

Zusatzübung zu Computergraphik II

– Übungsblatt Z –

Lehrstuhl für Computergraphik
und Multimediasysteme

Julian Bader

In dieser Aufgabe geht es darum einige Bereiche aus der Vorlesung weiter zu vertiefen. Dazu wird ein bestehendes Programm sukzessive um Funktionen erweitert. Am Ende soll eine Domino-Simulation entstehen. Darin sollen Steine anhand von mehreren Arten von Kurven gesetzt werden. Die Steine sollen in äquidistanten Abständen auf der Kurve stehen. Wenn man einen Stein umstößt, soll dieser mit benachbarten Steinen kollidieren und diese ebenfalls umstoßen.

Nehmen Sie als Ausgangsbasis für das Programm das auf unserer Website bereitgestellte Programmgerüst `ueb_Z.zip`. Um das Projekt für Ihre bevorzugte Entwicklungsumgebung auszuführen nutzen Sie das beigelegte CMake-Projekt ("`CMakeLists.txt`"). CMake kann von der folgenden Homepage heruntergeladen werden: <http://www.cmake.org/>. Nutzen Sie die Anleitung der Seite <http://www.cmake.org/cmake/help/runningcmake.html> und der Übungsseite um das Projekt zu generieren.

Aufgabe 1 [8 Punkte] Bézierkurven und Splines

Vervollständigen Sie die Klassen `BezierCurve`, `CatmullRomCurve` und `BSplineCurve`, welche Sie in den gleichnamigen CPP-Dateien des Projekts finden:

1. Implementieren Sie die Methode `getBPoints(float u)` in der Klasse `BezierCurve`. Für einen Parameter `u` soll der entsprechende Punkt auf der Bézierkurve berechnet werden.
2. Vervollständigen Sie den Konstruktor der Klasse `CatmullRomCurve`. Hierzu müssen die Tangenten und inneren Kontrollpunkte der Bézierkurvenstücke berechnet werden. Denken Sie auch über eine geeignete Methode zur Bestimmung der Tangenten an den beiden Endpunkten nach. Beachten Sie, dass die Klasse `BezierCurve` funktionieren muss, damit die Catmull-Rom-Splines fehlerfrei gezeichnet werden können.
3. Vervollständigen Sie die Methode `getBPoints(float u)` der Klasse `BSplineCurve`. Nehmen Sie hierzu eine Kurve vom Grad 3 an. Die Methode muss für einen gegebenen Parameter `u` den entsprechenden Kurvenpunkt berechnen.

Weitere Erläuterungen finden Sie in den Kommentarseiten im Code

Nachdem Sie die Klassen vervollständigt haben, sollten die zu Programmstart definierten Kurven denen von Abbildung 1 gleichen.

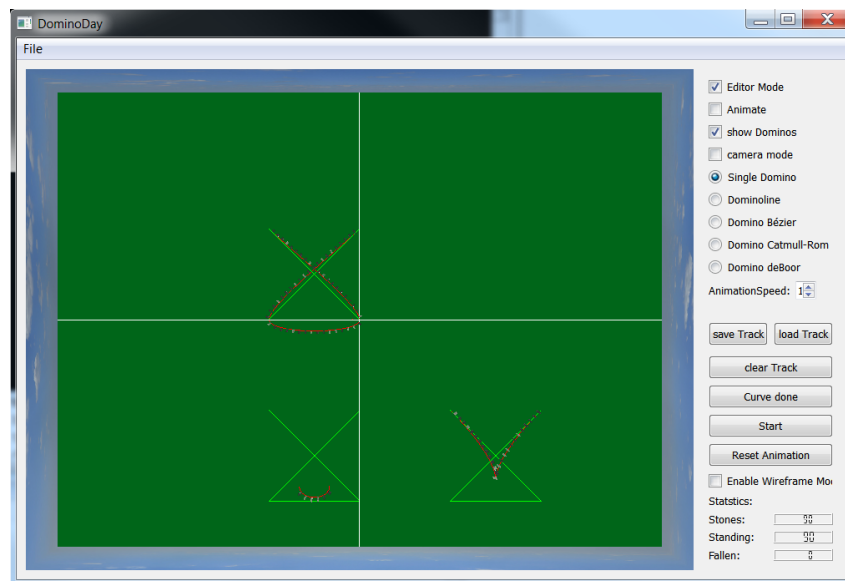


Abbildung 1: Catmull-Rom-Spline, B-Spline und Bézierkurve sollen bei Programmstart so gezeichnet werden.

Aufgabe 2 [5 Punkte] Äquidistante Abstände

Vervollständigen Sie die Methode `getDominos(float stepSize)` in der Klasse `Curve` (`Curve.cpp`). Diese Methode soll die Position und Ausrichtung der Dominosteine auf einer Kurve so berechnen, so dass sie alle den gleichen Abstand zueinander haben (vgl. Abb. 2). Die Länge des Abstandes wird über den Parameter `stepSize` definiert. Greifen Sie hierzu auf Methoden der Spline-basierten Animation zurück (vgl. Vorlesungsfolien: Kapitel 9.2).

Hinweis: Um Kurvenpunkte zu erhalten, müssen Sie die abstrakte Methode `getBPoints(float u)` der Klasse `Curve` benutzen.

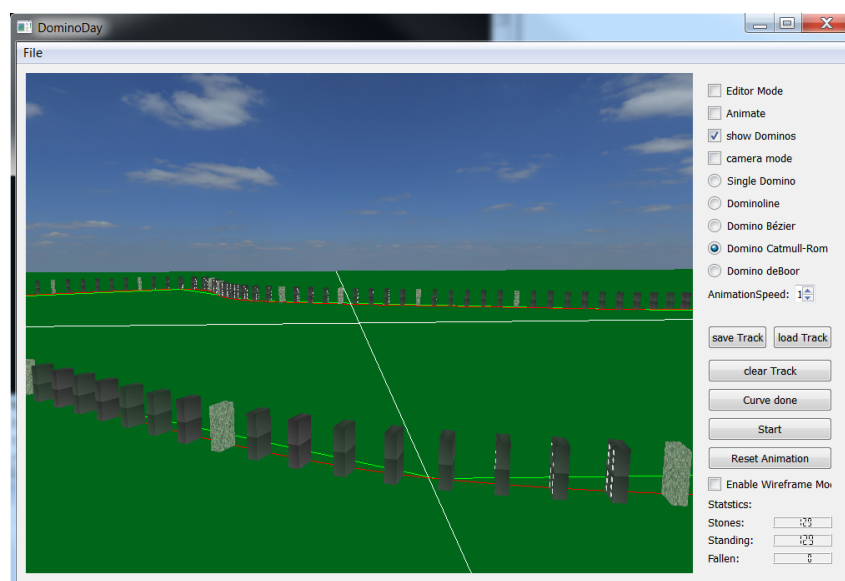


Abbildung 2: Die Dominosteine sollen alle den gleichen Abstand zueinander haben.

Aufgabe 3 [5 Punkte] Kollisionserkennung

Vervollständigen Sie die Methode `detectCollision` der Klasse `DominoManager`. Diese soll für zwei gegebene Dominosteine feststellen, ob diese miteinander kollidieren. Nutzen Sie hierfür die Konzepte der **Object Aligned Bounding Box** (OAB) und der separierenden Ebene. Bei korrekter Implementierung sollten die Steine der Reihe nach umfallen (vgl. Abb. 3).

Hinweis: Mit der Methode `Domino::getCornerPoints()` erhalten Sie die Eckpunkte des Dominosteins im Weltkoordinatensystem. Die Methode `Domino::center()` liefert den Mittelpunkt der Bounding Box. Weitere Informationen zur Benutzung der Methoden können Sie den Kommentaren im Quelltext entnehmen.

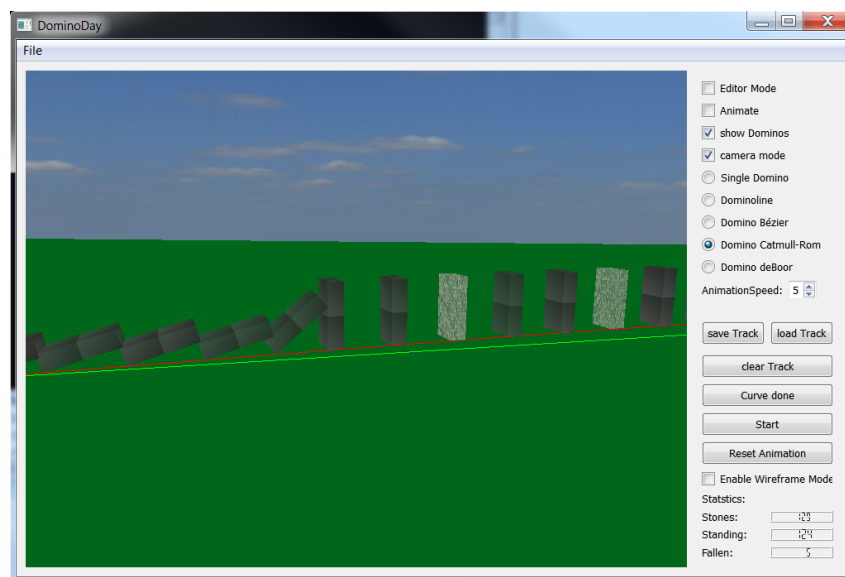


Abbildung 3: Dominosteine sollen miteinander kollidieren und nacheinander umfallen.

Abgabe von Aufgabe 1: Montag, den 10.12.2012, bis 8:30 Uhr

Abgabe von Aufgabe 2 und 3: Montag, den 28.01.2013, bis 8:30 Uhr

Schicken Sie uns sämtliche modifizierte Dateien oder auch das ganze Projekt zu. Wenn wir Ihre Lösung per Mail erhalten haben, senden wir Ihnen so bald wie möglich eine Bestätigung.

→ Email an: julian.bader@uni-siegen.de

In der Woche nach der Abgabe wird eine mündliche Abnahme stattfinden. Hierzu sollten Sie in der Lage sein Fragen zu Ihrer Implementierung und der zu Grunde liegenden Theorie beantworten zu können. Über den Termin zur Abnahme werden Sie zeitnah informiert.

Wichtig: Wenn Sie nach PO 2012 eingeschrieben sind und einen Schein über 10 LP haben wollen, müssen Sie, unabhängig von der regulären Übung, in diesem Übungsblatt mindesten 50% der Punkte erhalten um zur Klausur zugelassen zu werden.