



Zellomat3D

Testbericht

Projekt: 3D Cellular Automata Simulator – Diplomarbeit – SS/2005

Auftraggeber: Hochschule Rapperswil HSR

Betreuer: Eduard Glatz – Prof. Dipl. Ing. ETH eglatz@hsr.ch

Mitarbeiter: Michael Florin loop@loop.li
Andreas Weinmann a.weinmann@gmx.ch

Ablage: Testbericht - 02052005.doc



Inhaltsverzeichnis

1. EINFÜHRUNG	3
ZWECK	3
ÜBERSICHT.....	3
GÜLTIGKEITSBEREICH	3
2. RISIKOANALYSE.....	4
WICHTIGKEITSSTUFEN.....	4
WAHRSCHEINLICHKEIT	4
TESTSTUFEN	4
3. TEST DER NICHT FUNKTIONALEN ANFORDERUNGEN	6
TC1: FRAMEBUFFER	6
TC2: MINIMALE HARDWARE ANFORDERUNGEN	7
TC3: STABILITÄT DER APPLIKATION	8
BEGRÜNDUNG	8
HINWEIS.....	8
4. TEST DER FUNKTIONALEN ANFORDERUNGEN	9
TC4: HOHE FRAMERATE	9
BEGRÜNDUNG	9
TC5: KAMERASTEUERUNG	10
TC6: BERECHNUNGSZEIT	10
BEGRÜNDUNG:	10
TC7: UMSETZUNG DER ZA REGELN.....	11
TC8: ORIENTIERUNG	12
TC9: PARAMETRISIERUNG.....	12
5. TEST DER USE CASES.....	13
TC10: ÖFFNEN EINES ZA.....	13
TC11: RESET ZA.....	13
TC12: PLAY ZA.....	13
TC13: STEP ZA.....	13
TC14: PAUSE ZA.....	13
TC15: ANZEIGEINTERVALL EINSTELLEN.....	13
TC16: SCHNAPPSCHUSS EINES ZYKLUS.....	14
TC17: INFORMATIONEN ÜBER DAS PROGRAMM ANZEIGEN.....	14
TC18: HILFE ANZEIGEN / AUSBLENDEN.....	14
TC19: UMGEBUNG ANZEIGEN / AUSBLENDEN.....	14
TC20: KOORDINATENACHSEN ANZEIGEN / AUSBLENDEN	14
TC21: LEBENSRAUMKUBUS ANZEIGEN / AUSBLENDEN	14
TC22: ANZEIGE EIGENSCHAFTEN VERÄNDERN	14
TC23: WECHSEL IN FENSTER / VOLLBILDMODUS	14
TC24: ÄNDERUNG DER GRÖSSE DES FENSTERS	15
TC25: KAMERASTEUERUNG	15
TC26: BEENDEN DER APPLIKATION	15



1. Einführung

Dieses Dokument enthält die Testspezifikation sowie die **Zweck** Testberichte für das Projekt. Folgendes wird damit festgelegt:

- Welche Funktionen getestet werden
- Wie die Funktionen getestet werden
- Ob die Anforderungen eingehalten wurden
- Ob die Zuverlässigkeit und Qualität der Applikation Zellomat3D genügend gut ist.

Der Inhalt der Testspezifikation besteht im Wesentlichen aus **Übersicht** zwei Teilen. Diese sind:

- Risikoanalyse
- Testfälle

Die Risikoanalyse begründet, warum die einzelnen Funktionen verschieden ausführlich getestet werden. Damit wird festgelegt, wie detailliert getestet wird.

Zu allen Testfällen aus der Testspezifikation wird eine kurze Situationsbeschreibung geliefert, die aussagt, ob die Testfälle erfolgreich waren oder nicht. Trifft letzteres zu, wird beschrieben, was nicht funktioniert hat.

Dieses Dokument gilt für die Diplomarbeit "Zellomat3D", welche **Gültigkeitsbereich** im SS/2005 an der Hochschule Rapperswil HSR durchgeführt wurde.



2. Risikoanalyse

Alle Funktionen des verteilten Software-Systems werden in die **Wichtigkeitsstufen** drei **Wichtigkeitsstufen** Priorität 1, 2 und 3 eingeteilt. Ist ein Testpunkt mit Priorität 1 versehen, bedeutet das, dass diese Funktion besonders wichtig ist.

Die **Wahrscheinlichkeit für einen Mangel** (gross mittel oder klein) drückt aus, wie gross die Wahrscheinlichkeit ist, dass ein Fehler in der Funktion auftritt.

Die **Testzeit** wird aus dem Mittelwert der Wichtigkeit und der **Teststufen** Wahrscheinlichkeit eines Mangels gebildet und legt fest, wie ausführlich die Funktion getestet wird.

Testcase	Testpunkt	Wichtigkeit			Wahrscheinlichkeit für Mängel			Zeit		
		Priorität 1	Priorität 2	Priorität 3	gross	mittel	klein	systematisch	typisch	stichprobenhaft
Test der nicht funktionalen Anforderungen										
TC1	Framebuffer		X			X		X		
TC2	Minimale Hardware Anforderungen	X				X		X		
TC3	Stabilität der Applikation	X			X			X		
Test der funktionalen Anforderungen										
TC4	Hohe Framerate	X				X			X	
TC5	Kamerasteuerung		X				X			X
TC6	Berechnungszeit	X			X			X		
TC7	Umsetzung der ZA Regeln	X				X		X		
TC8	Orientierung			X		X				X
TC9	Parametrisierung	X			X			X		



Testcase	Testpunkt	Wichtigkeit			Wahrscheinlichkeit für Mängel			Zeit		
		Priorität 1	Priorität 2	Priorität 3	gross	mittel	klein	systematisch	typisch	stichprobhaft
TC10	UC1: Öffnen eines ZA	X				X			X	
TC11	UC2: Reset ZA	X			X				X	
TC12	UC3: Play ZA	X					X			X
TC13	UC4: Step ZA	X					X		X	
TC14	UC5: Pause ZA	X					X			X
TC15	UC6: Anzeigintervall einstellen			X		X			X	
TC16	UC7: Schnappschuss eines Zyklus			X		X			X	
TC17	UC8: Informationen über das Programm anzeigen			X			X			X
TC18	UC9: Hilfe anzeigen / ausblenden		X		X				X	
TC19	UC10: Umgebung anzeigen / ausblenden			X		X				X
TC20	UC11: Koordinatenachsen anzeigen / ausblenden			X		X				X
TC21	UC12: Lebensraumkubus anzeigen / ausblenden			X		X				X
TC22	UC13: Anzeigeeigenschaften verändern		X		X			X		
TC23	UC14: Wechsel in Fenster / Vollbildmodus		X		X			X		
TC24	UC15: Änderung der Grösse des Fensters			X	X			X		
TC25	UC16: Kamerasteuerung		X			X			X	
TC26	UC17: Beenden der Applikation	X					X		X	



3. Test der nicht funktionalen Anforderungen

Die Applikation darf die 64MB Grafikkartenspeicher unter keinen **TC1:**
Umständen überfüllen. **Framebuffer**

ZA Berechnung im Pause Modus: **Test erfolgreich**
ZA Berechnung im Play Modus **Test erfolgreich**
ZA Berechnung nach Reset **Test erfolgreich**

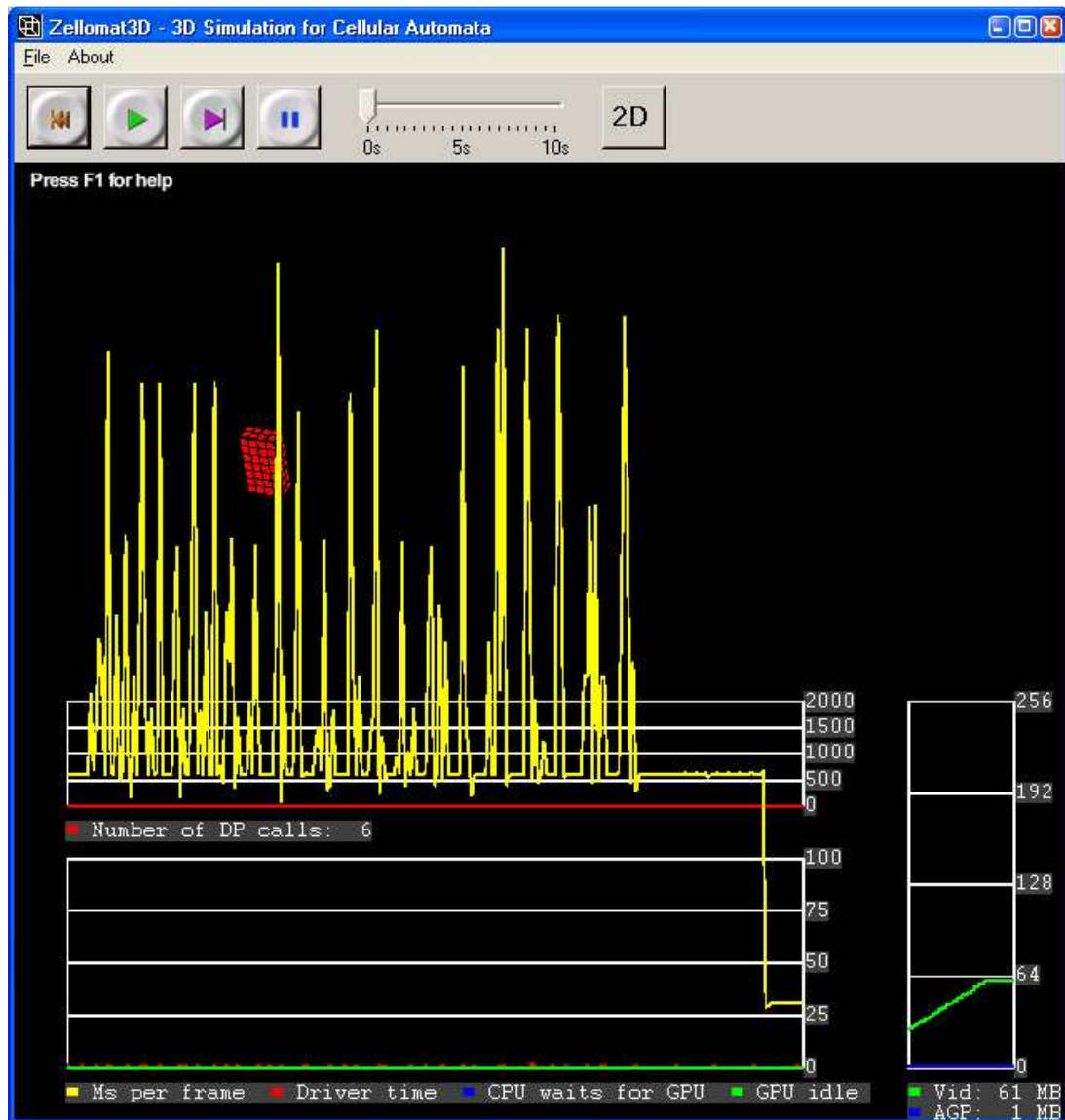


Abbildung 1: Der Grafikkartenspeicher wird fast vollständig befüllt (unten rechts)



Test, ob irgendwelche Probleme auftreten, unter Verwendung der minimalen Hardwareanforderungen.

TC2:
Minimale Hardware Anforderungen

Prozessorleistung
Grafikkarten Speicher
Grafikkarten Leistung
Arbeitsspeicher

Test erfolgreich
Test erfolgreich
Test teilweise erfolgreich (TC4)
Test erfolgreich

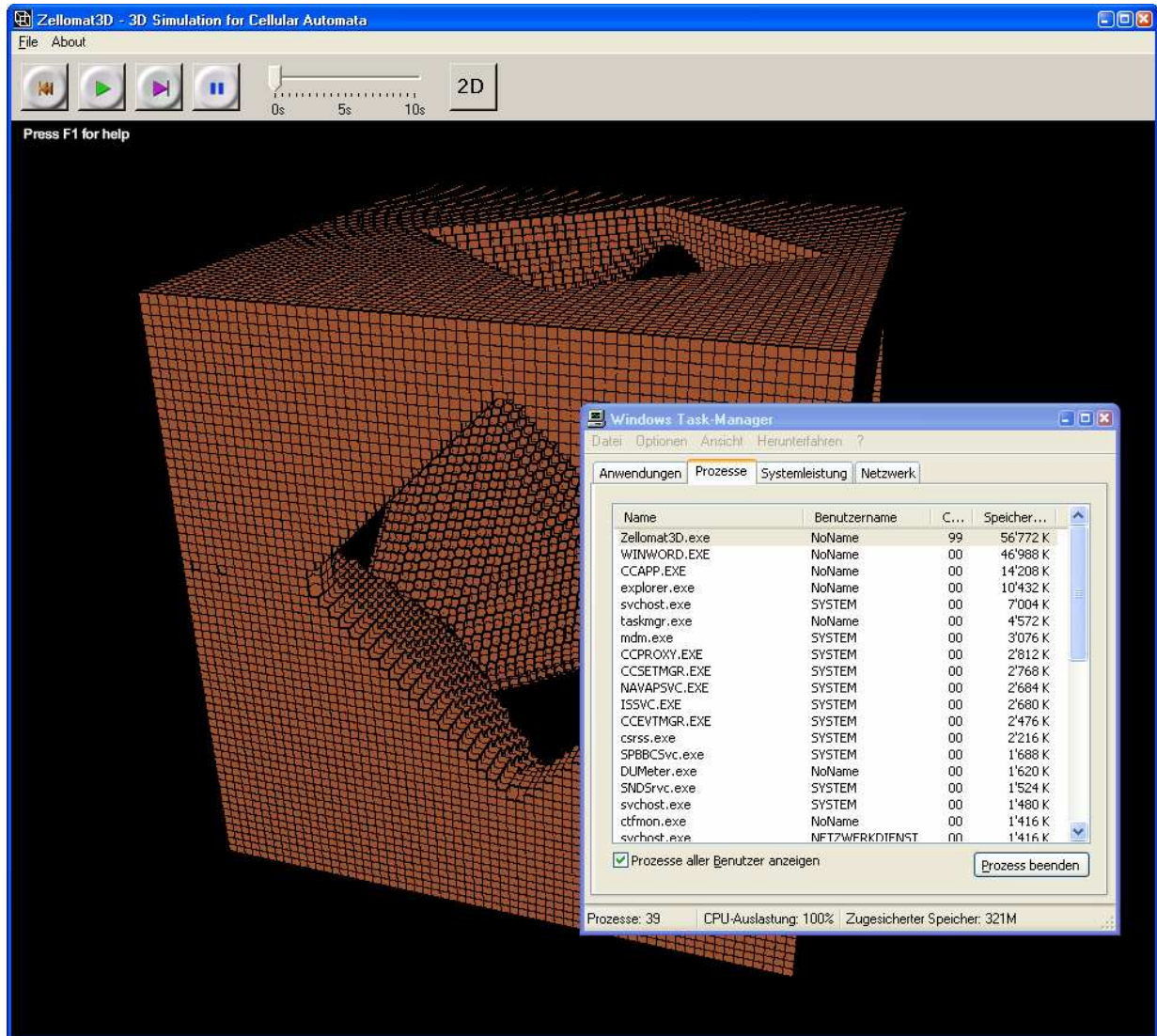


Abbildung 2: Die laufende Applikation, Speicherverbrauch 56'772KB



Test, ob durch Fehlmanipulationen oder falsche Parametrisierung die Applikation zum Absturz gebracht werden kann. **TC3: Stabilität der Applikation**

Öffnen eines neuen ZA **Test erfolgreich**
Bedienungselemente **Test erfolgreich**
Fehlmanipulationen **Test erfolgreich**
Auflösungsumschaltung **Test teilweise erfolgreich**

Die Applikation weist in nur einem getesteten Fall ein Fehlverhalten auf. Wenn sich das Programm im Vollbildmodus befindet und der Benutzer mit „ALT+Tabulator“ zu einem anderen Programm wechselt, bleibt die Applikation hängen. **Begründung**

Zu beachten ist, dass, wenn die Auflösung der 3D-Ausgabe auf eine andere Größe eingestellt werden muß (maximieren, minimieren, Fenstergröße ändern, Vollbild usw), dies mit einem erheblichen Zeitaufwand verbunden ist. Im Hintergrund werden grosse Initialisierungsroutinen durchlaufen, die die Grafikkarte auf eine neue Ausgabe vorbereiten müssen. Dies kostet verständlicherweise einiges an Zeit, in der der Benutzer die Geduld aufbringen muß zu warten. Diese Zeit liegt im Schnitt aber noch unter zehn Sekunden. **Hinweis**



4. Test der funktionalen Anforderungen

Die Framerate der 3D-Visualisierung darf 24 Frames pro Sekunde nicht unterschreiten **TC4: Hohe Framerate**

Lebensraumgröße: 8 **Test erfolgreich**
Lebensraumgröße: 16 **Test erfolgreich**
Lebensraumgröße: 32 **Test erfolgreich**
Lebensraumgröße: 64 **Test erfolgreich**
Lebensraumgröße: 128 **Test nicht erfolgreich**

Bei einer Lebensraumgröße von 128 sind es im Worstcase Szenario über 3'000'000 Punkte die von der Grafikkarte 24 Mal in der Sekunde neu berechnet werden müssten. Hier stoßen wir an die Hardwarelimite¹ der Grafikkarte. Die Grafikkarte braucht jetzt ca. 250ms pro Frame (Nvidia GeForce 4, 128MB Speicher) **Begründung**

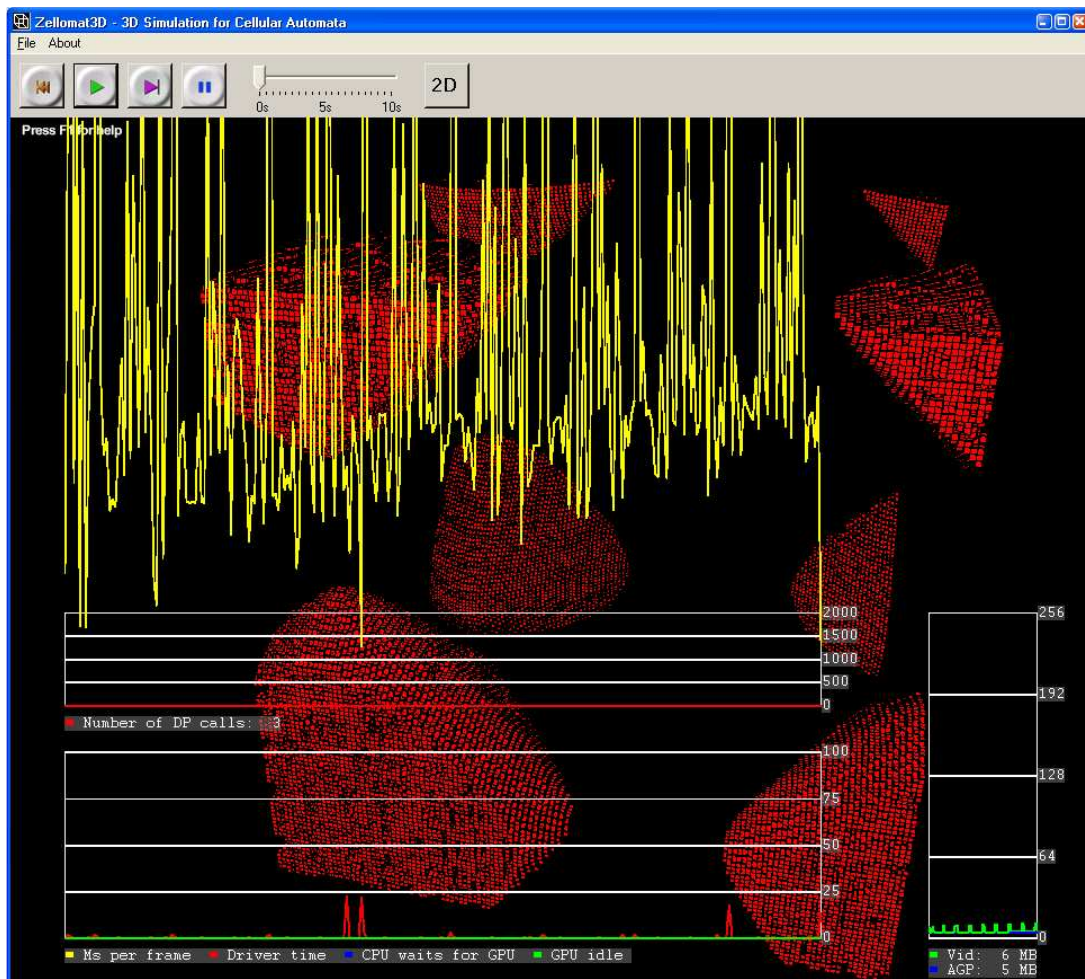


Abbildung 3 Lebensraumgröße von 128x128x128 Zellen, Framerate sackt auf 250ms (gelbe Linie) ab

¹ Mindestanforderungen der Hardware beschrieben in AnforderungsSpezifikation – 06042005.doc



Die Kamerasteuerung muss den Blickwinkel in alle Richtungen ermöglichen und die Verschiebung der Position in alle Koordinatenachsen ermöglichen. **TC5: Kamerasteuerung**

Kamerasteuerung: **Test erfolgreich**

Die Berechnungszeit darf nicht mehr als zehn Sekunden betragen. **TC6: Berechnungszeit**

Zellraumgrößen 8x8x8 bis 64x64x64 **Test erfolgreich**
Zellraumgröße 128x128x128 **Test nicht erfolgreich**

Die Berechnungszeit nur für die Zyklusberechnung der Zellen liegt noch unter zehn Sekunden. Um aber überhaupt eine Darstellung eines solch grossen Zellhaufens zu ermöglichen, musste ein komplizierter Algorithmus in der Datenaufbereitung implementiert werden. Dies braucht eine gewisse Zeit, um die Daten für die grafische Ausgabe aufzubereiten und zu komprimieren, da sonst die Grafikkarte mit zuvielen Daten überflutet würde und dann gar nichts mehr zu sehen wäre. Diese zwei Vorgänge zusammen überschreiten die Limite der zehn Sekunden. **Begründung:**



Die Applikation darf keine Fehler in der Berechnung der Zyklussteuerung aufweisen. Getestet wurden: **TC7: Umsetzung der ZA Regeln**

Zyklische Randbedingungen: **Test erfolgreich**
Korrekte Umsetzung der Regeln: **Test erfolgreich**
Farben entsprechen Zuständen: **Test erfolgreich**

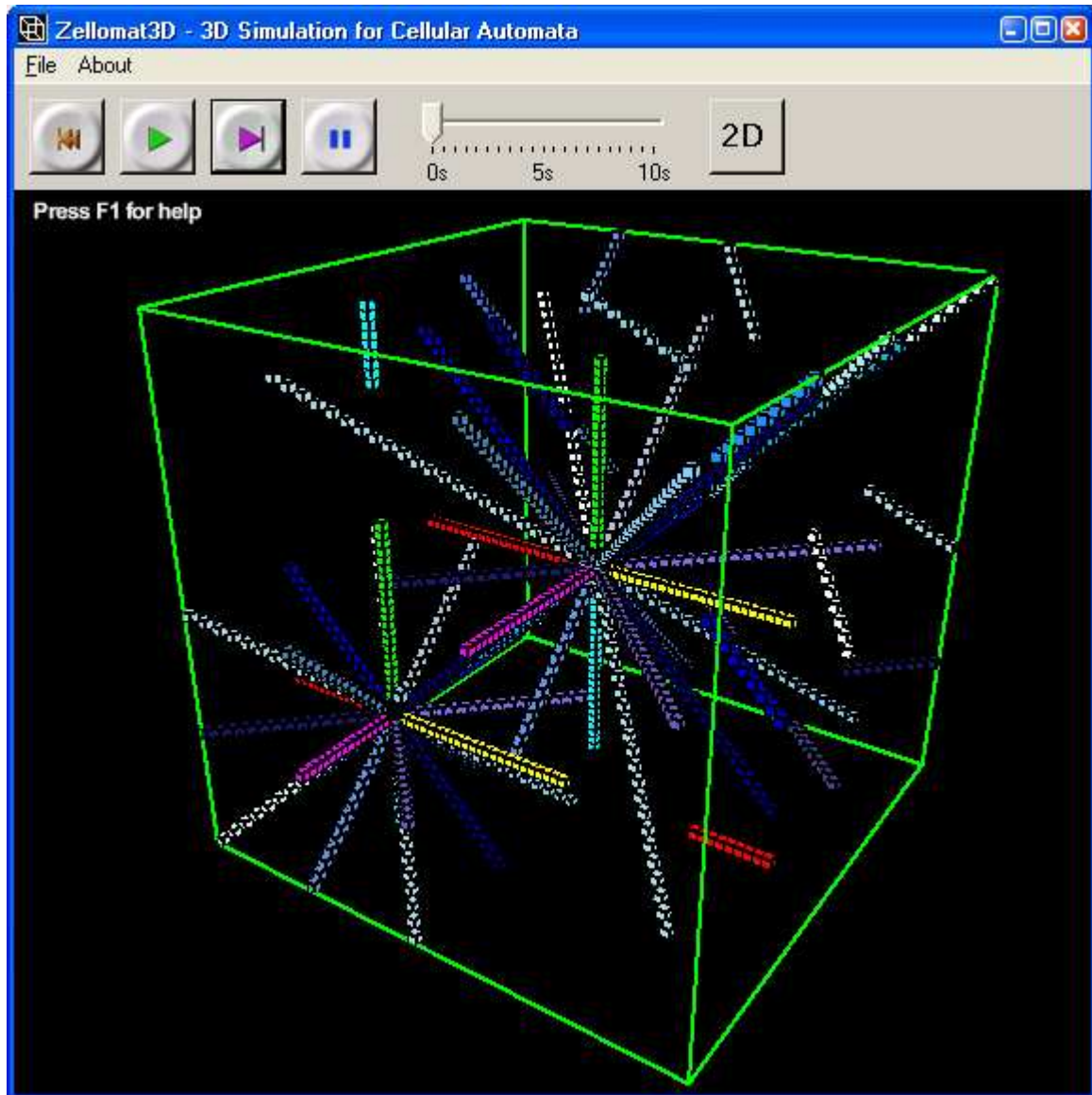


Abbildung 4: Test der der ZA Regeln, Randbedingungen, Farben, sowie die Korrektheit der Nachbarschaftsbeziehungen.



Dem Benutzer muss eine Möglichkeit zur Orientierung in der 3D- Welt gegeben werden. **TC8: Orientierung**

Hintergrundumgebung: **Test erfolgreich**
Koordinatenachsen: **Test erfolgreich**
Lebensraumkubus: **Test erfolgreich**

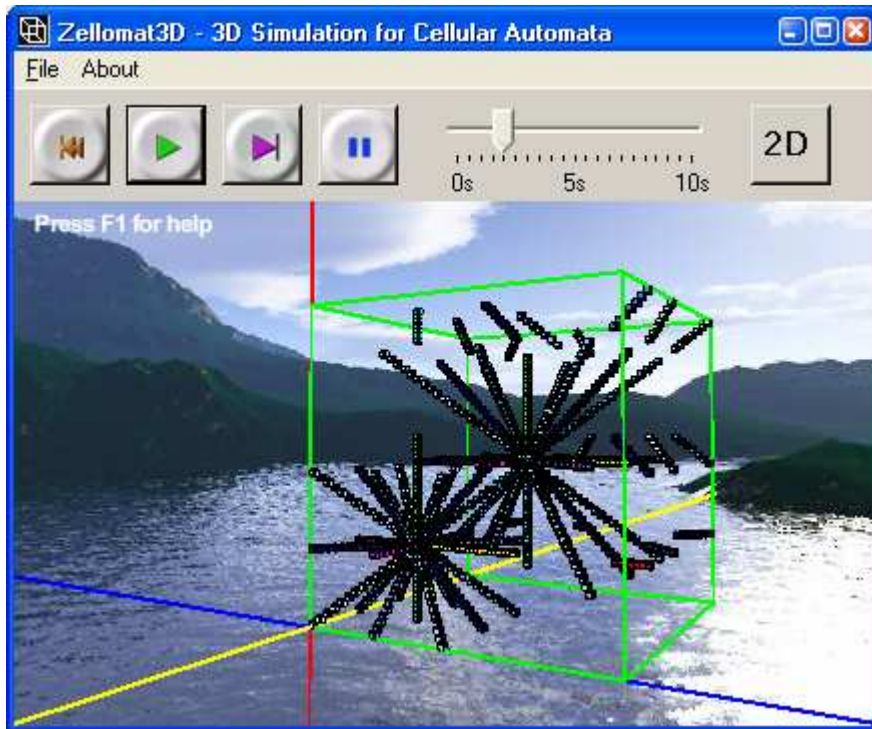


Abbildung 5: Applikation mit aktivierten Orientierungshilfen

Es muss sichergestellt werden, dass die minimalen Hardware- bedingten Grenzen nicht aufgrund falscher Parametrisierung der zellulären Automaten überschritten werden können. **TC9: Parametrisierung**

Überprüfung der ZA-Konfigurationsdatei: **Test erfolgreich**
Snapshot: **Test erfolgreich**
Lebensraumkubus: **Test erfolgreich**



5. Test der Use Cases

Der User lädt einen zellulären Automaten in die Applikation.

TC10:
Öffnen eines ZA

Menü **Test erfolgreich**

Fehlererkennung in der Beschreibungsdatei des ZA und Fehlermeldung an den Benutzer was genau falsch war:

Test erfolgreich

Der User oder die Applikation kann einen zellulären Automaten, wieder auf seinen Initialzustand setzen.

TC11:
Reset ZA

Menü **Test erfolgreich**

Button **Test erfolgreich**

Tastatur **Test erfolgreich**

Der User kann die einzelnen Zyklen automatisch nacheinander abspielen lassen.

TC12:
Play ZA

Menü **Test erfolgreich**

Button **Test erfolgreich**

Tastatur **Test erfolgreich**

Der User kann die einzelnen Zyklen von Hand nacheinander anzeigen lassen.

TC13:
Step ZA

Menü **Test erfolgreich**

Button **Test erfolgreich**

Tastatur **Test erfolgreich**

Der User kann die Anzeige der Zyklen pausieren.

TC14:
Pause ZA

Menü **Test erfolgreich**

Button **Test erfolgreich**

Tastatur **Test erfolgreich**

Der User stellt die Zeit ein, die zwischen zwei Anzeigen von Zyklen des ZA vergeht

TC15:
Anzeigeintervall einstellen

Regler **Test erfolgreich**

Tastatur **Test erfolgreich**



Der User schießt einen Schnappschuss des aktuell angezeigten Zyklus. **TC16:
Schnappschuss eines
Zyklus**

Button **Test erfolgreich**
Tastatur **Test erfolgreich**

Der User kann sich Informationen über das Programm anzeigen lassen. **TC17:
Informationen über
das Programm
anzeigen**

Menü **Test erfolgreich**

Der User lässt sich die Hilfe zur Steuerung des Programms anzeigen. **TC18:
Hilfe anzeigen /
ausblenden**

Tastatur **Test erfolgreich**

Der User kann die Umgebung als Orientierungshilfe einschalten / ausschalten. **TC19:
Umgebung anzeigen /
ausblenden**

Tastatur **Test erfolgreich**

Der User aktiviert / deaktiviert die Koordinatenachsen **TC20:
Koordinatenachsen
anzeigen /
ausblenden**

Tastatur **Test erfolgreich**

Der User aktiviert / deaktiviert den Lebensraumkubus **TC21:
Lebensraumkubus
anzeigen /
ausblenden**

Tastatur **Test erfolgreich**

Der User passt die Anzeigeeigenschaften seinen Bedürfnissen an **TC22:
Anzeige
Eigenschaften
verändern**

Menü **Test erfolgreich**

Hinweis: Die Applikation braucht ca. acht Sekunden um einen Moduswechsel vorzunehmen.

Der User wechselt in Vollbild oder Fenstermodus **TC23:
Wechsel in Fenster /
Vollbildmodus**

Menü **Test erfolgreich**
Tastatur **Test erfolgreich**



Der User vergrößert oder verkleinert das Fenster

Fenster **Test erfolgreich**

TC24:
**Änderung der Grösse
des Fensters**

Der User bewegt die Kamera

Maus **Test erfolgreich**
Tastatur **Test erfolgreich**

TC25:
Kamerasteuerung

Der User beendet die Applikation

Fenster **Test erfolgreich**
Menü **Test erfolgreich**
Tastatur **Test erfolgreich**

TC26:
**Beenden der
Applikation**