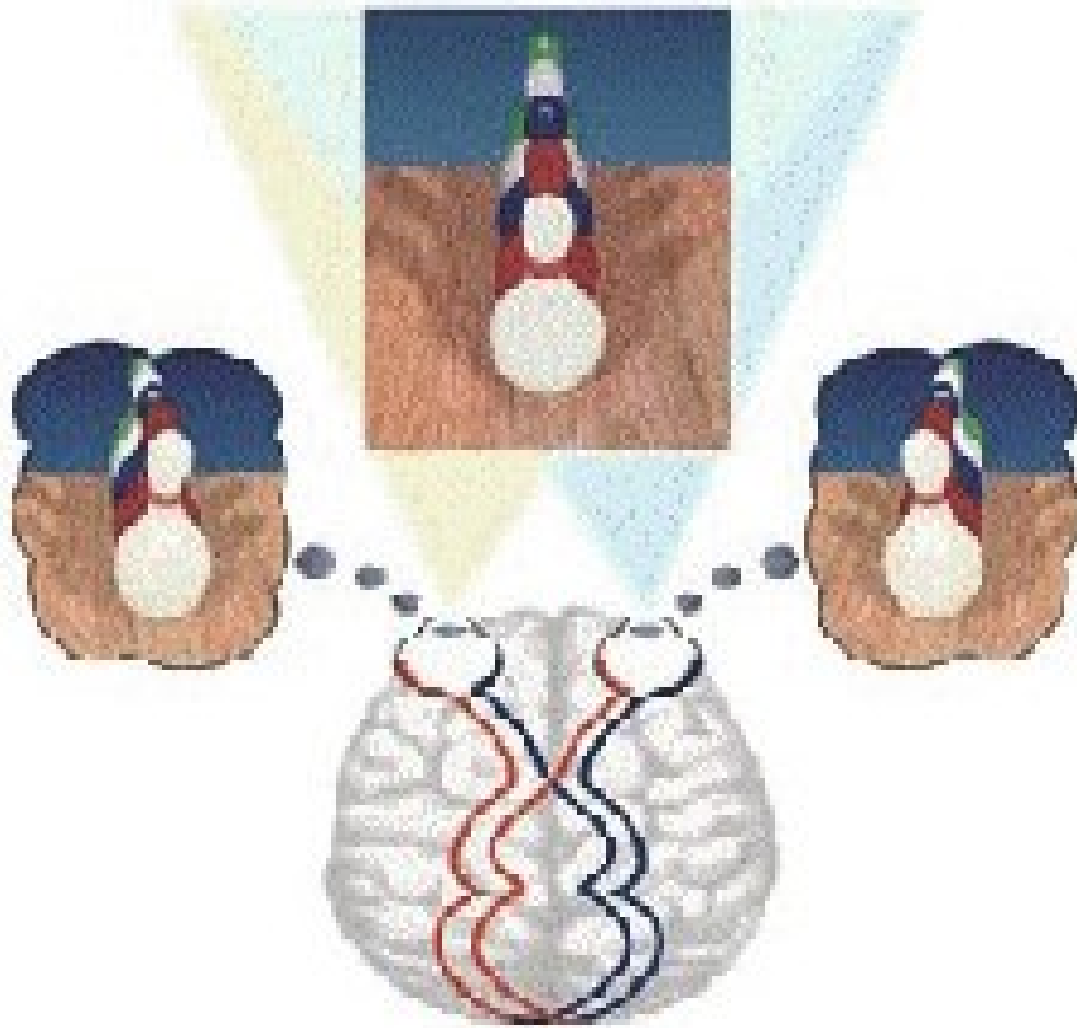


Infonode in CM3.1

- Swingkomponenten aus CM 3.0 können in Infonodecontainer gesetzt werden
- dadurch wenig Programmieraufwand
- Eigenschaften, wie Größe, Lokalität der Swingkomponenten werden überflüssig
- Durch ein integriertes LayoutUtil kann das Standardlayout des Zelleditors einfach eingestellt werden

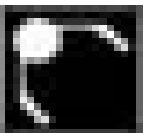


Stereo View



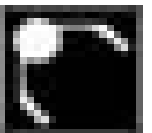
Stereo View

- Vorteile zu flacher Projektion des 3D-Raums
 - Tiefe der Objekte leichter zu erkennen
 - entspricht natürlichem Sehen
- Nachteile
 - nicht jeder Mensch kann das wahrnehmen
 - spezielle Hardware nötig
- Hardware für Stereoskopische Darstellung vorhanden



Stereo View in CM 3.1

- Java3D enthält bereits eine vollständige Implementierung für stereoskopische Darstellung
- Programmieraufwand gering
- Probleme mit Hardware zu lösen
 - Nicht lauffähig mit mehr als zwei aktivierten Monitoren



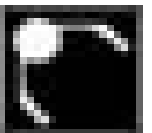
Platzierungsalgorithmus

- Laden der Zellkomponentenliste (Typ, Anzahl)
- Überprüfung der Position einer Zellkomponente
 - Membran -> wird zufällig gesetzt
 - Nucleolus -> wird in den Nucleus gesetzt
 - ...
- Erstellung der Bounds der Zellkomponente (Kugel)



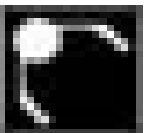
Platzierungsalgorithmus

- Vergleich der Bounds mit bereits platzierten Komponenten
 - Überschneidung mit einer anderen Zellkomponente
- Bei Überschneidung wird die Komponente an einen neuen zufälligen Punkt gesetzt und erneute Überprüfung findet statt
- Nach 1000 Fehlschlägen bricht der Algorithmus ab



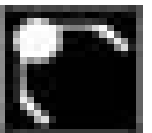
Laden/Speichern von Projekten

- Vorher:
 - nur Verweise auf geladene Zellkomponenten und deren Position wurde gespeichert
 - Fehler ab dem zweiten Laden von Projekten
 - VRML Dateien müssen in speziellem Ordner liegen
- Jetzt:
 - Verweis auf VRML-Datei, Position und weitere Metainformationen wird gespeichert
 - Ladefehler beseitigt
 - Neuer Speicherprozess

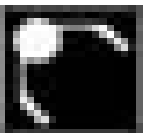


Laden/Speichern von Projekten

- Neuer Speicherprozess
 - Ordner mit Projektnamen wird erstellt
 - XML-Datei mit Informationen über Komponenten und deren Positionen wird erstellt
 - für Projekt benötigte VRML-Dateien werden in erstellten Ordner kopiert
- höhere Portierbarkeit der Projekte
- allerdings auch höherer Speicherbedarf



Live Demo



Ausblick

- Integration von Stoffwechselwegen
- Animation von Stoffwechselwegen
- Erweiterung um weitere Zelltypen
- Pfade für Kameraflug durch die Zelle definieren
- Komponentenmodellgenerator
 - Variable Zellmodelle
- verbesserten Platzierungsalgorithmus implementieren
- Modulare Algorithmen

