

## The Game Creators Newsletter Issue 31

Welcome to Issue 31 of the Game Creators newsletter. As another month of Summer goes by, we link ever closer to Autumn and that festive period. A time of year traditionally reserved for the best hardware and game releases, this year looking to be no exception. But for those that can't wait a few months we've got less than 53 stonking new games here for you! That's right, the Puzzle Game Compo deadline has been and gone, the games tested and verified and they're all presented here for you to play. So get downloadin', get coding and we'll announce the winners next issue.

In other news we've got a brand new FPSCom Creator available - a catch-up session with the head developer Lee Barber. The front cover this month is the great DOS game Alien Treasure. There are also tutorials for DBPro, PlayBasic and OmegaBasic, plus a guide from the PC Extreme team on getting old DOS games running again.

Before I go I'd just like to give a mention to the Razer Ironman mouse. Now this is a bit of a stretch, but they sponsored this or anything, but just because I bought myself a new limited edition Razer Diamondback chrome mouse + an ExactMatch mat to run this bat on, I have to tell you... they're staggeringly good!

And it all glows a cool blue colour too. The mouse is shaped with elegantly elongated buttons that your fingers nestle into after the sensitivity on the fly (up to a staggering 1600 dpi), and it also has a cool scroll wheel. The scroll wheel is built over a control or speed mode. The mouse itself has a very slick feel to it and its ergonomically shaped for long-haul gaming sessions (or just straight Windows use). Pulling off those wild head shots in Battlefield 2 has never been slicker :)

So if you're in the market for a new desktop critter, I'd heartily recommend them. And now, on with this issue!

## Contents

EZ Serv 1.4	Puzzle Games Galore!	PlayBasic Tutorial	FPS Creator News
DBPro Tutorial - Timing	Smart Particles	Review: Alien Treasure	Omega Basic Tutorial
IGF 2006	IDB Dreamer	New DBPro DLLs	From the Forums
gameSpace Mini Course	Synergy IDE	This months winner	Outro

## Puzzle Games Galore!

The Puzzle Game Compo deadline has come and gone. We've no less than 53 games entered, some of which are truly commercial quality.

The programmers have all done their hard work - and now it's up to us to judge the winners. This is a bit of a stretch, especially for those of you on S60. Even if you DO download them all, it will eat into your time to play the games. The most points wins.

Before I list all the games I will just like to point something out: there is 350 MB worth of games in total. Now this is a bit of a stretch, especially for those of you on S60. Even if you DO download them all, it will eat into your time to play the games. The most points wins.

I don't attempt to do this alphabetically! Otherwise some of the excellent titles towards the bottom of the list will never be seen, and you'll really miss out. Instead, I usually read you the game summaries that I've written and download game bases on those alone - whereas if they sound interesting to you, then do them. Graphics, especially for puzzle games, should be a deciding factor, though.

If you think you've got a game that would be a good candidate for the competition, then write me an email and let me know if you think it's interesting to others. Then I will add constructive comments about the game at the same time as voting if you wish. In approx. 3 weeks time we'll add up all the votes. The winners will be announced in the next newsletter.

We have a unique thread on our forum for each game. The first message of the thread is the game title, a screen shot, download link and a summary. You are free to add your comments to the thread about the game (please note that we'll actively delete all flameys) - go to the [Puzzle Game Compo Board](#) to access this.

How's it going?

Visit the [Puzzle Game Compo Board](#), click on the thread and cast your vote using the form at the bottom. You **MUST** be a TGC forum member, and yes, you can vote for 1 point on your own game if you pick 4 games out of those entered. The first game will be awarded 4 points, the second 4 points, the third will get 2 points and your final choice will get 1 point. You can leave constructive comments about the game at the same time as voting if you wish. In approx. 3 weeks time we'll add up all the votes. The winners will be announced in the next newsletter.



## PlayBasic - Types Inheriting the earth

In last months newsletter, we touched upon shape, circular and collision detection. While these are part of PB (and we installed them), these included Rotated, Vector shape, Circular and collision of Sliding Sprite Collision. While these collision features represent a vast leap forward to collision systems, my development effort is still focusing on improving the collision system beyond that. To do this means thinking not just about detecting intersections, but also fixing the user commands to handle things like Sticky Impacts (below), Los (line of sight), Nearest Colliders and Ray to Sprite intersections. These are slated for inclusion in the next 1.0b6, which should be available in a few weeks time.

In the meantime, let's dig a little deeper into PlayBasic types.

**Inherited Types**

One of the most tedious aspects of writing programs, often occurs when we need to write code that performs common tasks upon different typed arrays. This is particularly evident in game programming, if you quickly Aliens, Enemies, Alien Bullets, etc. items get into the mix. It's the most common dependency between objects. If you have four or more items (Players, Players, Aliens, Aliens, Bullets, etc.) of objects in the mix, then depending on what you do with them, you will end up writing some sections (often identical) to handle the base creation/deletion etc over and over and again.

To help solve this situation and allow you write more streamlined code, we introduced an inheritance feature into PlayBasic types. What this feature does is it lets you create parent types, from which multiple child types can be derived. Since the child types are descendants of the parent, they inherit the parent types fields. But more importantly, the descendant children type arrays can be passed into their parent's type code.

Ok, that's a real mouthful. So let's dissect it little more with an example. They were building a game and we want to have various typed arrays holding information about our Players, The Aliens and their Bullets. If you think about it, even though these arrays store information about totally different groups of objects, some information is going to be common across the board. Like the objects Status (Alive, Dead, etc), Position, Score, Lives, HP/Points/ Sprite/Index/Image Indexes etc. So these most common fields could be placed within a Parent type. So our basic Parent type could look something like this:

```
Click here to view the source code  
(opens a new browser window)
```

Since each of our different objects will need their own collection of unique fields, we'll base them off the BasicObject parent type, which might look something like this:

```
Click here to view the source code  
(opens a new browser window)
```

So far we've outlined some user defined types and created our arrays for our game, while we've already saved some effort, the real power comes through our ability to pass these types arrays into functions. To do this, we'll develop a centralized set our common functions, so that can be used upon all these objects. Things like creation / deletion / positionning / animation / movement etc, whatever you like. These functions use the original parent type (BasicObject) as an input parameter. Now, since our games objects (players/aliens/bullets) are all descendants of the parent type, PlayBasic will not accept BasicObject typed arrays as input, but it'll also accept it's children. So our Players, Aliens & Bullet arrays can be passed in an manipulated also. As an example, here's what's possible creation and position of functions might look like:

```
Click here to view the source code  
(opens a new browser window)
```

So now we have types and some basic functions we can set about creating and position the various objects like this:

```
Click here to view the source code  
(opens a new browser window)
```

Ok. While this example doesn't really do anything whiz bang visually, it does show how you can build a set of base functions that can be used across various object arrays, not only that, but this approach could be used to build your own base layer of functions that can used across a various games. This approach will not only reduce the size of your programs dramatically, but make your programming experience easier in the long run.

You can find another full example posted on our [PB forums](#).

**Sticky Collision Demo PB1.081**

This last second example shows one of the most recent collision modes called "sticky". What sticky collision does is calculate the first impact point of the sprite with the world. From this impact point you could simply destroy the object or even calc reflection angles to rebound sprites of the environment, rather than sliding them. The following tech demo shows Sticky Impacts being used to rebound some 250 sprites off their randomly generated environment. A bit like a pinball game without gravity.

Download Rebound Balls (Sticky Collision) Demo with Source Code included.

PlayBasic is on sale now for \$34.95 (\$26.99, £18.99)

[Web Site: http://www.thegamecreators.com/mnview\\_product?id=2001](http://www.thegamecreators.com/mnview_product?id=2001)

## PC Extreme - Special Subscription Offer

You know that video we promised a while ago on the FPSC forums? (sheepish grin) Well, it's finally here! FPSC Producer Rick Vanner sat down and recorded various gameplay sessions including demonstrations of physics, new AI and multiplayer and we've put them together into a short piece for you.

Here we have a catch-up Q & A session with Lee Barber about the current development of FPSC Creator:

Q: What do you feel is the biggest new feature since EA?

A: Physics, physics and physics. It transforms gameplay into something amazing!

Q: Can you give a small example of how it can extend FPSC users level / game design?

A: Imagine you have a switch on a wall, which when pulled opens a trap door, which makes a box fall through it and down some stairs, which hits a boulder, that crashes down ramp and squashes the enemy that you were behind. Now this was a lot more than just a cool effect, it was powerful.

Q: And users can edit all of these physics properties directly from the Windows interface?

A: Yes! The designer has complete control over object weight, friction, delta, whether the object explodes when damaged, etc - but if it spins you can change that yourself.

Q: What other changes are regular updates going to bring?

A: We have released almost 50 patches since the last update, both to fix bugs and to add new features. Some of these patches are quite major, such as the addition of a new physics engine.

Q: Can we expect more intelligent opponents?

A: In the EA era, they had only one response, which was to run at you and shoot. In V1 we have added behaviours which allow characters to duck behind crate and pop up to shoot you from time to time, and you have the strafe behaviour where the enemy dodges when hit, and randomly strafes to avoid being hit in the first place.

Q: All without complex FFI script changes?

A: Correct, just enter the Entity properties, click the MAIN field, and change it to CROUCH/FPI or STRAFE/FPI. Naturally we will increase this library with add-ons after launch.

Q: Talking of launch, I know it's the million-dollar question, but are we getting close now?

A: As I write this I'm pulling together the finer aspects of BE74, which is 98% function complete and represents an almost final product. We have the test group signed up now to help test it out over the next few weeks, during which the software will get become more stable and faster. It is already well tested, but now we the beta group are testing the final build versions.

Q: We know the new build supports multiplayer - but getting massive numbers working in multiplayer would have been extremely difficult, so are there any considerations users should take when setting up a multi-player session?

A: It's important to understand what the MP features are in relation to deathmatch levels, or what we have termed Arena Games. This means we have limited things that would normally be used to tell a story, or challenge the single player. For example, only doors open in MP, whereas lifts remain static. Also only arms, weapons, health and characters are dynamic; everything else is static. There are other limits too such as one level per Arena to keep the file size and game complexity down. The idea is that the restrictions allow the end user to create very fast arena levels that provide the best gameplay when played with your friends.

Q: What do you call your player in on-line PBSC frag mode?

A: I think you mean the frag mode in the PBSC frag mode, so there's no chance you know who to tag now!

Q: Any final tidbits you'd like to share with readers?

A: The main focus has been on doing Physics and Multiplayer, as best we can. You can review the details in the FPSC forums. I hope you enjoyed reading this article, and I hope it has given you a better understanding of what's involved in creating a game.

Review: [http://www.thegamecreators.com/mnview\\_product?id=2001](http://www.thegamecreators.com/mnview_product?id=2001)

More details: [http://www.thegamecreators.com/mnview\\_product?id=2001](http://www.thegamecreators.com/mnview_product?id=2001)

Back to design

You may have noticed that the characters in last week's demo were a little bit like the ones in the first few levels. The idea was to make them look a bit more like the ones in the later levels.

Q: What do you feel is the biggest new feature since EA?

A: Physics, physics and physics. It transforms gameplay into something amazing!

Q: Can you give a small example of how it can extend FPSC users level / game design?

A: Imagine you have a switch on a wall, which when pulled opens a trap door, which makes a box fall through it and down some stairs, which hits a boulder, that crashes down ramp and squashes the enemy that you were behind. Now this was a lot more than just a cool effect, it was powerful.

Q: And users can edit all of these physics properties directly from the Windows interface?

A: Yes! The designer has complete control over object weight, friction, delta, whether the object explodes when damaged, etc - but if it spins you can change that yourself.

Q: What other changes are regular updates going to bring?

A: We have released almost 50 patches since the last update, both to fix bugs and to add new features. Some of these patches are quite major, such as the addition of a new physics engine.

Q: Can we expect more intelligent opponents?

A: In the EA era, they had only one response, which was to run at you and shoot. In V1 we have added behaviours which allow characters to duck behind crate and pop up to shoot you from time to time, and you have the strafe behaviour where the enemy dodges when hit, and randomly strafes to avoid being hit in the first place.

Q: All without complex FFI script changes?

A: Correct, just enter the Entity properties, click the MAIN field, and change it to CROUCH/FPI or STRAFE/FPI. Naturally we will increase this library with add-ons after launch.

Q: Talking of launch, I know it's the million-dollar question, but are we getting close now?

A: As I write this I'm pulling together the finer aspects of BE74, which is 98% function complete and represents an almost final product. We have the test group signed up now to help test it out over the next few weeks, during which the software will get become more stable and faster. It is already well tested, but now we the beta group are testing the final build versions.

Q: What do you call your player in on-line PBSC frag mode?

A: I think you mean the frag mode in the PBSC frag mode, so there's no chance you know who to tag now!

Q: Any final tidbits you'd like to share with readers?

A: The main focus has been on doing Physics and Multiplayer, as best we can. You can review the details in the FPSC forums. I hope you enjoyed reading this article, and I hope it has given you a better understanding of what's involved in creating a game.

Review: [http://www.thegamecreators.com/mnview\\_product?id=2001](http://www.thegamecreators.com/mnview_product?id=2001)

More details: [http://www.thegamecreators.com/mnview\\_product?id=2001](http://www.thegamecreators.com/mnview_product?id=2001)

Back to design

You may have noticed that the characters in last week's demo were a little bit like the ones in the first few levels. The idea was to make them look a bit more like the ones in the later levels.

Q: What do you feel is the biggest new feature since EA?

A: Physics, physics and physics. It transforms gameplay into something amazing!

Q: Can you give a small example of how it can extend FPSC users level / game design?

A: Imagine you have a switch on a wall, which when pulled opens a trap door, which makes a box fall through it and down some stairs, which hits a boulder, that crashes down ramp and squashes the enemy that you were behind. Now this was a lot more than just a cool effect, it was powerful.

Q: And users can edit all of these physics properties directly from the Windows interface?

A: Yes! The designer has complete control over object weight, friction, delta, whether the object explodes when damaged, etc - but if it spins you can change that yourself.

Q: What other changes are regular updates going to bring?

A: We have released almost 50 patches since the last update, both to fix bugs and to add new features. Some of these patches are quite major, such as the addition of a new physics engine.

Q: Can we expect more intelligent opponents?

A: In the EA era, they had only one response, which was to run at you and shoot. In V1 we have added behaviours which allow characters to duck behind crate and pop up to shoot you from time to time, and you have the strafe behaviour where the enemy dodges when hit, and randomly strafes to avoid being hit in the first place.