Trimesteroverschrijdend Project: Curve Editor

Groep 13: Sibrand Staessens en Sibren Polders

Donderdag 22 mei, 2008

Inhoudsopgave

1 Voorwoord

Op welke manieren kan je een set van punten op een gladde wijze met elkaar verbinden? Dat was de vraag achter Curve Editor. Het gemakkelijkste pad dat getekend kan worden tussen de punten is natuurlijk lineair. Maar er zijn nog zoveel andere mogelijkheden, waarvan we er slechts twee hebben verwerkt in curve editor (nl. Bezier en Hermite). Deze algorithmes zullen vlakke kromme maken tussen de interpolatiepunten. De toepassingen van de algortitmes die we gebruikt hebben beperken zich niet enkel tot het tekenen van "lijntjes" tussen punten, maar ook op bijvoorbeeld camerabeweging of "AI"-beweging in games. De interpolatie tussen een gegeven set punten is een uitgebreide en interessante studie die op vele vlakken in de informatica/wiskunde zijn nut kan bewijzen. Curve editor geeft er de basistoepassing (ofwel de toegepaste wiskundige) ervan.

"If the path be beautiful, let us not ask where it leads." - Anatole France

2 Wiskundige Voorkennis

2.1 lineair

2.2 Bezier

2.3 Hermite

3 Implementatie

3.1 Packages

3.1.1 Java packages

Een Java package is een mechanisme binnen Java om klassen te organiseren in namespaces. Java broncode die binnen eenzelfde categorie of functie vallen kunnen hierdoor gegroepeerd worden. Dit kan door middel van een package statement bovenaan het beginbestand om aan te geven waartoe ze behoren. Dit is om twee redenen handig omdat de sources gegroepeerd zijn onder hun categorie, wat het geheel overzichtelijker maakt. Verder kunnen er nu twee verschillende klassen eenzelfde naam krijgen en toch uniek bepaald worden door er zijn package name voor te zetten. Wat zeker handig is als de programmeur een klasse dezelfde naam heeft gegeven als een klasse uit een library die hij wil gaan gebruiken.

Voor ons project hebben we een algemeen pakket src gemaakt die verschillende sub-pakketten bevat. Wat duidelijk zichtbaar is in figuur BLAH. In wat volgt wordt een korte beschrijving gegeven van al deze pakketten. Zonder al teveel in te gaan op de technische details.

3.1.2 CurveEditor

Dit is wellicht het kleinste pakket van de reeks (figuur BLAH). Het bevat slechts een klasse, namelijk de main klasse. Deze klasse zal, zoals wellicht duidelijk is, als bootstrap dienen voor de applicatie curve editor. Er wordt ook de mogenlijkheid geboden om al rechtstreeks vanuit de commandline een file mee te geven. Dit is enkel ter volledigheid vermits de gebruiker tijdens de loop van het programma zeer makkelijk bestanden kan inladen en opslaan.

3.1.3 Algorithms

Dit pakket voorziet allerlei klassens die voor de interpolatie tussen punten zullen zorgen. Elke klasse in dit pakket implementeerd de Algorithme interface. Dit is handig voor de groepswerk vermits deze interface vastlegt welke functies de programmeur zal implementeren. Zodoende weet je al op voorhand welke methodes je moet aanroepen om een bepaald resultaat te verkrijgen.
De klassenamen zijn triviaal gekozen: “Lineair, Bezier, BezierC1, BezierG1, Hermite, Hermite Cardinaal, Hermite Catmull Rom”(figuur BLAH). Zoals de namen al verdaden zullen deze de verschillende interpolatie methodes die besproken werden in het deel ’Wiskundige voorkennis’ implementeren.

Hiervoor werd natuurlijk altijd geoogd op de meeste optimale implementatie van degene die besproken werden.

3.1.4 Core

Dit pakket bevat enkele noodzakkelijk klassens(figuur BLAH).

De klasse CurveContainer zal ervoor zorgen dat ingegeven punten kunnen opgeslagen worden, samen met hun door interpolatie berekende punten.

Een eerste idee was het subdivsion principe toe te passen. In de beginsituatie is het tekenveld dan een grote rechthoek. Van zodra de gebruiker een curve begint te tekenen worden de secties waar punten geplaatst zijn onderverdeeld in steeds kleiner wordende rechthoekjes. In elk zo'n rechthoekje zat dan juist een punt van een curve. Zodat er gemakkelijk gezegd kon worden welk punt waar stond en tot welke curve het behoorde.
Dit algorithme bleek echter niet zo efficiënt te zijn wanneer we te maken hadden met een groot aantal input punten. Dit kwam voornamelijk doordat er telkens opnieuwe een kleinere rechthoek moest berekend worden bij ingeven van een nieuw punt. En een grotere wanneer er punten verwijderd werden. Uiteindelijk was de applicatie meer bezig met het berekenen van rechthoekjes dan met zijn doel: voorstellen van curves.

De tweede poging leek beter te lukken. We stelden een veld op waarvan elke pixel een mogenlijke houder kon zijn van een punt. De houders werden geïnitialiseerd met de waarde null zodat ze geen plaats innamen. Het toevoegen van punten is zo simpel uit te voeren door het new commando toe te passen. Verwijderen is dan gewoon de juist holder op null zetten (de garbage collection van java lost de rest op). Het zoeken gaat simpelweg door de positie van de muisklik om te zetten naar coördinaten die gebruikt kunnen worden op de vector waarin alle punten worden opgeslagen. Om daarna in een bepaalde vooraf bepaalde range te kijken of een punt houder niet op null staat, is dat zo dan zal de informatie van dat punt teruggeven worden, zoniet is er op die plaats in het veld geen punt beschikbaar.

De klasse Editor is het hart van de curve editor. Deze zal zorgen dat de data die uitgewisseld moet worden kan en ook in de juiste richting zal stromen. De uitwisseling van data zal voornamelijk bestaan uit het zoeken of verwijderen van punten uit de CurveContainer klasse. Maar ook het opvangen en afhandelen van excepties. Deze klasse zorgt er dus voor dat de verschillende andere klassen zo autonoom als mogenlijk kunnen werken. Dit verhoogd natuurlijk enkel de leesbaarheid en onderhoudbaarheid van de code.

Een laatste klasse van dit pakket is de FileIO klasse, deze zal niet alleen files opslaan en inladen, maar ook zal hij de funtonialiteit van undo en redo implementeren, vermits deze van dezelfde functies gebruik maakt.

3.1.5 Curve

Dit pakket bevat de de twee datatypes die doorheen het programma gebruikt worden. Point zoals de naam doet vermoeden geeft de mogenlijkheid een punt op te slaan. Curve geeft dan weer de mogenlijkheid om een verzameling van punten (lees een kromme of curve) op te slaan. De technische details van deze laatste klasse wordt beter uitgelegd verder in de tekst. Voorlopig is het voldoende om te weten dat Curve een vector van Point's bijhoudt en enkele basisvoorzieningen voorziet (punten opvragen, toevoeg, transleren, ...).

3.1.6 Exceptions

Een foutloos programma schrijven is al een hele opgave, vermits er altijd wel kleine bugs kunnen opduiken na langdurig gebruik. Een fool proof programma schrijven daarentegen is een onmogelijke opgave. Daarom hebben we gebruik gemaakt van exceptions om "verkeerd" gebruik van curve editor op te vangen. Onder verkeerd gebruik valt bijvoorbeeld het inladen van een onbestaande file, of een verkeerd fileformat. Het toevoegen van een punt zonder er de coördinaten van op te geven, ...

Er zijn ook twee HandleException klassen voorzien. Eentje in dit pakket, deze zal gewoon het exceptie bericht in de console uitprinten. In het GUI pakket is een HandleException klasse voorzien die in een dialog scherm de exceptie zal uitprinten.

3.2 Uitwerking van de GUI

3.3 Java Listeners

3.4 Datastructuren

3.5 Extra's

4 Planning

5 Taakverdeling

6 Appendix

A Handleiding

B Screenshot

C Referenties