



DEV.MAG

CREATE • DEVELOP • EXPERIENCE

DEV.MAG
DOING THE
ROUNDS

IMMERSION

BACK OF A
NAPKIN
PART 1

WHAT TO KNOW
ABOUT
FRAMEWORKS

NEW WHAT MAKES
BLIZZARD
SO GOOD?

EMBARKMENT 2005



18



04

Contents

Regulars

3 - Ed's Note

4 - News

Feature

6 - Dev.Mag Doing The Rounds

Review

8 - Sin

Spotlight

10 - National Defense 2

12 - Shotbreak Games

Design

14 - Back of a Napkin: Part1

16 - Immersion

18 - What makes Blizzard so good?

19 - What to know about Frameworks

Mobile

21 - Games on the go

Tailpiece

22 - A little from I.T Intellect

CONTENTS



07



05



17

Ed's Note

The first issue proved to be a learning experience. You simply can't start an e-magazine and hope it's an immediate success. It's something you have to work towards. As simple as it may sound, it took me awhile to figure that out.

From the hosting problems to misunderstandings, the first issue was plagued with problems. This issue, on the other-hand, flowed together nicely. None the less thanks to Nandrew and all the effort he's put into this project. Not to mention the combined effort of our other contributors.

This issue will see some changes to the magazine. The first and most obvious being the layout change. The second is the launch of our review section, which will expose you to projects other developers have been keeping themselves busy with. As I have said before, we're still growing. The magazine will be changing quite a bit during the next few issues as we begin to become use to writing and editing Dev.Mag.

As a final note, we've also started an advertising partnership with PressX -- a fellow free e-zine, filled with previews and reviews of the latest games. Hopefully this will be the beginning of a long friendship and we wish them the best of luck.

That's all for now

Stuart 'GoNz0' Botma

THE TEAM

TEH_BIGWIGZOR

Stuart "GoNz0" Botma

TEH_L33T_2ND

Rodain "Nandrew" Joubert

TEH_DESIGN_PWNAGE

Claudio "Ch1ppit" de Sa
Brandon "CyberNinja"
Rajkumar
Paul "Higushi" Myburgh

TEH_RITERS

Simon "Tr00jg" de la Rouviere
Ricky "Insomniac" Abell
William "cairnswm" Cairns
Bernard "BurnAbis" Boshoff
Danny "dislekcia" Day
Andre "Fengol" Odendaal
Yuri "knet" Oyoko
Heinrich "Himmler" Rall

**To join, make suggestions
or just tell us we're great,
contact:**

devmag@gmail.com

This magazine is a project of
the NAG Game.Dev forum.
Visit us at www.nag.co.za

All images used are copyright and belong
to the respective owners

DIGITAL MAYHEM



"Unfortunately, in the dying days of the "Chuck" craze, our own poor Higushi still thinks about him way too much. Makes for a good comic, though"

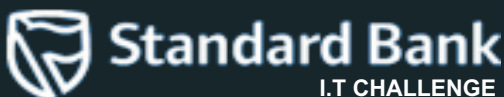


DIGITAL STOMPIES

Programming guru Bruce Eckel



(below) The Standard bank I.T Challenge



Thinking in C eSeminar Available

Bruce Eckel, renowned author of programming books such as *Thinking in Java*, *C++ Inside & Out* and the presenter of hundreds of seminars and lectures worldwide, is releasing a beta version of a free eSeminar based on *Thinking in C*. The purpose of this downloadable seminar is "to prepare people for C++ and Java by teaching them the necessary subset of C" Good stuff.

More details are available on Eckel's blog, at <http://www.artima.com/weblogs/viewpost.jsp?thread=154029> .

Standard Bank IT challenge

Hooray for the encouragement of developing new IT intellectuals! Standard Bank is now in their second year of hosting their annual IT challenge, which brings the top students from various universities around the country to duke it out with their knowledge of computer programming.

This year, competitors were given six questions that they had to answer in teams of four, using the Java programming language. As with last year, UCT emerged as the winners of the competition, successfully answering five out of six questions and bagging themselves each a laptop, while the runners-up got their own PDAs. With the amount of participating teams almost doubling from the last competition, this challenge looks set to grow well in the years to come.

Now, if we could only get a few of these champs to develop some games ...



New Book on Game Making

If you're interested in a bit of game development literature, take a look at *The Game Maker's Apprentice*, a new book written by Jacob Habgood and Game Maker creator Mark Overmars.

Available in June this year, the book promises to teach people of all ages the art of game creation, helped along by a bundled CD containing a copy of Game Maker and a few example games. More details on this publication can be found at:

<http://book.gamemaker.nl/>.



(above) The GDC 2006 held in San Francisco.

GDC 2006

The annual Game Developers Conference (GDC) held this March in San Francisco has reported vast success with both attendance and lecture lineup. This high prestige event covers just about all aspects of game development in today's world, with lectures taking place over several days and a variety of big names and organisations getting involved. Plans for GDC 07 are already underway, and for those who didn't make it to this year's event, plenty of resources are available on the event's website (<http://www.gdconf.com/>) and Gamasutra (<http://www.gamasutra.com/gdc2006/>)

DirectX Game Development Workshops

Looks like IT Intellect will be playing host to some DirectX Game Development Hotlabs! D# (said "D-Sharp" is a collaborative effort between IT Intellect and the SA Developer.NET authority on DirectX Development, Pieter Germishys (<http://www.pieterg.com>). There will be Game Development Hotlabs held once a month at IT Intellect in Musgrave Centre, Durban, which focus on 3 levels of DirectX Game Development, gradually progressing from beginner to advanced. These Hotlabs are free and open to anyone who wishes to find out more about Game Development using C# on the DirectX SDK. To book your seats or for more detailed information, email:

bernard@itintellect.com. Additional info is also available in this month's ITI article.

(above) i.t. intellect promoting game dev in S.A



Dev.Mag doing the rounds

The first tentative issue of our wonderful little publication had barely hit the proverbial shelf, it seemed, before we got busy and went places with it. Here's just a small smattering of what we've been getting up to in the past month or so:

Dev Mag at Scifest

On the 26th and 27th March, Dev.Mag was able to get their very own stand at Sasol's annual Scifest in the Western Cape. This is a high-prestige event that takes place over the course of about a week, and attracts scientists from just about all walks of life (or scientific life, at least) to come and either lecture or exchange knowledge on current scientific trends. Scifest also boasts international contacts and sponsors, so of course Dev.Mag had to get a piece of the pie. A stand was kindly organised for us at the last minute and, although humble, the amount of new subscribers we gained from the event puts us in good stead for a more prepared effort next year. Prepare yourselves for takeover, foolish mortals!

Dev Mag at I.T Intellect

We've had the incredible fortune of reeling a representative of IT Intellect into the Dev.Mag staff, and have taken full advantage of this by draining him of as much life and energy as possible to feed into our own humble mag. IT Intellect (the company behind the promotion of a new game-dev IT degree) is releasing a promotional CD to show people just how easy and cool game development can be. Featuring on this CD will be a copy of Dev.Mag itself, as well as interviews and games from members of our community. Definitely something that everybody here is looking forward to.

Sasol SciFest[®] 2006



(above) "Hey, what are you doing with that camera?"
Our loveable Dep-Ed, desperately trying to adapt to the bright lights and people at Scifest.



Nice colours, cool website and they've got a game degree too. IT Intellect is definitely a winner in our books.



(above) NAG is full of disciplined and hardworking journalists. It's also the home of Bowser, who's pictured here on the Dep-Ed's head. Also shown is the one and only dislekcia!

Dev.Mag at Nag

On the 29th of March, our rather careworn deputy editor flew over to Johannesburg and had the great fortune of meeting a few of the equally careworn people who were responsible for creating and publishing the well-loved NAG magazine. Although he didn't get a shot at the indoor swimming pool or private tennis courts that the NAG office sported, it was a great afternoon spent chatting with the people behind the face of this popular mag. Dev.Mag was able to gain some professional feedback from some of the staff, which proved to be invaluable. Of equal importance was the stunning picture taken of Bowser – the (unofficial) NAG mascot!

Dev.Mag and PressX

You can't keep a good dog down ... PressX is another local online publication which has been running for about half a year now, and as soon as they got wind of the new mag in town, they launched themselves at us with questions regarding what flavour of jellybeans we enjoyed eating. It wasn't too long before the subject of an advertising agreement was reached, and with hardly any paperwork at all, we are now officially

sister publications. Three cheers for the alliance! PressX deals with all things computery and gamey, making them a perfect complement for our game-dev-oriented e-zine ... and they're free, too! Expect to hear a lot from them in the issues to come.



Coming to an agreement with PressX was pretty easy -- blue is our favourite colour as well.

So ... watch out, world! We plan on pushing Dev.Mag at just about any place that's not going to throw us out, and we'll be pulling it off with style. Keep an eye out in your neighbourhood --we might be reaching you sooner than you expect! Nandrew



SIN

Into a pot add a few million demons, a billion more zombies, hell and a man trying to reclaim his soul from Lucifer himself. What do you get? SiN...

SiN is about a man named Balthazar, who is killed in a car accident. Lucifer then takes Balthazar's soul in exchange for a second chance at life. Balthazar can't live a life without emotions, feelings or a soul. He is an abomination of gods law.... a sin. He must search for Lucifer hoping to reclaim his soul. Aided by an old man named Victor, Balthazar literally goes through all levels of hell in his quest to retrieve his soul and put an end to Lucifer and his infinite army of demons and zombies.



"Can you spot the cross-hair? You may see it now at the top left, but when you're busy killing zombies, you don't have time to search"

Modes of play

SiN comes with 3 modes of play: Survival, Teleportation and the main quest. The main

quest is basically what it is. You take Balthazar through hell in search of his soul and Lucifer. You encounter a virtually endless line of zombies and demons that are hell-bent [*love the pun - Nandrew*] on making you a permanent resident in hell.

The main quest takes you on a journey through many levels that leaves your trigger finger twitching. The game's 5 weapons may seem few, but when dozens of demons and zombies are rushing at you, you don't have time to complain about the number of weapons you have. Anything that shoots will have to do, especially later in the game when the enemies start shooting back. The quest is made even more difficult because there are no bonuses or health pick ups that may have really made things easier.

Survival mode is probably the most difficult mode. You're given all the weapons available and you have to try and survive as long as you can. Within the first 10 seconds, you'll find yourself overrun by the massive flood of unfriendly

zombies. Balthazar can, however, create a shield around himself, provided you've collected enough souls to create the shield.

Teleportation is basically the same as survival, the only difference is that when things get too tough on one side of the battlefield, you can teleport yourself to another part of the map, but don't think it's any easier. You can still get swamped in a flood of unrelenting demons and zombies.

One of SiN's biggest problems is the cross-hair. You often lose sight of it in the heat of battle because the colour of the cross-hair is similar to the colour of some of the demons, causing you to lose a lot of life because you don't know where you are aiming.



"Zombies spawn anywhere on the field and are merciless in their quest to destroy you".

REVIEW

continued

The games visuals are pretty basic, but just like the few weapons, you don't have time to complain about visuals when you're fighting.



"Welcome to hell, I'll be your boss in this life." Even before you can enter hell, you must face your first major foe.

The sound is pretty bland, if it exists at all. You can hear music, but you don't hear your weapons when you fire them. The control scheme is straight forward. SiN adapts the movement controls of a First Person Shooter (WASD) while the mouse is used for aiming and firing.

Expect to have a sore finger from the all the clicking (I estimated that at the games peak, my clicking rate averaged 6 mouse clicks per second).

A good thing about SiN's

*"You encounter a
virtually endless line
of zombies and demons
that are hell-bent on
making you a
permanent resident in
hell"*

zombies and demons is that they spawn randomly, which means the games life span is greatly increased because you won't always be shooting the same things in the same place. But it's also a bad thing because it means zombies can spawn right under you and you won't be able to do anything about them before they get you.



FACT BOX

Developer: Zodiac Games

Released: January 2006

Current Link:

<http://h1.ripway.com/GoNz0/375393-SiN.rar>

Final Comments

SiN may not have great graphics or sound, but its difficulty compensates for the lack thereof. Playing SiN reminded me of the torture tool called Ninja Gaiden and the endless nights of trying to finish it.

In the end, SiN proved itself to be a really enjoyable game (even though I won't be playing it again any time soon).

Hopefully, there'll be an expansion or sequel with more weapons, modes (a co-op/multiplayer mode would have really been great) and of course better graphics and sound. **Knet**

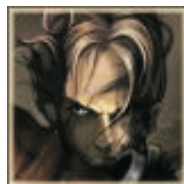


REVIEW



National Defense 2

Interviewing Ch1ppit, the creator of National Defense



SPOTLIGHT

How did you come up with the idea for the original National Defense, and how long did it take to create?

National Defense was created for NAG's comp 03, and the theme that time around was to remake an old classic game. I was, and still am, very fond of using the mouse for control, since it gives a lot more depth than the keyboard. That was when I thought of Missile Command, a game I played a long time ago at an arcade, and realised that it would work perfectly with my plans. While I changed quite a few things, like making each turret unique and changing enemies from missiles to aircraft, ND was rather similar to Missile Command. I made a lot of changes to it during its production, even after comp 03, and I believe the total time I worked on it would be about 3 months, from concept to completion.

Did you encounter any problems during production, were any gameplay changes made in the process?

I had a problem with the development of the original National Defense, which had a very persistent bug that had me on the verge of admitting defeat before I finally fixed it. That was pretty much the only problem I had throughout the production of the series. I had a lot more game-programming experience by the time I made ND2, though, gained from creating the original, and from the loads of programming techniques that I experimented with during the production of Commander. I believe it was thanks to my increased experience that the production of ND2 went off without any major problems. No changes were made to the game as a result of programming issues. The only

continued

gameplay changes made from the original to ND2 were to make the game fit the theme of 05, and to make it better than the original.

What's changed with ND2?

It's a much faster paced game that the original. You now face many more enemies at once and have a larger arsenal to use. I made enemies more varied than they were originally, and they behave differently from each other, removing some of the monotony of the original. From a technical point of view, ND2 is much more advanced than the original. It is capable of handling many planes at once, and it runs far smoother too. This is mostly due to my using OpenGL for graphics which gave me many more options to use to make the game look and run far better than the original ever did.

How do you feel overall about the production of this game? What have you learned while making it?

I'm rather pleased with how it turned out. It contains a lot of extra features that none of my other games have, with more user-friendly options to make the game as fun as possible to play. I added a lot of polish to the game, making it as complete and functional as possible, yet still streamlined and simple to play.



In the words of FPS Doug: "BOOM! HEADSHOT!" Some of the effects in this game are brilliantly carried out.

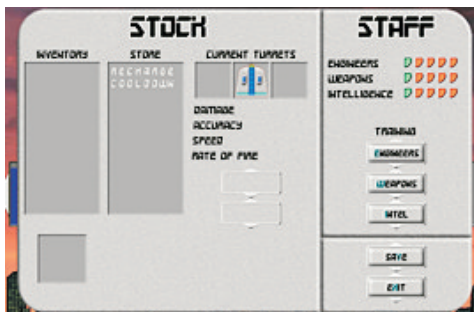


ND2 pulls off its use of OpenGL and particle systems with great success, showing that a mixture of good gameplay and technical aplomb really does pay off.

Throughout the production, I realised the value of community feedback on the quality of the game. Feedback is very important because some of the best ideas used in the game came from the community, not from me.

What message do you have for the game development community of today?

Add little touches to your games, since the smallest details are the important ones. They influence the impression the game leaves on the the player. And, most importantly, listen to the people who play your games to see what they like and what they don't. Then make sure that the next time they play it, they'll find fewer things to hate. **DM**



Great care and detail has gone into the making of the game. When purchasing, the numerous options available introduce a tactical element alongside the arcade action.

Shotbreak Games is dedicated to bring you innovative games!

Shotbreak Games

Interviewing Tr00jg the creator of Shotbreak Games



How long have you been developing games?

Almost a year and a half.

What first inspired you to get into game development?

I have the passion to create things and I love games. Mix those together and you have game development! I just never did it before, because all the code was baffling...

Tell us a little bit about Shotbreak Games.

Shotbreak Games is an indie development company (well, not really company). After developing, I decided to get myself noticed. So I established Shotbreak Games and its website. From there on, it has grown ever since and shows no sign of stopping!

What are your more popular projects? Could you tell us a little about them?

I would say Roach Toaster is my most popular (most known) project. It is a turn-based strategy game where you must thwart roach infestations with an elite group of militants!

What is your most recent work? Any explanations regarding it?

Well, I am working on 2 things now. CONSUME

and Nanos. Both are experimental prototypes, which means they aren't totally games, but rather short, experimental jabs at something. My recent completed (not totally) work is a remake of the legendary Excite Bike.

Do you have any particular plans or hopes for the future?

Well, at the moment, I wish to get noticed by the global indie community not via the so called "casual" games, but rather by something experimental and new.

Could you provide us with any links to your website and/or games produced?

You can get everything at <http://www.shotbreakgames.za.net> It contains my games, design articles and the usual blog hooah.

What's your message to game developers out there?

All your base are belong to us ... okay, not really. Start small. No matter what you do, you should always start small. Ask for help! Once you've gotten the gist, then you should consider taking on something more ambitious, like an RPG (which we all want to do!). **DM**



YOUR FREE E-MAGAZINE

WWW.PRESSX.ZA.NET





ON THE BACK OF A NAPKIN...part 1

Wondering why your PC doesn't play games so well anymore? Trying to decode the latest marketing-speak on the back of a game box? This series is designed to give you the knowledge you need to understand the tweaks and trade-offs you can make to improve your gaming experience. We'll go through the concepts behind the gibberish on the options menu, without going into all that math!

The Basics

The first thing anyone needs to understand about 3D games is that it's all fake. Everything you see on your screen is some kind of trick. Even the newest ultra-realistic computer generated effects in your favorite movie are based on tricks. Very clever little tricks, but tricks all the same. An ideal graphics system would simulate light interacting with the atoms that make up objects and then hitting the back of your eye... Good luck writing a perfect 3D world, unless you're Einstein or Stephen Hawking.

But even if our favourite quantum geniuses wrote a 3D engine, it'd be incredibly slow... Because that's what we all care about in the end, the speed our games run at. We need the image on the screen to change between 30 and 60 times a second to fool our brains into seeing smooth animation. (Unless you're a pro-gamer, gifted with super-human eyes and reflexes) Most games are perfectly playable if they achieve around 35 *frames per second* (or *FPS* for our twitchy, light-sensitive friends) -- dip too far below 30 FPS and we lose our suspension of disbelief and the game turns frustrating.

The 3D Engine

We've all heard of the term "3D engine" that's the arcane lump of code responsible for giving us our graphics fast enough. In the days before 3D accelerator cards, unwashed programmers tamed strange beasts like the S-buffer, active

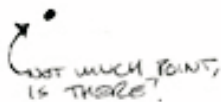
edge lists and BSP trees to perform their mathematical magic on underpowered CPUs. Nowadays, we simply throw data at DirectX or OpenGL (both are APIs, Application Program Interfaces, that sit between programmers and your 3D hardware) which do most of the magic on a graphics card to produce modern visuals. All a 3D engine has to do now is figure out which objects need to be displayed when and then tell your 3D hardware how it should display them, there's a little bit of setup involved, but we'll get to that later...

So what gets displayed?

Remember that everything in 3D is some kind of trick. So, to display something that looks like it has volume and shape on a 2D screen, we need an approximation that'll work. Enter the humble triangle. A triangle is the least amount of information you need to define a flat surface: three corners, which we call vertices (singular = vertex), define a plane in space; one corner is simply a point, two define a line and four don't always lie on the same plane; five is right out! [been watching Monty Python again, dis? - Nandrew] To a 3D card, the order in which the corners are sent matters (it's used to calculate which direction the surface is facing) but logically the corners can be specified in any order and you still get a triangle.

So, we now have a way to define a flat surface that can have any orientation, great. If we put

ONE POINT:



TWO POINTS:



THREE POINTS:



FOUR POINTS:



enough of these surfaces together, we can approximate pretty much any shape we can think of. Stick a whole bunch of vertices describing triangles together, lob them at your 3D card and *POOF* you have a mesh. Meshes display everything in a 3D game, some are generated on the fly, like the terrain in FarCry, others are animated (which means their individual vertices move around relative to each other) and still others are responsible for displaying effects such as shadows.

So graphics relies on the idea of a vertex, which we'll cover next month. *Dislekcia*

So what have we learnt? It's all a lie.

35 or more FPS = suspension of disbelief.

Modern 3D engines rely on clever APIs (DirectX or OpenGL) and hardware.

Anyone who writes software 3D engines is crazy or Carmack.

Graphics = Mesh data = Triangles = Vertices.



The creation of a monster from triangles, texturing and lighting effects.



Immersion

By reading this awesome magazine you should already be thinking about or already be making games. When designing a game your main priority ought to be to make the game as fun as possible -- after all that's what gaming is all about, having fun. But how do you achieve this? Using fancy, shiny graphics or the latest fads and gimmicks like no loading times? Well no, the way is to immerse the player in another world.

Think about the jolt in your heart as you are spotted by a guard when sneaking around or the intense concentration you feel as you are trying to shake off 10 cop cars AND a helicopter. These emotions are what gaming is about, taking the player and immersing them so they are not thinking about what's going on around them in the real world or that they need to tap the square button next -- they instinctively feel what to do in the game world they are immersed in. But then how do you achieve this? All the games I talked about might be Triple-A titles, yet immersion is easy to achieve using anything like Game Maker. Look below. It's not too hard, you just need to be a bit creative.

Storyline

Creating a world which your player cares about and becomes emotionally attached to can really draw them into thinking only about how to save this "world" they are playing in. Another way is to create very believable characters with strong personalities who steadily change and mature as the game goes on. If you can get the player to become emotionally attached to the character and feel a connection with them, then you've achieved in immersing the player in another world.

"It's not too hard, you just need to be a bit creative."



The cast of Final Fantasy providing gamers with epic storylines.

Gameplay

Producing gameplay which makes the player have to concentrate so that they are only thinking about what they need to do next in the game is effective. However, note that to get them to concentrate and get all their attention doesn't necessarily mean you need to make the game hard. By giving the player options like choosing different paths to take in a short space of time or making the player need to work out strategies and think ahead will place their mind in the game

world. The Artificial Intelligence in your game doesn't have to be super complex, keep it simple with depth and make sure your little monsters don't do something dumb like walking into a wall. There's not a more painful reminder that it's not real than when your opponent does something plain stupid. You can also create situations where everything goes quiet, so the player feels a false sense of security and then BAM! something jumps out and the player will be so surprised that they are sucked back into the game world and will only be thinking, "Aah! I need to take this thing out!"

Graphics

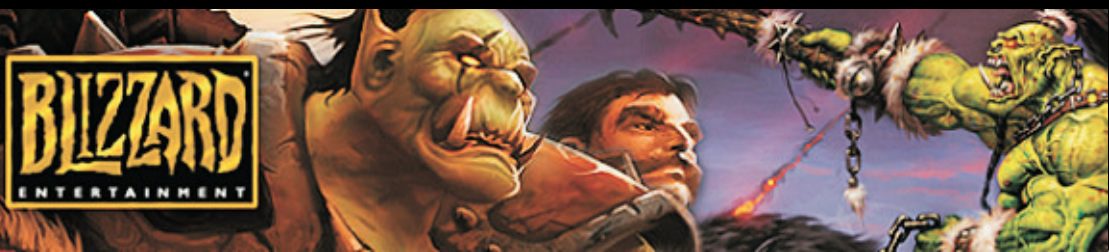
While the number of polygons or how detailed your character is can slightly assist in immersing the player, there are many easier ways to make them forget they are just sitting still on a couch instead of another place. In Game Maker you could use a technique seen in the game King Kong where there is no HUD (the on screen display depicting life, ammo etc.) so the player forgets they are playing a game and instead just focuses on being in the game world.

With this technique the player has to rely on their own personal intuition to tell how much energy they have left just like real life.

These are just a few simple techniques, you can take those points much further. Just remember that when designing a game every element you add should only make the game more enjoyable and assist in immersing the player in a different world. *Insomniac*



(above) The mighty Oblivion and FarCry, two benchmarks in the world of computer game graphics.



What makes Blizzard so good?

With most games released over the last couple of years, when you have installed the game after buying it, you can expect to find a patch on the internet for that game already. A good example is the nightmare that is Battle Field 2. With every patch matching 120 MB of downloads, and an extra load of bugs, it is no wonder I don't play this game anymore

Okay then why is it I still play Diablo 2 today since its release in 1996? Why can I not find one person who plays games that has not played at least one of Blizzard's titles, and loves it? What makes them so special?

It's quite simple, actually. Two things make Blizzard games timeless. The first is polish. Blizzard's games shine with workmanship and detail.



Diablo 2

There is no patch needed, or flaws in an initial releases. They work smoothly on the systems specified on the box. There is no clutter, no unnecessary content in game, and everything is there for a reason. They run fast, they demand little bandwidth (for multiplayer) and are simply brilliantly put together. But you already knew that, anyone who has played

any of their games will agree. But what does this mean to the Indie Game Developer in us? This should give us a few helping guidelines. "Less is More" Only add if it adds to the game play. At the same time, think *detail*. What makes your game shine?

The second thing about blizzard games is replayability. Diablo 2 has random maps, random monsters and, like most RPG's, random loot. It also has 3 difficulty levels appropriate for the levels gained when playing the previous acts. It has a great story, but the game doesn't rely on it. When you reach the level cap, it has loads of items left for you to find to make the ultimate character.



World of Warcraft

This is exactly why World of Warcraft has sold over 5 million

copies worldwide. With the same mind set as Diablo 2, just on a much larger scale, it is a huge success.



World of Warcraft

So remember, make your game fun to play, but always make it fun to play the second and third and fourth time as well. Keep it random. Make it so that you don't get tired of the game while play testing it.

Not everyone has hundreds of employees and million dollar budgets, but then again companies that do don't get it right either. The most popular online FPS game ever was made by a modding team, and they created something new and replayable.

It was called Counter-strike, and I bet they didn't think of a job at Valve when they made the mod -- they were thinking of how much fun they were going to have playing this game. **Himmler**

fRAMeWorKs

What to know about frameworks

A framework is a program design along with the methods that are used to make it work. Frameworks can be structured to support rapid game development methods. Game Maker is designed as a framework, it supports most of the key aspects of a good framework. If the Game Maker framework suits you then use it. If for some reason the Game Maker framework does not suit your development style you probably need to develop your own framework in the development language of your choice.

By using a framework creating a new games becomes a whole lot easier. The framework gives you the advantage of having a large part of the game already developed. By following the structure of the framework and by following the methods defined for making use of the framework it makes it quicker and easier to add functionality into the game without worrying about redeveloping the basic structure each time. Game Maker has a very well defined structure of using images (sprites) along with objects in scenes (rooms).



Game Maker, the popular game development software

Once you as a game developer find the game structure that suits your development style its pretty easy to strip out all the game specific aspects of the game and be left with a basic shell that can act as a framework. By understanding your own framework and structuring it around

the way you normally use to develop games your new games will be so much easier to develop. Game Maker has already developed the structure required to start developing. In many ways the Game Maker tutorials define the methods that need to be followed when using the framework.

Typically aspects of game development that should be included in a framework are: the game loop, the management of game states, the loading and management of images and sounds, and where possible the basic structure of a sprite engine or something similar. By having common methods to manage these aspects of the game within your framework it means that you can reuse the code instead of re-developing it each time. Game Maker defines these aspects of game development in its own way.



Game Maker allows you the option of customizing your loading screens

A framework should enable you to build your entire gaming world and all other aspects such as output options. The framework should be able to create advanced objects and well as simple objects like a loading screen.

fRAMeWorks

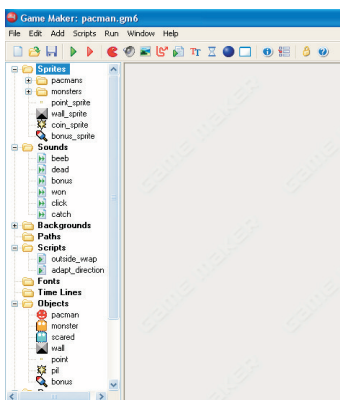
continued

Whenever you start a new game development its time to pull out the framework and fill in the blanks. Typically this would be to first redefine the game states you need and the triggers that will move the game between the various states. This gives you a game structure that delivers the outline of the game you want. By using the triggers to help structure the relevant portions of the game you can concentrate on only the parts of the game you want to. Later the other game states can be filled in allowing a simpler and more logical development path. Game Maker does this in a very structured way. When opening a new game maker project you immediately have the game loop defined as well as support for image management, object movement and through the creation of multiple room even state management.

It is also important to understand that a framework can be used for more than just structuring the way you develop a game. A framework can be developed to support multi developer games by structuring the framework to support individuals working on various smaller sections of the code. A framework like this is based on well defined interface between the various components of the system.

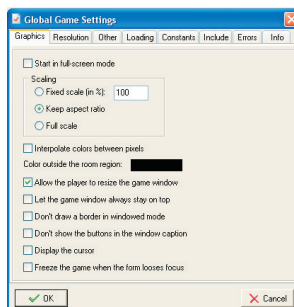
Game Maker has the option to switch between advance and beginner modes.

With its friendly interface and easy to learn tools Game Maker has quickly become one the the most popular game development tools for the beginner.

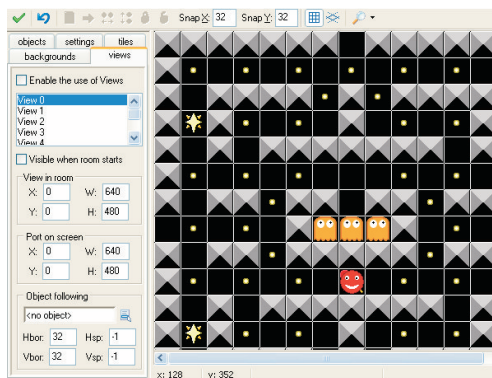


The Game Maker framework is very good in supporting a sprite based game involving numerous interactions between the objects, however when it comes to developing a MMOG the Game Maker does not deliver the sort of framework that you would need.

Frameworks save time and effort, they make rapid game development a reality. In some cases a good framework can simplify game development to just a few lines of code. Go ahead, make your framework and let it work for you, or use the predefined framework that Game Maker gives you. **Cairnswm**



(above) Game Maker's Global setting screen. Providing options for quick and easy game development.



Game Maker comes with a built in Sprite Editor

MOBILE GAMES DEVELOPMENT

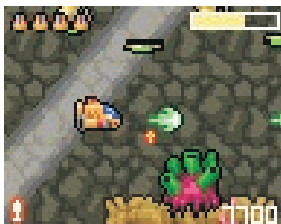


Games on the go...

It's great fun creating games, and being able to show them to your family and friends on your PC, but how cool would it be to be able to show them off to anyone, anywhere? Well, it's possible and you won't even need to lug your trusty 'rig' around with you to parties!

How is this possible? Well, with your cellphone of course. Modern cellphones are actually extremely powerful little computers, and are capable of running some great games, similar to those you'd create with GameMaker. Mobile game development has become big business over the last few years.

Of course we're not interested in the 2 billion cellphone users worldwide, we just want to make games for our own enjoyment and for that of our buddies and family.



While doing this is not quite as straightforward as developing with GameMaker (yet) and does require quite a bit of programming, if you can use scripting in GM, you can learn Java (the language you'll most likely be using).

If you know how to use Flash, you could choose that as a tool as well. Many new phones are Flash enabled, and it's support is growing, however unlike Java, the tools are not free.



There are some unique challenges when creating mobile games, for example storage space is limited and you have to be smart about how you use images and sounds. The many different

screen sizes you have to cater for may also affect your game



if it have lots of graphics, you might need different versions to cater for different screens. Another big difference is input- you obviously don't have a full keyboard and mouse, and you have to design around this.

These challenges can also add to the fun though!

That's it for now, but if this sounds like your thing, check out some of the links below, as well as googling some of the following: J2ME, Flash Lite, Brew. Happy hunting! **Flint**

www.j2me.org
www.jbenchmark.com
www.java.sun.com
www.smallfrymobile.com



The mobile game 'Abyss' displayed at various resolutions (96x65, 128x128 and 176x208).



IT Intellect has had a long-standing relationship with the crew at SA Developer.NET and we're very stoked, to say the least, that two of their brightest and most talented Lead Developers have decided to choose ITI as a base for their wonderful vision of educating and supporting a hobbyist community of Game Developers. These include the very tight community at Game.Dev, headed by the freelance aficionado and Telecommunications activist Danny "dislekcia" Day, as well as the highly revered SAgamedev.co.za community.

The two developers I speak of are one Andre "Fengol" Odendaal and Pieter "Armaddon" Germishuys. Both are established authorities in DirectX Game Development and have very kindly offered their knowledge and expertise in their field to the public at no charge!

IT Intellect is looking to promoting and supporting the Game Development sector in South Africa at all levels, as anything we can do to help grow this small community will be beneficial to the industry in the long run.

IT Intellect are very excited as our first intake for Game Development students of the UAT curriculum will be joining us in June 2006 at our Bryanston Campus. They will be overseen by UAT Graduate; class of 2004 Scott Snider who will be the first of many UAT lecturers to be relocated to our beautiful country to assist in the education and skill development of aspiring local game developers. This coupled with the Hotlabs will make for an exciting year ahead, one filled with many fresh ideas and new experiences.

Thanks to Game.Dev and the team at SA Developer.NET for their active involvement. We are looking forward to the future! *Burnabis*

Here is a list of the D# workshops that iTi are holding along with SA Developer.NET. All workshops are held from 9:00 - 12:00 on the morning of the given date at IT Intellect in Musgrave Centre, Durban.

Email bernard@itintellect.com for further details.

SETTING UP A WINDOW AND DIRECT3D

DATE: 29th of April 2006

Probably one of the most interesting parts in the DirectX set of APIs is Direct3D. Setting up the window and the device is our main goal for the beginning of this course.

WORKING TOWARDS TETRIS

DATE: 27th of May 2006

Tetris, probably one of the most well known games on earth. We are going to attempt to recreate this awesome game and learn some of the DirectX components that can help us on the way.

WORKING TOWARDS BREAKOUT

DATE: 24th of June 2006

Dropping blocks was fun but we need to get some real interaction going. Breakout was a game that allowed the player to control a paddle and to prevent the ball from dropping. In addition to not dropping the ball, the player had to hit blocks the resulted in bonuses dropping from the blocks that were hit.

WORKING TOWARDS PACMAN

DATE: 29th of July 2006

Pacman introduced some very interesting concepts such as Artificial Intelligence and maps. Although Pacman only has one map, we still need to do boundary checks.

WORKING TOWARDS SPACE INVADERS

DATE: 26th of August 2006

Space Invaders added an exciting twist. The player has a few barriers and has to eliminate all the enemies before they reach the bottom and take out the player. The enemies fly/march down a row and keep on speed up as they progress.