

# PROJECTBLITZ MAG

Das Magazin der deutschen BlitzBasic-Community

# don't panic!

**SUPER MARIO**  
Deathmatch

STARTRADE



## Inhalt

	Seite
<u>Community</u>	
Startrade, von kurz nach früher bis jetzt	3
Daran arbeiten unsere Blitzer	8
<u>Test</u>	
Framework Assistant	10
Super Mario Deathmatch	11
Don't get Angry 2	12
	13
<u>Coden</u>	
Onlinehighscore mit BlitzMax und PHP	17
BlitzMax --- Wir programmieren ein Spiel - Teil 2	21

## Startrade, von kurz nach früher bis jetzt

### Ein Interview mit Fetze

**hamZta**

Grüße, Fetze.

**Fetze**

hi auch ^^.

**hamZta**

Gleich mal im Voraus, danke, dass du dir die Zeit nimmst um für das PBM über StarTrade zu reden.

**Fetze**

Kein Problem, hab ja schließlich genug davon  
\*g\*

**hamZta**

Das ein Schüler mal so was sagt... Sachen gibt's :) Naja, genug abgeschweift, wir wollen ja schließlich über dein Projekt, StarTrade reden. Die erste News auf deiner Website wurde am 26.1 2003 geschrieben. Startschuss für StarTrade?

**Fetze**

Nein, StarTrade gibt es schon rund 2 Jahre länger. Die erste StarTrade-Alpha wurde noch mit VisualBasic geschrieben und war in nahezu allen Hinsichten mehr oder weniger lächerlich. Ich hab dieses „Ur-ST“ auch nie veröffentlicht, daher setze ich den „eigentlichen“ Startschuss dort an, wo ich damit angefangen hab, die Blitzbasic-Version von StarTrade zu schreiben. Das dürfte irgendwann in 2002 gewesen sein

**hamZta**

Mit Visual Basic? Gibt's dieses „Ur-ST“ denn noch?

**Fetze**

Ich hoffe nicht \*g\*

**hamZta**

Na, wäre doch sicher interessant die zwei Versionen miteinander zu vergleichen ;) Der Grundstein von ST liegt also in Visual Basic... oder existiert die Idee auch auf einem Blatt Papier? Hast du mit Zettel und Stift begonnen?

**Fetze**

Nein, ein Konzept hat es nie gegeben. Ich hab einfach drauf los programmiert und das ganze nach und nach in die Form gebracht, die ich haben wollte. Das funktioniert auch erstaunlich gut, wenn man es richtig angeht.

**hamZta**

Erstaunlich, ein Projekt ohne Konzept so lange durchzuhalten ;) Erzähl doch mal wie du dir StarTrade zu Beginn vorgestellt hast!

**Fetze**

Meine Vorstellung von dem, was am Ende dabei herauskommen sollte, war immer sehr schwammig. Ich hab einfach immer als nächstes das eingebaut, worauf ich gerade Lust hatte. Das hat sich mit der Zeit und wachsender Resonanz natürlich verändert, mittlerweile achte ich auch darauf, was die Leute in ihren Kommentaren so über dies und jenes schreiben. Das geht jedoch immer nur bis zu einem bestimmten Punkt, nämlich so weit, bis sie nicht mehr mit meinen eigenen Vorstellungen vereinbar sind. Wie dem auch sei. Für das Gameplay gab es immer zwei Vorbilder, beides alte DOS-Spiele: Das eine war StarHammer, das andere Privateer. StarHammer war ein reines Weltraum-Actionspiel, bei dem man nach und nach Missionen à la „Flieg die Wegpunkte ab und schieß auf alles, was dir in den Weg kommt“ absolvierte, das ganze aus der Vogelperspektive und mit toller, Situationsdynamischer Hintergrundmusik. Das andere Vorbild, Privateer, war ebenfalls ein Weltraumspiel, allerdings bereits in gefakter 3D-Grafik. Man konnte entweder der Storyline nachgehen, Handeln oder sich mit der Kopfgeldjagd Geld verdienen. Von beiden Spielen hat StarTrade ein paar Dinge geerbt. Von StarHammer die Vogelperspektive und die situationsdynamische Hintergrundmusik und von Privateer das Genre. Vielleicht auch die Pixelgrafik, die ich in ST um jeden Preis erhalten will.

**hamZta**

Gab's dabei auch irgendwelche, vielleicht kuri-

osen Ideen, die später wieder verworfen wurden?

### Fetze

Verworfen Ideen gibt es wohl annähernd genauso viele wie umgesetzte. Aber falls kuriose darunter sind, fallen sie mir gerade nicht ein, tut mir Leid \*g\*

### hamZta

Erzähl doch mal ein wenig über die VisualBasic Version! Hatte ST schon so ein großes Universum wie heute?

### Fetze

Es gab 5 hardgecodete Stationen, keinerlei Ini-Dateien, 3 Raumschiffotypen, keine Waffen oder Ausrüstungsgegenstände, 4 festgelegte Waren und vielleicht etwa 20 KI-Händler. Es war wohl das statischste Spieluniversum, das man sich vorstellen kann. Nein, warte, ich glaube, es waren sogar 4 Raumschiffotypen. Selbstverständlich hardcoded.

### hamZta

Hehe, Also sozusagen das Gegenteil von der aktuellen StarTrade Beta. Wann hast du das Spiel im Blitzforum vorgestellt?

### Fetze

Das muss kurz nachdem ich mit der Blitzbasic-Version begonnen hatte gewesen sein. Von „vorgestellt“ kann aber kaum eine Rede sein, da ich zu der Zeit ein ziemlicher Internet-n00b und noch dazu 13 war und mich dementsprechend verhielt... ähm... reden wir nicht drüber. ^^

### hamZta

\*lacht\* Gut, ich frag nicht weiter ;) War die Resonanz eigentlich sofort positiv?

### Fetze

Im Gegenteil. Es lag mit ziemlicher Sicherheit auch an meinem damaligen Verhalten, aber die Resonanz war durchweg negativ.

### hamZta

Ich stell mir gerade vor, wie der junge Fetze dem Blitzforum sein Meisterwerk vorstellt :) Wurde ST, wie man sagt, „in der Luft zerrissen“ oder gab's auch gutes Feedback, Hinweise und Tipps?

### Fetze

Naja, es war damals nicht wirklich ein Meisterwerk. Abgesehen von einem bisschen Handel, einem unvollständigen HUD und ausschließlich handelnden NPCs gab es nichts. Nicht mal Waffen. Ich glaube, ich bin noch relativ gut weggekommen, da dürften tatsächlich einige ernst gemeinte Tipps dabei gewesen sein.

Und, um das mal anzumerken, ob es tatsächlich ein Meisterwerk ist, sei mal dahingestellt. \*g\* Mir gefällt's jedenfalls ^^

### hamZta

Und nicht nur dir, wie mir scheint! 80 angemeldete Benutzer im StarTrade-Forum sprechen für sich. Wann begann denn die Community rund um StarTrade zu wachsen?

### Fetze

Nachdem ich das Forum eingerichtet hatte - das erste war ein werbeverseuchtes Rapidforum - haben sich nach und nach ein paar Leute dort angesammelt, die zwar selten, aber regelmäßig da waren. Und da ich selber quasi permanent im damals noch recht leeren Forum war und mich auf jeden neuen Beitrag gestürzt hab, hatten Neulinge auch nie lange auf eine Antwort zu warten, was einen natürlich dazu motiviert, noch mal hereinzuschauen.

Nach und nach haben sich dann eben immer mehr Leute angesammelt. Größtenteils typische 1-Post-Kandidaten, aber auch ein „stabiler Kern“, der regelmäßig postet. Mir fällt gerade auf, dass ich die eigentliche Frage gar nicht konkret beantwortet hab. Ich müsste das erste Forum zum Start der Website eingerichtet haben, also am 26.01.03.

### hamZta

Dieser „stabile Kern“.. inwiefern haben dir User die geholfen?

### Fetze

Zunächst einmal haben sie das Forum aufrechterhalten, ich musste mir keine Sorgen machen, dass dort nach und nach wieder Ruhe einkehrt. Man konnte und kann sich immer nett mit ihnen über verschiedenste Themen unterhalten und ein gewisses Niveau bleibt im Normalfall immer erhalten. Gibt natürlich auch den einen oder anderen Spammer \*g\*

Abgesehen davon hatte ich immer Feedback, wenn ich gerade zu einem bestimmten Feature oder Thema welches gebraucht hab. Das



ist einfach was, worauf ich mich verlassen kann. Wenn ich mir in einem wichtigen Gamedesign-Aspekt unsicher bin und sonst nirgendwo jemand seine Meinung kundgibt, in meinem Forum bekomme ich genug davon.

**hamZta**

Also der Traum eines jeden Programmierers. Sozusagen.

**Fetze**

Könnte man so sagen, aber man sollte das nicht verallgemeinern \*g\*

**hamZta**

Diese Community entstand sicherlich auch, weil StarTrade „modbar“ ist, also dank den ini-Dateien selbst für Nichtprogrammierer leicht zu verändern ist. Wann fiel dir denn ein, die Daten auszulagern?

**Fetze**

Das hab ich, seit ich mit Blitzbasic angefangen hab, grundsätzlich bei jedem Projekt gemacht. Die Idee ist aber nicht direkt von mir, wie ich zugeben muss. Vielleicht fällt dem ein oder anderen eine gewisse Ähnlichkeit mit den Ini-Dateien aus den „Command & Conquer“-Teilen 2 bis 4 auf. Die Command&Conquer-Serie war bis zu Teil 4 mein absoluter Favorit in Sachen Echtzeitstrategie und irgendwann hab ich dann beim zweiten Teil die Ini-Dateien entdeckt und angefangen zu modden. Modden rockt. Ich mag Spiele, die jeder auf Anhieb verändern oder sogar mit eigenen Inhalten erweitern kann. Und dementsprechend versuche ich eben, StarTrade zu einem solchen zu machen, was mir scheinbar gelungen ist, denn es gab bereits mehrere Mods, darunter auch die ein oder andere Total Conversion für StarTrade. Leider wurde bisher kein Mod fertig gestellt, was sehr wahrscheinlich auch daran liegt, dass das Originalspiel bisher nicht einmal fertig ist \*g\*

**hamZta**

Lass uns mal über die Technik hinter ST reden. Könnte man den Aufbau als „modular“ bezeichnen?

**Fetze**

Überhaupt nicht. StarTrade ist innerlich immerhin teilweise rund 3 Jahre alt und als ich damit begonnen hatte, war meine Erfahrung in Sachen

Programmiertechniken noch stark begrenzt. Ich finde hin und wieder alte Passagen, bei denen es mir kalt den Rücken herunter läuft. Erst vor ein paar Monaten hab ich das letzte „Goto“ entfernt. Ein heilloses durcheinander. Aber man könnte sagen ich bin auf dem besten Weg der Besserung.

**hamZta**

Ein Projekt ohne Konzept. Ein Programm ohne durchdachten Aufbau. Ich hör schon die Leute schreien: „Wie? Das geht doch nicht!“. Sehr zukunftsreich hört sich das nicht an. Naht Rettung?

**Fetze**

Jap.

**hamZta**

\*lacht\* Das lässt viel Platz für Spekulationen, danke Fetze ;) Verwendet StarTrade denn irgendwelche herausragenden Techniken? (Anmerkung d. Redaktion: Im Zeitraum zwischen Interview und Release dieser PBM-Ausgabe hat Fetze in seinem [Forum](#) bereits bekanntgegeben, worum es sich handelt )

**Fetze**

Nein, wohl kaum. Aber es ist denke ich erwähnenswert, dass StarTrade das Universum nicht nur um den Spieler herum berechnet. Es ist nicht einfach wie bei diversen anderen Spielen, wo man einen Gegenstand ablegt, 10 Meter weg läuft und den Gegenstand nicht mehr vorfindet. Wenn man in StarTrade ein Handelsschiff zerstört und die Waren und Trümmer herumfliegen, dann sind die nicht verschwunden, wenn man kurz weg war. Jedenfalls nicht einfach so. Es könnte natürlich sein, dass in der Zeit ein Pirat vorbeigekommen ist und sich über herumliegende Waren gefreut hat. Diese würde man dann allerdings an einem in der Nähe liegenden Piratennest im Angebot vorfinden, denn dort hat er sie zuvor aller Wahrscheinlichkeit nach verkauft. Es sei denn natürlich, er fiel auf dem Weg einem ihm feindlichen Allianz-Raumschiff zum Opfer. Alles wird simuliert.

**hamZta**

Bleiben wir gleich beim Thema „KI“ und „NPCs“. Im Spiel fliegen ja Piraten, Händler und Polizeikräfte herum. Steckt dahinter eine ausgetüftelte KI?

**Fetze**

Für KI gilt normalerweise, dass sie gar nicht

so intelligent ist, wie sie aussieht. Das gilt auch für StarTrade. Jedes Raumschiff hat nur ein paar simple Verhaltensmuster. „Patrouilliere“, „Kämpfe“, „Ruf Hilfe“, „Handle“, „Rüste dich neu aus“ und noch ein paar mehr sind es. Je nach Situation wählt ein NPC eine dieser Aktionen aus und

führt diese mit mehreren „Unteraktionen“ aus. Kämpft ein Schiff beispielsweise mit einem anderen, wird es dieses zunächst ins Visier nehmen und feuern, wenn eine Trefferchance besteht. Ist es zu nah am Ziel dran, fliegt es erst einmal auf Distanz, um sich dann umzudrehen und neu anzufliegen. Das war's schon. Daraus entstehen aber

teilweise wirklich fordernde Kämpfe. Die KI umfasst allerdings auch komplexere Bereiche wie z.B. den intelligenten Handel. Ein Händler bewertet zunächst sämtliche im Spiel vorkommenden Stationen auf Kauf- und Verkaufspreise, setzt diese in Beziehung zu seinem Laderauminhalt und zur Umgebung der jeweiligen Stationen („Riskiere ich einen Ausflug zum Piratennest, weil es da echte Schnäppchen gibt?“) und trifft am Schluss eine Gesamtentscheidung.

#### hamZta

Lass uns mal näher auf die Piraten eingehen. Kaufen, bzw. verkaufen diese auch Waren?

#### Fetze

Piraten rüsten sich neu aus, wenn es notwendig wird, reparieren ihr Schiff und halten immer Ausschau nach feindlichen Schiffen oder umherliegenden Waren. Daraus ergibt sich ganz von selbst das Verhalten, Händler

anzugreifen und danach das auszusammeln, was von ihrem Laderauminhalt übrig geblieben ist. Das verkaufen sie anschließend im nächstgelegenen Piratennest. Es gibt auch einen Piratentyp, der sich auf das Sammelverhalten spezialisiert hat. Er jagt Händler, hält sie zunächst mit einem Traktorstrahl fest, zerstört ihn anschließend mit seinem

Waffenarsenal und hält dann im Trümmerfeld an, wo er mehrere Dronen startet, die die Trümmer aufsammeln und in den Laderaum des Sammlers bringen. Ist der voll, das Trümmerfeld leer oder ein Feind in der Nähe, holt er seine Dronen zurück und begibt sich je nach Laderauminhalt ent-

weder wieder auf Patrouillie oder auf den Weg zum nächsten Piratennest.

#### hamZta

War das schon immer so? Wie sah denn die KI in der VisualBasic-Version aus?

#### Fetze

In der Visualbasic-Version gab es so gut wie keine KI. Die Händler zeigten schon ihr Auswahlverhalten, was allerdings noch nicht so komplex war wie jetzt. Und Piraten oder Allianz-Raumschiffe gab es nicht. Als ich mit der Blitzbasic-Version von ST begonnen hab, war die KI natürlich auch erstmal wieder bei Null. Aber nach und nach hab ich sie eben erweitert und letztendlich auf den jetzigen Stand gebracht.

#### hamZta

Ich stöbre gerade nebenbei im Forum und lese wie ein stolzer Spieler erzählt er habe „mit einem Catcher 6 Mammut 1-3 geschrottet“. Weiter unten fragt einer nach einem Multiplayermodus, eine Antwort gibst du in dem Thread aber nicht. Wie sieht's denn mit einem Multiplayermodus aus?

#### Fetze

Ich hab nicht geantwortet? Muss ich übersehen haben. Normalerweise ist meine Antwort „Nein“. Wird es wohl auch erstmal bleiben, was vor allem daran liegt, dass es ein Riesenaufwand wäre, wenn technisch überhaupt jetzt noch möglich. Abgesehen davon halte ich es für besser, erstmal einen fertigen Singleplayer abzuliefern als einen unfertigen Singleplayer und einen nicht ausgereiften Multiplayer. ..aber da ich selber sehr gerne einen Multiplayermodus für StarTrade hätte, werde ich mich sofort dranmachen, wenn es realistische Chancen auf Erfolg und ein gutes Aufwand/Nutzen-Verhältnis gibt ;) ...was aber leider, wie ich zugeben muss, nicht sehr wahrscheinlich ist.

#### hamZta

Warum findest du das nicht sehr wahrscheinlich?

#### Fetze

Es ist eben leider nicht sehr wahrscheinlich innerhalb eines akzeptablen Zeitraumes



Ein Screenshot aus der Visual-Basic-Version von StarTrade ...

ohne jegliche Multiplayer-Erfahrung mit einer unstrukturierten, nicht darauf ausgelegten Engine einen funktionierenden Multiplayermodus hinzubekommen.  
^^

## hamZta

Laut dir dürfen wir ja noch hoffen, es naht Rettung, wer

weiß, vielleicht bekommen wir eines Tages unseren Multiplayermodus ;)

Ich bin dir für dieses Interview wirklich dankbar, bessere Antworten kann man sich eigentlich gar nicht wünschen.

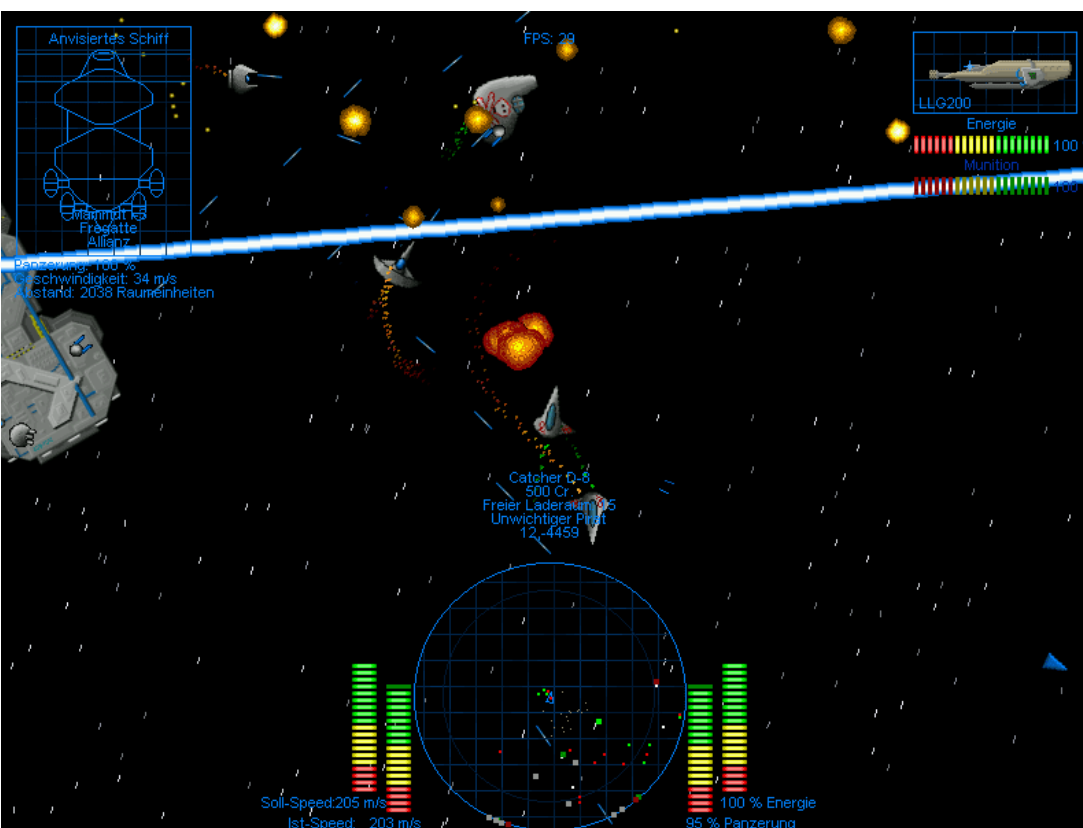
Ich denke, dass dieses Interview den Lesern gefallen wird, mir hat's Spaß gemacht und ich hoffe dir auch.

Danke Fetze!

## Fetze

Kein Problem, hat Spaß gemacht ^^

Ich wünsch euch noch viel Glück mit dem PBM :)



... und einer von der heutigen BlitzBasic-Version



## Daran arbeiten unsere Blitzer

**Latte!** von Muck



Im nächsten Jahr gedenke ich Latte! zu veröffentlichen, einen Fußballmanager, wie zum Beispiel das gute alte Anstoss 3. Im Gegensatz zu anderen Spielen des Genres soll Latte! nicht in Arbeit ausarten, sondern Spaß machen. Mein Grundsatz lautet "komplex - aber nicht kompliziert". Man kümmert sich hier wirklich nur um den Managerpart, dadurch entstehen aber auch neue Möglichkeiten wie die Interaktion mit dem Trainer. Bei einigen Sachen habe ich mich von anderen Spielen inspirieren lassen, aber natürlich auch ein paar frische Ideen mit eingebracht.

Das Spiel befindet sich nun bereits seit einem halben Jahr in der Entwicklung. Dabei mache ich fast alles alleine, nur die Musik hat je-

mand anders beige-steuert (Glück gehabt!). Anfang September konnte eine kleine Demo fertig gestellt werden, eine Art Grundgerüst, in dem schon wichtige Dinge wie Spielertransfers und ein Textmodus für die Spiele integriert sind. Seitdem wurde zwar fleißig weitergearbeitet, aber es müssen trotzdem noch einige größere und kleinere Spielelemente eingebaut werden - u.a. mehr Ligen & Vereine oder die gleichzeitige Berechnung aller Spiele für Live-Ergebnisse. Auf [www.fussball-management.com](http://www.fussball-management.com) könnt ihr euch immer über den aktuellen Entwicklungsstand informieren.

**Alm-Life** von Blitzmaker

Alm-Life. Lebe, liebe, Hacke! Alm-Life ist eine Aufbau-, Lebens- und Wirtschaftssimulation in einem. Es geht darum das der Spieler eine Alm leitet, sich Versorgt und eine Familie gründet. Das alles in 2D UND in 3D!! Alm-Life lehnt sich stark an Die Sims und Harvest Moon. Das Spiel wird seit 1 Jahr entwickelt und ist in der Beta Phase. Der Release Termin der neuen Demo steht leider noch nicht fest. Aber wir Arbeiten dran. DK-Software





## Daran arbeiten unsere Blitzer

### Unreal Tournament 2D von TheRealSephiroth

Beschreibung: Stellt euch Unreal Tournament vor, wie Ihr durch eure Action geladenen Levels läuft. Jetzt stellt euch die selbe Szene vor... mit einem Unterschied: es existieren nur 2 Dimensionen ! Gesteuert wird deine Figur aus der Vogelperspektive !

#### Bereits enthaltene Features:

- Bombing Run inkl.Bots
- Team Deathmatch inkl.Bots
- 3D Sounds
- Mapeditor
- Console
- Items(Health Packs,Armor)
- UT Waffen
- UT Botnamen
- Multiplayer(LAN,Internet)
- Singleplayer(mit Bots)

#### Geplante Features:

- Deathmatch inkl.Bots
- Capture the Flag inkl.Bots
- "Record" Befehl zum Aufnehmen von In-Game-Videos
- automatischer Mapdownload
- Updater

Name: Unreal Tournament 2D

Genre: 2D-Action-Shooter

Entwickler: Dreamware

Homepage: [www.dreamware.eu.tc](http://www.dreamware.eu.tc)

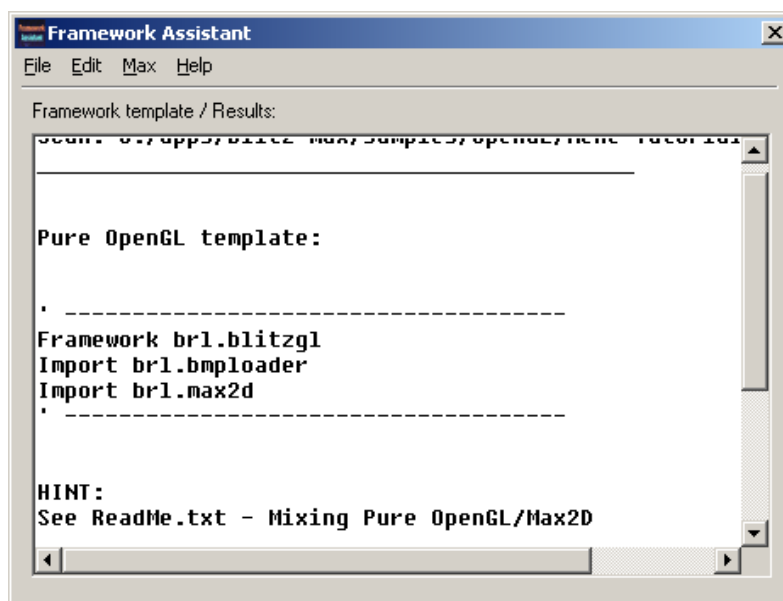
Entwicklungsbeginn: 19.10.2004

Besonderes: Erlebe UT aus einer neuen Perspektive kennen! UT2D

basiert auf der Engine von CS2D

Vergleichbare Spiele: CounterStrike2D

## Framework Assistant



### Framework Assistant

#### Was ist der FA?

Ein Programm das es BlitzMax Codern erleichtern soll, ein Framework aufzubauen. Der Coder muss nun nicht mehr wissen, welche Module welche Befehle enthalten und kann sich getrost auf die Arbeit des FA stützen.

Er durchkämmt den eigenen BlitzMax Ordner und stellt eine Liste aller Befehle und Module auf. Danach lässt sich eine BMX Datei wählen, für die ein Framework erstellt werden soll. Das klappt sogar rekursiv!!!

Gibt es mehrere Möglichkeiten ein Framework zu bilden, so offenbart es das Tool ebenfalls.

#### Wofür ein Framework?

Ein Framework ist dafür gedacht, um nicht alle Befehle der Module mit in die EXE schreiben zu müssen, sondern nur die, die auch wirklich benötigt werden. Das erspart bestenfalls mehrere 100Kbytes.

#### TIPP am Rande:

Natürlich sollte man das Tool erst beim release des Programms einsetzen, da sich der Code ja immer weiter entwickelt und so auch neue Befehle eingesetzt werden die im aktuellen Framework evtl. noch nicht implementiert sind. - IMurDOOM

# SUPER MARIO

## Deathmatch

Seit einiger Zeit kursiert in der deutschen BlitzBasic-Community eine programmierte Mischung aus UnrealTournament-Sounds und süßen Mariografiken.

Passt nicht?

Falsch.

Das sagt zumindest die Mehrzahl der mittlerweile 155 Beitragsschreiber und bekundet lauten Beifall zur Abschlachtorgie des kleinen, unschuldigen Klempters.

Doch nicht nur dem wahrhaft spritzigen Humor von Super Mario Deathmatch hat Programmierer „Ninja“ es zu verdanken, dass sein Spiel so erfolgreich ist:

Anders als die Quelle der Inspiration, „Super Mario War“, enthält SMD einen Multiplayermodus, der das Gruppenschlachten über LAN und Internet möglich macht.

Es macht nicht nur mehr Spaß, die Spielfiguren menschlicher Gegner zerplatzen zu sehen, sondern es ist auch eine Abwechslung zu den KI-Gegner, auch wenn die meiner Meinung nach gut gelungen sind. Anfängliches Verabreden gehört mit der Serverliste auf der SMD-Homepage der Vergangenheit an, auch wenn dem Spiel mangels weltweiter Bekanntheit etwas an Spielern mangelt, die sich bekriegen könnten.

Doch mit einer Veröffentlichung wird dieses Problem wohl auch bald kein Thema mehr sein.

Doch dazu mehr von Ninja selbst:

### **PBM: Eine kurze Beschreibung zu deiner Person bitte!**

Ninja: Dazu kann ich nicht viel sagen, bin halt ein normaler 16 Jähriger, und gehe noch zur Schule. Was wollt ihr noch Wissen? Ach ja mein Lieblingsfilm ist Bambi! ;)

### **PBM: Was war die Initialzündung für "Super Mario Deathmatch" und was hat "Super Mario War" für eine Rolle dabei gespielt?**

Ninja: In meiner Klasse haben immer alle SMW gespielt und sogar ein Turnier veranstaltet da das aber nicht über Netzwerk ging und auch nicht mehr weiterentwickelt wurde hab ich mir gedacht ich mache dort weiter wo die Entwickler aufgehört haben und bringe Features wie eben den Netzwerksupport ein.

### **PBM: Was denkst du, macht SMD so spaßig? Warum hat der Thread 153 Posts und 3647 Views?**

Ninja: Das liegt wohl vor allem daran dass das Spiel mit 2-4 Leuten an einem PC echt Spaß macht!

### **PBM: Hast du vor das Spiel zu veröffentlichen (Softwarearchive, etc.) und wenn ja, wie willst du das \*hust\* Problemchen mit den Grafiken lösen?**

Ninja: Ja habe ich vor. Wie ich das mit den Grafiken löse? Vorerst mal gar nicht, immerhin gibt es 100e Spiele die die Super Mario Sprites

nutzen und Nintendo behält sich im Moment zumindest vor gegen Urheberrechtsverletzungen vorzugehen.

### **5. Wird SMD noch weiterentwickelt und welche Features werden noch eingebaut?**

Ninja: Vorerst bin ich erstmal fertig, mit bugfixes werdet ihr natürlich noch versorgt und neue Features gibt's wenn mir welche einfallen und sich diese auch mit dem netcode vereinbaren lassen ;). Ich hab da aber so eine Idee mit items...

### **PBM: Welche Projekte hattest du vor SMD und hast du schon konkrete Pläne für etwas neues?**

Ninja: Vor SMD hatte ich vor allem Velocity - ein Weltraumshooter - der zwar nicht so gut ankam wie SMD aber trotzdem Spaß macht, sonst hatte ich vieles aber nichts richtig großes :D Pläne habe ich viele - aber keine Konkreten im Moment ;) .

### **PBM: Einen Satz über Blitz Basic, Mark Sibly, das Forum und die Welt, bitte!**

Nutzt Blitz, es ist eine tolle, einfache Programmiersprache und mit Blitzmax könnten auch die letzten Limits von BlitzBasic übersprungen werden!

### **PBM: Und eine letzte Message, die du unseren Lesern mitteilen möchtest!**

Ninja: Programmieren lernt man durch Programmieren (Ich weis der Satz ist scheisse aber so ist es nun mal! :D )



## Don't get Angry 2

Do Okt 27, 2005 10:20 pm lautet es im Thread mit dem Titel „Don't Get Angry! 2“. Dieses Datum war der Startschuss für eines der größten Projekte, die je in BlitzBasic gefertigt wurden. Auf ein neues begeistert(e) Mike Dogan alias X-Pressive mit dem 2. Teil seiner 3D-Antwort auf „Mensch ärgere dich nicht“ und zieht die deutsche BlitzCommunity in seinen Bann. Man zeigt sich beeindruckt, wie man in einer Zeit von drei bis vier Monaten ein solches Spiel so professionell entwickeln kann und spart nicht an lobenden Kommentaren. Noch erstaunlicher ist es, dass kein Team dahinter steht, sondern alles aus einer Hand stammt...

Das Ziel in Don't Get Angry! 2 ist es, seine Spielfiguren gemäß den wohl bekannten Regeln von „Mensch ärgere dich nicht“ über den Weg aus Spielfeldern auf die sicheren Plätze zu befördern. Doch damit nicht genug: Verrückte Extras und irre Effekte machen Laune und sorgen für unerwartete Wendungen (und das ist kein Werbetext, auch wenn es sich vielleicht so anhört)! Zum Einsatz kommen natürlich die hauseigenen Funktionsbibliotheken SpriteCandy und ParticleCandy, die X-Pressive einiges an Arbeit abgenommen haben und das Spiel mit imposanten X-Plosions- :) und Partikeleffekten füllen.

Doch auch ein noch so ausgereiftes Spiel kann trotz KI-Gegnern langweilig werden – wäre da nicht der Multiplayermodus, dank dem man sich via LAN und Internet fordernde und lustige Gefechte liefern kann und das einmalige Gefühl dieses Spielprinzips voll vermittelt bekommt und man sich aus dem Taumel von Trauer über die eigenen, zermatschten Spielfiguren und Schadenfreude, den Gegner am gewinnen gehindert zu haben, kaum noch befreien kann. Natürlich ist das LAN-Spiel dank (meistens) direktem Sprach- und Blickkontakt noch einen Tick spannender.

Sicher, das Spiel ist sehr beeindruckend, doch mich hat auch der Mann und die Abläufe dahinter interessiert, was vielleicht daran liegt, dass ich mehr programmiere als spiele ^^ Da hatte ich bei Don't Get Angry! 2 sofort ein gefundenes Fressen, so wie ich bereits das Vermarktungskapitel aus „Spiele Programmieren mit BlitzBasic“ verschlungen hatte, weil mich schon immer interessiert hat, wie man Spaß (am Programmieren) mit Geld verbinden könnte. Zugegebenermaßen wusste ich bisher nicht, dass Mike „X-Pressive“ Dogan Blitz3D tatsächlich zum Brötchenverdienen benutzt, doch es ist nicht nur ein lukrativer Nebenjob, Mike Dogan proggt hauptberuflich (BlitzBasic-)Spiele!

Als mir das dann in den Sinn gekommen war, konnte ich nicht mehr an mich halten und habe den Künstler persönlich angeschrieben und es sind bei unserem „kleinen“ Interview sehr interessante Informationen ans Tageslicht getreten und eröffnen der deutschen BlitzBasic-Community einen Einblick in das Leben und Wirken eines professionellen, unabhängigen Programmierers!

### 1. Bitte erzähl etwas über dich; Beruf, Hobbies, Alter, alles was interessant ist

Ich bin Baujahr '74 -also noch gänzlich ohne Konsolenschrott aufgewachsen (wenn man mal von der trashigen Atari2600 absieht) und gehöre somit auch der typischen Commodore64-Generation an. Obwohl ich meine ersten Programmiererfahrungen in den Achtzigern auf dem C64 gesammelt habe, hat es letztlich doch zehn weitere Jahre gedauert, bis ich mir meinen ersten PC anschaffte.

Meine Hobby ist vor allem das Motorradfahren (ich fahre das ganze Jahr hindurch und habe mein Auto aus Mangel an Fahrspass schon vor längerer Zeit verschert), aber auch Segeln, Reisen (ich bin ein Asien-Fan) und gute Filme. Letztere werden leider, wie gute Computerspiele auch, immer seltener. Wenn ich nicht unbedingt "muss", verbringe ich die Zeit nicht unbedingt vor dem Computer.

Obwohl ich schon auf dem C64 programmiert habe, habe ich "erst" vor ca. zehn Jahren angefangen in eigener Regie das Programmieren in verschiedenen leistungsfähigeren Sprachen auf dem PC zu erlernen (das Internet mit seinem unbegrenzten Zugang an Wissen machte es möglich) und mich nach ein bis zwei Jahren autodidaktischem Studium mit geringem Aufwand (ein Computer und ein bisschen Software) als Programmierer selbstständig gemacht. Seitdem entwickle ich Computerspiele, übernehme aber auch grafische Aufträge, Webseiten, Multimedia- und Datenbank-Projekte, wobei letzteres so knochentrocken ist, dass ich mir die Freiheit nehme, nicht jeden Auftrag anzunehmen.

In der Regel bilde ich mich fast ausschließlich über das Internet weiter. Ich habe nur zwei Programmierbücher zuhause herumliegen: das eine ist eine PHP-Referenz, das andere die

BlitzBasic-Fibel von Rene Meyer 😊 Aber auch die kommen nur noch selten zum Einsatz.

## 2. Was hast du in deiner Programmiererkarriere alles zustande gebracht?

Anfangs hatte ich mich eher auf Online-Games und Internet-Inhalte spezialisiert, später dann endgültig auf Spiele für den Offline-Bereich. Während dieser Zeit habe ich zahlreiche Webseiten programmiert, größere Portale wie z.B. eine Online-Partnerbörse und ein DVD-Filmportal in Norwegen realisiert (beides Auftragsarbeiten), viele Onlinespiele erstellt und ein halbes Dutzend umfangreichere Spiele für den Vertrieb auf CD-ROM.

Mein Credo war von Anfang an, so wenig finanziellen Aufwand wie möglich zu betreiben - seit heute habe ich z.B. keinen einzigen Pfennig für Werbung ausgegeben. Meine Webseite habe ich damals ausschließlich mit dem Streuen von Online-Games promotet, die inzwischen auf allen bekannten Flash-Seiten zu finden sind (u.a. Mars Massacre, Bushido Fighters, Mahjongg 3D, Click Clack, Puzzle Maniax, BrickBreaker, Missile Strike, Arcatris und viele andere).

Inzwischen sind diese Spiele, die ja eigentlich nur für den Online-Einsatz gedacht waren, auch auf mehreren CD-ROMs erschienen, Mars Massacre z.B. wurde damals von NVidia auch als Werbegimmick auf der Games Convention in Leipzig verteilt. So kamen also immer wieder Aufträge rein, ohne daß ich viel Geld in Werbung investieren musste. Allerdings stecken in all diesen Spielen insgesamt einige Jahre Arbeit, das darf man nicht vergessen.

Eine originelle Website war mir auch sehr wichtig - schließlich musste man ja für Mund-zu-Mund-Propaganda sorgen. Darin habe ich viel Zeit und Arbeit investiert. Damals war x-pressive.com eine der abgefahrensten Flash-Seiten, die es gab und die Seite sorgte für ziemliches Aufsehen und Artikel in Zeitschriften (u.a. der PC-Action). Dadurch erhielt ich wiederum viele Anfragen und konnte wertvolle Kontakte knüpfen.

Heute ist Flash nicht mehr so populär (was vor allem daran liegt, dass es fast nur noch für Werbezwecke und Banner missbraucht wird) und inzwischen habe ich meine Website wieder zu einer eher konventionellen abgerüstet, letztendlich auch, weil ich ja nur noch Offline-Spiele erstelle. Einige BlitzForum.de User kennen die alte Seite aber vielleicht noch. Zwei Jahre habe ich auch fest in einer Agentur gear-

beitet, wo ich zwar wertvolle Erfahrung sammeln konnte, was den Vertrieb von Spielen angeht, meine Kreativität aber ganz und gar nicht ausleben konnte. Weil ich schon immer gewohnt war, effizient alleine zu arbeiten, habe ich mich im Laufe der Jahre darauf spezialisiert, alles selbst zu machen - Grafik, Programmierung, Sound, bis hin zu Werbetexten und dem Cover-Artwork der Spiele. Nur so kann man mit sehr geringen Kosten ein Spiel realisieren, behält möglichst viel Kontrolle und bleibt unabhängig. Allerdings bedeutet das aber auch einen sehr großen Aufwand, was Zeit und Nerven angeht.

In den letzten zehn Jahren habe ich eine Menge Erfahrung gesammelt. Als ich damals meine ersten Gehversuche im Internet unternahm, ahnte ich nicht, dass meine Spiele irgendwann in den Läden stehen würden. Damals noch der große Traum eines Programmierers, gehen heute meine Spiele den regulären Weg durch die Regale von Saturn, Karstadt & Co., sind in der ComputerBild, in Deutschland, Österreich, der Schweiz, Italien, Polen, Grossbritannien, den USA und demnächst auch in Russland zu finden und hauchen das Licht ihres ein- bis zweijähriges Verkaufs-dasein irgendwann wie alle Spiele im Pearl-Katalog aus oder gehen mit einem kleinen Tusch von der Bühne, wenn sie letztendlich noch bei ALDI verkauft werden, wie z.B. der erste Teil von Dont Get Angry! ab Januar nächsten Jahres.

Allerdings bringt all das bei weitem mehr "Ruhm" als Geld. Reich wird man dadurch in der Regel nicht - ein Autor bekommt, wie z.B. in der Plattenbranche auch, im Schnitt nur ca. 5% der Verkaufseinnahmen eines Spiels. Nur, wenn man es schafft, ein bis zwei wirkliche Hits in Folge mit sehr guten Verkaufszahlen zu produzieren, kann man auch mal einen wirklich guten "Deal" für eine Fortsetzung mit besonderen Konditionen aushandeln.

Und die letzten Zehn Jahre haben auch Nerven gekostet, das schließt z.B. auch zwei Urheberrechtsprozesse mit ein, mit denen ein ehemaliger Arbeitgeber an die Rechte meiner Spiele gelangen wollte. Die verlor er zwar mit Pauken und Trompeten - was aber geblieben ist, ist ein großes Misstrauen in alle Leute, die einen Anzug tragen.

## 3. Wie bist du zu BlitzBasic gekommen?

Am Anfang hatte ich mich ja noch auf das Erstellen von Online-Games spezialisiert. Diese wurde im Lauf der Zeit immer ausgefeilter und kamen dadurch irgendwann an ihre technischen Grenzen. Auch die Geschwindigkeit, die miserable Performance, war hier immer wieder

ein Hindernis, das wirklich fetzige Spiele, vor allem Actiongames, meist nur schwer zu realisieren oder gar unmöglich machte. Bei mir muss es in Spielen einfach "krachen" -da müssen Dinge rumfliegen, Explosionen sollten Bildschirmfüllend sein und eine Spielwelt aus vielen kleinen Details bestehen. All das war nur sehr eingeschränkt und mit viel Ruckelei möglich.

Man konnte so nicht wirklich das umsetzen, was man "im Kopf" hatte -schon gar nicht in 3D-Grafik. In C++ zu programmieren, lohnt sich aber für einen einzelnen Entwickler allerdings auch nicht -der Zeit- und Codeaufwand ist einfach zu groß. Professionelle Studios, vor allem die Japaner im Konsolenbereich, benutzen hier seit langem eigene Entwicklungstools, die das Erstellen von Spielen vereinfachen. Nur so lässt sich in möglichst kurzer Zeit und mit möglichst wenig Aufwand ein gutes Resultat erzielen.

Aus diesem Grunde war ich immer auf der Suche nach einer wirklich guten Entwicklungsumgebung, die zwar vor allem schnell und leistungsfähig ist, aber auch den Programmieraufwand in Grenzen hält, so dass man sich vor allem auf visuelle Effekte und das Gameplay konzentrieren kann, ohne sich ständig mit der Umsetzung von technischen Problemen ärgern zu müssen.

Irgendwann fiel mir dann auch BlitzBasic in die Hände. Ein großer Nachteil war für mich zwar zunächst, dass kein objektorientiertes Programmieren damit möglich war (was das Umsetzen von Ideen meist einfacher und strukturierter macht), aber ich war auch fasziniert davon, wie viel man aus ein paar einfach wirkenden BASIC-Kommandos herausholen kann und wie sauber (ja, sauber!) BASIC-Codes im Vergleich zum verschnörkelten C++ oder anderen OOP-Sprachen aussehen. Ein Code muss bei mir auf jeden Fall ästhetisch aussehen. Wenn eine Sprache wie "gehäkeltes" oder Erbrochenes aussieht, verliert sie viel von ihrem Reiz.

Wenn man ein Spiel in den Handel bringt, ist es nicht zu umgehen, dass man nach vielen Monaten auf Wunsch eines Lizenznehmers immer mal wieder etwas daran verändern muss - und da ist es eben wichtig, schnell wieder in den eigenen Code hineinzufinden. Das geht nur, wenn er möglichst sauber, knapp und übersichtlich ist.

Vor allem ist es mit Blitz möglich, mit 3D-Grafik, und das sogar mit sehr guter Performance, zu arbeiten. Mir fiel auf, dass man mit BB in kürzester Zeit Projekte realisieren kann.

Das war vor ca. zwei Jahren, seitdem habe ich mich eigentlich ganz auf BB3D konzentriert, inzwischen zwar mit einem schielenden Auge auf BlitzMax, aber das wird wohl noch ein wenig dauern, bis es wirklich ausgereift und sicher genug für kommerzielle Spiele ist, vor allem im 3D-Bereich.

#### **4. Du hast vor kurzem dein aktuelles Spiel "Don't Get Angry 2" vorgestellt, sag uns etwas über den Verlauf der Entwicklung, Höhen und Tiefen, deine Idee dahinter, Motivationen und wichtige Fakten!**

Ich habe Brettspiele schon immer gemocht -sie sind klar strukturiert und bei rundenbasierten Spielen erhält der Spieler, der gerade am Zug ist, die ganze Aufmerksamkeit der anderen Spieler. Solche Spiele haben einfach eine soziale Komponente und tragen nicht, wie z.B. Ego-Shooter und Spielkonsolen, zur Vereinsamung des Spielers bei. Ein Brettspiel als Computerspiel ist aber noch interessanter, da sich hier ja noch Einfälle hinzufügen lassen, die bei einem gewöhnlichen Brettspiel (aus physikalischen Gründen) ja gar nicht möglich wären.

Das Regelwerk des Original-Brettspiels zu implementieren war recht einfach und nahm wenig Zeit in Anspruch -das lag vor allem daran, dass ich mir schon für den ersten Teil ein recht gutes und ausbaufähiges System ausgedacht hatte. Weil das Grund-Gameplay relativ leicht von der Hand ging, konnte ich mir viel Zeit für alle möglichen Extras und Gags nehmen, die das Spiel ein bisschen "peppiger" als das Original machen sollten.

Die Programmierung selbst war dabei eigentlich das kleinste Hindernis. Vergleichsweise mehr Zeit hat eigentlich das Artwork (Texturen, Modelle, Grafiken) in Anspruch genommen. Eine solide Grafik ist bei Spielen, die es in den Handel schaffen sollen, unentbehrlich. Und gerade, wenn einem viele Möglichkeiten wie Shader etc. bei Blitz nicht zur Verfügung stehen, ist es wichtig, das mit allerlei künstlerischen Tricks zu kompensieren, damit die Grafik nicht angestaubt und wie ein Relikt aus vergangenen Zeiten wirkt. Mit ein bisschen Kreativität kann man hier aber noch viel herausholen, vor allem, was das Zeichnen der Texturen angeht. Dieser Punkt wird von vielen gerne übersehen.

Ich habe vor allem viel Wert darauf gelegt, dass DA2 wie ein Konsolenspiel wirkt -alles sollte sich bewegen, "fetzig" sein, vor allem was das Interface und das Menü angeht. Hier waren die beiden Candy-Libs eine unersetzliche Hilfe und



haben eine gute Bewährungsprobe hingelegt.

Ich habe mir während der Entwicklung oft vorgestellt, das Spiel würde in einem Laden wie z.B. Saturn als "Testspiel" auf den ausgestellten PCs laufen, an denen die Kunden dann vorbeilaufen und sagen "Guck mal -das ist ja witzig!". Versucht das mal -dieser Gedanke motiviert enorm und bringt gute Einfälle! 😊

Wirklich Kopfzerbrechen hat mir eigentlich nur der Multiplayer-Modus bereitet, mit dem ich bisher nur experimentiert hatte. Inzwischen ist mir klar geworden, dass hier nicht unbedingt die Stärke von Blitz liegt. Hinzu kommt noch, dass das Testen in diesem Punkt recht aufwändig und umständlich ist.

Vom Multiplayer-Modus einmal abgesehen, hat mir die Entwicklung des Spiels viel Spaß gemacht -beim Einbauen der Gags habe ich mir selbst oft den Hintern abgelacht. Zum Beispiel, als es darum ging, die Furz-Sequenz einzubauen. Ich habe mich auch gefragt, welche Farbe wohl ein Furz hat. Seltsam, warum giftige Gase in Spielen und Cartoons immer eine grüne Farbe haben, nicht? Es ist aber so.

Einige Ideen waren allerdings etwas "krass", so dass sie es nicht bis ins Spiel geschafft haben. So war anfangs auch ein Team von Arabern eingeplant (weil es einfach originell gewesen wäre), aber spätestens dann, wenn diese den Sprengstoffgürtel gezündet hätten, wäre das ganze ein wenig "delikat" geworden.

Damit das Spiel später eine unbegrenzte Altersfreigabe bekommt, musste man sich gut überlegen, was man Kindern (und FSK-Prüfern) gerade noch zumuten kann. Wahrscheinlich wird das Spiel nun eine Freigabe ab sechs Jahren bekommen -damit kann ich aber noch leben.

## 5. Warum hast du dich für das Entwickeln von Shareware entschieden?

Shareware, weil ich damit meine Brötchen verdiene 😊

Was mich aber generell am Entwickeln von Spielen an sich fasziniert, ist die Möglichkeit, eigene kleine Welten zu erschaffen. Man kann das eigentlich auch mit dem Schreiben von Büchern, dem Zeichnen oder dem Drehen von Filmen vergleichen. Es geht immer darum, seine Phantasie auszuleben, eigenes "Leben" zu erschaffen. Das Programmieren an sich ist ja nur ein Mittel zum Zweck, so wie eben ein Maler mit Pinseln hantiert. Wer sich dabei zu sehr nur auf die technischen As-

pekte konzentriert, verliert seine Phantasie und Kreativität.

## 6. Wenn man deine Homepage, deine Produktpräsentationen, etc. ansieht, so strahlen diese Professionalität aus. Wie denkst du hast du das erreicht? Und war das schon immer so?

Der Fluch des Perfektionismus. Manchmal kann er einen zur Verzweiflung treiben, aber letztendlich sorgt er immer dafür, dass man sich nicht blamiert 😊

Wenn man an einem Projekt arbeitet, egal, was es ist, muss man immer versuchen, es mit den Augen anderer zu sehen. Das ist sehr wichtig, wenn es darum geht, Produkte zu entwickeln, die "massentauglich" sein sollen. Ich werkle dann einfach so lange an den kleinsten Details herum, bis der "Bauch" endlich sagt, dass auch "fremde" Augen das gut finden würden. Vorher darf man einfach nicht aufhören und muss sich zwingen, weiterzumachen. Bequemlichkeit und Faulheit zählen hier nicht.

Manchmal geht das recht fix, an anderen Tagen sitzt man eine Ewigkeit an kleinsten Dingen, bis der "Bauch" grünes Licht gibt. Das kostet viel Zeit und Nerven, aber im Lauf der Zeit lernt man viele Kniffe und gewöhnt sich Arbeitsmethoden an, die es einfacher machen.

Jedes Projekt, das in die Öffentlichkeit gelangt, kann einem Ruhm oder Schande bereiten. Also gibt man in solchen Dingen immer sein Bestes, um nicht Schande über sich zu bringen. So muss man das sehen. Vielen zählt ihre eigene Bequemlichkeit leider mehr, als irgendein Perfektionismus.

## 7. Wie viel Zeit investierst du ins Programmieren und was stellst du dafür zurück?

Da ich selbstständig bin, kann ich mir die Zeit Gott sei Dank recht frei einteilen. Ich achte auch darauf, möglichst keine Auftragsarbeiten anzunehmen, die einen schnell zum Sklaven machen.

Trotzdem braucht man sehr viel Disziplin, wenn man von seinen Spielen leben will. Wenn ich an einem Projekt arbeite, hat ein Arbeitstag bei mir 12-16 Stunden, die ich am Stück programmiere und erst aufhöre, wenn ein vorgegebenes Ziel erreicht ist.

Wenn ein Spiel aber fertig gestellt und "verpackt" ist, nehme ich mir dafür wieder einige Wochen oder manchmal auch Monate

Auszeit, düse mit dem Bike herum und relaxe einfach, um wieder neue Ideen zu sammeln. Da haben dann Menschen, Sonne und Natur wieder absolute Priorität. Während eines Projektes powere ich also voll rein -aber dafür muss man danach auch wieder ausspannen können, sonst reibt man sich auf und brennt irgendwann durch.



Das musste ich aber auch erst lernen. Anfangs hatte ich immer ein schlechtes Gewissen, wenn ich mal einen Tag nicht gearbeitet habe. Das geht aber irgendwann auf die Gesundheit.

### 8. Deine Message! Sag uns etwas über Mark Sibly, BlitzBasic, de Zukunft & das Universum!

Um es ganz klar zu sagen: die Zukunft der "großen" Spiele und Ego-Shootern gehört den Konzernen wie Atari und Electronic Arts, die mehr und mehr den ganzen Markt auffressen. Wer also versucht, in diesem Bereich zu entwickeln und Vollpreistitel zu imitieren, verschwendet seine Lebens-



zeit und kämpft gegen Windmühlen.

Aber gerade durch die großen Konzerne und ihre Fokussierung auf Monumentalspiele und Action- und Gewaltlastige Konsolentitel entsteht immer mehr ein Nischenmarkt, der in Zukunft immer gefragter sein wird: so genannte "casual games", also "Gelegenheitsspiele" für die breite Zielgruppe, vor allem Frauen, Kinder und Familien. Das sind die Zielgruppen, die bislang von den Konzernen vernachlässigt werden und genau hier müssen wir als unabhängige Spieleentwickler ansetzen. Kleine, feine Spiele mit originellen Ideen und überschaubarem Entwicklungsaufwand.

Und gerade für diesen Bereich ist BlitzBasic sehr gut geeignet. Das Problem, dass ich sehe, ist u.a. aber, dass die Technik nicht still steht. Während sich die technischen Möglichkeiten immer schneller ausdehnen, entwickelt sich BlitzBasic nicht immer schnell genug mit. Vielleicht wird sich das mit BlitzMax in Zukunft ja ändern, aber dann müsste auf jeden Fall ein 3D-Modul her, dass mindestens so einfach und zu gleich leistungsfähig und kompatibel ist, wie Blitz3D. Mal sehen, ob das gelingt.



## Onlinehighscore mit BlitzMax und PHP

In diesem Tutorial möchte ich erklären, wie man in BlitzMax unter Zuhilfenahme von PHP einen eigenen Onlinehighscore programmieren kann. Natürlich kann man auch auf einfachere und bessere Methoden zurückgreifen und z.B. das recht prominente i.Score<sup>2</sup> von Bruzard verwenden, doch es lohnt sich immer, das Grundprinzip dahinter zu verstehen.

### Was braucht man?

- PHP-fähigen Webpace
- Kleine Kenntnisse in der BlitzMax-Programmiersprache
- Internet-Grundwissen, wie FTP-Upload, etc.

Beginnen wir mit PHP:

PHP ist eine serverseitige Skriptsprache, das heißt der Webserver, auf dem die PHP-Datei abgelegt ist, verarbeitet ihren Inhalt und befolgt die Befehle, die wir ihm mittels PHP geben.

Dabei ist darauf zu achten, dass ein Server überhaupt PHP-fähig ist.

PHP-Dateien haben, wie könnte es anders sein, die Dateiendung „.php“.

Man kann sie sehr einfach mit einem Texteditor (z.B. das Windows beigelegte „Notepad“) oder mit komfortablen PHP-Editoren schreiben.

Um dem Server aufzuzeigen, wo er ein PHP-Skript verarbeiten („parsen“) soll, muss man folgende Struktur verwenden:

```
<?PHP
    // Befehle
?>
```

Kommentare, die sich über eine Zeile erstrecken, werden wie hier im Beispiel mit „//“ gekennzeichnet und ignoriert.

Jetzt müssen wir erfahren, wie man Daten an ein solches PHP-Skript weitergeben kann, denn schließlich müssen unsere Punktzahlen von unserem BlitzMax-Programm zum PHP-Skript gelangen!

Am einfachsten ist es, die GET-Methode zu verwenden; doch man kennt es ja nicht anders: sie ist eben auch am einfachsten zu manipulieren. Neben GET gibt es auch noch POST. Der Einfachheit halber werden wir jedoch bei GET bleiben.

Sicher habt Ihr es schon mal im Internet gesehen: Manchmal sind an normale Internetadressen (noch) seltsame Sachen angehängt. Zum Beispiel:

<http://www.deine-domain.de/skript.php?punkte=50&name=„Justus“>

Wie ihr seht, folgt nach der Webadresse des PHP-Skripts ein Fragezeichen, was die Wertübergabe einleitet. Will man mehrere Werte übergeben, so verknüpft man das mit einem &-Zeichen.

Soweit die Theorie, doch für die Praxis fehlt uns noch etwas: Befehle im PHP-Skript:

Wir müssen nun Variablen erstellen, um sie mit den übergebenen Werten (punkte, name) zu füllen.

Dazu benutzen wir GET:

```
<?PHP
    $punkte = $_GET['punkte'];
    $name = $_GET['name'];
?>
```



So, jetzt haben wir Variablen mit Werten darin. Variablen erkennt man an ihrem Dollar-Zeichen, was sie immer vor sich stehen haben.

Jetzt müssen die Werte irgendwo abgespeichert werden. Und das machen wir genau wie in BlitzMax: In Dateien.

Man kann Daten zwar auch in MySQL-Datenbanken speichern, doch das ist für uns erstmal unnötiger Firlefanz ^^.

Wer jedoch unbedingt will und seinen Highscore verbessern möchte, dem sei kein Bein gestellt; am Ende dieses Tutorials stehen hilfreiche Links.

Das Speichern funktioniert eigentlich fast wie gewohnt:

```
<?PHP
    $punkte = $_GET['punkte'];
    $name = $_GET['name'];
    $datei = fopen(„score.txt“, „a“);
    fputs($datei, „$punkte\n“);
    fputs($datei, „$name\n“);
    fclose($datei);
    die(„Hier spricht das Skript, ich habe fertig“);
?>
```

Keine Angst, dabei bleibst es vorerst. \$datei ist die Variable für die .txt-Datei, in der wir unsere Highscores ablegen. Das „a“ sorgt dafür, dass unsere Highscores nur an die Datei angehängt werden und nicht den alten Inhalt überschreiben.

Wichtig ist außerdem, dass Ihr die Datei mit dem so genannten CHMOD-Attribut 777 freigibt, damit das PHP-Skript darauf Zugriffsrecht bekommt. Das macht Ihr in eurem FTP-Program (z.B. Filezilla) unter „Datei-Attribute“ oder einer ähnlichen Option.

Jetzt sollte alles funktionieren: Ladet das PHP-Skript auf euren Webspace hoch und probiert es aus, indem Ihr die Adresse in eurem Browser eingibt und die benötigten Werte anhängt. Zum Beispiel:

<http://www.domain.de/score.php?punkte=3000&name=Justus>

Ruft Ihr jetzt die .txt-Datei auf, sollte dort ein neuer Eintrag stehen! Außerdem sollte in eurem Browserfenster die Meldung zu sehen sein, die wir vorher mit die(„bla“); festgelegt haben.

Doch wir wollen die besten Punktzahlen aller Zeiten der Öffentlichkeit nicht vorenthalten, sondern uns dem Skript wenden, das die besten Spieler der Weltbevölkerung bekannt macht.

Zuerst der Code:

```
<?PHP
    $datei = fopen(„score.txt“, „r“);
    Do {
        $punkte = fgets($datei,200);
        $name = fgets($datei,200);
        echo($punkte)
        echo($name)
    } While ($punkte != „“);
?>
```

So. Wie Ihr sicher bereits aus BMAX kennt, handelt es sich beim Hauptteil des Skript um eine Schleife; und zwar um eine, die so lange, die innen angegeben Befehle abarbeitet, solange die Variable \$punkte Daten enthält.

Mehr brauchen wir nicht zum Auslesen. Außerdem haben wir mit diesem Skript zwei Affen mit einem Hammer geschlagen (oder so ähnlich): Einerseits können wir

es benutzen, um die Scores auf einer Website ausgeben zu lassen, andererseits ist es aber auch fähig, die Punktzahlen und Spielernamen direkt an unser Spiel zu senden, damit wir ggf. direkt ingame eine Liste anzeigen lassen können!

Endlich geht es ans Eingemachte: MaxIDE gestartet und in die Tasten gehauen, jetzt beginnt der BlitzMax-Part!

Dazu müssen wir uns die neuen Netzwerkbefehle von BlitzMax ansehen, die jedoch leicht zu verinnerlichen sind. Ich behaupte einfach mal unwissenderweise, dass das Vorgehen auch mit Vertex' Bnet-Modul geht; die Art und Weise sei jedem selbst und dem persönlichen Codegeschmack überlassen.

Mit CreateTCPSocket stellen wir eine Verbindung zu unserem Highscoreserver her. Natürlich sprechen wir ab sofort das socket mit der gleichnamigen Variable socket an. Anschließend richten wir einen Stream auf diesem Socket ein.

Unser Code bisher:

```
'Socket und Stream
socket = CreateTCPSocket()
stream = CreateSocketStream(socket)
```

Jetzt müssen wir die IP-Adresse unseres Servers herausfinden, wie es jedoch einfacher nicht geht: Mittels des Befehls HostIP() speichern wir die IntegerIP in der Variable ip.

HostIP() übergeben wir die Adresse unseres Servers/Webspace. HostIP liefert ein kryptisches Wirrwarr zurück, dass sich jedoch dank der Funktion DottedIP(ip) in die lesbare Form mit den drei Punkten (z.B. 248.145.25.69) verwandeln lässt. Dazu sei gesagt, dass es Freewebspaceanbieter, es verständlicherweise nicht gerne sehen, dass stark frequentierte Skripte teuren Traffic verursachen, doch ich glaube nicht, dass für euch ernsthafte Gefahr besteht, sofern du nicht DarkCorner heißt und Stranded 3 programmierst ;).

Doch verbunden sind wir mit dem Server noch nicht. Das erledigen wir über das HTTPProtokoll, das die Welt von den URLs kennt. Wir verwenden HTTP 1.1 und den dafür standardisierten Port 80.

Diese ganzen Informationen übergeben wir ConnectSocket(). Damit wird unser Socket mit unserem Server verbunden und wir können unser dort heimisches Skript fragen, ob es unsere Punkte einträgt.

```
'Verbindung herstellen
Global socket = CreateTCPSocket()
Global stream = CreateSocketStream(socket)
Global ip = HostIP(„score.freehoster.de“)
Global connection ConnectSocket(socket,ip,80)
'Ergebnis im DebugLog speichern
DebugLog „IP: „+DottedIP(ip)
Select connection
    Case 0 RuntimeError „Verbinden fehlgeschlagen“
    Case 1 DebugLog „Verbindung hergestellt“
EndSelect
```

So weit so gut. Sollte die Verbindung erfolgreich hergestellt worden sein, habt ihr alles richtig gemacht ^^

Wenn wir es besonders bequem haben möchten, können wir das Ganze am besten in eine praktische Funktion packen und mit Init\_Connection() oder Ähnlichem betiteln.

Eine Funktion verwenden wir auch für das Senden der Punkte an das PHP-Skript:

```
Function Send_Score(player$,punkte)
WriteLine(stream,"GET /writescore.php?name="+player+"&punkte="+punkte" HTTP/1.1")
WriteLine(stream,"HOST: score.freehoster.com")
WriteLine(stream,"")
EndFunction
```

Das war's auch schon. Der Funktion werden der Name des Spielers und die Punktzahl übergeben und dann mit GET an unser Skript übermittelt. Danach noch schnell die Hostadresse geschickt und einmal „“, um zu sagen, dass fertig gesendet wurde, und es ist vollbracht.

Natürlich ist auch der Server nicht stumm, sondern teilt uns gesprächig all seine Gefühle und Empfindungen mit, die wir mittels schwachsinn\$ = ReadLine(stream) empfangen.

Doch nicht nur unwichtiges Zeug kommt, sondern auch die wichtige Meldung unseres Skript, das „ich habe fertig“ wird uns zugesandt. Das funktioniert mit der PHP

Funktion echo() als auch mit die() (quasi als letztes Sterbenswörtchen ^^ miserabler Witz).

Ihr ahnt es, ich habe keinen Bock mehr zu schreiben, sondern gebe euch lieber einen kräftigen Tritt in den Allerwertesten, die Methode zu erweitern:

#### AUSBLICK:

Auch wenn das jetzt für das erste Mal genug war, ist es lange nicht zu Ende. Sollte jemand gewieft mit den diversen Überwachungstools zurechtkommen und prüfen, wann was an wen verschickt, ist es ihm eine Leichtigkeit, es euch nachzutun und sich per Browser ohne Spiel Highscores zu erschummeln. Die Übergabe eines Codes oder einer irgendwie geheim von eurem Spiel gebildete Sicherheitsvariable sollte übergeben werden, damit dumme Gedanken nicht zu dummen Taten werden.

Ebenso seid Ihr auf euch allein gestellt, wie ihr die Rückgabe des Auslese-Skipts im Spiel verarbeitet und ausgeben.

Doch dieses Tutorial hat euch hoffentlich geholfen, auf dem Parkett des Internets und mit BlitzMax weiterforschen zu können.

Justus



## BlitzMax --- Wir programmieren ein Spiel - Teil 2

game.bmx

```
'Programmer: IMurDOOM

Graphics 800,600,0,60
SeedRnd MilliSecs()

'Zum einstellen...
Global cfg_spawntime=400
Global cfg_scrolly=2
Global cfg_alienhealth=2

Include "init.bmx"

'MAINLOOP
Repeat
    'DRAWING
    render_backdrop()
    aliens=render_aliens()
    render_fire()
    render_shots()
    render_player()
    render_explosions()
    render_hud()
    '/DRAWING

    check_mouse()
    check_times()

    scroll_y=(scroll_y+cfg_scrolly) Mod 200
    ResetCollisions()
    Flip ;flips=flips+1
    Cls
Until KeyHit(key_escape)
```

Init.bmx

```
'init.bmx

Include "check.bmx"
Include "render.bmx"

'ARRAYS
Global img_explosion[3]

Global fps
Global Flips

'IMAGES
SetMaskColor 0,0,0
    Global img_background=LoadImage("gfx/stars.bmp")

SetMaskColor 255,0,255
    Global img_alien=LoadImage("gfx/alien.bmp")
    Global img_player=LoadImage("gfx/player.bmp")
    Global img_shot=LoadImage("gfx/shot.bmp")
    Global img_protect=LoadImage("gfx/protect.bmp")
    Global img_fire=LoadImage("gfx/fire.bmp")

    ;MidHandleImage img_alien
    ;MidHandleImage img_player
    ;MidHandleImage img_shot
    ;MidHandleImage img_protect

    ;MidHandleImage img_fire
```

```

    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 pro_alpha:Float

```

```

    Field dead

```

```

    Field angle

```

```

    Field spindir

```

```

    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

```

```

    S_Alien.Spindir=Rand(0,3)

```

```

    S_Alien.Pro_alpha=0

```

```

    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 I_explosion
        s_explosion.frame:+1
        If s_explosion.frame>=25 Then ListRemove(I_explosion,s_explosion)
    Next
End Function
End Type

Global L_Fire:TList=CreateList()
Type T_Fire
    Field x
    Field y
    'Farbe
    Field red
    Field green
    Field blue

    Field speed_x
    Field speed_y
    'Field size
    Field alpha:Float

    Function NewFire(vx,vy,vsx,vsy,vr=230,vg=60,vb=10)
        s_fire:t_fire=New t_fire
        s_fire.x=vx
        s_fire.y=vy
        s_fire.speed_x=vsx
        s_fire.speed_y=vsy
        s_fire.alpha=1
        s_fire.red=vr
        s_fire.green=vg
        s_fire.blue=vb
        ListAddLast(I_fire,s_fire)
    End Function
End Type
'/TYPES

'TIMES
Global time_spawn=MilliSecs()
Global time_fps=MilliSecs()
Global time_expl_update=MilliSecs()
Global time_firespawn=MilliSecs()
Global time_reload=MilliSecs()
'/TIMES

'OTHER
Global scroll_y
'/OTHER

```



## Check.bmx

'Check.bmx

```

Function check_times()
  If MilliSecs()>=time_spawn+cfg_spawntime Then
    time_spawn=MilliSecs()
    T_Alien.newalien(Rand(100,700),-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
End Function

Function check_mouse()
  If MouseDown(1) Then
    If MilliSecs()>=time_reload+100 Then
      time_reload=MilliSecs()
      T_Shot.newshot(MouseX()-22,481)
      T_Shot.newshot(MouseX()+22,481)
    End If
  End If
End Function

```

## Render.bmx

'render.bmx

```

Function render_aliens()
  'Aliens
  aliens=0
  SetBlend Alphablend
  For S_Alien:T_Alien=EachIn L_Alien
    'Update der Gegnerposition
    aliens:+1
    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

    'Schild anzeigen
    SetBlend alphablend
    SetAlpha s_alien.pro_alpha
    SetBlend lightblend
    DrawImage img_protect,s_alien.x,s_alien.y
    SetAlpha 1
    SetBlend maskblend
    'Alpha abrechnen
    If s_alien.pro_alpha>0 Then s_alien.pro_alpha:-0.1
  End For
End Function

```

```

'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
    If s_alien.spindir=0 Then s_Alien.angle:+5
    If s_alien.spindir=1 Then s_Alien.angle:-5
    If s_alien.spindir=2 Then s_Alien.angle:+2
    If s_alien.spindir=3 Then s_Alien.angle:-2

    If MilliSecs()>=s_alien.deadtime+1000 Then
        If s_alien.scale>=1 Then t_explosion.newexplosion(2,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...
'Partikel erstellen
If s_alien.dead=0 Then t_fire.newfire(s_alien.x,s_alien.y-ImageHeight
(img_alien)/2+5,0,-1)s_alien.speed-2)
'/Partikel erstellen

'Deleten wenn das Alien y>=600 hat
If s_alien.y>=650 Then ListRemove(L_Alien,s_alien)
Next
SetBlend maskblend
'/Aliens
Return aliens
End Function

Function render_fire()
'Fire
For S_Fire:T_Fire=EachIn L_Fire
    SetColor s_fire.red,s_fire.green,s_fire.blue
    SetBlend lightblend
    SetAlpha s_fire.alpha
    DrawImage img_fire,s_fire.x,s_fire.y
    SetAlpha 1
    SetColor 255,255,255

    s_fire.alpha:-0.02
    s_fire.x=s_fire.x+s_fire.speed_x
    s_fire.y=s_fire.y+s_fire.speed_y

    If s_fire.alpha<=0 Then ListRemove(l_fire,s_fire)
Next
'/Fire
End Function

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

Function render_shots()
'Shots
For S_Shot:T_Shot=EachIn L_Shot

```

```

'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)
                'Alpha des Schildes auf 1 setzen
s_alien.pro_alpha=1
            End If
            'Hat das Alien unter einem Lebenspunkt
            If s_alien.health<=0 Then
                s_alien.dead=1
                s_alien.speed=-(s_alien.speed/2+1)
                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

SetColor 255,255,0
SetBlend lightblend
    DrawImage img_shot,s_shot.x,s_shot.y
SetBlend maskblend
SetColor 255,255,255

'Außerhalb des Bildschirms: Löschen
If s_shot.y<=-10 Then ListRemove(L_Shot,s_shot)
#end1
SetScale 1,1
Next
'/Shots
End Function

Function render_player()
'Eigenen Spieler zeichnen
SetBlend maskblend
DrawImage img_player,MouseX(),600-55
t_fire.newfire(MouseX()-15,570,0,1,50,50,255)
t_fire.newfire(MouseX()+15,570,0,1,50,50,255)

t_fire.newfire(MouseX()-35,570,0,2,50,50,255)
t_fire.newfire(MouseX()+35,570,0,2,50,50,255)
'/Eigenen Spieler zeichnen
End Function

Function render_explosions()
'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
    End Select
Next
SetBlend maskblend
'/Explosionen zeichnen
End Function

Function render_hud()
    'HUD und sonstiges zeichnen
    DrawText "FPS: " +fps,10,550
    '/HUD und sonstiges zeichnen
End Function

```