

## The Game Creators Newsletter Issue 33

After many years of development, changes, texturing, modeling, animating, sound effect creation, music composition, PR, artwork, testing and evaluation - FPS Creator is finally released! Over 3 years in the making we firmly believe it brings a new level of game creation to the masses. This new release issue you can read all about the release of FPS Creator, about the first of what we hope will be many community built content packs, and also an interview with Lee Bamber, the lead developer. Find out just what the release of FPS Creator means for the community as a whole. DBPro and SDK users may wish to dig into this interview, because the completion of FPS Creator now means changes and updates across the entire spectrum of products.

This month we've got two competitions running. One where you could win an ATI X800XL graphics card by designing an innovative FPS Creator map. The other where you can win some software by creating a greyscale map to go with Steve Vink's excellent DBPro Transitions tutorial. There are also some new games worth checking out, lots of new DBPro DLLs, a whole stack of free media, an exclusive (slightly strange!) video of the staff up Mount Snowdon, a 20-liner, forum chat and more!

As becoming the norm I seem to wind-up recent newsletters by giving out cool product or game recommendations. This is no exception, apart from the fact that it was TGC community member Grant (geece3) who first recommended this item to me! It is an X-ARCADE control pad. Picture the entire control section of an arcade machine ripped out and slapped onto your desk. All USB'd up to your PC and then run NAME. This is retro heaven! This is a board made out of genuine arcade machine parts and is fully utterly perfect in use. You can buy it one or two player variants and the joystick and 8 buttons can take a real hammering. The white "1 Player" start button just sets it off. There are even buttons on the sides for arnibal games. Yes, I truly had to die and gone to retro heaven. Check it out, and, if you're in the UK I cannot recommend strongly enough this excellent company who supplied mine next day, with faultless customer service.

Ok sales pitch over. Just before I go however, if you live in the UK and want to attend this years DB convention then check out the details [here](#). That's all, have fun with this issue and see you again in October.

Contents			
Editorial	FPS Creator Press Release	Interview with Lee Bamber	FPS Level Competition
Jo-Go	Advanced Animation Plugin	Slayer: Battle Tanks II	Sprite Collision
SnowDoom	Enhanced Audio Plugin	FPS Forest Pack	DBPro Tutorial - Transitions
TGC IRC Chat	Massive Media	Digger	Ripe for Remarks
From the Forums	20-Liner - Cave Flyer	This month's winner	Outro

### FPS Creator gives power to the people

Thousands of computer fans dream of making and selling their own games but lack the know-how to actually do it. Now there's a revolutionary new way to turn those dreams into reality with a PC game creation package that not only cuts out the need for programming knowledge but also gives you free licence to profit from selling your games.

Eliminating the need for 3D modeling knowledge, FPS Creator's intuitive 3D editor enables you to virtually paint your game world onto the screen, using Prefabs or Segments of drag and drop walls, doors and windows that click together with pixel perfect precision to construct 3D buildings of many stories and levels.

Adding entities is just as easy with 800 3D game objects to choose from, including 62 different enemies and 18 weapons. Dramatic sound effects and lighting increase the atmosphere and you can also import your own sounds, music, graphics or 3D models. Logical scripting makes it easy to customise every aspect of your game and direct the action across some 50 large scale levels with realistic physics, explosions and dynamic effects.

Once the game is finished it can be saved to CD and given to friends for multi-player death matches across LAN or Internet and for the first time with any creation package of this kind, you can sell your creations 100% royalty free!

"This is the first game creator that produces results of such a high standard," said Rick Vanner, Commercial Director of The Game Creators. "With a full physics system, great artwork, multiplayer, excellent graphics styles and a whole lot of other clever features, the creative possibilities are both endless and comparatively easy to achieve, and that's a combination that no other product offers. There are already 1,000 users of our early adopter version and they're voting a great hit."

FPS Creator is available today priced \$44.95 / €43.46 / £29.99

Web Site: [http://www.thegamecreators.com/?mview\\_productid=2001](http://www.thegamecreators.com/?mview_productid=2001)

### A Conversation with Lee Bamber

With FPS Creator finished we took some time to sit down and query Lee Bamber, the head developer, about the development process, future plans and what may lie in store for DarkBASIC Professional and the Dark Game SDK.

#### About FPS Creator...

**FPS Creator is finally out, how long has the development process taken?**  
Lee: It started just over three years ago, and has gone through many internal revisions before we arrived at the design you see today.

**Where did the original idea come from?**  
Lee: We have always created game creation tools, and we always seek to make the process easier and more flexible. FPS Creator is simply the next step in this evolution. The ideas within FPS Creator have been contributed by the whole team inside a three year melting pot.

**Which part of the project would you say provided the most enjoyment in terms of development?**  
Lee: The very best moment in the project was the testing of the physics system. The test part about the system is that the outcome is not known. You make a structure made-up of boxes, exploding barrels, furniture, etc and then throw grenades at it, and wait to see what happens! It transformed my test games from a predictable A to B design into a magical universe where anything could happen.

**Which physics system are you using in FPS Creator?**  
Lee: We are using the Open Dynamics Engine, ODE for short, within the FPS Creator game engine. It has the benefits of being very fast and easy to code for. The extra feature that did it for us was the fact the license allows you to distribute the final games for free, which means any games made you can sell royalty free. A vital ingredient for a tool that ultimately produces redistributable executables!

**Notice you said that the "FPS Creator Game Engine" uses it. What components are there for the game engine specifically?**  
Lee: One of the outpots of finishing FPS Creator was to prove that commercial products could be created using DarkBASIC Professional, our own programming language. The FPS Game Engine was written entirely in DBPro. During the next Upgrade session we shall be releasing the FPS Creator Game Engine source code to the DBPro community for free, allowing them to customise the actual engine itself.

**So this means any DBPro program could take the game source and modify it to add their own features? I.e. particles, a 3D HUD system or even RPG elements?**  
Lee: Exactly. You can create everything about the engine. Strip it down to a mere level loader, expand it to include vehicles, destructible scenery, anything. Tweaking elements to your requirements on a game by game basis, and then you may share your modifications with the community if you wish. We place very few restrictions on what you can do with the source code.

**Does that include selling any mods you make?**  
Lee: To a certain degree, yes. There is one restriction we must place to protect our interests and that is you cannot make "game creation" software using our source code (in whole or in part). DBPro programmers can create and sell their own source code that links into the game engine source code, but it must clearly not include parts of the source code that we release for free. Just to clarify though, you CAN sell your own games you make! You just can't call FPS C 2 with it.

**FPS Creator will have the source code released to the game engine, free for modifications, no license or royalty restrictions and no watermarks or splash screens - why the change to this business model from the traditional "license" one?**  
Lee: The "commercial license" is best employed with technology and tools that service the professional commercial industry, who typically have money to spend. FPS Creator was designed and priced for wide acceptance. Our intended customer would not appreciate a license, or anything, that impedes their progress to making and selling games. We wanted to remove all obstacles from FPS Creator to create the best product we could.

**If there was one feature you would really like to see in FPS Creator that it doesn't currently have, what would it be?**  
Lee: Vast populated exterior terrains and seas! Everything from green lush forest planets to the stark alien sand planet of Elk (Editors note: We have no idea where Elk is either).

**Will we be seeing any updates to FPS Creator in the near future? Is there some kind of release roadmap?**  
Lee: We have a short-term plan to release the FPS Creator Game Engine source code, and a Modern Day Graphics Update available in the future. We hope to continue making graphics packs for the product, as we anticipate this will be the number one request from the community. We will also be making updates available in the course of fixing any bugs that may arise.

**Which is your favourite character in the V1 release?**  
Lee: The character I tend to choose the most during my testing is Conker. The guy has spikes coming out his head and wears nipple clamps! Not sure what that means, but in my mind this one stands out as a real "character".

**What plans are there to educate the FPS Creator community?**  
Lee: One of the most effective tools I have found useful during the beta testing and PR events was the Video tutorial. I personally plan to create a series of quick video tutorials covering the more advanced aspects of FPS Creator. There is no better way to learn how to use a tool than watching someone else show you how to get around the interface.

#### About DarkBASIC Professional / Dark Game SDK

**What is the immediate plan of release / action with regards to DBPro?**  
Lee: The immediate plan is for Mike and I to take part in another bug week from which will be produced Upgrade 5.9. This upgrade will also include the new ODE and Multipolymer DLLs. These two DLL's will be unsupported command sets and are required for the release of the FPS Creator Game Engine source code, which we shall be releasing at the same time.

**What does "unsupported" mean?**  
Lee: Unsupported means they will be fully functional, and any bugs will be fixed during future upgrades, but there will be no official documentation of the command sets added into DBPro at the time of release. There are 76 ODE commands and 35 Multipolymer commands in total. It is also worth noting that there are some ODE related bugs that cannot be fixed due to the incomplete status of the original SDK, but hopefully over time these will be addressed.

**So would now be a good time for users to start collecting any DBPro bugs in the bug forum?**  
Lee: Yes. We are now planning when the bug week will commence, and our goal fee is sometime in the next two weeks (early October). So any users who want their bugs fixed first to release them in the bug forum, visit [http://www.thegamecreators.com/bugforum](#) and let us know.

**If 5.9 can be considered an FPS Creator integration build + bug fixes, what do you have in mind for 6.0 and beyond?**  
Lee: Upgrade 6.0 will include the language by dealing with functionality/incompatibility issues that have been raised by the community. It will address instances where the help says one thing and your experience suggests another. We will also be working on the compiler and debugger, focusing on speed and stability. We have no schedule on when this upgrade will begin.

**As DBPro is upgraded will we see those changes move across to the Dark Game SDK?**  
Lee: The relationship between DBPro and Dark Game SDK is a very close one. They share the same technology source code, and only differ in how they are compiled for end user usage. Any updates to one will be made available in the other, and only require compiling and releasing to make the benefit public.

**Are there any changes on the horizon for the SDK?**  
Lee: I feel there is a great potential for the SDK, and it opens up a world of possibilities for those programmers who want to profess to a language that is an accepted industry standard. One of the first changes will be the renaming of the product to the "Dark GDK" (Dark Game Development Kit), which better defines the identity of the product.

**Although there are no firm plans regarding this - do you see TGC going down the Windows Vista / DX10 route sooner rather than later?**  
Lee: We are going to take the first step in that direction when we all upgrade our development tools to the new .NET 2005 editions, which natively support the 64bit compilers. As stated there are no plans other than to conduct research into the viability of supporting future technologies. As for DirectX, as soon as the April 2004 DirectX 9.0c end user runtime is released, we will be supporting that asap. Visually an important move for Microsoft, and we will follow where the market leads us. There's no point being stuck with old technology. The computer world waits for no man!

**You've been creating "game making" software for most of your professional life, if you could make any other type of application, what would it be?**  
Lee: A tough question. Most of my brain has been hard-wired for game creation these days. I would probably write "life programs". Applications that are designed to learn on their own. You can buy it one or two player variants and the joystick and 8 buttons can take a real hammering. The white "1 Player" start button just sets it off. There are even buttons on the sides for arnibal games. Yes, I truly had to die and gone to retro heaven. Check it out, and, if you're in the UK I cannot recommend strongly enough this excellent company who supplied mine next day, with faultless customer service.

### FPS Creator Level Design Competition

Win an ATI X800XL Graphics Card + 5 Free Custom-built Models!

Here's a competition specifically for FPS Creator V1 users - create us a stunning level and we'll give out some prizes for the best one! We are looking for interesting use of physics in your level. So don't concentrate too much on how complex or large your level is, just make sure it's a really good / fun physics demonstration! Some ideas to get you thinking: 1) Create a large "domino run" style map and let the user start it off. (2) Donkey Kong! Have the player located on a floating platform with spawning barrels and get him to kill the enemies by hurting them below. (3) A Bridge too Far! Create a map where the player has to build a bridge from the pieces you provide to cross a chasm.

Your only restriction is that you must only use media provided with FPS Creator V1 (full or demo version), so no including your own sounds, images, models, etc. - not in this competition anyway :)

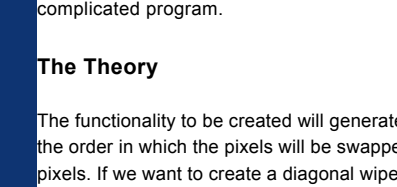
We've two really cool prizes for the winners: 1) First of all you'll get a brand new state of the art ATI X800XL graphics card. This beat can't only show polygons and shaders as good as there is no tomorrow. You will also win FIVE CUSTOM BUILT MODELS, created by our 3D artist Simon Benge. You will get to specify exactly what the model is and Simon will model and texture them for you. The final models will be released free to the FPSCC community. The only restriction on this is you cannot have a character or weapon, it has to be a single entity object. But you get to pick what it is, with no restriction on the game either.

Do your best and get map making. Keep your eye on the FPS Creator forum to find out where to send your finished FPM files and we'll pick a winner at the end of the month. Full competition details will be posted to the FPS Creator web site in the next couple of days, so use this time as a head start to get creative. The deadline is the last day of October 2005.

Web Site: [http://www.thegamecreators.com/?mview\\_productid=2001](http://www.thegamecreators.com/?mview_productid=2001)

### Jo-Go

Well what do you know, just one month after announcing the winners of our Puzzle Game Competition we receive a message from Living Tree Software about their DarkBASIC Professional created puzzle game: Jo-Go. Described as "very similar to Chess in the approach required (both defensive and offensive playing strategies)", Jo-Go allows you to play against the computer or against friends via its Play-By-Email feature. The following is from the authors announcement:



"Jo-Go is the ultimate tactical board game. Some believe it to be the perfect "off-season" trainer for games such as chess since it teaches you to position your pieces for both offensive and defensive action and quickly shows you the consequences of a misplaced capture of an opponent's piece. If you like chess, you'll love Jo-Go. If you hate chess, you'll still love Jo-Go. The game is deceptively simple yet complex enough to challenge your skills for ages to come. Plus, it's a proven fact that visual games foster intellectual development. Perfect for kids and adults. It's a thinking game that will get your right and left brain working together."

Both players receive eight pieces with 1-5 beads on each piece's pole. Each piece moves in a set pattern. If your opponent passes one of their pieces over to you in the course of its assigned movement pattern, you lose 1 bead. If they land on you, you lose 2 beads and you switch places if your piece still has newcomers! If the players are at different playing abilities, you can even handicap the more experienced player by giving the weaker 1 or 2 extra beads on each piece at the start of a game.

The computer opponent has the integrated PBEEM play-by-email system and with the unique handicapping system you can try out the harder levels with a little extra help in the form of extra beads for your pieces. The levels of difficulty (PBEEM play-by-email) system lets you play anyone in the world. All this you can do using the handier levels by Jo-Go and our server. Sign up for a free account and start playing today with the unregistered version of Jo-Go. If you like the game after 10 turns, you'll be asked to register to further development of Jo-Go and help cover the costs of our server expenses.

Jo-Go features 10 different game boards, 5 different player colors, and an incredibly easy to use interface that actually helps teach you the game. Jo-Go is available to buy on-line now for \$19.95.

Web Site: <http://www.jo-go.com>

### Advanced Animation plugin for Dark Basic Pro

DBP Advanced Animation plugin (AA) supports all of the modern boned animation techniques. For example:

- Loading animation sequences separately from model.
- AA can perform animation blending. This means that you can execute multiple animations at the same time and AA will blend them together smoothly.
- With AA you can make smooth transition from one animation to another.
- With a little work, you can fully manipulate the individual parts of your characters.

This is useful when you want to create truly dynamic motion at runtime without the aid of predefined animations. For instance, if you wanted your character's head to turn as a car goes past him, you could control the heads rotation directly to keep facing the moving car. Over 84 new powerful commands! Simple for understand and coding! Features:

- Load unlimited animation sequences
- Blend animated sequences
- Management of set of layers
- Cut/Copy/Paste/Delete layers
- Layers blending
- Advanced stringset functions
- Set of additional functions

Download a free demo or buy the full plugin for only \$5.99 from the web site.

Web Site: <http://www.accode.com/dbp.htm>

### Slayer: Battle Tanks II

We featured the original game last issue in a games round-up, well the author Rhett Lowler is back and it again with Slayer: Battle Tanks II, a full 3D sequel to the original title. Created with DarkBASIC Classic you can challenge your friends against a variety of tanks (21 to be exact) and battle your way through 4 arenas, as well as game scenarios. You can play in arenas such as Abandoned Island, Fort McDonald, Castle and City Streets. Game styles include Capture the Flag, Shoot-out, King of the Hill, Single Player Challenges and Goalender.

You can play with a joystick or two players on keyboard and mouse. If you were to play one different game (with differing combinations of tanks, arenas, scenarios, etc) per day, then you could play a game for over 154 years! Grab the free download and give it a blast.

Web Site: <http://www.slayerbattletanks2.tk>

### Sprite Collision Tutorial

Some time ago Andy Igone of Banshee Studios wrote a complete tutorial for DarkBASIC Professional about how to handle pixel perfect collision detection with sprites. Originally charged-for, the tutorial has been released free of charge and is now a free download. The main difference with this tutorial is that the code still works perfectly even when sprites are rotated. Visit the Banshee web site for lots of fine game downloads including DBPro and Blitz created titles.

Tutorial Download: [banshee\\_collision.zip](http://banshee_collision.zip) (30KB)

### SnowDoom

Who says games developers never leave their coding dens? On September 3rd both Lee Bamber and Rick Vanner of The Game Creators, along with 30 other staff, set off to Snowdon in North Wales for charity. Snowdon is the highest mountain in England and Wales, rising 1085m (3560 ft) tall, also known by its local Welsh Language name Yr Wyddfa (The Tomb). The event was sponsored with all proceeds going to the Entertainment Software Charity, an organisation devoted to help disadvantaged young people realise their potential by providing training, education and support of youth charities. In total over £6000 was raised and as part of their sponsorship agreement both Rick and Lee acted out a "Live" scene from FPS Creator at the summit of the mountain! If you feel your sanity can take it, we've uploaded a video of this event. Grab the video from a link below.

Video: [snowdon.wmv](http://snowdon.wmv) (640 KB WMV format)

### DarkBASIC Enhanced Audio Plugin

Tomey has released an extremely powerful DLL for DBPro that includes playing Ogg Vorbis, MP3, FLAC, WAV (uncompressed), AIFF, MOD, S3M, XM, and IT files. Extra features include the use of 3D world Trigger Zones with audio that you load into your game. You can load, play and loop audio, set the volume, the panning, the pitch shift, return the audio length and position (as well as being able to set the position), alter the playback device and create trigger zones. All in all, this is a superb free package for anyone looking to use the above audio formats in DBPro. It's a fully automated installer, so there's really no excuse not to use it!

We've mirrored the downloads here (709 KB) and here (144 KB) (this version for those that can't get the installer working). But check out the Forum Thread below for the latest news and details, and post your comments.

Visit the Audio DLL Forum Thread

### FPS Creator Forest Pack

Expand a field of your vision! Do not limit itself to wait! Go to open space! Go to the forest. The FPS Creator Forest Pack includes 32 objects (ground, various herbs, stones, trees). Also seasons (autumn, winter, summer) are submitted.



You can purchase the Forest Pack online for only \$3.99 from the AC-Code web site.

Web Site: <http://www.accode.com/psc.htm>

### DBPro Tutorial - Design and Deliver by Steve Vink

This month's tutorial is all about thinking like a programmer, rather than thinking like a games player. The difference between how a game works is the secret to opening up a world of opportunity, and involves a degree of lateral thinking. Some would say my mind is in need of a serious psychological makeover, I prefer to think I'm creative.

Last month, we took a step into the world of memblocks, and manipulating the display to produce some nice transitions. Using the knowledge we already have, we'll take this a step further, and create scene transitions that are limited only by your imagination! This tutorial is not just about the code, but also the design. We are looking at concepts such as preprocessing, compromise, optimisation, advanced memblock manipulation and graphical effects. We cannot achieve the special effects without finding a way to process the data, and this design process is what we will be concentrating on.

The code you will see and test is a more advanced form of last month's code. All of the basic commands are the same, but we will be putting them to better use, in a more complicated program.

#### The Theory

The functionality to be created will generate a transition from one screen to another, using a mask. The mask is greyscale, and represents the order in which the pixels will be swapped from one image to the next. Black pixels will be affected first, sweeping through to the white pixels. If we want to create a diagonal wipe from top left to bottom left, the mask image would look like the illustration opposite.

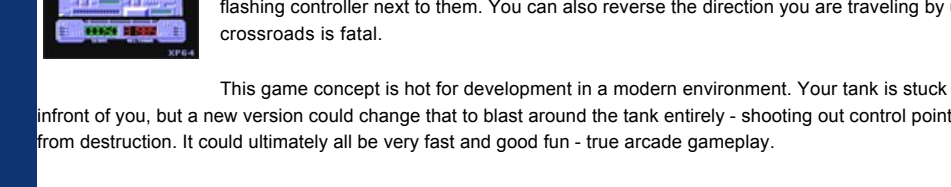
The 3 components of this function are the screen image, the new image and the mask. The mask contains 255 levels of grey. So we must take 255 snapshots of the new image, as dictated by the mask, and overlay them one by one. To put this in perspective, 255 frames in a movie or TV program equates to around 10 seconds. Once finished, our Dark Basic Professional code will achieve the transition as quickly as it can, in a second if we desire. Quite a task, and if you want to see how it's done, read on and be amazed at just how powerful Dark Basic Professional is!

#### Interpreting the mask

The mask is a simple image, and can be created in any graphics program, and saved as a 32 bit image. There would be a pixel, it will follow the same structure we investigated last month. That is, 3 double words to describe the width, height and depth. The rest of the data is then converted to a memblock, it will follow the same structure respectively. We will use the red channel to extract the greyscale information, as this is the accepted standard; the remaining data is irrelevant. In a greyscale image, all colour channels have the same value. The framework of the code to read the mask will look like this:

```
for m = 0 to 479599
if memblock byte(mask@memblock, m*4 + 14) = n
pixel = m
pixelline = pixel / 800
pixelcolumn = pixel - (pixelline * 800)
screenPosition = (pixelline * 320) + (pixelcolumn * 4)
newImagePosition = (pixelcolumn * 320) + (pixelcolumn * 4)
copy memory get pixels pointer() + screenPosition, ...
get memblock ptr(newImage) + newImagePosition + 12, 4
endif
next m
```

In this code, we are reading a memblock representing an 800 \* 600 image. That is 480,000 pixels multiplied by 4 bytes per pixel, plus 12 bytes for the header information. n is the step we are working with at any given time, 0 to 255. As we analyse the full image pixel by pixel, we must step through 4 bytes at a time, add 12 bytes to bypass the header, and add 2 more bytes to get to the red channel.



#### Refining the Approach

At the simplest level, we could do exactly as described. We could take 255 separate images of just the pixels we need for each step, and paste them one by one at a reasonably fast process. In the example we will create, the images are 800 pixels by 600 pixels. That equates to 480,000 pixels, multiplied by 4 bytes per pixel. 1.82 megabytes of data to represent the image. Storing 255 snapshots would require just short of 500 megabytes of memory. It's time to think outside the box, and find a suitable compromise. This approach is not feasible.

Thinking about the sequence of events, we could actually achieve this effect with just one image, not 255. We can store cycle through the 255 levels irrespective, so why not do this in the fly? We could pick out all of the black pixels, map them to the new image, pick them and transfer them to the screen. After the SYNC results, move on to the pixels with a value of 1 and repeat the whole process. This eliminates the need for multiple copies of the image.

#### Example 1: Working On-the-fly

Example 1 shows this effect working "on-the-fly", just as we have described. Review and run the code, and you will see a good representation of the effects we are trying to achieve. The images follow the diagonal mask supplied. The first observation is the speed. My 3 GHz computer averaged 30 to 50 frames a second. To use the full range of 255 levels would take 5 to 8 seconds to complete. It's good, but not that good. We need to think again.

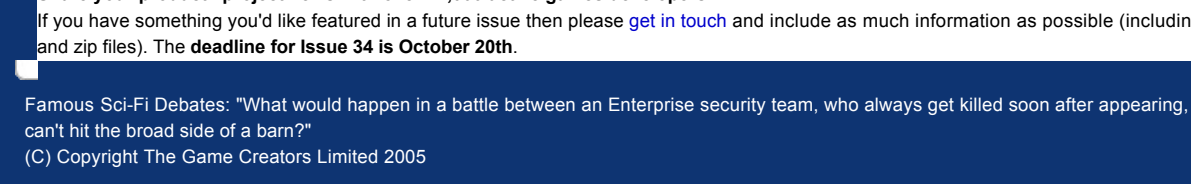


We can assume from this test that 5 seconds is required to process the mask. We can, however, hide those 5 seconds elsewhere by preprocessing the effect. This is normal, and you encounter it every time you play a game. Loading screens, level information and title screens all hide the preprocessing required to run the game. We need to find a way to make an efficient screen transition, running much faster than our current solution, and far more practical than saving 255 screen images.

Right now, our screen data is organised by row and column. We have no idea which pixels are to be processed first, which are to be processed last, not mention all of the steps in between. The solution to our problem lies in this last statement. What we need is a chronological view of the data. Once the pixels are processed by time rather than position, we can process them sequentially, and only process the ones involved in the current cycle.

#### Chronological Processing

So how can this approach be implemented? As with the previous attempts, there is always a trade-off of resources: a compromise. We will require a second copy of the image, in a memblock, with the data arranged in chronological order. We will need to store the pixel numbers as double words, which means that the new memblock will be the same size as the image itself, minus the 12 header bytes. We also need a structure which records the break between each of the steps of the transition, as we cannot record this in the chronological order. We can do this by using a small array of 255 elements. If, for example, there are 3,000 black pixels in our mask, we will record these 3,000 locations in our new memblock, and write the value 3,000 to the first element of our array, instructing our program to SYNC after 3,000 pixel replacements. We will then continue with pixels in the mask of a value of 1, adding them sequentially to the memblock and recording their position in element 2 of the array.



Our compromise has allowed us to optimise our data. In the second example project, the function has been divided into 2. The first function loads the mask, converts it to a memblock, and then creates a second memblock with the data rearranged chronologically. An array is created and the quantity of each pixel is recorded. The second function iterates through the 255 levels of the mask. It reads the quantity array, replaces the specified number of pixels by reading the mask memblock sequentially, and SYNCs the display. It then reads the next level of mask data and so forth.

Has the addition of transitions worthwhile? You will notice that the second example has had the frame rate clamped, whereas the first was allowed to run as fast as possible. On my computer, the effect ran at a staggering 650 frames per second! That is more than a tenfold speed increase. This allows me to create the effect I want at the speed I want. The effects are also more uniform in their motion, and the images are 800 pixels by 600 pixels. That equates to 480,000 pixels, multiplied by 4 bytes per pixel. 1.82 megabytes of data to represent the image. Storing 255 snapshots would require just short of 500 megabytes of memory. It's time to think outside the box, and find a suitable compromise. This approach is not feasible.

#### Alternative Solutions

The final solution to this line-based code was not the only one considered. To highlight how "brainstorming" a problem can produce many solutions, here are a couple of other possibilities, and the reason why they were rejected.

- Halving the number of transition steps to 128. This would half the workload when creating the mask. There would be a slight increase in each step, having to look for pairs of values in the mask, rather than just a single value. In my personal opinion, 128 steps is not as smooth a transition as I would like, and, results in loss of smoothness in the effect.
- Halving the Mask size. Essentially, reducing the height and width of the mask reduces the data size to just 25% of the original. This is certainly a huge reduction in workload, slightly offset by the fact that each pixel must represent 4 replacements in the screen image. Again, I decided against this because the net effect would be a more blocky transition, with "pixelation" of the display.

When considering the above options, another important point comes to light, regarding the optimisation of the code. In a loop, you can maximise performance by reducing the code to a bare minimum. To add 1 line of code in order to half the inner loop of 480,000 iterations, we are asking the computer to loop out 240,000 additional lines of code. If we add 1 line of code to the outer loop of 255 iterations in order to facilitate more efficient code, we are only adding an additional workload of 128 lines of code in order to improve our performance. Just 0.05% when compared to modifying the inner loop! Had we chosen either of these two alternatives, this is another factor that would need consideration.

The net effect of each of these solutions, including the one we chose, is identical, but the performance and quality of each one varies enormously. I hope this has given a rewarding insight into thinking laterally to improve your code. I also hope you can get creative with masks and create some unique and interesting effects. You could create a mask using text, your signature or silhouetted objects. How about a couple of characters in pure black, and a lighter, more vibrant background?

#### Transitions Competition

In conjunction with this tutorial The Game Creators are offering up two prizes for the two best-looking transitions created. Check out the Forum Thread about it and get greyscaling!

Until next time, Happy coding!  
Steve Vink

Download the tutorial + resources: [DBPro\\_Tutorial\\_Memblocks\\_2.zip](http://www.thegamecreators.com/?mform_view&id=61135&id=2) (477 KB)

### TGC IRC Chat Session

On the 26th of September (that's the Monday coming folks) The Game Creators team will be having an irc.dohat.net in channel #darkbasic and a Q & A session. Post your questions to the team. The session starts at 9pm GMT politely (not sure what time that is in your country? Then look up here). Lee, Rick, Mike and Rich will be attending. Go to the Forum Thread if you need help obtaining and configuring an IRC client.

IRC Chat Thread: [http://forum.thegamecreators.com/?mform\\_view&id=61135&id=2](http://forum.thegamecreators.com/?mform_view&id=61135&id=2)

### Massive Media

We've featured his number of times in this newsletter and this month sees a welcome return as geece3 (aka Grant Clark) hits back with no less than a whopping 500 seamless tiles and 15 seamless landscapes, all sorted into packs for easy downloading. All titles in PNG format, most are 256x256 pixels. More amazingly they are all hand-drawn (no "generator" program in sight, this is pure Corel Photo Paint handwork, people). Grab them from the site below and leave your feedback in the forum thread.

You'll find all graphics here at <http://members.thegamecreators.com/geece3/>

#### Tronic Tiles