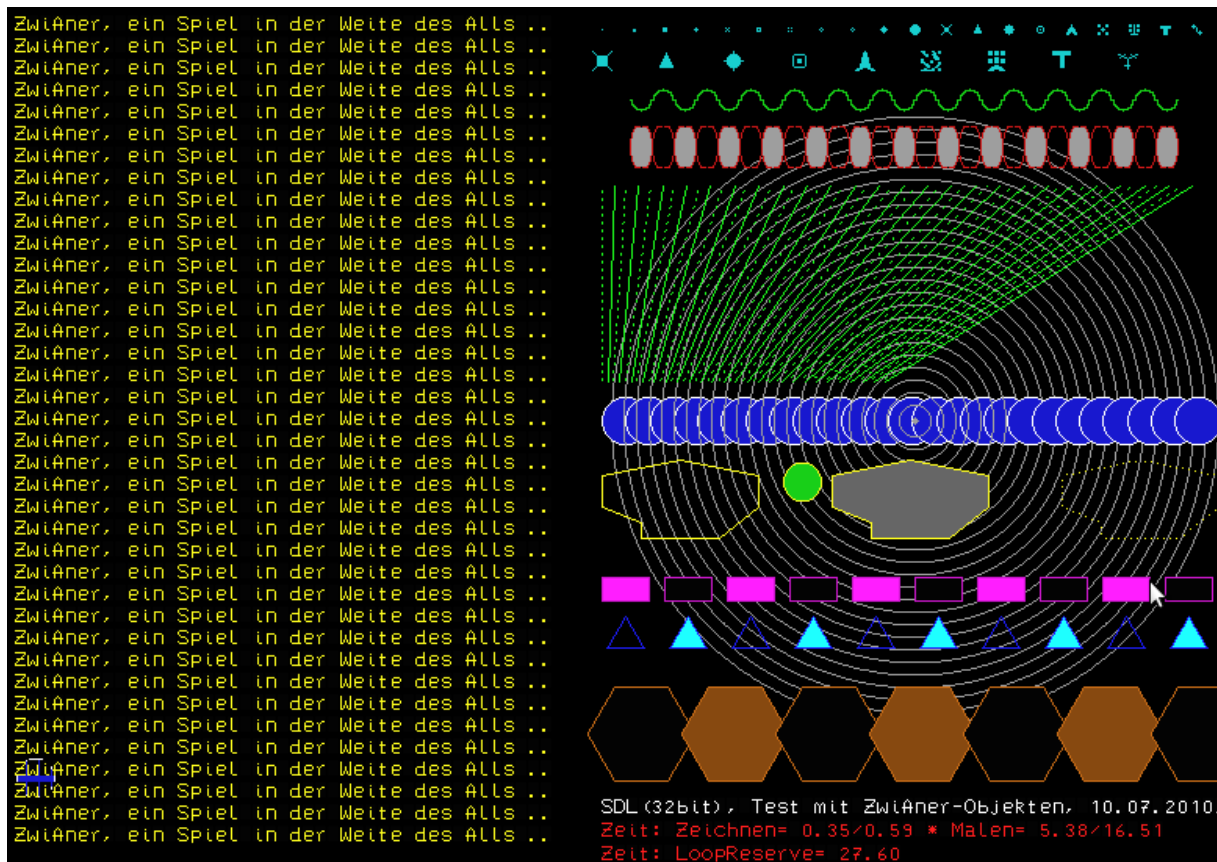


Auf dem Weg von DDraw nach OpenGL

08.05 .. 25.07 2010, ein Erfahrungsbericht von Heinz Kempfer



Im Moment brummt mir der Kopf, wenn ich an die letzten Monate zurückdenke. Ich hatte tausend Fragen und wusste nicht wo anfangen. Aber Dank der tatkräftigen Unterstützung von ‚developia.de‘ und ‚zfx.info‘ bin ich nun am Ende nicht nur bei OpenGL sondern auch in UBUNTU angekommen.

Begonnen hat es damit, dass ich am 18. September 2009, 16:20 die **ZwiAner** in den **Projekten** auf **developia.de** vorstellte, weil ich dachte, mein Spiel sei nun reif für die Öffentlichkeit. Die Resonanz machte mir Mut und ich habe es auf weiteren Internetseiten veröffentlicht.

Nun hatte ich viele Tester und es kam der erste Schreck. Es gab Probleme mit Windows Vista. Ja, damals gab es noch kein Windows 7 und schnell war klar, es lag an DDraw. (Also, ich habe Vista nie gemocht!)

DDraw wird zwar seit DirectX10 von Microsoft nicht mehr separat weiterentwickelt aber die COM garantiert das zeitlose weiterfunktionieren .. so dachte ich damals. Gut, es war nicht DDraw im allgemeinen, sondern nur eine spezielle Anwendung davon, ‚Palettentechnik‘ (8Bit-Grafik), was die Probleme verursachte. Also habe ich mich hingesezt und meine Grafik auf 32bit erweitert. Dabei konnte ich inzwischen dazu gewonnene C++ Kenntnisse einarbeiten und mein Grafiksystem auf eine allgemeinere Basis stellen. Ohne diese Vorarbeit, wäre mir der Weg nach OpenGL nicht in so ‚kurzer‘ Zeit möglich gewesen.

Palettentechnik, ich höre euch stöhnen, wie kann man mit so uraltem Zeug arbeiten? Das hat mehrere Gründe. Mein Ziel war es ursprünglich, meine Kenntnisse in C++ (vor allem Klassentechnik) zu verbessern. Dieses Vorhaben wollte ich aber außerhalb meiner täglichen Berufspraktik durchziehen. Es sollte etwas sein, das mir erlaubte, immer wieder mit neuen Erkenntnissen von Vorne zu beginnen (ohne sich bei Chef oder Kunde rechtfertigen zu müssen).

Und, da habe ich mich entschieden, in meiner Freizeit ein Spiel zu entwickeln. Ja, auch damals (4.4.2000) hatte ich tausend Fragen. Also habe ich Bücher über C++, Bücher über Spieleprogrammierung und Bücher über Daten und Algorithmen gekauft. Unter anderem auch Bücher über DirectX9 von Sirotin und eines von Stephan Zerbst.

Im Buch von Sirotin ging es unkompliziert und direkt zur Sache. In kurzer Zeit konnte ich mit DDraw, DSound und DInput umgehen. Alles schön ‚gekapselt‘ (Klassen) wie ich es lernen und besser verstehen wollte. Beim Lesen von Zerbst habe ich dann den Wunsch entwickelt, alles vom ersten Pixel an selbst zu machen.

OK, nun war klar, wie ich mein Spiel aufbauen werden. Es wird ein Spiel in ‚zwei‘ Dimensionen und die Bewohner werden ‚**ZwiAner**‘ heißen.

Ich bin seit jeher ein Fan von ‚Science Fiction‘ und habe immer davon geträumt, mein Raumschiff in eine Umlaufbahn zu bringen! Und dazu brauchte ich nur ‚Raum‘ und ‚Gravitation‘. Der Raum war begrenzt durch ‚double‘ und die Gravitation durch die vorhandene ‚Rechenzeit‘.

OK, das mit dem Raum ist nur ein Maßstabsproblem, aber die Rechenzeit machte mir Kopfzerbrechen. So habe ich viel getestet und einen Weg gesucht, die notwendige Grafikdarstellung auf ein Minimum an Zeit zu trimmen. Dabei bin ich auf die 8Bit-Grafik, Palettentechnik und die eigenen Grafikroutinen gestoßen. Whow, war das schnell! Und genauso schnell war auch meine Entscheidung gefallen, genauso mach‘ ich‘s! (Man muss dabei noch bedenken, im Jahre 2000 waren die Rechner noch nicht so schnell und die Grafikkarten noch nicht so speziell wie heute.)

Inzwischen ist die 32Bit Variante ohne Palettentechnik nur noch eine kleine Idee langsamer als 8Bit und Palette. Aber bei meinem ersten Spiel bleibe ich bei meinen eigenen Grafikroutinen! Pasta! (Die Entwicklung der Grafik hat soviel Spaß gemacht, dass diese Methode mindestens ein Spiel verdient hat.)

Warum wollte ich eigentlich noch mal von DDraw nach OpenGL? Ja, da waren einige (oder auch einige mehr) untern den Testern, die von so etwas wie Linux geschwärmt haben. Gehört hatte ich schon davon, aber bisher keinerlei praktische Erfahrung damit gemacht. Und unter Linux nimmt man OpenGL, da ist DDraw reines Gift! Ich will doch, dass alle auf der Welt mein Spiel spielen können, Also machte ich auf, auf den Weg von DDraw nach OpenGL .. und hier sind meine Erfahrungen:

Zuerst mal welches Linux? Ich hatte nicht die Zeit und Muße alle Varianten mal durchzuprobieren. Wie soll das auch gehen? Aus dem Bauch heraus habe ich mich für Ubuntu entschieden und es funktioniert. Ich kann heute damit die Dinge erledigen, die ich damit machen wollte. Was will man mehr?

Nach dieser Entscheidung wird es allerdings zuerst einmal sehr kompliziert. Wo und wie bringe ich Ubuntu zum laufen? Aha, da gibt es eine ‚Virtual Box‘, und darin eingesperrt soll Ubuntu dann arbeiten. Auch hier gibt es wieder eine Menge Anbieter und auch hier habe ich mich aus dem Bauch heraus für VMWare entschieden und es funktioniert. Was will man mehr?

Ich kann mich nicht mehr genau an alle Details der Installation von VMWare und Ubuntu 9 erinnern. Aber eigentlich klappte es auf Anhieb. Bis auf die Organisation des Zugriffs auf Laufwerk C:, brauchte ich dazu keinerlei weitere fremde Hilfe. Bei dieser Aktion hatte ich den ersten Kontakt mit ‚sudo‘. Nach jahrelanger Windowserfahrung nun so was! ‚SUDO‘! Gut, das ich noch Freunde habe. Inzwischen habe ich mich auch daran gewöhnt, nein, eigentlich noch nicht.

Ok, Stand der Dinge. Ich kann den Rechner einschalten, VMWare starten und darin mit Ubuntu arbeiten!

Jetzt brauche ich etwas ähnliches wie Visual Studio. Wie schon zweimal zuvor bleibt nichts anders übrig als .. ‚aus dem Bauch heraus zu entscheiden‘ .. und ich habe mich für Code::Blocks entschieden und es funktioniert. Was will man mehr? Einiges! Hier muss ich nochmals sagen: Einiges! Ich komme damit klar, aber wenn man einmal mit Visual Studio 2008 und VC 9.0 gearbeitet hat, vermisst man eine Menge an lieb gewordenen Hilfen. (Vor allem legt man beim Debuggen immer wieder in sinnloser Weise die Maus auf eine Variable .. und nichts passiert.)

Ja, plötzlich ist man in einer anderen Welt. Gewohnte Knöpfe tun nichts oder nicht das was man von ihnen gewohnt ist. Gewohnte Schreibtechniken funktionieren nicht mehr .. man verliert viel Zeit mit Einrichten und Umdenken. Vor allem mit Umdenken ist man auf dem rechten Weg, glaub‘ ich zumindest. Bin ich in Windows mache ich es so, bin ich unter Ubuntu, mache ich es eben anders. Da bleibt der Geist in Bewegung!

Ok, wieder einmal Stand der Dinge. Ich kann Code::Blocks öffnen und fast so arbeiten wie mit Visual Studio.

Aber wie komme ich nun zu OpenGL? Hier braucht man Infos von Google und den Rat von ‚devolopia.de‘ und ‚zfx.info‘. (Manche meinen auch, dass einer von den beiden letztgenannten genügen würde.)

Inzwischen weiß ich, wohin die Post ihren Paketdienst verlagert hat. Nach Ubuntu! Paketdienst, jetzt weiß ich Bescheid. Damit werden fehlende Sachen nachinstalliert. Nein, keine Setup.exe mehr! Am schlimmsten für den Anfänger ist außer des gewaltigen Inhalts, die Machtlosigkeit. Man drückt ein Knöpfchen und es tut sich was. Aber was geschieht, wohin wird

‚paketiert‘ und wieviel davon? Man muss es machen lassen und dann funktioniert es auch. Was will man mehr?

So habe ich dann nach und nach (oder Thread auf Thread) alles zusammen, was ich zum programmieren brauche und los geht's! Denkste!

Nicht nur DDraw ist Gift für Linux, ich glaube absolut tödlich ist das MFC (ATL). Nichts wie zurück in meine Windowswelt. Nie mehr schau' ich nach Ubuntu!

Aber es reizt halt' doch. Nun will ich es wissen. Ich habe meine ersten Programme auf einen TI geschrieben, Magnetkarte, 999 Befehlsschritte. Viele davon sind mit auf den CPC 464 gewandert. Das war ein Fortschritt! Nur wenige habe es auf meinen ersten AT-Rechner geschafft. Das war ein Rückschritt! (Den XT habe ich ausgelassen, musste damals viel arbeiten). Auf DOS und Pascal war ich sehr erfolgreich und konnte meine Familie erfolgreich ernähren. Dann kam Windows. Die Situation war ähnlich wie heute, nur wichtiger. Damals ging es um die berufliche Existenz. Heute „nur“ um ein Spiel. Damals, von DOS nach Windows habe ich es geschafft! Also weiter im Text ..

CObject, gibt es nicht und damit auch kein ‚Serialize‘, jammerschade.

CString, wer's nicht kennt, vermisst es nicht. Aber ich bin daran gewöhnt. Eine Welt ohne CString, nicht denkbar. Soll ich etwa wieder mit char-Zeigern arbeiten?

CStdioFile, muss auch ersetzt werden.

TRACE, wer vermisst schon TRACE. Ich schon!

CPtrList und Co. Alles nicht da!

QueryPerformanceFrequency, natürlich nicht vorhanden.

__DEBUG ist nicht definiert.

#pragma, auch dich vermisse ich.

Hilfe über Funktionen: Bitte Google benutzen oder ‚sudo man‘. Nein, nicht Superman, es heißt ‚sudo man‘.

Und das sind nur die großen Dinge. __max(..) oder __min(..) .. unbekannt.

Die gute Nachricht ist, es gibt für alles einen Ersatz. Die schlechte Nachricht ist, es kommt in manchen Fällen viel Handarbeit auf dich zu.

Z.Bsp.: __max(4.0, 1) kann nicht so einfach durch std::max(4.0, 1) ersetzt werden. Entweder std::max((int)4.0, (int)1) oder std::max((double)4.0, (double)1)! Was wollte ich damals nur im Quelltext vergleichen? Also ohne genaues Hinsehen geht das nicht. Nein, keine Automatik!

Merke, alles was mit ,_' beginnt, das gibt es in der ,neuen', ,anderen' Welt nicht.

Nu zu den Lösungen:

Das fehlende CObject habe ich noch nicht ersetzen können. CString heißt nun CKString und baut im wesentlichen auf std::string auf. Auch CStdioFile habe ich mit CKStdioFile erfolgreich wrappen können. Aber bei CPtrList und Co. musste ich passen. Diese wurden und werden noch durch viel Handarbeit mit std::list und Co. ersetzt, erfolgreich ersetzt!

Für meine Pixel habe ich mit DrawPixels und einem zusätzlichen eigenen Pixel-Array eine brauchbare, wenn auch sehr langsame Lösung gefunden. Sie ist so langsam, dass ich diesen Weg mit OpenGL nicht weiter verfolgen werden. Dafür habe ich eine vergleichbare Lösung mit der SDL gefunden. Diese steht an Geschwindigkeit meinem DDraw in nichts nach. Also OpenGL, ich habe dich zwar gefunden, aber du musst noch auf ZwiAner 3.0 warten. Dann aber, wie die Versionsnummer schon sagt, im vollen 3D ,Look And Feel'.

Das ist also meine Geschichte, von ,Auf dem Weg von DDraw nach OpenGL'.

Den letzten Stand der Dinge kannst du dir downloaden und auf deinem System ausprobieren:

Linux/OpenGL:

<http://www.zwianer.de/Download/TestLinuxKZeichnenOpenGL.exe>

Linux/SDL:

<http://www.zwianer.de/Download/TestLinuxKZeichnenSDL.exe>

Windows/OpenGL:

<http://www.zwianer.de/Download/TestWindowsKZeichnenOpenGL.exe>

Windows/DDraw:

<http://www.zwianer.de/Download/TestWindowsKZeichnenDDraw.exe>

Diese Beispiele zeigen die Leistung meiner Grafikroutinen auf Windows, einmal mit DDraw (Vollbild, aktuelle Auflösung, 32bit oder 8bit) und mit OpenGL (800, 600) oder auf Linux mit OpenGL (800, 600) oder der SDL (800, 600) in der linken unteren Ecke an.

Was kommt als nächstes? Antwort: ,Auf dem Weg von DSound nach SDL' und danach? Antwort: ,Auf dem Weg von DInput nach SDL'!

Und dann, dann kann man die ZwiAner auch auf Linux spielen!

Wer mehr über mich und die ZwiAner wissen will:

<http://www.zwianer.de> oder Heinz.Kempter@ZwiAner.de, Betr. ZwiAner.



25.07.2010 Heinz Kempfer

PS:

Ich würde mich freuen, wenn die Downloads (vor allen für Linux) getestet werden würden. Dadurch bleibt mir dann hoffentlich ein zweites ‚Vista‘ erspart. Und wenn ihr dann noch einen Screenshot, mit Angabe von Grafikkarte, Rechner und Betriebssystem an mich sendet (Heinz.Kempfer@ZwiAner.de), werde ich die Daten in einer Excel-Liste auf ‚developia.de‘ und ‚zfx.info‘ unter meinem Thread veröffentlichen.