

PROJECT BLITZMAG

01/05



INHALT

Community

Clowns und Kekse 2

DeiDe 3

Preview

Texel 7

Test

AntiWar 10

Test zum Jumping Clown Contest 11

Coden

BlitzMax --- Wir programmieren ein Spiel 13



Clowns und Kekse

Wieder einmal – um genau zu sein zum dritten Mal – ist ein Wettbewerb im ProjectBlitz-Forum, bei dem die Coder ihren Fähigkeiten freien Lauf lassen konnten, zu Ende gegangen.

Ziel des Wetteiferns war es, einen Clown mittels eines Trampolins in luftigen Höhen Ballons einzusammelbn bzw. zerplatzen zu lassen.

Nach zweiwöchiger Programmierzeit (die für manche(n) immer noch zu kurz war :) trudelten 3 hübsche und spaßige Exemplare bei Aufsichts- und Durchführungsperson Hot-Bit ein, um dann von der Community begutachtet zu werden. Auch die PBM-Redaktion hat sie genauer unter die Lupe genommen. Mehr dazu im „Test“-Teil.

Sieger aus der Runde von Simi, Shakespine und mir ist letztendlich Simi Gewinner geworden und durfte sich über 20 Euro freuen.

Doch genug des Lobes; kommen wir nun zum zweiten Teil dieses Artikels: Contests im Allgemeinen und die Teilnahme an ihnen.

Virtuelles Kräftermessen mit anderen Blitz-Programmierern ist eine sehr schöne, spaßbringende, aber auch sehr lehrreiche Angelegenheit. Egal wie das Ergebnis ausfällt: Man kann nur gewinnen: Eine Fülle an unschätzbar wertvollem Wissen steht mit nahezu jeder Contestteilnahme ins Haus, und wenn man diese gut umgesetzt hat, vielleicht auch noch Ruhm und Ehre und/oder ein attraktiver Preis.

Dennoch schienen sich aus der PB-Community eher wenige dazu angehalten am JumpingClown-Contest teilzunehmen.

Zwar hatten sich im Vorfeld viele angemeldet und lauthals ihre Teilnahme verkündet, so viel die Auswahl zur Stunde der Wahrheit mit 3 Beiträgen doch recht mager aus. - (Justus)

DreiDe

Die freie deutsche **BlitzMax-3dEngine**

BlitzMax ist schon seit einiger Zeit zu haben, doch ein offizielles 3D-Modul von BB-Erfinder Mark Sibly lässt noch auf sich warten. Für BlitzMax-Besitzer, die das 2D-Geplänkel nicht mehr aushalten, ist jetzt eine erste Leidenslindeurung vorhanden:

Community-Mitglied Vertex hat auf ein neues mit seinen Programmierkünsten überrascht und präsentiert seit einiger Zeit sein Selfmade-3D-Modul, das er liebevoll „DreiDe“ taufte.

Belächelnde Blicke sind angesichts folgender, sehr überzeugender Features fehl am Platze:

MIPMapping

- Spheremapping
- Cubemapping (ungetestet)
- Multitexturing (funktioniert noch nicht ganz)
- Dot3 Bumpmapping

3DS-Loader

und und und...

Weiterentwickelt wird (zumindest im Moment) sehr schnell, was auf baldigen 3D/DreiDe Nachschub hoffen lässt.

Dazu einige Worte des Meister persönlich:

PBM: „Vorab: Eine kurze Selbstbeschreibung, bitte ^^“

Vertex: Ich heiße Oliver Skawronek, bin 18 Jahre alt, lebe in Altenburg (Thüringen, in der Nähe von Leipzig) und mache z. Z. meine Ausbildung zum techn. Assistenten für Informatik. Berufsmäßig habe ich so in Richtung Spieleprogrammierung oder Prozessorentwicklung gedacht. Freizeitmäßig: Freundin, Freunde, Partys und Saufen.

PBM: „Vorab2: Eine kurze Beschreibung für Leute wie mich, die bei B+ stecken geblieben sind und keine Ahnung von BMax haben: Was macht



DreiDe und was macht DreiDe so geil?

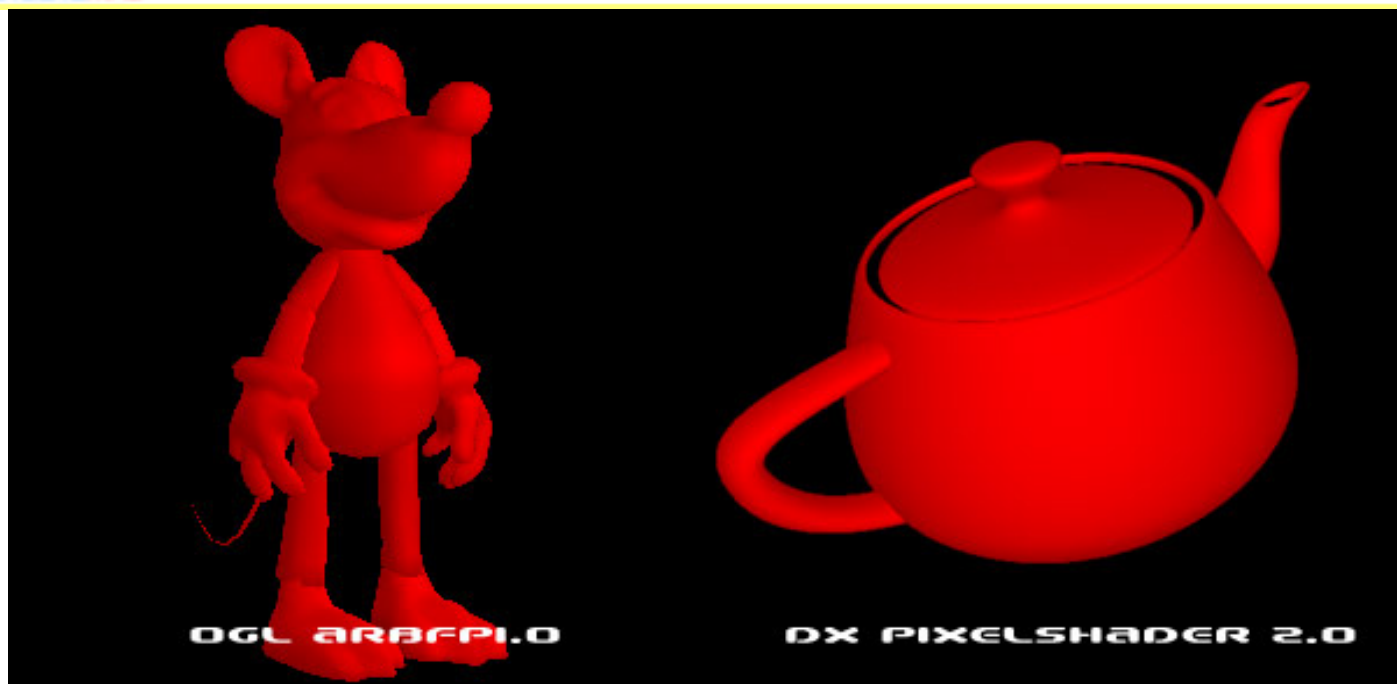
Vertex: DreiDe ist eine 3D Engine speziell für Spielegrafik. Sie ist so geil, weil sie auf Windows, Linux und MacOS läuft, Killfeatures wie Cubemapping, Bumpmapping, Vertex/Pixelshader usw. besitzt, Open Source, und recht leicht zu erlernen ist.

PBM: „Wie kamst du auf die Idee DreiDe zu programmieren?“

Vertex: Das erste fertige Netzwerkmodul für BlitzMax war meine Schöpfung, also wollte ich auch einer der ersten sein, der eine vernünftige 3D Engine auf die Beine stellt.

PBM: „Wie motivierst du dich selbst dazu, immer weiter zu machen?“

Vertex: Es ist hammergeil, wenn ein selbstgemachter Grafikeffekt vor der Linse flimmert. Mein Wissensdrang hinter die Kulissen zu blicken und das ganze zu verste-



hen ist unerschöpflich, darum bietet es sich an, das ganze in eigenen Projekten selbst umzusetzen. Wenn ich mir Grafikfeatures von Spielen ansehe, will ich diese auch selber umsetzen. Es ist ein Gefühl der Überlegenheit etwas machen zu können, wofür Leute hoch bezahlt werden(dennoch will ich

auch bald zu diesen Leuten gehören 😊). Ansonsten liegt es nahe, für sein Berufswunsch so viel Erfahrungen zu sammeln, wie möglich.

PBM: „Was möchtest du mit DreiDe erreichen?“

Vertex: Wenn ich mich dann nächstes Jahr für ein Praktikum in einer IT-Firma bewerben muss, brauche ich was zum Vorzeigen. Ansonsten will ich natürlich viele viele Spiele von DreiDe-Usern sehen.

PBM: „Wie lange programmierst du schon an DreiDe?“

Vertex: Zu Beginn von Version 1.00 (DreiDe hieß damals noch "Sitio", span. für "Raum") hatte ich mir dann auch die ersten OpenGL Kenntnisse aneignen können, die über das üblich Anzeigen eines bunten Dreieckes hinaus gingen. Das war so Mitte März 2005. Dann war es ersteinmal still um DreiDe. Mich hat dann im Juli doch die Lust wieder gepackt und ich habe alles von Vorne begonnen.

PBM: „Du hast ja bereits einige andere Programme und BB-Erweiterungen höherer Programmierkunst geschrieben - eine kleine Zusammenfassung deiner bisherigen Errungenschaften bitte!“

Vertex: Ich glaube mit meiner BASS-Userlib hat alles so begonnen. Auch wenn manch Schandmaul anderer Meinung ist, ich habe Rythmus im Blut, und mit BASS konnte ich endlich auch mal Objekte nach Musik auf dem Bildschirm tanzen lassen.

Da ich in Übung war, Userlibs zu entwickeln, habe ich dann auch gleich Devil übersetzt. Ist eine kleine Imagebibliothek die viele Bildformate laden und speichern kann.

Als nächstes kam eine Software-Rendering-Engine. Sie macht praktisch das, was eine 3D Grafikkarte leistet. War nur Wireframe, aber sicherlich interessantes Projekt. Wenn man mal an Quake1 zurück denkt... Da ist das Retroherz auf Extasy.

Wer sich etwas näher mit Blitz3D/2D auseinander gesetzt hat, wird wissen, das der Zugriff in Pixelebene ziemlich lahm ist. Also habe ich mit anderen Kollegen begonnen, Blitz auseinander zu nehmen, direkt gesagt zu hacken. Plötzlich war es möglich, das man z. B. eine DLL schreiben

konnte, um direkt Bilder seiner Webcam in ein Image zu übertragen. Das ganze wurde dann doch etwas zu weit getrieben, und es wurden Sachen gehackt, die völlig sinnlos waren. Schlussendlich habe ich dann ein BlitzHacker geschrieben, der aus Blitz erstellte Exe-Dateien alle Variablen, Funktionen usw. auslesen konnte.

Eine Zeit lang war ich in einem Team genannt "TriStar" die noch gute alte Retrodemos produziert hatten. Ein Kollege "Pixel" steht auf die alten C64 Sounds, und da gibt es die SID Musikdateien die er in Blitz abspielen lassen wollte. Also habe ich mich dran gesetzt, einen Player für das Zeug zu coden. Mir war bis dahin nicht bewusst, das SID Dateien Programmdateien wie Exe-Dateien sind. Man musste die C64 CPU und den SID Chip der im C64 drin war, emulieren! Also setzte ich mich ran, und entwickelte einen C64 CPU Emulator mit Blitz. Leider stellte sich dann tausende Zeilen Code später heraus, das die komplette Adressierung falsch emuliert wurde. Aber von nun an wusste ich, wie ein Emulator arbeitet, und habe wichtige Assemblerkenntnisse erhalten.

Mit diesen Kenntnissen wollte ich dann eine ultimative Hochsprache, sprich Compiler á la Blitz für einen kleinen günstigen Handheld genannt PMini entwickeln. Naja zu wenig Zeit gehabt, um was vernünftiges daraus zu machen. Es ist dann doch nur beim gepushten Lexer geblieben.

BlitzMax flog Weihnachten 2005 in mein Haus. Was neben einer 3D Engine fehlte, war ein Netzwerkmodul. Also Kenntnisse zum Thema Sockets gesammelt und mich dran gemacht, alle UDP- und TCP-Befehle von Blitz-Basic nach BlitzMax zu portieren. Das gute Ding habe ich dann "BNet" getauft. Auf Windows lief es recht gut, doch eine Portierung auf Linux und MacOS musste her. Hier entstand mein Hass auf Linux und meine Meinung, dass Linux garnicht Open Source ist (nicht ernst nehmen). BigMichi hat mir dann aber später angeboten, das ich auf seinem



Eine Bumpmaptexur verleiht diesem alten Mann ein realistisches Gesicht.

Linuxrechner programmieren kann über ein Remotecontrol. Das ganze war erfolgreich und konnte mit einer kleinen Änderung dann auch auf MacOS portiert werden. Kurz darauf kam dann Marks Socket Modul heraus, wobei ich meins geiler finde

PBM: „Seit wann programmierst du in BB?“

Vertex: Ich glaube, etwas später, als Blitz3D released wurde.

PBM: „Wie bist du zu BB gekommen?“

Vertex: Hier sei SucoX gedankt, der mich aus den Klauen von DarkBASIC befreit hat

PBM: „Was denkst du sind die größten Stärken/Schwächen von BB/ BlitzMax?“

Vertex: BlitzBasic ist eine Sprache, die man mal programmiert haben sollte. Generell sollte es in Schule gelehrt werden (zur Hölle mit TurboPascal!!!). Sie ist einfach, bugfrei, stabil und performant. Hier wurde großes Potential ausgeschöpft. Mit Blitz3D gehts dann richtig los!

BlitzMax: Wer objekt orientiert programmieren will, der ist sicher der Meinung, das er mit C++ oder C# besser beraten sei, zumal das kostenlose Sprachen sind. Die Module von BlitzMax machen diese Sprache aber so interessant. Auch wenn ich kein Freund von Mark's Codestil bin, ist es super, das er die Module Open Source lässt. Die Help kann man getrost wegwerfen, aber die Sourcecodes erklären alles.

Generell ist BlitzMax für mich die ultimative Sprache, und das kann ich nach über 12 erlernten Sprachen mit Sicherheit sagen.



Texel

Ich fliege mit meinen Kameraden weit draußen im Universum Patrouillie. Mit unseren High-Tec Raumschiffen, die bis zu mehrfache Lichtgeschwindigkeit an Speed zulegen können ein Schmauß. Alles ist so perfekt, das Antriebssystem welches weniger Kraftstoff verbraucht als jemals zuvor, aber auch das modulare Waffensystem, das es erlaubt je nach gebrauch Waffen an meinem Raumschiff anzubringen.

Dafür werden Droiden benutzt, die aussehen wie ein kleines leuchtendes Ei. Benutze ich es, legt es sich wie maßgeschneidert an meinen Flieger an und verstärkt meine Kraft um das vielfache.

Versunken in Gedanken bemerke ich nicht, wie sich feindlich Gruppierungen von Raumschiffen an uns anpirschen und uns angreifen. Mein Team kommt nach einem Minutenlangen Kampf sehr gut davon, aber mich triffts am Heck und ich manövriere in die Atmosphäre des nächsten Planeten.

Mein Verfolger bleibt dich hinter mir. Umso näher ich den nebeligem Boden komme, desto mehr erkenne ich. Es gibt Bäume, es sieht belebt aus. Ich fliege nur noch wenige Meter über dem Boden als mich mein Verfolger erneut trifft. Diesmal an meinen Degravitationsmodul. Es ermöglicht mir entgegen der Schwerkraft zu fliegen. Mein Verfolger setzt sich ab und ich verliere ihn aus dem Auge.

“Weiß er etwas über diesen Planeten?” denke ich mir. Wieder zu mir gefunden sehen ich vor mir mehrere außerirdische Wesen auftauchen. Ich muss einen Platz finden mein Degravitationsmodul zu reparieren. Ich glaube nicht, das es mir die Aliens hier leichter machen werden *seuftz*.

So oder anders könnte die Vorgeschichte des spannenden Spaceshooters TEXEL sein.
Der Namenlose Held in einem Raumschiff wird nach altem Prinzip durch die wunderschön gestalteten Levels gesteuert. Ein Horizontalshooter wie man ihn seit langen nicht mehr gesehen hat.
Dermaßen detaillierte Texturen und Animationen sind selbst bei Vollprestiteln mangelware wie es genügend Beispiele zeigen.

Die Levels sind voll von schönen Bäumen, brütend heißer Lava und atemberaubend gut animierte Aliens. Diese verfügen über eine KI, die Formationen ausführt und auf den Spieler schießt. Schön geblendete Explosionen machen die Sache rund.

Schöne Upgrades wie Waffen und Extraleben ver-süßen das Spielen und bringen gleichzeitig eine Menge Abwechslung mit. 4-5 verschiedene Waffen, die verschieden zu benutzen sind. Mal schießt sie nach hinten, mal nach Vorne, schräg u.Ä..

Viele verschiedene Aliens je nach Terrain müssen sie besiegen. Es warten schwere End- und Zwischengegener auf sie die am Ende eines jeden Terrains sind.

Aber auch Eigenschaften bzw. Features, die Verbesserungswürdig sind müssen angesprochen werden. Als erstes fällt auf, dass das Spiel wirklich nichts für Gamer mit niedriger Frustgrenze ist. Nach einem Treffer wird ein Leben abgezogen was es wirklich schwer macht die Level zu bestehen. Außerdem wird die Tastenbelegung zur Steuerung des Raumschiffes nicht gespeichert. Bei jedem Start des Programms muss die Steuerung neu eingestellt werden, wenn nicht die Standardkonfigurierte benutzt werden möchte. Ich hoffe diese Mängel werden in den folgenden Versionen berücksichtigt und bereinigt.

Rundum ist TEXEL ein gelungener Shooter mit viel Action und einer Menge Geschwindigkeit. Er befindet sich – was ja zu bedenken ist – in einer sehr früher Version. Auch wenn das Spielprinzip langsam ein wenig ausgelutscht ist macht es trotzdem Spaß das

Spiel zu spielen. - IMurDOOM

Interview mit Crack

PBM: Vorabfrage: Einen Satz zu deiner Person, bitte!

Crack: Zu mir, mein Name ist Gregor bin 32 Jahre alt und bin Alleinerziehender Vater . Meine sonstige Interessen ausser Coden und Grafik sind: Kampfsport und mit meinem Mountainbike fahren.

PBM: Mit welchem BlitzBasic "Ableger" (B+,B3D,BMax) programmierst du TEXEL?

Crack: Ich benutze für Texel BlitzPlus und die ExtendedBlitzLib von Patrick Lester.

PBM: Wie lange programmierst du schon mit BB?

Crack: Es müssten bald 2 Jahre sein, fast eineinhalb Jahre auf dem PC.

PBM: Und wie lange sitzt du bereits an TEXEL?

Crack: Die ersten Ideen und Grafiken sind schon über 10 Jahren alt, noch auf dem guten alten AMIGA. Ich war nur der Grafiker und hatte mit Programmieren nichts am Hut. Der Programmierer wollte nicht mehr daran Weiterarbeiten und so lag das Projekt lange Zeit auf Eis. Danach erstellte ich fast nur Raytracing Bilder. Die Meisten waren von der Auflösung her zum Ausdrucken gedacht, einige verkaufte ich auch (z.B. Rendergrafiken mit Fotos von Haustieren, Kinder,...) in meinem Bekanntenkreis.

Vor ca. 2 Jahren entdeckte ich den kostenlosen Download von AmiBlitz und begann mit den alten Grafiken TEXEL selbst zu Programmieren. Kurze Zeit später bekam ich ein Mainboard mit einer 450MHz AMD-Cpu geschenkt und baute mir einen PC auf, klar das ich mir dann BlitzPlus kaufte und auf dem PC mit dem Projekt begann.

PBM: Eine große Stärke von TEXEL ist u.a. die Grafik - wer hat die so gut hinbekommen und mit welchen Programmen?

Crack: Die Grafiken sind auch von mir, ich benutze Cinema 4D V6, Particle Illusion SE, Paintshop Pro 7 und World Builder Pro 2.3.0. Die Grafikprogramme sind alle von Heft-CDs.

PBM: Die Demo ist in ihrem Umfang noch recht beschränkt. Welche Features werden uns im fertigen Spiel erwarten?

Crack: Es wird 5 Level geben, in jedem min. 10 "normale" Gegner, je ein Mittel und Endgegner. Vier Extrawaffen und ein Armor, nach einem Treffer von einem Gegner wird kein Leben abgezogen und die Extrawaffe nicht abgestuft (hab ich aus Galaga Deluxe). Die Geschwindigkeit des eigenen Raumschiffes ist in 4 Stufen einstellbar.

PBM: Was gefällt die besonders gut an Blitz Basic und was sollte verbessert werden?

Crack: Die leichte Erlernbarkeit und die (glaube ich, den ich kenne keinen direkten Konkurrenten von BlitzBasic) hohe Geschwindigkeit der erzeugten Programme.

PBM: Wie bist du zu BB gekommen?

Crack: Das steht im Großen und Ganzen schon in Punkt 3. - (Justus)



Screenshots aus dem Spiel Texel





Zitat von der AntiWar-Homepage <http://awar.de.vu>: „AntiWar ist ein simples Ballerspiel. Böse Ami-Flugzeuge greifen dich und deine Brüder an. Jetzt heisst es sich zu verteidigen.“

AntiWar

Bereits die Betaversion des Shooterspiels AntiWar schlug ein wie eine Bombe:

Selbst ich hämmerte gerne auf meiner Maus herum, um die kleinen niedlichen Turbanträger vor den bösen Flugzeugen zu beschützen.

Vor bösen **US**-Flugzeugen?

Diese Frage stellen sich auch einige englische BB-Programmierer und es entbrannte eine hitzige Diskussion.

Schließlich wurde das Spiel sogar von der BlitzBasic.com-Hauptseite entfernt, was die Debatte zusätzlich anfachte:

Es kam bei einer Mehrzahl der Mitglieder der englischen Community die Meinung auf, AntiWar sei nicht nur Anti War, sondern auch Anti America.

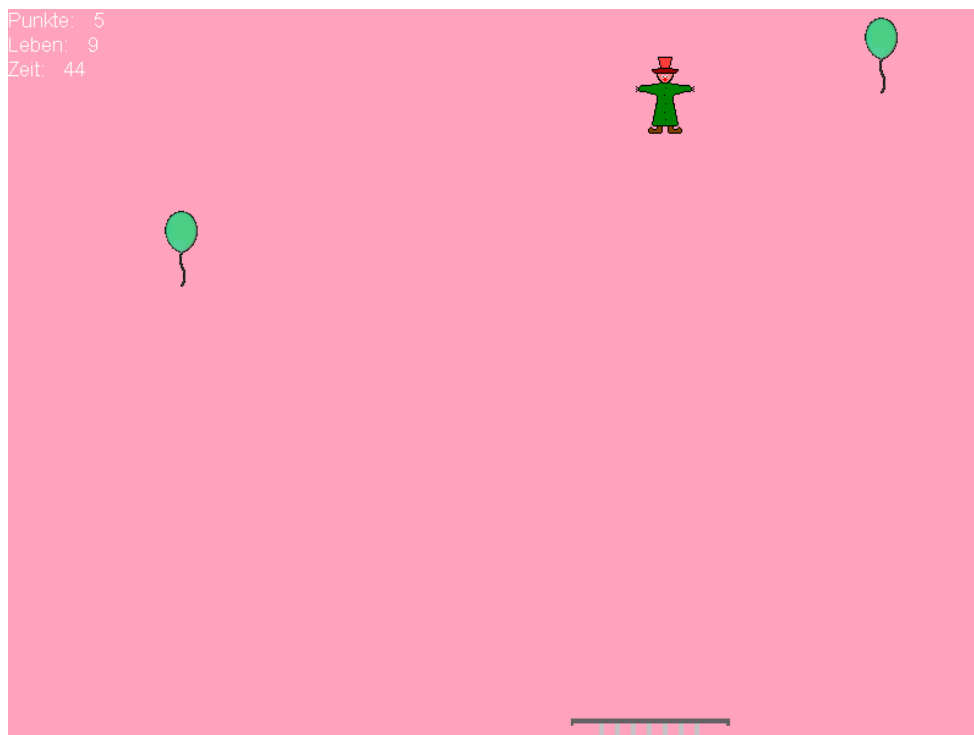
Leider machte es AntiWar-Programmierer mangoos mit seiner Aussage, Amerika sei paranoid und sicherlich kein Land der Freiheit nicht gerade besser.

Später revidierte er seine Aussage jedoch und erklärte, er habe damit nicht ganz Amerika ansprechen wollen.

Wer die Diskussion mitverfolgen möchte, kann dies unter <http://blitzbasic.com/Community/posts.php?topic=49302> tun.

Doch trotz aller inhaltlicher Kritik ist AntiWar eines der besten BlitzMax-Spiele, die es bisher gibt – über den politischen Gehalt des Spiels darf sich jeder seine eigene Meinung bilden. - (Justus)

Test zum Jumping Clown Contest



Spiel von Justus

Der Beitrag von Justus zum Jumping Clown Contest ist leider noch unfertig. Das Spiel ist in Level, welche sich an der Geschwindigkeit der Luftballons unterscheiden, unterteilt. Beim ersten Level brauch man nur die 60 Sekunden warten dann ist das Level geschafft und man bekommt trotzdem an die 20 Luftballons. Diesen Trick kann man aber ab dem 3. ,manchmal sogar ab den 2. nicht mehr verwenden weil man ,wenn die Luftballons den linken Bildschirmrand berühren , Leben abgezogen bekommt. Die stimmige Musik passt zum Spiel.



Spiel von Spikespine

Das Spiel von Spikespine ist anders als die anderen Beiträge : Statt das einem ein „Game Over“ angezeigt wird wenn der Clown stirbt werden einem Punkte abgezogen., denn das Spiel ist immer nach 90 Sekunden zu Ende. Es gibt Power-ups wie z.B. 10 Sekunden mehr zeit oder mehr Luftballons . Beim Trampolin kann man den Neigungswinkel ändern ,so kann man den Clown besser steuern .



Spiel von Simi

Das Spiel von Simi ist ein sehr ausgeglichener Beitrag. Anders als bei den anderen Beiträgen kommen die Luftballons nicht von der Seite sondern einzeln von unten. Kleinere Bugs wie das der Clown ,wenn er gegen die wand trifft immer schneller wird oder das die Luftballons an der wand hängen bleiben und so keine anderen Luftballons mehr kommen trüben den Eindruck. Was aber durch die schöne Musik schnell vergessen wird.

BlitzMax --- Wir programmieren ein Spiel

Über:

Um euch BlitzMax ein wenig näher zu bringen, haben wir uns vorgenommen ein Spiel in den kommenden Ausgaben des ProjectBlitzMags zu entwickeln bzw. fortzusetzen.

Das Spiel wird komplett OpenSource und der Quelltext wird komplett in diesem Mag stehen.

Als Beispiel für einen leichten Einstieg in BlitzMax habe ich ein Spiel gewählt, das sich auch in anderen Programmiersprachen leicht umzusetzen lässt, ein Space-Invaders Klon.

Informationen:

Für die erste Ausgabe habe ich erst einmal nur das "Grundgerüst" des Programms geschrieben.

Es beinhaltet:

- Bewegung des eigenen Spielers
- Bewegung der gegnerischen Spieler
- Schießen des eigenen Spielers
- Explosionen werden erstellt

Programmiermethoden:

Die gegnerischen Spieler werden via Types gehandelt genau wie die Schüsse und die Explosionen wobei es bei den Explosionen bereits 3 verschiedene Arten gibt.

Explosionen:

Alle Arten werden bei verschiedenen Vorkommnissen ausgelöst. Welche Explosion erscheinen soll wird über eine ID weiter gegeben. Die Explosionen werden über LIGHTBLEND gezeichnet. Ist das Framemaximum erreicht, wird die Explosion gelöscht.

Gegner:

Die Gegner werden nach einer bestimmten Zeit neu erstellt. Standardmäßig alle 750ms. Diese bewegen sich in dieser Version nur abwärts. Werden sie – standardmäßig – 2 mal getroffen, wird die Geschwindigkeit *-1 gerechnet und fangen an sich zu drehen. Nach einer Sekunde drehen wird eine Explosion erstellt und das Alien wird kleiner. Ist es auf den Wert 0 gestreckt worden, wird es gelöscht.

Player:

Das Funktionsprinzip des Spielers ist leicht. Er soll sich an der unteren Kante des Bildschirms bewegen und sich nach MouseX() ausrichten. Mit einem Mausklick wird ein neuer Schuss erstellt, der bei einer Kollision mit einem Alien oder beim Austreten aus dem Bildschirm gelöscht wird.

Das Spiel:

Lässt sich mit Quellcode inkl. Mediadateien und EXE von

www.d-dreams.com/blitzmag/game1ausg.zip

herunterladen.

Quelltext:

```
Graphics 800,600,0,60
'SetGraphicsDriver GLMax2DDriver()
SeedRnd MilliSecs()
```

'Zum einstellen...

```
Global cfg_spawnTime=750
Global cfg_scrollY=2
Global cfg_alienHealth=2
Global cfg_debug=1
```

'ARRAYS

```
Global img_explosion[3]
```

IMAGES

SetMaskColor 0,0,0

img_background=LoadImage("gfx/stars.bmp")

SetMaskColor 255,0,255

img_alien =LoadImage("gfx/alien.bmp")

;MidHandleImage img_alien

img_player =LoadImage("gfx/player.bmp")

;MidHandleImage img_player

img_shot =LoadImage("gfx/shot.bmp")

img_explosion[1]=LoadAnimImage("gfx/explosion_1.bmp",100,100,0,25) ; MidHandleImage img_explosion[1]

img_explosion[2]=LoadAnimImage("gfx/explosion_2.bmp",100,100,0,25) ; MidHandleImage img_explosion[2]

SetMaskColor 0,0,0

/IMAGES

TYPES

Liste für den Type T_Alien deklarieren

Global L_Alien:TList=CreateList()

Type T_Alien

Field x

Field y

Field health:Float

Field speed

Field dead

Field angle

Field deadtime

Field scale:Float

Function NewAlien(vx,vy,vs,vh=1)

S_Alien:T_Alien=New T_Alien

S_Alien.x=vx

S_Alien.y=vy

S_Alien.health=vh

S_Alien.speed=vs

S_Alien.dead=0

S_Alien.scale=1

ListAddLast(L_Alien,S_Alien)

End Function

End Type

Liste für den Type T_Shot deklarieren

Global L_Shot:TList=CreateList()

Type T_Shot

Field x

Field y

Function NewShot(vx,vy)

S_Shot:T_Shot=New T_Shot

S_Shot.x=vx

S_Shot.y=vy

ListAddLast(L_Shot,S_Shot)

End Function

End Type

Liste für den Type T_Explosion deklarieren

Global L_Explosion:TList=CreateList()

Type T_Explosion

Field id

Field x

Field y

Field frame

Function NewExplosion(vid,vx,vy)

S_Explosion:T_Explosion=New T_Explosion


```

    S_Explosion.x=vx
    S_Explosion.y=vy
    S_Explosion.frame=0
    S_Explosion.id=vid
    ListAddLast(L_Explosion,S_Explosion)
End Function

Function UpdateFrame()
    For s_explosion:t_explosion=EachIn l_explosion
        s_explosion.frame:=+1
        If s_explosion.frame>=25 Then ListRemove(l_explosion,s_explosion)
    Next
End Function
End Type
'TYPES

'TIMES
Global time_spawn=MilliSecs()
Global time_fps=MilliSecs()
Global time_expl_update=MilliSecs()
'TIMES

'OTHER
Global scroll_y
'OTHER

'MAINLOOP
Repeat
    FlushMem()
    'DRAWING

    TileImage img_background,0-MouseX()/2,0+scroll_y*2
    TileImage img_background,100-MouseX()/4,0+scroll_y*1

    'Aliens
    SetBlend Alphablend
    tmp_alien:=0
    For S_Alien:T_Alien=EachIn L_Alien
        'Update der Gegnerposition
        s_alien.y:=s_alien.speed

        'Drawing
        SetRotation s_alien.angle
        SetScale s_alien.scale,s_alien.scale
        DrawImage img_alien,s_alien.x,s_alien.y
        SetScale 1,1
        SetRotation 0
        'Aliencounter erhöhen
        tmp_alien:=+1

        'Collisionsmap schreiben
        If s_alien.dead=0 Then CollideImage(img_alien,s_alien.x,s_alien.y,0,0,1)
        'Collisionsmap schreiben

        'Bei tot...
        If s_alien.dead=1 Then
            s_Alien.angle:=+5
            If MilliSecs()>=s_alien.deadtime+1000 Then
                If s_alien.scale>=1 Then t_explosion.newexplosion(3,s_alien.x,s_alien.y)
                s_alien.scale:-0.01
                If s_alien.scale<=0 Then ListRemove(l_alien,s_alien)
            End If
        End If
        'Bei tot...

```

```

'Deleten wenn das Alien y>=600 hat
If s_alien.y>=600 Then ListRemove(L_Alien,s_alien)
Next
SetBlend maskblend
'/Aliens

'Shots
tmp_shots=0
For S_Shot:T_Shot=EachIn L_Shot
    'Shotcounter erhöhen
    tmp_shots:=+1
    'Position updaten
    s_shot.y:=-5

    'Collisionen überprüfen
    If CollideImage(img_shot,s_shot.x,s_shot.y,0,1,0) Then
        For S_Alien:T_Alien=EachIn L_Alien
            If ImagesCollide(img_shot,s_shot.x,s_shot.y,0,img_alien,s_alien.x,s_alien.y,0) Then
                'Dem Alien 1 Lebenspunkt abziehen
                s_alien.health:-1
                If s_alien.health>=1 Then t_explosion.newexplosion(1,s_shot.x,s_shot.y)
                'Hat das Alien unter einem Lebenspunkt
                If s_alien.health<=0 Then
                    s_alien.dead=1
                    s_alien.speed=-s_alien.speed/2
                    s_alien.deadtime=MilliSecs()
                End If
                'Der Schuss soll Standardmäßig gelöscht werden.
                ListRemove(L_Shot,S_Shot);Goto end1
            End If
        Next
    End If
    'Collisions überprüfen

    'Schuss zeichnen
    DrawImage img_shot,s_shot.x,s_shot.y

    'Außerhalb des Bildschirms: Löschen
    If s_shot.y<=-10 Then ListRemove(L_Shot,s_shot)
    #end1
Next
'/Shots

'Eigenen Spieler zeichnen
DrawImage img_player,MouseX(),600-55

'Explosionen zeichnen
SetBlend lightblend
tmp_expl=0
For s_explosion:t_explosion=EachIn l_explosion
    tmp_expl:=+1
    Select s_explosion.id
    Case 1
        SetScale 0.5,0.5
        DrawImage img_explosion[1],s_explosion.x,s_explosion.y,s_explosion.frame
        SetScale 1,1
    Case 2
        SetScale 0.5,0.5
        DrawImage img_explosion[2],s_explosion.x,s_explosion.y,s_explosion.frame
        SetScale 1,1
    Case 3
        DrawImage img_explosion[2],s_explosion.x,s_explosion.y,s_explosion.frame
    End Select
Next
SetBlend maskblend

```

```
'Explosionen zeichnen
```

```
'HUD und sonstiges zeichnen
```

```
If cfg_debug=1 Then
```

```
    DrawText "Aliens: "+tmp_aliens,10,505
```

```
    DrawText "Shots: "+tmp_shots,10,520
```

```
    DrawText "Explosions: "+tmp_expl,10,535
```

```
    DrawText "FPS: "+fps,10,550
```

```
End If
```

```
'HUD und sonstiges zeichnen
```

```
'DRAWING
```

```
If KeyHit(1) Then
```

```
    T_Shot.newshot(MouseX()-22,481)
```

```
    T_Shot.newshot(MouseX()+22,481)
```

```
End If
```

```
If MilliSecs()>=time_spawn+cfg_spawnntime Then
```

```
    time_spawn=MilliSecs()
```

```
    T_Alien.newalien(Rand(700),-50,Rand(5),cfg_alienhealth)
```

```
End If
```

```
If MilliSecs()>=time_expl_update+10 Then
```

```
    time_expl_update=MilliSecs()
```

```
    t_explosion.updateframe()
```

```
End If
```

```
If MilliSecs()>=time_fps+1000 Then
```

```
    time_fps=MilliSecs()
```

```
    fps=flips
```

```
    flips=0
```

```
End If
```

```
scroll_y=(scroll_y+cfg_scrolly) Mod 200
```

```
ResetCollisions()
```

```
Flip;flips=flips+1
```

```
Cls
```

```
Until KeyHit(key_escape)
```

Das wars mit der ersten Folge von unserem Projekt. In dem nächsten Magazin werden wir weiter an diesem Projekt arbeiten. Schickt uns Leserbriefe wie es weiter gehen und was weiteres hinzugefügt werden soll. Ich werde versuchen sie in der dann nächst folgenden Ausgabe mit einzubauen.

Was soll noch rein?

- KI? Gegner schießen und weichen aus
- Shöne Partikeleffekte
- Ein ansprechendes HUD
- Eine Highscoreliste
- Ein Hauptmenü
- Upgrades

Wir werden nicht alle diese Themen in der nächsten Ausgabe ansprechen bzw. einprogrammieren aber das ist eine kleine Übersicht von dem, wie das Spiel später aussehen soll.

IMurDOOM