

MARKUP

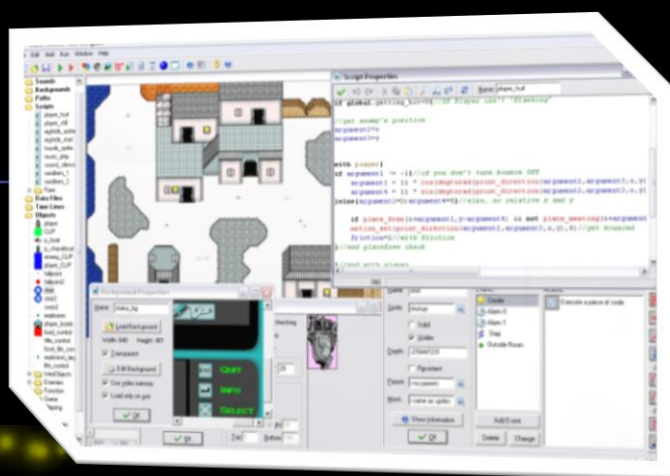
MAGAZINE

FEELING THE BEAT OF THE NEW YEAR

THE GAME MAKER RACE
IS OVER! RESULTS INSIDE

+ THE MAKING OF ARK 22

THE EFFECTS
CREATOR



EXACT GAUSSIAN DISTRIBUTION

+ AND A REVIEW OF THE GAME MAKER RACE WINNER

2008 SPECIAL
EDITION

Welcome!

Table of Contents

Editorials

Editor's Desk	3
Testing: Vital to any game.....	13

Development Entries

The Making of Ark 22	4
Development Journal: God of Rock	16

Monthly Specials

Script of the Month	12
The Game Maker Race Results	14
MarDar	17

Reviews

Tools: Effects Creator	10
Sim City Societies.....	11
The Game Maker Race Winner	15

Why so little?

This New Year issue is a mini-special intended only to announce the Game Maker Race results. Don't expect to see a lot, but we've put together some of the exciting articles and exclusives like 'The Making of Ark 22' to keep this issue interesting, regardless of its length.

Contributors

Eyas Sharaiha	Sr. Editor
Robin Monks	Editor
Andris Belinskis	Editor
Philip Gamble	Writer
Sreh	Writer
Leif Greenman	Writer
Veeti Paananen	Writer
Darragh Tobin	Writer
Robert Colton	Writer
Suhaib Al-Dari	Cover Designer

Special Thanks

Again, a special thanks to guest judges of [The Game Maker Race](#) for their work. While no commentaries of theirs have been included in this issue, their efforts must be recognized:

KCLC	Rhys Andrews
Snabela	TGMG

About MarkUp

MarkUp is a project of [GMking.org](#), a network dedicated to support and serve game developers of all IDEs. For more information on [GMking.org](#) and all the other projects it has, visit our main site!

Contribution

For more information about contributing to MarkUp Magazine, visit our forums [here](#).

Welcome!

Social Game Development

So lately, the world has been shifting towards online gaming rather than local gaming. Whether it with MMORPGs on the PC, like World of Warcraft or Guild Wars, or if it is Live Gaming on Consoles, gaming in general is shifting to become more of a social experience.

Communicating with the world and others is really becoming a need for many gamers.

Such type of communication could be achieved in multiple ways. While 'online games' as a concept is the most social type of games, it is not the only way to achieve a social gameplay experience.

One of the simplest ways to achieve some sort of a social gameplay experience would be online highscores, where the player stops trying to break his/her own highscore, but rather to beat others as well.

To build on social highscores would be chat functionality in games. So that a person could discuss game strategies, etc. with others who play the game, such as those who broke his highscore, etc.

Multiplayer modes in games could be a "bridge" between the classical single-play format and MMORPG games. Providing lobbies and rooms for players to challenge each other is a good thing, too.

Just remember, whenever you're making a game, to try and provide functionality that would make it as social as possible. I'm talking about ditching your game idea in favor of an MMORPG that might never be complete –that's unrealistic, but instead to incorporate social features into your existing game and vision.

Eyas Sharaiha■

Issue 12 Coming in February!

In February 19, 2008, the next issue of MarkUp: MarkUp Issue 12 – the 1 year anniversary issue, will be released!

We will be taking January and the beginning of February off, to get rested, AND to prepare an AMAZING 1-year anniversary issue for you all to read!

Contribute to the 1 Year Anniversary Special!

If you want your work to be present in the 1-year anniversary special: Issue 12 of MarkUp Magazine, then you can contact us at the following address: staff@gmking.org and send in your submission or queries.

Writing at MarkUp: Goodies

You get 1/4 page ad for every 400 words written [up to 1 page ads].

After writing 2 more articles for MarkUp, you can request FREE Game Development BOOKS and REVIEW them

The Making of Ark 22

So, a bit of history; I guess this is where traditionally I'd list a few earlier projects and what I learnt from them in order to be able to put Ark together. The trouble is, Ark 22 was my first ever Game Maker project, and most of the mistakes made (and there are some truly awful ones) are still in the game somewhere.

Introduction

The first thing I thought of when I got my hands on this brand new thing called 'Game Maker' back in 2002 was, "I wonder if it's possible to make my favourite game in this?". If you've not already guessed, that was Zelda. For a few weeks I played around with getting a Zelda engine set up – a bit of movement, some sprite animation, a sword attack and so on. Foolishly I assumed the rest of the development of the game would be as smooth and painless, so I decided to bolt on an original story, some of my own graphics, some altered weapons, etc. Simple, right?

Development lifetime

Getting one obvious thing out of the way first; developing Ark 22 took a *long* time. There are a few GM games I know of which have been simmering away for a while longer (one of them should be out in a few months – you'll see), but as far as I know, Ark is still 'up there' as something of an epic – for the right or wrong reasons, perhaps. Development started in 2002, and I made my first attempt at a 'final' release in October 2005. That's about three years for the first build, but it doesn't stop there.

A lot of the game's features to date were added long after this, in response to user feedback or as a reaction to the increasingly technically impressive state of GM6+ games (remember, Ark 22 is GM5.0) – things like the Mortar service, boss replay symbols, updated NPC graphics, updated tiles, etc. As a point of reference, the last update

– although mostly cosmetic fixes such as spelling errors and tile inconsistency – went up last month (November 2007).

What would be most useful here would probably be to try and answer "how" I kept a game project going "for so long", or where the "dedication" came from. This sort of thing gives me *far* too much credit. I don't remember exactly the point where I realized just how much time or effort the rest of the game was going to take, but it was definitely long after I realized I *had* to finish it, as I'd spent too much time on it already to give up. That's not really something you have control of, I don't think.

Thinking back, I probably would have done things roughly the same way if I'd known how much time and effort it would all take, but I certainly didn't the first time round. Some other GMers I've talked to are in the same situation. The idea slowly decays from being the Best Thing Ever to a real chore. There are occasional moments of self-rewarding magic, but otherwise it'd work, pure and simple. This is all pretty obvious I suppose, but the trick is to think of it *before* you're waist-deep in something whose only exit is downwards. It's not bad – you still get it done – but it's just not all cake and balloons either. No holding your breath forever. You have to drown in it eventually.

Implementation

I have a feeling this is going to get a bit ugly, but it seems like fun. I spent a long time thinking about this, when

The Making of Ark 22

being asked "How did you make Ark 22?", what on Earth I would answer... There's nothing I can think of which stands out as particularly shiny or clever in there – nothing which would hold a reader's attention for more than a minute, at any rate.

On the other hand, there are plenty of things I probably shouldn't have done, or didn't do and wish I had. These are probably a lot more entertaining. Seems a bit narcissistic to go on about 'great things I did' like this, so I guess I'll share a bit of what I did wrong. Here goes my reputation, haha...

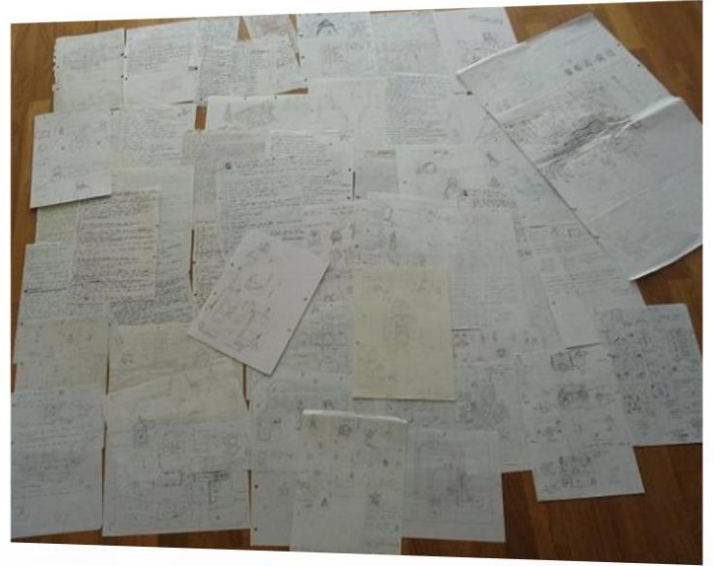
Sreh's top ten horrific mistakes in Ark 22 development;

10. The mountaintop in the clouds

Ever climbed a mountain? If you have, you'll know how much *fun* it is when they're arranged as a series of tiny 'fake' summits. Every time you think you're almost there, you clear the top of the hill and... no. It's the next one. Eventually you decide to stop thinking about the top and *just walk*. It will end when it ends.

Same thing with a game like this; basically what I just waxed on about for a page or so, so I'd rather not bore you all with the details all over again. This is **project underestimation**; failing to realize what sort of demands building a game of this size and scope is going to put on you, the tools you're using, the game engine itself – everything's affected. It's not easy to keep it all trim and efficient – I didn't. As a result Ark 22 is huge, runs a little slowly on some machines, and loses big points in a lot of areas for not displaying the sort of stylish smoothness we're starting to see all over the Game Maker scene. Is this a necessary consequence of building a game this size, or using GM5 to do so? Maybe. Nonetheless there's still a lot which could have been avoided or improved by just a little more **planning** and thought. Don't dive in when all you have is a walking demo. Know in advance what you're going to need, and roughly how you'll achieve it.

Plan your game. Know your limits and those of the tools



you're using. If your game turns out bloated and sluggish – even if it plays well – at least you saw it coming.

9. Paper plates and plastic forks

Similar to the "don't just dive in" line. GM gives you a lot of things for free – a rendering engine, a configuration/state system (rooms), a snap grid, etc. This can be both a little overwhelming and wonderful at first – "all these great things I have, and just the way I need them to make my game!" Proper sweet-shop feeling, at least it was to me. This is about **over-reliance on defaults**. A room resolution of 640x480 is fine, but is it the *best* option? A refresh rate of 30FPS is again perfectly palatable – but what other options are available? I didn't consider any of this, and as such Ark has a lot of artifacts in which *scream* "Game Maker!". A smaller resolution could've allowed me to focus on tighter level design. Something in widescreen might've been more cinematic. Smaller tiles, maybe. These are all things which are worth considering. Even if something's there, it doesn't mean you have to use it. My current game project doesn't rely on automatic draw – it refreshes the screen 'manually'. When it breaks, it breaks badly, but it's worth it.

8. A comfortable tortoise

Anyone who's ported a medium- to large-scale project from GM5.* to GM6.*+ will know there's a lot of things it

The Making of Ark 22

shouts at you for when you do. What is this *font_color* you keep talking about? Scale that up a bit to a project which isn't terribly well-partitioned in the first place and the amount of work changing the underlying system incurs is a horrible thought. This is **complacency**, being so used to something you think you can squeeze it into doing what you want. I was used to GM5 and didn't want the weeks of headache it would have taken to harmonize all Ark's source code with the new-and-shiny GM6. I'm not sure if I would actually make the jump after all if I had a second chance, but there's no denying GM6 is stronger, faster, harder and certainly better than version 5, performance-wise. Something to bear in mind then; if technology changes and you think you're "too far in" with your project to change with it, think again. It may be the best thing you can do. It may be the worst. But at least consider it.

7. Somebody think of the children

I'm not suggesting anyone undertake a project in their school years so massive that they bear children before it's finished, no. This is **inheritance** (note from Eyas: awesome pun!!). If you don't know what that is, take a day or so to look it up, play with it, and *use it*. If you do know, put on a wry and knowing smile. Inheritance saves you an absolutely silly amount of work if you use it properly. It's about identifying common elements and patterns between the objects in your game. For instance, an NPC who sells you some ammo is just a specific type of NPC-who-sells-you-stuff. That NPC is just a specific type of NPC you can walk up to and talk to with a key press. That object is just a specific type of object you can press enter while near and it'll do *something*. And so on.

Now taking guesses as to how many of the 1000-ish objects in Ark 22 form an inheritance tree. Clue: less than 1.

6. Stone, paper, orange

One of the most liberating things about using Game Maker is the simplicity with which you can express your

intention without being constrained by the constructs of 'real' to-the-bones programming languages. It's a fantastic way to focus on the *game* and not the *program* and I love it to pieces. The trouble is, your game *is* a program, so it makes sense to take a little time to think about how it's organized internally. The idea is **maintainability** – the degree to which you can painlessly change or add elements to the game's content or engine *after* you've already built them. The simplest example is text. In a game like Ark 22, there's a lot of text – NPC speech, cut scenes and so on. The obvious approach is to write the text strings straight into the code where they appear. When you want to change one, you have to find the object in which it appears and change it there. This is what I did, everywhere.

What if I'd stuck all my strings in a big switch-statement script, like `get_textstring(id)` or something? Clearly it's all in one place, so I only need to go to that script to change things. But then the code in which it appears is less readable, people are talking in numbers, so I don't really

Winskin Extension

'Winskin' is a Game Maker extension developed by DFortun81 to add skinning capabilities to Game Maker. The extension does not skin the actual game window, but rather allows the creation of virtual 'windows' within the game, with a certain skin.

The extension basically adds capabilities of RMXF window skinning to Game Maker.

To be honest, the whole implementation for the extension seems clean and rather attractive looking. Definitely recommended.

Get it now!

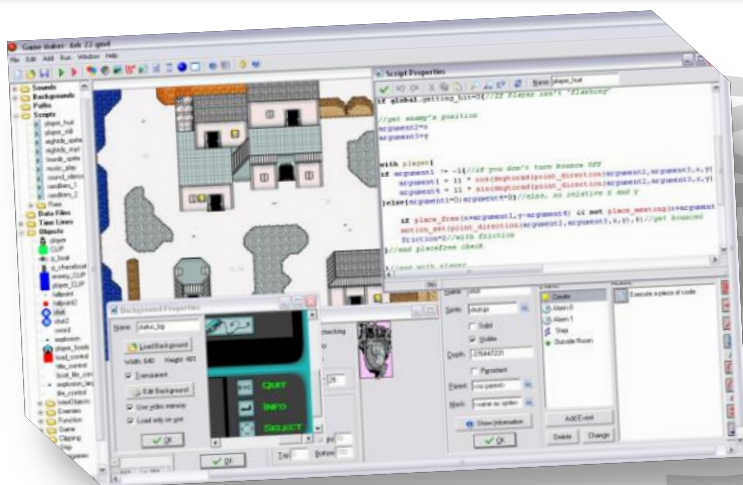
gmc.yoyogames.com/?showtopic=282667

The Making of Ark 22

gain anything immediately. But what if three years later I want to release "L'Arc 22" by translating everything to French? Make your game 'look up' its own data from variables, scripts, etc. wherever you can identify somewhere where these may change.

5. Sand in a jar

One extension to the 'lookup' idea is to override some common elements of the GM system – controls and input is usually a safe bet. For example, if I want players to be able to redefine their controls, storing each input key in a variable and reading from there is the obvious solution. That's fine, but my objects are still all going `keyboard_check(global.key_action1)` or something. What about joysticks and gamepads? What about changing the way a certain action is even performed – two buttons instead of one, for instance? It's not strictly essential, but a **second-layer system** can help leave such decisions until later. In my current project I have an object which populates an array `global.pressing[]`. Each index corresponds to a logical 'action', and the objects all read from this – never the `keyboard_checks` or `joystick_~` scripts directly. In Ark 22 I not only had static `keyboard_checks`



everywhere, but the input keys were hardcoded too. Not clever.

This made writing `obj_force_joystick` a horrifying experience. Don't do it. If you can abstract something away like this, as long as it's not trivial, do it. It may never change and you may have incurred another unnecessary level of lookup. But if it does change, you'll be glad you were ready.

4. Crash mat

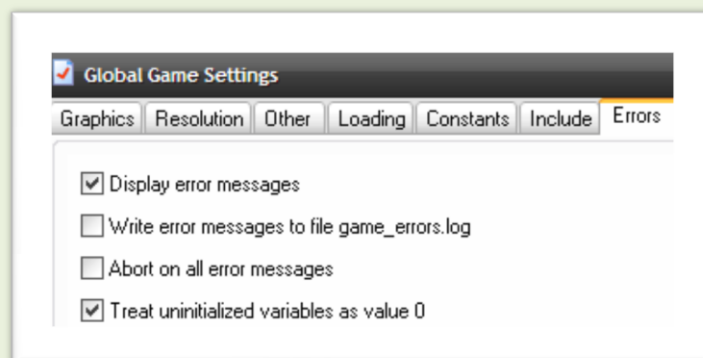
Simple, this one: **make backups**. I only needed to be told

Treating Uninitialized Variables as Value 0

I will be blunt: this is a bad idea. If you are using this option, make sure you know what you are doing. Treating variables that have not even been created yet as 0 creates several problems, first and foremost it makes code hard to debug. If you set 'iVar' to 5 somewhere in your code and later seek to add 4 to 'iVar,' it is quite easy to type something like the following: "iVra += 4;" this leads to the obvious (yet not so obvious in practice) problem that 'iVar' will never be increased, instead, a new variable "iVra" is created (as 0) and 4 is added to it giving it the value 4. Since 'iVra' is considered every bit as proper as 'iVar' no error message will be generated and you'll be hard pressed

to find the problem. Let's get this straight: there are no benefits to using this option unless you are an expert user; for new to advanced programmers this will only cause you pain.

Leif Greenman■



The Making of Ark 22

this once. But I learned it the hard way. Save often too. If you don't, you'll find out why very quickly.

3. Ao cartography

This isn't something that I have any authority on, nor that can be properly detailed in just a few lines, but in retrospect it's well worth thinking about: **design small**. We usually split games into the engine and the content. For something like a puzzle game, the two things may be practically inseparable; the engine knows to increase the difficulty as the game progresses, and that's what drives the content – the playable bit. For an adventure/RPG game, the engine is a very small part – character motion, enemies, interaction, and so on; the real meat of the game comes from designing locations, puzzles, *adding* things everywhere to use the same basic constructs again and again.

This, clearly, is what I did with Ark 22. A big overworld, complex dungeons, trade-quests, NPCs, optional upgrades, all of it. But it doesn't have to be this way. A game can be small, but just as rich. You can keep coming back to the same few places as they change and evolve, made possible by the fact that the designer has packed each pixel with detail and multiplicity. Ark is full of rooms which serve only to link two locations, to fill a square on the map – to be walked across. Things should never be that way.

2. Multi-pass

This is fairly obvious, but it still didn't occur to me. From

the looks of things it's still quite common in general, too; If you've only just started using an unfamiliar tool, and you want to get the hang of how it all works, **make your first game simple**, or, as nearly every Software Engineering textbook will tell you, **plan to throw one away**. Making your first-ever game an adventure/RPG is only a few insanity steps away from making it an MMORPG. It's not impossible by any means, clearly, but a large proportion of your development time will then be spent on retrofitting and maintenance – changing things you didn't realise were inadequate the first time round. Nearly everything I've mentioned here is something I later had to 'bolt on', and that's never as effective as designing it properly from the beginning. It's not my 'fault' as such, I didn't know half these techniques were even possible – some of them weren't – but it's still a factor. Overall this is what hounds Ark's source code – bits and pieces of miscellaneous nonsense from before I knew any better. This is in the same vein as the 'planning' point – if you can't do it yet, don't. Wait, and do it properly later. Or be ready to spend a large slice of your time building the same parts of your game again. I did. It's not much fun.

1. All tied up with string

Imagine this thought: "A lot of the objects I'm creating are only used in one or two places. Things like story-specific NPCs who only appear once, or one-time visual effects. Instead of creating a separate object for each of these, I could create one 'template' object and specify its actions through strings and `execute_string()`."

Write for MarkUp Magazine

[CLICK HERE](#)



The Making of Ark 22

Don't ever, *ever* use this function in an automatically-scheduled action. This means NOT in **Step** or any of its siblings, NOT in **Draw**, NOT in a fast-reset alarm.

I did, and it hurt. I've been able to fix most things post-release, but with this one the best I could manage was to reduce its effects.

The good things

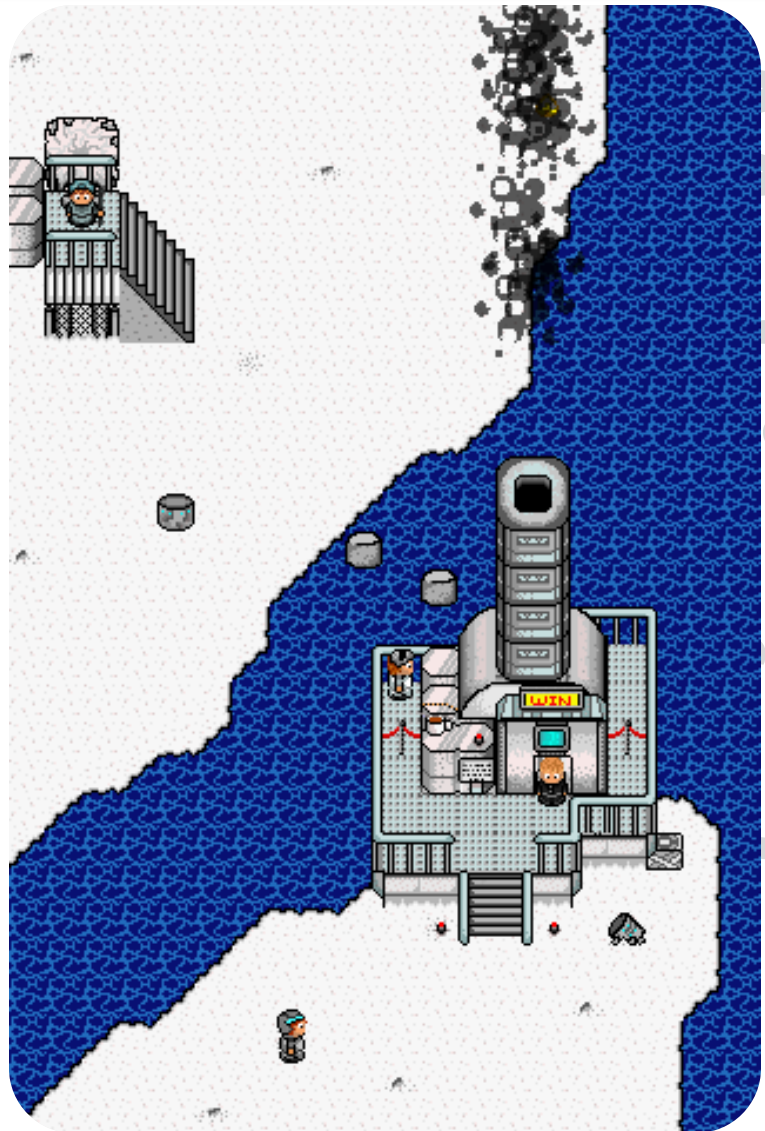
There are, despite all this, things I'm overall pleased with, perhaps even proud of, with Ark. They're not all as 'front-line', nor as numerous, but they matter to me;

1) The writing. I enjoyed giving NPCs their own mannerisms and idiosyncrasies a lot more than I enjoyed building rooms or coding enemy behaviour patterns. There are still things in there which make me smile.

2) Post-project support. Not that this is something I had a vast amount of choice about, but I like how much further the game came after being 'finished'. Joystick support, a proper message system, a re-skinning of most graphics, a better title and config screen... these are all things I really like, and I'm glad I bothered to add them in so late in production; they make more of a difference than I thought they would.

3) The game itself, I think, turned out well for what it is. I get quite a bit of criticism nowadays for things like the graphics and clunky engine, or the fact that the gameplay isn't "original". These things aren't what I was focusing on – the heart of the game, the 'gravity' which pulls everything together; the adventuring aspect is. I wanted to make a solid adventure game which would hold players' attention and provide an incentive to delve further in and finish it. Cosmetic aspects, and some fantastically *stupid* decisions aside, I think I managed, more or less. Close enough.

None of these is something reading a development log or



'words of wisdom' is going to help you do yourself. With any luck you might now have a clearer idea of what **not** to do.

An Easter egg

"I want some proper Coffee."

Sreh■

The Effects Creator

There was once a time where creating particles to gloss up our games was a formidable challenge. Must we always hunt in vain for particle tutorials that suit our needs? Thankfully, the answer is no. With Weenom's 'Effects Creator' our search has finally ended.

The basic idea of this utility is to provide users, beginners in particular, with some sort of means of experimenting with the limits of the particle system like an artist's palette without writing reels of code. And when you're happy with the particles, just copy the code into your game. Simple.

I, evidently, am not the only person who has taken to this most useful tool. The general response from the community has been quite good, regardless of several misguided users who have mistaken it for a poor game. Clearly this tool, when used properly is of great benefit. It has been described enthusiastically as "SOOOO COOL!" and "Very good and extremely useful."

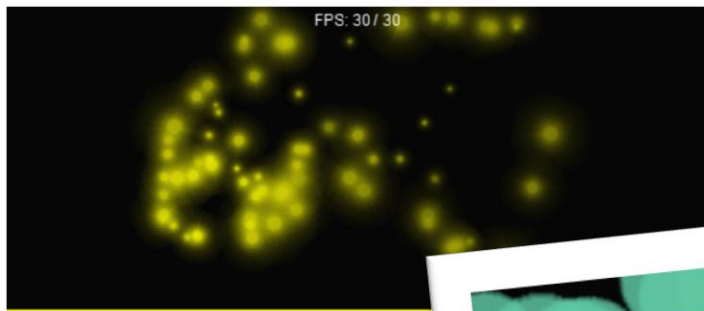
The utility is, to me, clear and well written, as it uses some

rather familiar buttons and colour palettes to some it is rather unpolished and lacks several features but it still provides a fabulous insight to the power of particles for the developer. Especially beginners, as the author of the program himself added:

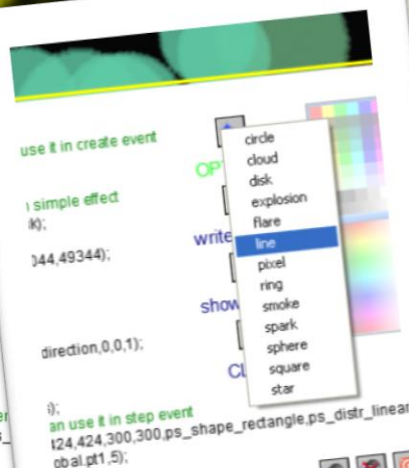
"This utility doesn't have all features, but others ones you can use very easy if you will learn this."

This utility is great for beginners, starting off learning particles, but as for the more advanced users, it lacks some of the pro features, which can admittedly be a bit of a pain. But overall, a great utility, one I'd recommend to anyone.

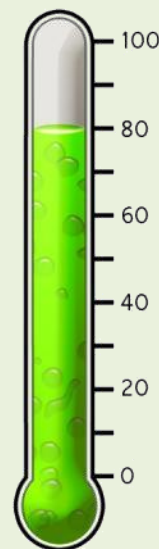
Darragh Tobin ■



```
{
// this 2 lines are used only once, you can use it in create e
global.ps = part_system_create();
global.pt1 = part_type_create();
// this 10 lines are used to create your own simple effect
part_type_shape(global.pt1,pt_shape_flare);
part_type_size(global.pt1,0.20,1,0,0);
part_type_color3(global.pt1,65535,32896,16448);
part_type_alpha3(global.pt1,1,0.50,0);
part_type_speed(global.pt1,5,6,0,0);
part_type_life(global.pt1,50,50);
part_type_direction(global.pt1,0,360,0,0);
part_type_orientation(global.pt1,direction,direction,0,0,1);
part_type_gravity(global.pt1,0,0);
part_type_scale(global.pt1,1,1);
global.em = part_emitter_create(global.ps);
// this 2 lines are used to cast effect, you can use it in step even
part_emitter_region(global.ps,global.em,172,172,166,166,ps_
part_emitter_burst(global.ps,global.em,global.pt1,5);
}
```



Ratings



Graphics: 4/6
Sound: 3/6
Design: 5/6
Helpfulness: 5/6

Developer: Weenom
Version: 1.0
Game Maker: Version 7
File Size: 2.56 MB
Instant Play:

Here

Sim City Societies

Sim City Societies is now the latest installment in the 'Sim City' line, but the first not to be developed by Maxis. The game proves to be largely different than the previous ones, and is suffering from negative press by many the Sim City fans.

What they say

After reading several reviews across the internet I have come to say that most users are quite disappointed. I quote a man on Amazon.com who said "It's Like A Beginners Version" which basically he goes on to say that the game is quite boring and there is not much replay value at all ("it is for little kids"). Others are speculating that giving the project to *Tilted Mill* was the mistake, because they are not as experienced and focus less on the realism and simulation feel that most look for in *Sim City*. So what do you, the reader, think? Only you can comprehend that.

My Opinion

Don't get me wrong I still bought this game mind you, but only as a *Sim City* fan. This is not even a *Sim City* game in my head. Yes for a while it is entertaining, finally 3d, but after awhile placing the cartoony buildings that do nothing over and over again pertains to a small and easy infrastructure and is not fun at all; it takes away everything *Sim City* stands for. I do like the professionalism of *Tilted Mill* they got this game done quickly and on time, however I am not much a fan of their games. I did play *Caesar IV* and loved the strategy in it, but it just feels like they took that building engine and ported it to *Sim_City_Societies* and just changed around the features. It is for one EA's fault for handing this project to such a new and inexperienced team. I believe this project should have been handed to the *Command_&_Conquer* team maybe *Firaxis*, however I don't want *Sim_City_Civilizations*, it should at least be people who know the *Sim_City* game and people that are

within the *EA_Games* payroll, not some team that is just eyeballing the game to quote *PC_Gamer*. In the end this game will fall to \$20.00 US Dollars just like *Caesar IV*, and that is when you should buy it. The goal that was focused on here was an old in the works game called *Sims Ville* which was a cross between *Sim City* and *The Sims*, but was dropped due to the upcoming *Sim City 4* and other Sims games.

Game Play

The game focuses on controlling the posterity and welfare of your society. It does not focus on the building of a law enforcement system, the construction of a massive El-Train project in downtown, or the management and construction of a new high school. Basically six(6) social parties power you city in many different ways, the technique you use will combine into *your* social society not an urban simulation. You must still fight disasters such as fires possibly being caused by global warming in affect from the type of power you chose to run your city on. You can create and design new buildings so cities no longer continuously and painstakingly look like each other. You design and create to shape *your* society.

Interesting Tid Bit - Tilted Mill claims it is not *Sim City* and that it is *theirs* when it is actually just a finished version of the *Sims Ville* Concept

Thanks for reading; I will see you in Issue 12 of Markup Magazine!

Robert Colton ■

Exact Gaussian Distribution

Script of the Month

Powered By GMLscripts.com

Returning random numbers in Game Maker can be usually easily done using the existing random functions in Game Maker. However, it has never been natively possible to create a random number based on the exact Normal (Gaussian) Distribution, which represents the normal way in which values are distributed, such that numbers in the middle have a higher probability than very high or very low numbers.

The Script

'Yourself of' the GMC gives us the awesome script, `exact_gauss()`:

```
/*
** Usage:
**     exact_gauss(mean, deviation)
**
** Arguments:
**     mean        mean value of the distribution
**     deviation    standard deviation of
distribution
**
** Returns:
**     a pseudo-random number with an exact Gaussian
distribution
**
** Notes:
**     See
http://en.wikipedia.org/wiki/Normal\_distribution
**     for more information about Gaussian
distributions
**
** GMLscripts.com
**/
{
    var x1, x2, w;

    do {
        x1 = random(2) - 1;
        x2 = random(2) - 1;
        w = x1*x1 + x2*x2;
    } until (0 < w && w < 1);

    w = sqrt(-2 * ln(w)/w);
    return argument0 + argument1 * x1 * w;
}
```

The script is fairly short, but it provides some awesome capability for Game Maker.

The function allows you to add unlimited realism to the game. Suppose you want the player to be random each time, with a different mass, height, physical abilities, etc. Then rather than making such properties completely random (the game would then lack coherence), you can randomize them according to the Gaussian distribution, in which a certain value (the mean) has a larger chance of occurrence than any value. The further away the value is from the mean, the lower the probability of achieving it becomes.

Such script could be applied to the strength of the player or other characteristics of the player himself, it could also be applied to the strength and behaviour of enemies, etc.

Closing thoughts

The script seems rather short and simple, but it is far from that actually. Allowing the generation of a random number according to the Gaussian distribution is actually pretty valuable in games that want to achieve realism, because they define the 'normal state' of the game, and all properties usually lie close to that state. It still allows rare occasional 'big changes' to occur to the game's behaviour, and that's just what we want if we aim at achieving an exciting game.

Eyas Sharaiha■

Testing, vital to any game

Last month I answered an invitation in the team request section of the Game Maker Community forum asking for people to help test the early beta of a game. The team request section isn't normally a place I frequent since it tends to be full of repetitive baldy worded posts requesting an entire game development team or people looking for help to construct their first game, an MMORPG.



When responding I was directed to the team's privately hosted forum where I was instructed to register and download the latest version of the project, a top down shooter game named [Excalibur War: Norak](#) developed by the Excalibur Gaming team.

From what I read at the GMC I assumed the game was in its late stages since external testers were being sought, however I was surprised to find that the entire download was a little over 1.5 MB. At my first play I quickly discovered that the game was earlier in its development than I imagined as it contained a number of bugs.

Defects included the ability to shoot through walls when you were adjacent to them and the huge bonus that is invincibility. You didn't die regardless of how many times you were shot, the green circle representing your health just seemed to grow and grow the more you got shot. There was a very basic AI commanding "terrorists" to walk directly towards you shooting, they were therefore extremely predictable and easy to defeat. Big problems I am sure you will agree.

Excalibur Gaming had a small but dedicated bunch of people testing the game, and after feedback reached a certain level a new release was made available for the testers to rate with the suggested changes implemented.

Getting feedback in the early stages of game development is extremely important as it opens the door to suggestions of new features and lets you fix bugs in their earliest stages. For this reason I recommend posting an early demo in the Work In Progress section of the Game Maker Community. As well as providing valuable feedback you also get a chance to plug your final release of the game.

All of the problems originally encountered have been fixed and Excalibur War has been released and is available to download from the Excalibur Gaming website.

<http://excaliburgaming.org/games.html>



Philip Gamble■

The Results Are in... THE GAME MAKER RACE

Special

The Game Maker Race was initially revealed at October 15th 2007, and gave the Game Maker Community a chance to enter the competition and possibly win two PC games, a domain name for two years as well as hosting, free USB Keys, and free copies of Commercial Game Maker Games. We've given the community a one month only to create a game and submit it. Since the deadline of submissions, we've been working hard to eliminate the games until we created the list of top 5 entrants. The top 5 games have been – since the beginning of December – displayed on the site, and a public vote was going on. Now, for the first time: we reveal the winner of the first 'Game Maker Race', as it has been **decided by the public**.

We have developed a special voting system for The Game Maker Race, to ensure that public voting would be both an easy process and a secure, fair system.

We've made sure to record all voting transactions, e-mail addresses, IP-addresses, time of voting, and more!

We've reviewed the results and audited the entire thing, checked the votes, one by one, made sure they were valid, and made sure no cheating ever occurred.

We have studied the results thoroughly and believe that we have a valid uncorrupted result.

The Voting

The final stage of the competition: the public vote on the top five competing games got a good amount of activity: 186 votes, of which 61% went to the first place winner of this competition.

I've tried to build up some tension, but it is now time to reveal the winner of the entire competition, and I cannot be more excited!

So, are you ready?

The Results

The Results are as shown below:

Mount Contradiction	%9
The Meltdown	%61
Orbiter	%4
Triton	%24
Freeze, Melt, Vaporize	%2

Accordingly, the winner of the first ever 'Game Maker Race' competition is:

The Meltdown

By Jelle Straatsma

The game got a whopping 61% of the votes, far ahead of any other game.

Indeed, the Meltdown is a simple game and relies mostly on its clever concept and gameplay. The game truly deserves the prizes!

Congratulations,

Eyas Sharaiha■

The Meltdown, by **Jelle Straatsma**, is the winning game of 'The Game Maker Race' competition. Veeti Paananen reviews the game to tell us what he thinks of it: does it really deserve the first place prize for the competition?

What they say

"*fun game, but gets a bit repetitive*", says MT. I have to concur on this.

Description

The Meltdown is a game designed for the *MarkUp/GMTech Fire and Ice* contest. Since the contest requires a theme related to the subject of fire and ice, the game obviously somehow relates to it: in its case, the player controls, surprise, the fire and ice. One can change to fire or ice at any time: fire is the offensive, ice is the defensive: the first one can shoot fire at enemies; the second can build ice blocks (and freeze existing ones).

The levels consist of solid blocks that cannot melt, and ice blocks. The player occasionally needs to make ice blocks to get over high obstacles, but most of the time needs to use fire to kill enemies, that are the following: a flying enemy, that shoots at the player, a flying enemy that can't cause direct offense to the player, but drops ice-melting blocks, and a walking enemy in the style of a Goomba from Mario games.

The game is controlled with the arrow keys and the two mouse buttons. It lacks WASD controlling, which is a small minus, but the game is comfortable enough to play with the arrow keys. The background music consists of a few decent tunes that keep the player entertained. There are no sound effects.

The graphics consist of particle effects and solid colors. They fit the theme of the game, although could be less repetitive: *The Meltdown* can get a bit repetitive during the course of the game in other aspects too. It certainly needs some improvement to keep the player entertained: for example, powerups. There are also some bad examples of English in the text. The engine is very bug-free.

Pros and Cons

All in all, *The Meltdown* can be an entertaining game for a small break, or even for a little bit longer time, and a definite try for anybody.

Pros

- *Good engine*
- *Graphics fit the theme*
- *Addictive*

Cons

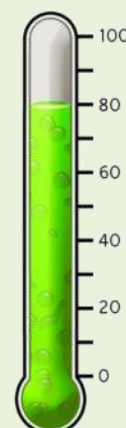
- *Poor sound effects*
- *Graphics need a bit more improvement*
- *Gets repetitive*

Conclusion

This is a game anybody should check out.

Veeti Paananen ■

Ratings



Graphics: 5/6
Sound: 4/6
Gameplay: 5/6
Design: 5/6

Developer: Jelle Straatsma
Version: 1.0
Game Maker: Version 7
Instant Play:

Here

God of Rock Development Journal

I'm working on the online lobby menu at the moment. The menu for the lobby features a cool 3D globe and some shiny gray buttons to the left of the globe. I bet you wondering why I'm mentioning this. I am mentioning it because that's where I had some annoying problems in making.

The first problem I had with that was getting the 3D globe model to fit the texture correctly. To fix that I wrote a very complicated script which draws the sprite outside the room, stretches it, and saves that new image, and sets that as the texture. The next problem was getting the background look 2D. After hours of messing with Z coordinates I finally got that effect. After solving those I had to figure out how to draw 2D menu buttons in a 3D world. I spent countless hours messing with the `'d3d_set_projection_ortho'` function and found out I just had to place the object like normal but, lower the depth.

The next thing I worked on doing was the note system, since I have to redo it. Right now I'm working on making a system where the room size can be 64x480 and the notes will still be synced. I'm about 25% done with it. This has to be done because the game kind of lags on less powerful computers.

The next function I added was a system where players can play career mode together, that mode is about 10% complete. When that mode is complete players will have the option to share a fretboard or play bass and lead cooperatively. (Please note: If I find this mode doesn't really need to be in the game, it will be removed.) The final problem I faced was in the game data file the problem being sometimes game data does not get written. To fix this I made a 'controller' object that does all file writing. This method works but the FPS may slow down, but not a lot. Small price to pay to make sure you data gets saved, don't you agree? The next problem is the in the downloader sometimes it freezes or never finishes the download. It took me ages to fix that but it's been done. Well, there's the Development Journal of God of Rock, for the month of December.

Mathew Malone■

Advertisement

Get a free book by writing to MarkUp!
Choose a book related to game development and review it for MarkUp Magazine!

Free Book - Free Shipping
Click for Information

[CLICK HERE](#)

MarDar: MarkUp Radar

Creation: Sun Sun Der



"Sun Sun Der" is Martin Crownover's entry for YoYo Games' winter challenge competition. The game is a SHUMP in which you play as "Friga" who fights against the monsters of summer trying to restore winter back and therefore get everything into its natural cycle.

The game is full of eye candy and gameplay! On the eye candy-side, the graphics and effects are just amazing: the darkness, fire, attacks, it just comes together too well. For gameplay, the experience becomes exciting and rewarding, with lots of monsters and enemies, power-ups, special attacks, etc. Not to mention, the musing is amazing – AMAZING! It has been compiled by David Peritte, the same guy who was responsible for the 'Aces High Over Verlor Island' background music.



Work in Progress: Super Game Maker Arena

The Game is good in principle, and appears to be heading on the right track, but nevertheless isn't in a shape right now to be considered 'amazing' as is – but it's going there if all goes well.

Super Game Maker Arena is a Super Smash Bros.-style game with some 'Game Maker elements' thrown in it. What Game Maker elements, you ask? We don't know, we have to wait and see. So what's good about the game? Well, on the graphics side, the game has good animations and characters. The gameplay is fun and the concept is nice, but nevertheless it gets confusing and is buggy here and there, with things like jumping and physics.

Goodbye!

Thank you for reading our New Year Mini special issue! The issue was created solely to release the results for the Game Maker Race, and therefore the issue might appear to be limited in many ways – but remember, we released it very early just to give everyone a peak at the results.. As soon as possible.

We're coming back with our 1-year anniversary special: Issue 12. We're planning on making this a very special issue, so if you have anything to contribute, please send it along!

MarkUp Magazine is supported by contributors! If you are a reader of MarkUp Magazine, then you can help us become better by sharing your experience with us and the readers! You can help by either joining the MarkUp [forum](#), or e-mailing the MarkUp [staff](#). You don't *need* to apply to become a staff member, you can just write for us!

The MarkUp Staff■

Be sure to Check Out...

[GMking.org](#) is the parent network for MarkUp magazine. It is constructed as to behave like a centralized portal that links to the four main aspects of GMking.org's projects: The GMking.org Site [which is now a sub-site of the main gmking.org page], The GMking.org forums, GMPedia.org, and MarkUp magazines. Visit the site for all of MarkUp's sister projects!

One of MarkUp's sister projects, also developed and maintained by GMking.org, is GMPedia.org. To learn more information about your Game Platform of choice, you could check out [GMPedia.org](#). GMPedia is a game development wiki with a growing community-base and content. GMPedia is not limited to Game Maker, but expands to include all forms of game development, including Flash, etc.

GMking.org

Let them make games!

MarkUp is an open publication made possible by the contributions of people like you; please visit markup.gmking.org for information on how to contribute. Thank you for your support!

©2007 Markup, a GMking.org project, and its contributors. This work is licensed under the Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA. Additionally, permission to use figures, tables and brief excerpts from this work in scientific and educational works is hereby granted, provided the source is acknowledged. As well, any use of the material in this work that is determined to be "fair use" under Section 107 or that satisfies the conditions specified in Section 108 of the U.S. Copyright Law (17 USC, as revised by P.L. 94-553) does not require the author's permission.

The names, trademarks, service marks, and logos appearing in this magazine are property of their respective owners, and are not to be used in any advertising or publicity, or otherwise to indicate sponsorship of or affiliation with any product or service. While the information contained in this magazine has been compiled from sources believed to be