

MARKUP

1ST ANNIVERSARY



Artificial Intelligence:
a conceptual approach

**Windows Vista's Game
Explorer: Revisited**

Plus... Game Maker Compatibility with Windows Vista

WINTER COMPETITION

Interviews with the Winners!

THE MAKING OF...

The Expelations of Pod Man

STEP-BY-STEP TUTORIALS

360° Platform Gravity

GAME MAKER INSIGHT

IDs in Game Maker

AND MUCH MORE!



MarkUp Magazine

Table of Contents

Editorials

Editor's Desk	3
Game Maker Experiences	36
Game Maker and Windows Vista	37
More than Just Games.....	38
Games I'd Like to See	40
The Default Message Box	42
YoYo Games: What would make the site much better.....	43

Development Entries

The Making "The Expelations of the Little Green Pod Man"	4
Development Journal: Cities at War	8
Development Journal: God of Rock	11

Previews

65 Million + 1 BC: Three Months Later	13
---	----

Tutorials

Artificial Intelligence for Beginners: A Conceptual Approach.....	15
Storing and Retrieving Binary data using Bit Masks.....	19
360° Platform Gravity.....	20
Vista's Game Explorer: Revisited	21
IDs in Game Maker	28

Monthly Specials

Extension of the Month	31
Script of the Month	32
MarDar: Creations	37
MarDar: WIP	38

Interviews

YoYo Games Winter Competition Winners.....	33
--	----

Reviews

Space Explorer Deluxe.....	44
Pyramids	45
Books: Game Maker Apprentice	46

Contributors

Eyas Sharaiha	Sr. Editor
Robin Monks	Editor
Andris Belinskis	Editor
Bart Teunis	Writer
Darragh Tobin	Writer
David Yates	Writer
Dougie	Writer
Jelle Straatsma	Writer
Josh Hummel	Writer
L4ntom	Writer
Leif Greenman	Writer
Mathew Malone	Writer
Philip Gamble	Writer
Rick Slick	Writer
Robert Colton	Writer
Robert-Jan Bruintjes	Writer
Schlyer	Writer
Sean Mounce	Writer
Snailfox	Writer
Stephen H.	Writer
Thomas	Writer
Veeti Paananen	Writer
Suhaib Al-Dari	Cover Designer

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!

This issue's cover...

Cover created by Suhaib Al-Dari. The model in the cover is created by Andris Belinskis. The background image is created by Dirk van der Made, and is licensed under Creative Commons Attribution 1.0. View the original image [here](#).

Contribution

For more information about contributing to MarkUp Game Development Magazine, view the contribution page on our official website [here](#).



Welcome!

Editor's Desk

For those of you who haven't enjoyed an episode of Star Trek at some point, I'm speaking of games of logic. I've been thinking a bit lately about thinking in games, and how it can be done right and wrong. I mean, take a look at Tetris; it's probably the best example out there of an addictive logic puzzle game. It's been the subject of a huge number of clones, a [Tetris Board Game](#) (I want one!), and many strategy guides. So what has made Tetris so popular? I propose a theory:

Re-playability

Tetris has the opportunity to advance levels with increasing difficulty, and each and every game will be entirely different. It encourages the player to think fast, and tests their mental skills and stamina. It drives the players will to achieve better and better scores, and encourages attempt after attempt as the player tries to beat his or her old score.

Simple Implementation

Tetris isn't complex, it doesn't sport highly complex graphics, it's simple enough for a young child to play and understand; yet, it's addictive enough for adults to be interested. In my opinion the best puzzle games don't have complex instructions, or many player controls, they are simple tests of skill.

There is no doubt more reasons these sorts of games succeed, but, you get the point.

We have a great issue of MarkUp prepared for you this month as we hit that magical 1 year mark! This makes us the only Game Maker magazine to ever reach 12 issues! We couldn't have done it without the great support of the GMking community, and the wonderful team of writers who have supplied top-notch content over the past year.

Dig in, and celebrate 12 issues with us!

Robin Monks

In the News...

YoYo Games Competition

The YoYo Games Winter Competition's results have been released. The prizes totaled to \$1750, the first prize being \$1000 in cash. Frozzd by 2Dcube was the first place prize winner. The top-two winning game authors are interviewed in pages 33-35 of this issue of MarkUp Magazine.

YoYo Games have also started the [Ancient Civilization Competition](#), having the same total prize sum. Submissions have already started and end on April 27th, 2008.

GMplanet Launched

[GMplanet](#), a project of GMking.org, is a content aggregator of many Game Maker RSS feeds. The site was launched at the beginning of February.

Other Competitions

Other sites have also started working on their own competitions, not related to games, but instead tutorials and resources. [Game Maker Helper](#) and [GMTutorials](#) are both notable examples. Competitions to showcase your work are discussed in page 10.

Independent Game Festival

The [2008 Independent Game Festival](#) results were announced, and this year, we get a sufficient number of Game Maker games on the winner list! Congratulations to all winning games!

The Making of “The Expelations of the Little Green Pod Man”

By: Sean Mounce

“The Expelations of the Little Green Pod Man” is a game I created for a two week platform game creation competition. This article is intended to tell you the thoughts and techniques that I employed in the semi-frantic rush to get this game completed by the deadline.

Introduction

For the most part, I'll be going over the more artistic aspects of this game (The concept, the music, the graphics, etc.), as opposed to the more technical programming aspects of it. There is a very simple reason for this: I never employed any complex scripting to the game; what made it succeed where the more artistic and creative portions, rather than its technical side.

The Concept

As I said before, this game was created for a platform game creation competition. Nothing more specific than that was provided; just the much generalized “Platform Game”. This left me with a very large amount of freedom for the game, but it also resulted in a general lack of direction and inspiration.

For all of us, there will be times in our game making (or any other type of creative endeavors) where we lack a useful starting idea. Therefore, I shall do my best to give good suggestions on how to come up with one.

The first thing I thought about when coming up with the concept was a definition of a platform game that one of the contestants mentioned: “A game based on difficult jumps”. I focused on that one small point and started trying to think of ways to expand on it. So, when trying to come up with a game concept from scratch, it is best to think of a simple word for word definition of the genre you are making. You, obviously, do not have to follow it at all, but it can create a good starting point for your game.

From here, try to think of ways of “making difficult jumps”

more *interesting*. For me, this was by “aiming and launching yourself from something to different platforms scattered across the level”. This thought developed into such things as having to choose between different floating platforms and, essentially, creating a maze. Then, perhaps, adding some interesting puzzles throughout the game would add some diversity and a bit more interest to the game. In fact, the puzzle level (level eight, I believe) was the first level I created for the game, and originally, the game was going to be an entirely puzzle based game, with each level being a different puzzle with you had to solve.

However, due to time constraints (for it took quite some time to get level eight functioning well, and I knew I would not be able to make ten such levels in the given time), I decided to abandon the idea of an entirely puzzle-based game. In the end, I am glad this was the case, for it added much more diversity to the game.



The Making of “The Expelations of the Little Green Pod Man” Cont.

So, I had the very basic concept down, but what about the details? What would the game look like? A tree answered these questions of mine, looks-wise, at least. As I was thinking about the concept of my game, I looked out my window and took notice of the nicely shaped and semi-twisted tree in my front yard. It was also raining outside, and the combination of these two things brought about the wet and forest like feeling/setting of Pod Man. So, when coming up with the concept of a game, it helps to look at, listen to, or do anything to anything you find inspiring, and follow your imagination.

From here the rest of the concept just began to flow. The idea of launching yourself from a plant pod came logically to me from the wet plant setting of the game, and I just continued to think both logically and abstractly from there: Pods eat bees; pods grow; larger pods launch farther; birds pop out of apples from time to time.

As I said, near the end of the game, I was in a massive rush to finish, and I was having difficulty coming up with ideas for levels that were not just reiterations of previous ones. However, through a lack of doing anything else in my day, and forcing myself, through various means, to come up with new level ideas, I managed to finish all ten of the required levels and a completed game.

What I have just explained to you is the best attempt I could make of putting a thought process into words, but it is impossible to know everything that goes on inside your own head. The creative process is one of the few things that you learn from yourself and your experiences instead of from the words of others. However, I hope that these paragraphs are able to help you think of a starting point from which the ideas in your head may grow.

The Graphics

The graphics for Pod Man are the first graphics I have ever done that I was truly pleased with, which is most likely a result of them being the first graphics that I actually spent a lot of time and work on! However, more importantly:

they were the first graphics that I thought out in my head before drawing them on the computer. This, I believe, is one of the most important aspects of creating good graphics: See it in your mind before you see it on the computer. If you do not, you, I know from my own previous attempts at graphics, will come out with boring and excessively two dimensional graphics.

I still am not good at super detailed pixel/graphic work. If I try to make the super detailed sprites with intricately placed pixels that others can do so nicely, I simply leave a garbled mess of pixels behind me. As a result, I logically decided to make simpler graphics, but how could I get them to look nice? Colors and shading would be the answer. Choosing the right colors and adding shading to your sprites are very important (whether the graphics be simple or complex), and the ideas for both are best to be thought of in the “see it in your mind” step that I mentioned earlier. Although, you should definitely experiment with them once you have them on the



SinazXInput

The SinazXInput is a Game Maker Extension that adds support for Xbox 360 Gamepads when connected to a computer.

The extension adds full support for the Xbox controller, its triggers, buttons, and analogue thumbsticks (the input will be mapped on separate axes). In addition, the GEX supports rumble, which could make the Xbox controller vibrate.

While the GEX is mainly targeted at Xbox 360 controllers, it is capable of supporting any XInput controller (XInput is a component of DirectX), except for headsets, etc.

The GEX also comes with an example GMK file.

Get it now!

The Making of “The Expelations of the Little Green Pod Man” Cont.

computer. Just do what looks right.

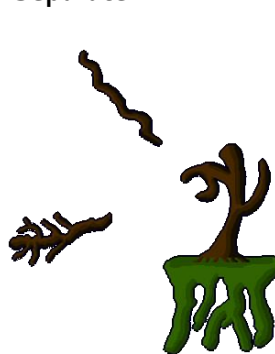
For the colors, I used a darker, more ‘wet’ feeling for the colors, such as what you might see in a rain forest. This was, evidently, attempting to amplify the wet, jungle feeling of the game. The shading in Pod Man is what made the graphics what they were. All of the graphics were based around this idea I had of using only three shades of one color for each object. This idea came simultaneously with the concept with the game, and the first platform I pictured was a floating grass platform with a rocky bottom, using this three-shade shading idea. I sprited the idea and looked at the result. This sprite, which was the first sprite I made for the game:



That was it. After seeing the result of the idea, I was certain that this was what the game would have to look like, so I just followed my new three-shade rule for all the sprites I did. When creating shading for any sprites, I have learned from various sources that it is a given for the light source always comes from the top-left of the sprite, so the farther down and to the right you go, the darker it grows.

For the most part, it was easy to follow these rules, but, sometimes, all the twisted branches created some confusion, in which case I simply experimented with them until it looked right. Another problematic circumstance the trees created was the fact that, unless I made hundreds of tree sprites, it might become tiring for the player to see the same trees over and over again, and if I added hundreds of tree sprites, the game file would become too large. What did I do? I only created about fifteen tree and branch sprites, some for specific purposes, and others just to exist. However, I made the shading on the majority of these tree sprites to be able to line up with each other. That way, I could take two or three tree sprites and place them on top of each other and make it look like one tree, such as shown here:

Separate:



Conjoined:



It is still fairly obvious that I am recycling sprites, but now, hopefully, the fact shall not be glaring at the player.

In summary, first think of what you intend for your sprite to look like. Most importantly, the design of the creature, character or object you are going to sprite. Then put what

hamburgerspel.com



The Making of “The Expelations of the Little Green Pod Man” Cont.

is in your head onto the computer. Lastly, if it needs it, edit it until it looks how you want it to.

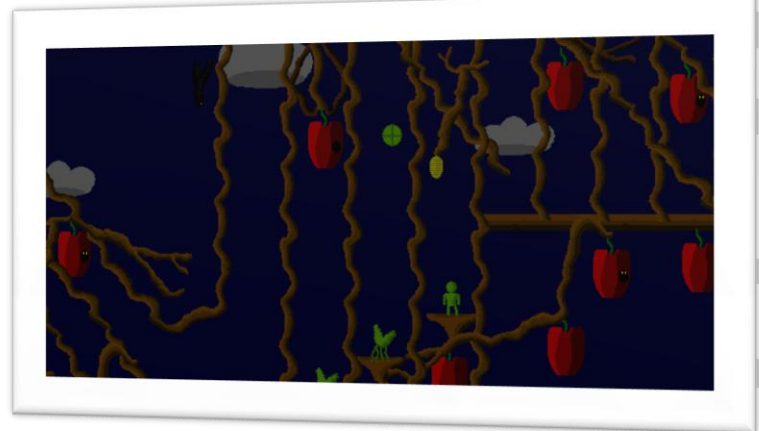
The Music

I very much enjoy making music, be it for a game or just a melody in my head that I decide to make into a song. However, it will also be, like the game concept, difficult to describe where the song idea came from.

The creation/recording process of the song for Pod Man was different than that of other songs I have made for games. With the other songs, I would usually make the song up as I went along, writing and recording section at a time, changing it until I liked it, and then moving on to the next section. With the Pod Man song, however, I was required to do it differently. The reason is because I, through a series of circumstances, did not have any multi-track music recording software available, so I began to think how I could make a single-track song as full and mood-fitting as a multi-track one. I picked up my guitar and began to experiment.

After a while, I came up with a little section I liked, and that made use of the delay effect. I liked the delay on there, because, to me, it seemed to give a raindrop feeling to the song. I continued to expand upon the song, creating two different sections: one in the high octaves of the guitar, and another strumming down low. I had many ideas for the song that I knew would sound quite nice as the main or backing tracks, but because of my lack of recording software, I knew I had to just pick out the ones I liked the most. I continued to polish the song, picking out the different ideas that I thought sounded the best, and until I had something I was happy with.

Over the next few days, I was becoming so busy with working on the game that I did not have a chance to record the song, and, as the days until the deadline lessened, I began looking for a good opportunity to record the song. I knew a single track would not take too long. However, by the day of the deadline, I still did not have it



recorded, nor did I have the game fully completed. I still had the final level to do, and a few polishing.

Then another thing happened that added interest to the situation. I got the recording software back and installed it with no fuss. Now I could add those other tracks I wished to add. But I had four hours until the deadline, and a game still to finish? So, for the next four hours, I went back and forth between finishing the game and recording the song. I finished the game half an hour before the deadline, and spent the time after that finishing what I had left of the song. With just a few minutes to spare, I finished that as well, and listened to it. I was very pleased with how it sounded (possibly one of my favorite songs that I have made), and put it into the game. From there, I uploaded the game, and submitted it fifty seconds past the deadline. Fortunately, they still accepted it as an entry.

Conclusion

That is the story of The Expelations of the Little Green Pod Man! I hope you were able to learn something from reading this article, and if not, I hope you enjoyed reading it.

Thanks for reading!

■

Cities at War

By: Robert-Jan Bruintjes

Development Journal

Do you know how hard it is to make a decent RTS-game? If not, then keep on reading, since that is what I'm going to tell about. Every month, I will write an article here about the development of Cities at War, and the technical sides of developing a RTS-game.

The Game

Cities At War is a Real-Time Strategy-game by Xsoft Games. I started this project about a year ago. I play a lot of RTS-games, and I always wanted to make one myself. So I started this new project, and I knew it would take a long time. But I didn't know I would still be working on the engine a half year later. I lost my motivation to work on the project, so I started another project. I joined Xsoft Games, which was then still not so big of a team. After a while of working on my own project, the Xsoft Games team caught my attention again, and a bit later I decided to do Cities at War with the people from Xsoft Games. The first big project of Xsoft Games had started.

From then on, the development went quite fast. Paul23 almost rewrote the whole engine, which was actually needed, since the original one wasn't quite bug-less. I continued working on small bugs, and built in some extra things. And that's where we are now. More information about the game itself can be found at the [website](#) of Cities At War, at [the GMC topic](#) and the website of [Xsoft Games](#).

Features

In a year of developing, a lot of features already have been built in. That includes a Fog of War engine, a movement system using GM's `mp_grid`-functions, a minimap system with Fog of War, and an inventory system. You can check them all out yourself by playing the latest version at [YoYoGames.com](#).

The current movement system we're using uses the standard `mp_grid` functions in Game Maker, as I said earlier.

But paul23 is trying to find a mechanism that works better than the current one. With the current system, when you make multiple units, and try to move them at the same time, they get stuck in each other. That is, if we would let the units collide. In the current version we have disabled that, since that just doesn't work. Units will move over each other, to avoid the getting stuck. More about this in the next issue.

The minimap is actually the FOW combined with a normal minimap system. It uses the same surfaces as the Fog of War does, but it resizes it to the size of the minimap. Underneath the FOW, every object draws its own sprite rescaled in the minimap. You can also click in the minimap to change the view to a specific position in the map. I will discuss the inventory in the next issue.

We're planning to build in some great features. There will



for example be a huge multiplayer online system running. We're now thinking about a good reward system for this feature. I already thought up the story of the campaign. The plot of the story will be kept secret. We're also planning to make a new kind of mode available: Quick Battle. You start with an army and some training facilities, and you'll have to fight soon against the enemy.

Fog Of War

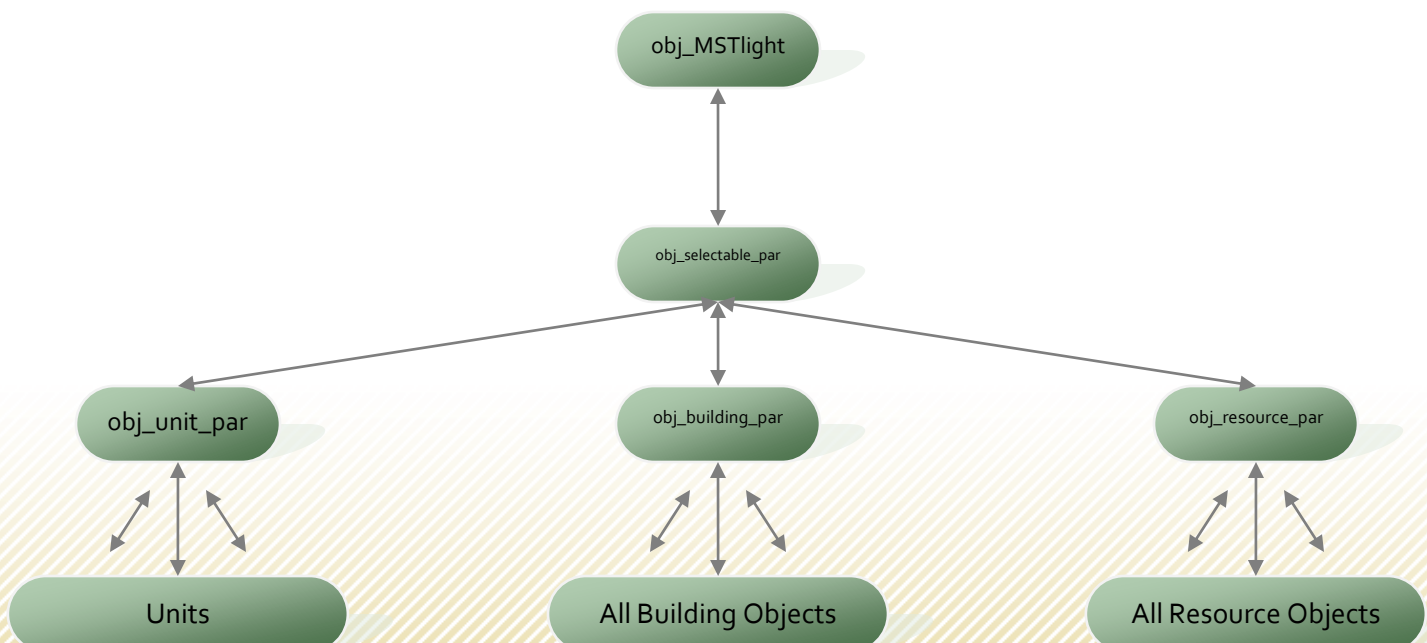
Fog of War is black fog that hangs over unexplored terrain in RTS games. The Fog of War engine uses surfaces. A surface is a sort of separate canvas, on which you can draw instead of drawing on the screen itself. Then later, you can draw the surface on the screen using different blend modes, to create certain effects. For the FOW, we use two different surfaces, one for covering up the unexplored terrain, and one for highlighting the area that is viewed by the objects. The first surface is as big as the whole room, while the second one is as big as the view. This is done because only places inside the view can be highlighted, since it is impossible to select anything outside the view.

This second surface is cleared every step, after it is drawn. The first surface is not cleared, since that one reminds all explored terrain. FOW comes with a complicated parent system, for the units that should cast light are all children of one object called 'obj_MSTlight'. In our game, the kinds of objects are grouped under special object. All units for example have as parent 'obj_unit_par'. All the buildings have as parent 'obj_building_par'. Then all these parents have one and the same parent: 'obj_selectable_par'. So all selectable objects are grouped under 'obj_selectable_par'. Then 'obj_selectable_par' has 'obj_MSTlight' as parent. So all selectable units cast light, as you can see in the schedule.

In the create event of another object, called 'obj_FOW', we have this code (after we made the surfaces):

```
surface_set_target(global.FOW_surface)
draw_clear(c_black)
surface_reset_target();
```

We first set the target to the surface we just made. Then we make the surface all black. The last action is to reset the drawing target back to the screen. Now in the step event, we have the following code (next page).



```
surface_set_target(global.FOW_surface)
draw_set_blend_mode_ext(bm_src_alpha,bm_inv_src_color);

with (obj_MSTlight)
{
    if (team == global.userteam){
draw_sprite_ext(FOW_sprite, FOW_image, x/global.FOW_quality,
y/global.FOW_quality, FOW_xscale/global.FOW_quality,
FOW_yscale/global.FOW_quality, FOW_angle, FOW_colour,
global.FOW_alpha)
    }
}
```

These `FOW_`-variables are all pre-defined variables. `FOW_sprite` for example is the sprite of the spotlight. What happens first is that we set the drawing target to the surface again. Then we set the blend mode, so we get the effect that we want: the white on the surface will be

transparent. In the code after that, for every child of `obj_MSTlight`, the spotlight is drawn at the right position, at both surfaces. It goes a little too far for this time to describe that in detail. In the draw event, both surfaces are drawn at the screen.

Conclusion

We've been working really hard on Cities at War, a lot of features have already been built into the game, but a lot is yet to come. Next month, we'll explain the inventory system and movement system better. See you next time!

■

Contests to Showcase Your Work

The past few months have seen probably the highest ever concentration of Game Maker competitions. First we had The GM Race, a joint venture between the GameMakerTech and MarkUp magazines which attracted 20 entries and then YoYo Games got in on the act just before Christmas. YoYo Games have recently announced details of their second \$1750 competition which runs until 27th April.

As well as the traditional game making competitions there are also a number of other contests currently open which focus on the game maker platform. In January GMTutorials.com ran their first tutorial competition giving away \$100, and they are back at it again this month with 3 domain names on offer. Game Maker Helper on the other hand has a \$35 competition which is open to any Game Maker scripts, example, libraries or extension that aren't currently on the site.

The first YoYo Games competition was well publicized, receiving attention on many Game Maker websites and even websites devoted to the larger area of Indie game development. The result being that the winning game "Frozdz" has currently been played over 8,000 times on

the YoYo Games website and is in the top 25 most played games, despite only being uploaded at the end of December 2007.

Even if you don't think you are in with a realistic chance of winning entering a competition will undoubtedly get your game or tutorial seen by more people than it would otherwise be, and can also be a source of valuable feedback. You have nothing to lose and a looming contest deadline could provide an incentive for you to crack on with your development. If it's any good there is also the chance that you could win some prizes!

Currently open contests:

- YoYo Games \$1750 Ancient Civilization competition
<http://www.yoyogames.com/gamemaker/competition02>
- GameMakerHelper \$35 competition
<http://gamemakerhelper.com/>
- GMTutorials.com tutorial competition
http://gmtutorials.com/article/category:news-name:february_2008_prize_competition/

Philip Gamble ■

God of Rock

By: Mathew Malone

Well, this time I don't really have any bugs since I haven't been working on God of Rock for a while but another fellow GMC Member is. So I'll just talk about what he's fixing and how he's going about fixing it.

Month 1

Bugs

The first glitch is the chord system in the not working correctly. So to correct this he is going to use two-dimensional arrays. By doing this he can basically store two values in one variable. For example let's say we had a variable titled "Game Over". We could store two values one being the amount of times the player has died and one being if the screen has shown or not. By doing that we can store the values, and those values, in this case would be if the chord was hit and which notes they were, in the form of a string. I have written a special piece of code to read several pieces of information from a single string which is why I only need one value for the note information.

He is also recoding the Band System since my coding was a little messy. Ever noticed how when creating a Band the game always fails to read your name out of the file, with the first boot of the game? Some people have been having that problem and it would soon be corrected.

The third thing he is fixing is lefty flip system. It's not complicated but I want to do it without any changing of y or x coordinates and that is why I haven't already done it myself. Well I hope you are looking forward to the next update of God of Rock.

Delays

I'd really like to apologize for the ridiculously-long wait time but, there are a lot of people on the GMC that didn't want to help me with development and that's what causes



the delay. Meanwhile make sure to read the information in the game topic it may tell you something good.

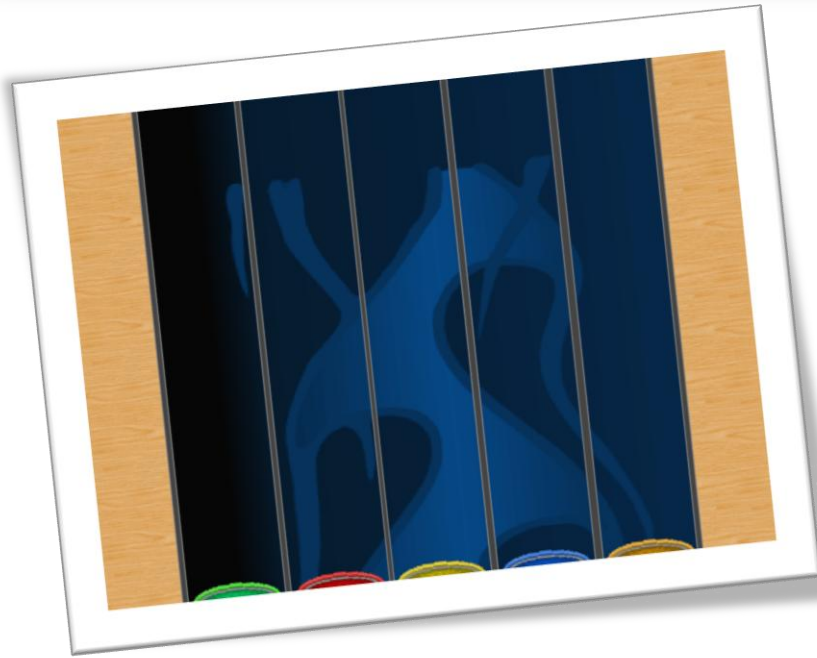
Month 2

Recoding Adventures

As you probably know, I have to recode the engine. I am about 60 percent done with that. The first thing I did was change the song sync files to an INI format. This INI format, is unique thou since it specifies periods of time.

God of Rock Cont.

By: Mathew Malone



For example, '8 = N <expression1> <expression2>' would mean make a note at the 8th tick. The INI files also contain other important information such as, BPM, Resolution, Time Signatures, Events, Lighting Events, etc. Lighting is a new feature by the way that has been half-implemented.

Important Improvements

When the recoding is done the game will feature a number of improvements also like 640X480 room size, Guitar Hero Guitar Support, and Customizable FPS Limit. This 'revamped' version isn't expected to be release for a while unfortunately, do to complicated coding problems and the fact that this project is a one-man project. When it is

released thou, I will be open to team member requests.

Fixes

Even though I have a lot of issues to fix I have fixed some conflicts, luckily. Some of those being the mysterious, 'Pause N' Freeze' glitch, The Invisible Menu Glitch, The Mono Music Issue, The Sound Frequency Glitch, The Long Note Glitch, and last but certainly not least, The Song Downloading Glitch! All those glitches (and probably more) have been corrected in the newer version. The last bug was a pain to fix, it took me days!

But at least I got some MASTER TRACKS!!! Some of the bands that are available for this include, Jonathan Coulton, 8763 Days Till' Extinction (My one-man band), and more... So that means all this waiting is going to pay off and you'll be rewarded with a better song list.

Conclusion

Well that's the development journal for January and February. I haven't done much but I am still working on this. I hope you will still support me in the development of this project.

■

Advertisement

New Indie Gaming Website...

GameApe

Advertisement

65M+1 BC: Three Months Later

By: snailfox

Hello. If reading about Earthly events that happened more than 6,000 years ago causes you undue stress, please skip this article immediately! I don't want to offend anyone. If, on the other hand, you are not crazy – i.e. you think computer games are not worth getting stressed about – then read on!

You might be wondering just what this article is about anyway. Well, about comets, giant machines, talking dinosaurs and, more broadly, the game I have been working on for quite a while now – 'Sixty Five Million And One BC'.

If you're already getting angry because you think the earth didn't exist that long ago, then I suggest you stop reading the article; that is, if historical "accuracy" of what scientifically happened before 'the creation' matters so much to you, then this probably isn't your thing!

The game is about what is generally thought to have happened 65 million years ago – the extinction of dinosaurs thanks to a comet slamming into Central America and blowing everything up. The story itself focuses on a pack of velociraptors who realize that Armageddon is on its way and set about building a giant machine to try and stop it.

How do they do that? Well that would spoil it. But safe to say it revolves around fighting enemies, solving puzzles and hack-slash-tail-whipping your way through the Mesozoic world as a fierce and terrifying velociraptor that talks.

MarkUp Magazine was kind enough to feature an in-depth preview of the first release of 65M+1BC, which you can find in Issue 9, [here](#).

"What's new?" in a nutshell

But the game has come a long, long way since then! It's very nearly finished. As well as the forest, raptor nest,

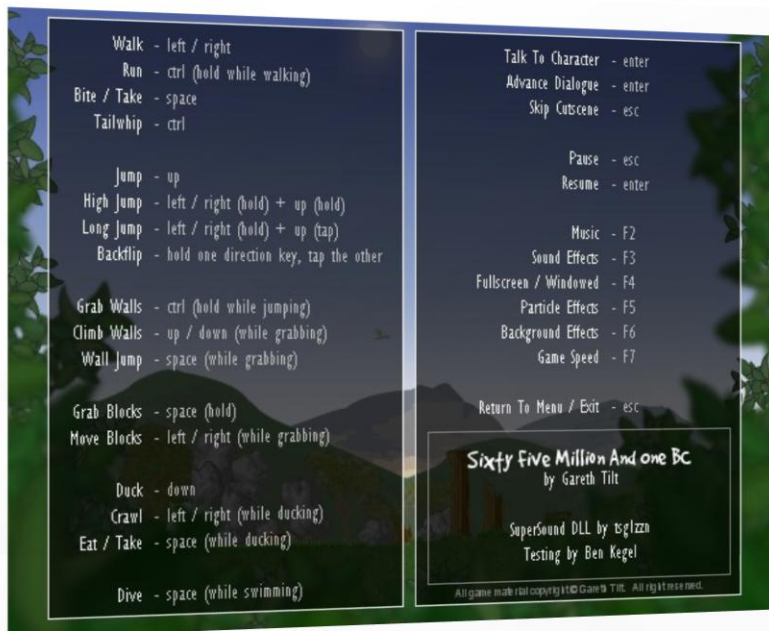


mountains and cave levels, there is a whole 'plains' area with a seriously massive tyrannosaurus-rex at the end. There are boss fights, cut scenes, and interesting animated characters throughout. There's also an epic battle between your raptor pack and a rival one, which shows off the enemy AI very nicely: POUNCE-POUNCE-POUNCE essentially, but still very fun.

Object interactions

I've spent a lot of time polishing – especially the interactions during game play. If you leap at an enemy that's bigger than you, you typically get knocked flat, but if it's a smaller one it'll go flying! Running through smoke and other particle effects makes it swish and swirl, while jumping into water creates ripples and spray.

65M+1: Three Months Later Cont.



Resources

In case you are wondering, the graphics, the engine, the music and even the dinosaur sounds are all original. So if anyone wants to know how to make a velociraptor roar sound, I can safely say that one way is to find buffalo grunts, puma snarls and walrus barks, then mix them all together. Another way would be to hit it on the tail with a stick, I expect. The music is composed around a couple of main themes, with different tracks for each area and the various cut scenes.

Conclusion

There are Easter eggs to find, of course, and features to unlock by completing the game.

Dynamic Nature of the Game

The game is designed as to be dynamic in nature. Nothing is static, not even the game play - Meteors streak through the sky during the final levels, the sky gradually grows red and the ground rumbles as that comet draws nearer. You even hitch a ride with some pterodactyls and dodge an airborne meteor shower at one stage. If this game doesn't make you feel like a talking dinosaur on the edge of global cataclysm, nothing will.

There is a lot of game play in Sixty Five Million and One BC – it takes even longer to play than it does to say, so you can expect to be kept busy for a while. Even if you skip all the cut scenes and go straight through from objective to objective, you'll need a good two hours to get to the end. If you're really bad at it, it might provide a whole year's worth of entertainment!

You can check out the latest demo and updates at:

gmc.yoyogames.com/?showtopic=266828

Status: 95% done, beta testing getting under way!



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

Artificial Intelligence for Beginners

By: Rick Slick

... a conceptual approach

One of the most challenging aspects of designing a single-player game is to ensure that the enemies that your player encounters along the way provide some degree of difficulty in avoiding or eliminating. Failure to achieve this will cause your game to become too easy to complete for many players, reducing the amount of interest in your game and minimizing the replay-ability.

This is why it's important to put a certain amount of thought into the design of your enemy AI first before you ever sit down to churn out the code. Not only will you end up with a more capable AI, but you will make the programming requirements a lot less daunting and frustrating by breaking everything up into more manageable pieces.

I'm not going to sit here and tell you that this is the only way to create a workable AI, nor is it probably the most efficient, but what you will read here is very easy to conceptualize and then translate into coding requirements. As you become more proficient with dealing with all of the different conditions and information that your AI has to analyze in order to make decisions on what actions to take, then you can begin trying more complex AI concepts such as command stacks and the ability to learn and adapt to the player's behavior.

Some purists will argue that "true artificial intelligence" requires nothing less than a highly advanced neural network, the ability to learn, and the ability to store, process, and analyze information about the AI's surroundings. However, keep in mind that these very advanced AI simulations are being conducted on supercomputers at universities operating with research grants and the like. Simple AI in terms of what we would want to create from Game Maker would satisfy the following question: "How do I make an enemy provide enough of a challenge to the player without significantly impacting performance?"

Designing from the Top Down

First, let's start with a very basic top-down view of artificial intelligence. For the purposes of this article, I will describe a simple tank game, where the player and the computer are both maneuvering tanks on a playing field that has multiple obstacles that will block both movement and

SLG Account System

The "SLG Account System" is an online multiplayer system for account management, etc. The system is similar to the "Reflect Games" online account system, but it is instead open to all developers for using it with their games.

SLAS (Star Light Account System) is almost a carbon copy of reflect games. It was created to help Gm (Game Maker) users redirect multiplayer info such as:

- IP
- Server Name
- Ports
- Other Information

The system just requires a registered copy of Game Maker and a php compatible web server. The SLG Account System is the creation of **Jack Liscom**, also known as **Fox USA** in the GMC.

Get it now:

gmc.yoyogames.com/?showtopic=352013

tank shells. The assumption is that the reader has a working knowledge of GM, variables, and the mechanics of making objects move while avoiding objects with the pathfinding commands in GML. I will also conceptualize the enemy AI as a "state machine" (more accurately known as a "finite state machine"), where at any given moment in time, the enemy AI will be in a particular "state", each of which having its own set of tasks that it will perform while it is in that state, as well as conditions for changing from one state to another. You can store this state as a local variable in the enemy object, and process the actions for that particular state in the object's step event, using a switch statement to select what to do depending on what state the enemy AI is in.

The fundamental question we want to ask ourselves is "If I were playing as the enemy unit, what are all of the different decisions that I would make about how to move my tank, rotate my turret, and fire?" Then, let us refine these questions with more amplifying information and what each of these questions mean for our particular example. Breaking this down in the conceptual stage will help us better understand each programming task in smaller, more manageable pieces:

- 1) Can I see the human player?
 - a) If he's closer than 500 units and I have line of sight, I can see him.
 - b) If I have seen him in the last 10 seconds and he's no longer in sight, remember where he was at.
- 2) Do I want to move towards him, stay put, or back off?
 - a) If his back is turned to me, I will move to him
 - b) If he's moving closer to me, stay put
 - c) If my health is too low, back off
- 3) Is my turret aiming at him? Should I rotate it?
 - a) Rotate my turret to face him or his last known position at all times
- 4) Do I have a clear shot?
 - a) Is there something between me and where I want to shoot?

- b) If there is, I should move somewhere else
- 5) Which way is the human moving? Where should I aim to lead the target?
 - a) Based on speed and direction of the human, and speed of my bullet, calculate the trajectory I need to fire at
 - b) Add/subtract some degree of error to this trajectory based on difficulty level of the AI
- 6) Am I ready to take a shot?
 - a) I can only fire one shot every 5 seconds, make it count
- 7) If I don't see the human player, which way should I go?
 - a) Can I go straight? If so, move forward
 - b) At random intervals, rotate the tank if the path is free to go in that direction
 - c) If I am stuck, determine which way I can move and rotate the tank in that direction
 - d) If my health is low, make sure there are obstacles between the human and me
- 8) Is there something blocking my path? Which way should I turn instead?
 - a) If I don't see the human, turn a random direction and try that instead
 - b) If I see the human, choose a new path to a clear shot
- 9) Is a tank shell coming at me? Where can I move to avoid it?
 - a) Determine the angle of the attack and move perpendicular to it if free to do so.
 - b) If not free to move perpendicular to the angle of attack, move to the closest point that puts an obstacle between me and the projectile (or player's tank?)

Information Availability

It would be also helpful at this point to think about what information the AI will have available to it in order to make

some of these decisions and perform these actions. Be careful about what information you let the AI have access to, because this will directly impact the difficulty of the AI. Also keep in mind that no one likes to play against AI that has unfair advantages due to the amount of information it can possibly be aware of, so you will need to consider all of this carefully.

The most important and powerful piece of information the AI can have is the location of the player. What is the visual range for the enemy AI? This would be a function of your AI difficulty. One option might be that if you can see the entire room on the screen, then since the player would be able to see the enemy AI at all times, so we should assume the reverse for the enemy AI. Or, to give the player an advantage, perhaps the enemy AI only knows the player's location if you are within 500 units of it. Or, if you are using views, perhaps the enemy AI's visual range is a bit farther than the radius of the view so that the enemy AI sees the player coming before the player sees it. We could ask: would the enemy be able to see the player if it has no line of sight? Can the player see the enemy unit without line of sight? Will the AI know how much damage it has inflicted on the player, and perhaps make decisions about how/when to attack accordingly? Can it tell when the player has fired a projectile at it? Does it know if the player's turret is even aimed in its direction?

Defining AI States

Now that we've thought about the information available to the enemy AI and what decisions it will be required to evaluate, we can start coming up with some of the AI states that we will want our enemy to be able to switch between. Here are some examples:

Search – In this state, the AI randomly patrols the playing field by setting random destinations and then using path-finding to avoid obstacles along the way, until it finds the player. At that point, it will switch to another state.



Attack Aggressive – In this state, the AI knows where the player is at, will path-find towards the player, and will keep its turret pointed in the direction of the player's projected path based on distance, speed, and direction at all times (or a slight variation of this direction to account for AI difficulty), firing whenever there is a clear line of fire. This state may be used if the enemy is not heavily damaged.

Standoff Attack – In this state, the AI knows where the player is at, will remain stationary, and will keep its turret pointed in the direction of the player's projected path based on distance, speed, and direction at all times (or a slight variation of this direction to account for AI difficulty), firing whenever there is a clear line of fire. This state may be used if the enemy is moderately damaged.

Retreat – In this state, the AI knows where the player is at, will path-find to the farthest destination possible from the player, and will keep its turret pointed in the direction of the player's projected path based on distance, speed, and direction at all times, firing whenever there is a clear line of fire. This state may be used if the enemy is heavily damaged.

Dodge – In this state, the AI knows there is an incoming projectile, and will path-find perpendicular to the angle of attack to quickly clear the projected path of the projectile. After the projectile is no longer a threat, it will switch to

another state.

Maneuver – In this state, the AI knows where the player is at, but does not have a clear line of sight. It'll path-find to a destination along a line perpendicular to the angle created by the player, the obstacle, and itself in order to clear the line of sight.

You can start to see our AI slowly taking form, and you may even start thinking about what conditions you would check for in order for the enemy AI to change state. If you have more than one enemy AI, you could also consider the following additional states designed to establish some rudimentary teamwork:

Request Assistance – In this state, the AI is damaged to a certain degree and knows where the player is at. It will send information on the player's location to the nearest enemy AI that does not already know where the player is at, and then switch to another state. To prevent from the enemy continually requesting assistance and pulling every tank in the room to the player's location, you may want to limit the use of this state to once per AI. You may also want to consider a percent chance that this state is used, based on the difficulty of the AI.

Flank Player – in this state, more than one AI knows where the player is at and the AI will path-find to a destination such that the player's tank is between that destination and another enemy unit. Again, you may also want to consider a percent chance that this state is used, based on the difficulty of the AI.

Conditions for State Changes

The last part of how to conceptually tie all of this together is thinking about the conditions which must be true for the enemy AI to switch between states. These conditions can be obtained from the exercise we performed above when we listed all of the different questions that we came up with in terms of what we needed to think about if we were

playing as the enemy AI.

The easiest condition is whether the location of the player is known, and this could be a simple local variable flag you set such as `PlayerLocationKnown`. If false, then the default state is "Search". Another easy one is whether a player's projectile is incoming... in that case, the state would be switched to "Dodge" regardless of whether the player's location was known or not. If the player's location is known, then the decision to switch to one of the various attacking and retreating states could be a function of relative damage between the player unit and the enemy unit. And lastly, the "Maneuver" state would be switched to if the player's location is known but there is no line of sight to attack, and the enemy unit is not currently in a "Retreat" state.

Conclusions

The actual code on how all of these tasks take place is left as an exercise to the GM programmer, but you can easily see how picking apart all of the decisions, conditions, and information that your AI has to process will make the actual programming a lot easier. The more planning and designing you put into your AI ahead of time, the more opportunity you will have to break each state into the fundamental basics so that you may more easily tackle the coding requirements for your AI. From a simple enemy that follows the player around and fires at the players current location whenever possible, to a more complex enemy that outflanks the player, lays in ambush, guards power-ups, works in tandem with other enemy units, and leads the target when firing, some of the methods and tools that I demonstrated here will be invaluable to you in designing a challenging single-player game. Good luck with your AI design and in programming your next exciting GM project!

■

Storing and Retrieving Binary Data Using Bit Masks

By: Leif Greenman

GreenmanGames.vze.com

Let us imagine we are creating a game, in this game the main character has several properties that need to be set, first of all it can be either male or female, secondly the main character can either be human or orc and last but not least he or she can wield either a sword or a mace as a primary weapon.

The Problem

The tendency of most developers is to immediately assign each property a variable and have the value of those variables read and interpreted when required; however, game maker uses double precision floating point numbers to store even simple integers, this can lead to a great deal of wasted space.

Luckily, if we store each piece of data as a single bit we can compress all these variables into a single integer.

The Solution

For instance, let us suppose we decide that male, human, and sword correspond to a value of 1. To describe a male human who wields a mace we could simply create an integer (I will leave out the extra zero's here for readability) 110_2 such that the first bit is the gender, the

second the race, and the third the weapon (this can be done in either big or little endian, but make sure you are consistent).

We have just represented all this data with the single integer 6. Of course we also need to be able to read this data back 1 bit at a time. To do this we create a bitmask.

The bitmask is simply a value which we can bitwise-and with our integer to retrieve a single bit from the integer. To retrieve the weapon we can use the bitmask 001_2 because $110_2 \& 001_2$ is 0 (predetermined to mean a mace). To retrieve the race we could calculate $110_2 \& 010_2$ which is not 0 (therefore the bit in question was 1, meaning human).

Using this technique one can store a great deal of binary data in a single integer.

■

Advertisement



Advertisement

360° Platform Gravity

By: Josh Hummel

360° platform gravity is a concept that has rapidly expanded in popularity due to the recent release of Nintendo's 'Super Mario Galaxy'. This popularity boom has only been embellished in the GMC by the fact that 2D Cube's 'Frozzd' won first place in the YoYo Games Winter Competition by operating on a similar premise.

Introduction

Now, of course, many budding Game Maker users want a 'piece of the pie'. But due to the math and concepts involved they can't do it. This is where the more advanced (and mathematically inclined) users, including myself, come into the equation.

I have released an example of this type of gravity in the 'Tutorial and Examples' section of the Game Maker Community (GMC).

gmc.yoyogames.com/?showtopic=351438

The example details various aspects of 360° platform gravity, such as:

- Aligning your character to the ground below
- Adjusting gravity to towards to nearest ground
- Jumping and moving in relation to gravity (on uneven ground)

Finding the nearest ground is probably the trickiest part to grasp but basically, the character just checks for collision in a circle constantly increasing in radius (using the `collision_circle` function). If a collision is found it then checks each point on the circle (360 default) to find the exact location of collision (using the `lengthdir_?` functions in combination with the `position_meeting` function). The `gravity_direction` is then adjusted to the

direction of the point.

Another tricky part is to find what angle the terrain slope is at. This example accomplishes this by sending two 'rays' downwards (one on either side of the character) in the direction of gravity (found by the previous step) until both hit the ground. A right-angled triangle is then formed between the two co-ordinates where the rays hit and the angle is worked out from that by using simple trigonometry.

The movement is relatively simple in comparison to the rest of the code. The character simply moves in a direction perpendicular (90°) to the gravity direction. Jumping is accomplished in much the same way by sending the character in a direction 180° from the `gravity_direction`.

That's all what requires understanding in 360° platform gravity, the rest is all relatively simple code that requires little explanation.



Vista's Game Explorer: Revisited

By: Eyas Sharaiha

Back in [the April issue](#) (Issue 2) of MarkUp Magazine, I discussed how to integrate your game with Windows Vista's **Game Explorer**. Today, we will be revisiting the topic to discover more about how to achieve this.

Introduction

The **Game Explorer** is a new feature in Windows Vista that aims to empower the PC-gaming experience by providing an organized, attractive portal to all of the games installed on your computer. The Game Explorer can be accessed by choosing "Games" from the Start Menu's Right Panel. It provides information about your game such as box-art, system requirements, ratings, parental controls, publisher information and websites, etc.

The problem is: creating a game that is added to the Game Explorer is not an easy task; the game needs to be installed on the computer using MSI or InstallScript Installation Software, several SDKs might be downloaded to create a DLL with the game information, and all of the resources created need to be put together with the installer. This article will go through the entire process of creating Game Explorer-compatible articles.

New tools have been created to make this process much easier. Unfortunately, to be able to use these tools, it will require us to download some software from Microsoft, including libraries, SDKs, and C++ header files. To make the article as simple as possible, we're taking advantage of [GMPedia.org](#), a sister project of MarkUp Magazine, to provide further detail about separate software, SDKs, and their installation.

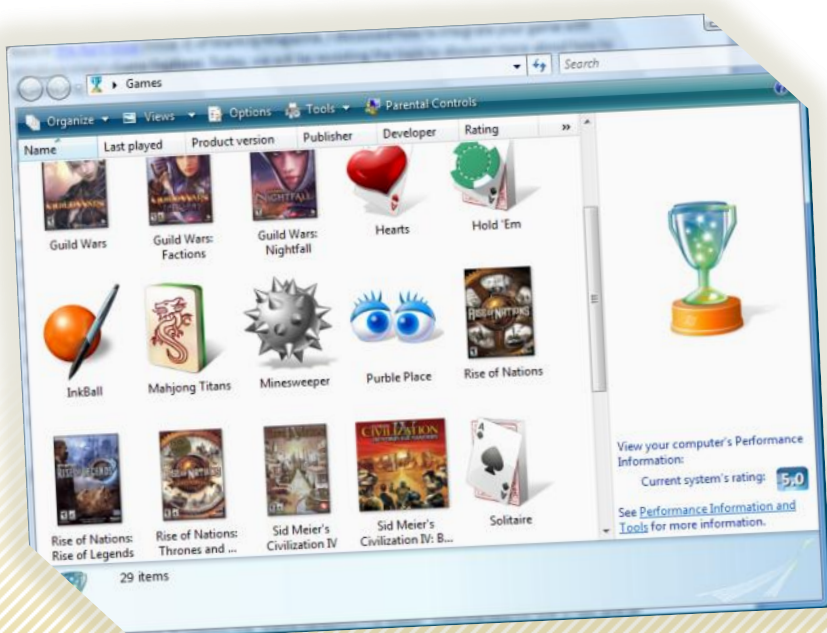
Now that I'm done with this article, I can say with all honesty that I've made sure that the article is completely fool-proof; all of you unfamiliar with Microsoft Visual Studio, or windows development, will – in the end – be able to create your own Game Maker game that is compatible with Windows Vista's Game Explorer.

Required Files

- 👑 [Microsoft Visual C++ 2008](#)
- 👑 [The DirectX November 2007 SDK](#)
- 👑 [Windows SDK](#) (Previously known as the Platform SDK, PSDK)
- 👑 [afxres.h](#) (Part of the MFC Library, an MFC-compatible file can be downloaded)

Overview

The overall process is as follows: the "Game Definition File Editor" from the DirectX SDK is run, and all the information is entered, and the project is saved. Then, the project is used to build an "rc file", which is basically the



Vista's Game Explorer: Revisited Cont.

"source" of a Resource file. The RC file will then be compiled into a RES file (resource file). The created RES file is now linked either to an EXE or a DLL.

Since we are using Game Maker, we cannot link the RES file to GM exe; instead, we will have to link the file to an additional DLL.

This DLL or executable file should be installed by the game's installer, and its path should be passed to the Game Explorer APIs during the installation process. The entire process will now be described in detail.

Creating Game Definition Files (GDFs)

Downloading the DirectX SDK

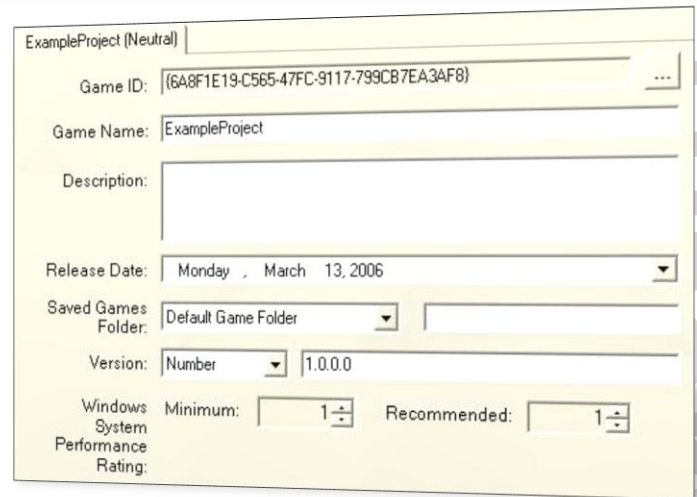
The creation of Game Definition Files was the primary focus of the previous article. Now, the **DirectX November 2007 SDK** provides a much easier way to do this, by providing a piece of software called the Game Definition File Editor.

Simply download the **~428MB** executable and run it; this will install various files on your computer that make it easier to work with Windows Vista's latest features.

 For More information on the **DirectX SDK** and its installation, view the [DirectX SDK](#) article on GMpedia.org

The November 2007 SDK of DirectX includes a **Game Definition File Editor**, found under the name '**GDFmaker.exe**', located in '**utilities/Bin/x86**'. This could be accessed in multiple ways:

- From the start menu: "Start Menu\Programs\Microsoft DirectX SDK (November 2007)\DirectX Utilities"
- Running "gdfmaker" from **Run dialogue**, the **command prompt**, or the **CMD shell**
- Navigating to "%DirectX SDK%/utilities/Bin/x86"



The process of creating a GDF file with the Editor is relatively easy: you could check out the [step-by-step guide](#) at the MSDN Network.

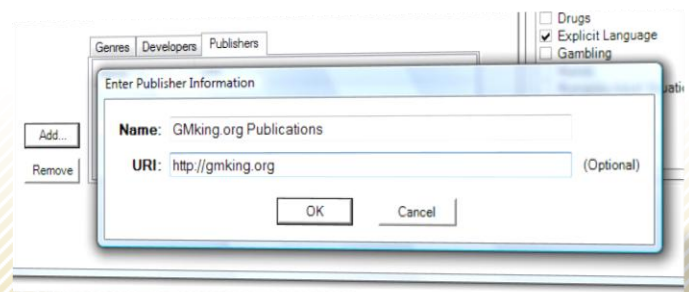
Entering the Information

The editor allows you to add all the images you need, all the ratings using the important rating boards, and different game metadata according to the different language settings, as well as the selection of supported languages, etc.

After opening the GDF Editor, create a new GDF File and choose to save it in the same directory as your game (the root directory).

Enter all the necessary information, including the developer and publisher information, directories, and URLs. The process is quite easy and the interface makes a lot of sense.

Note that all the information needs to be filled multiple times according to the amount of languages you have



Vista's Game Explorer: Revisited Cont.

chosen. You also need to realize that you may decide to choose **no languages**, which is, indeed, better; since that will make the GDF file only dependent of what is known as "The Language Neutral File", or, the LN file.

Creating the 'Resource File'

After all information and settings are finalized, a resource of the GDF must be built. On the "Resources" menu, choose "Build Script for Projects", and save the resource (.rc) file in the same directory as the GDF project.

Sherlock Jr.GDFMakerProject	1 KB	☆☆☆☆☆
Sherlock Jr.h	1 KB	☆☆☆☆☆
Sherlock Jr.rc	2 KB	☆☆☆☆☆

Note that a .h (header file) and .rc file have been created. The .GDFMakerProject is the project file itself, and will not be used in the actual process.


Now, the entire GDF data needs to be compiled either into an executable or a Dynamic-Link Library (DLL) in order to be used. For a Game Maker game, compiling the GDF into a DLL might seem more relevant.

A Step Sideways

One of the good things about game development with Game Maker is that you don't have to download several SDKs from Microsoft with incredible file sizes, however, if you want to advance into "commercial" or "commercial-like" game development, you'll have to eventually give in and download them.

Microsoft Visual C++ 2008

To compile the .rc file you just created into a .res file, certain tools and headers must be available. While the headers aren't imported directly from Microsoft Visual C++ 2008, the software is required to access and execute other header files required to compile the GDF file into a DLL.

 For More information on the installation of **VC++**, view the [Installing Microsoft Visual C++ 2008](#) article on GMpedia.org

The Windows SDK

Now, you **will** have to download the Windows SDK, since the .rc file links to certain header files that are present in the SDK (gameux.h). Make sure you have Microsoft Visual C++ 2008 installed from the previous step. The Windows SDK also provides us with an **RC Compiler** that we'll be using, and a **linker** that'll link our compiled .res file to a DLL or EXE.

 For More information on the installation of the **Windows SDK**, view the [Installing Windows SDK](#) article on GMpedia.org


The link above provides *very detailed* step-by-step information on how to download and install the Windows SDK. If the web installer doesn't work for some reason, then you're stuck with the full 1180MB version.

Use the instructions in the link above to also *fully configure* the **Windows SDK with Microsoft Visual C++ 2008**.

afxres.h

Other than gameux.h, afxres.h is also required by the .rc file. This file is originally included in the [Microsoft Foundation Class Library \(MFC\)](#), which does not come with the free version of Visual Studio. However, a similar file could be downloaded as a replacement.

The version of afxres.h we'll be using has been included in the Mozilla Songbird software, which is licensed under the GPL and therefore allows the use and reuse of its code. The link below provides full instructions on the downloading and installation of afxres.h as a header file.

 For More information on the installation of the **afxres.h** file, view the [afxres.h](#) article on GMpedia.org

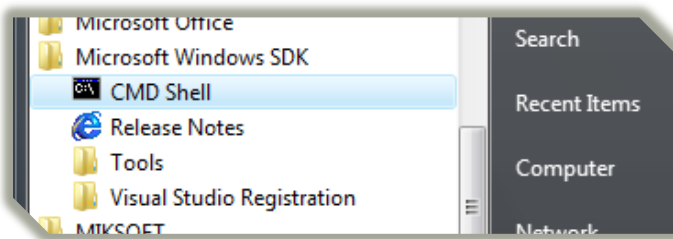
Make sure that the afxres.h you download is copied to a proper **include directory** that Microsoft Visual C++ 2008 recognizes. If you have installed the Windows SDK and configured it properly, then you can copy afxres.h to: "%Windows SDK%\Include\".

Vista's Game Explorer: Revisited Cont.

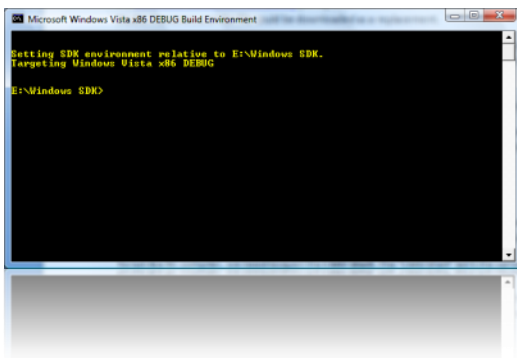
Compiling to a Resource File

To compile the .rc file into a .res resource file, we use the **RC Compiler** (RC.exe) found within the **Windows SDK** we downloaded earlier. The .res file will then be linked to a DLL or an EXE file.

To run the RC compiler, we need to open the **CMD Shell**. The "CMD Shell" isn't the same as the built-in "Command prompt" in Microsoft Windows, but instead is a similar tool that comes with the Windows SDK.



The CMD Shell can be accessed from the Start Menu, under the folder "Microsoft Windows SDK".



After running the CMD shell, type the following command inside:

```
RC "path/to/your/rc/file.rc"
```

In "Creating Game Definition Files", I created a .rc file under the title "Sherlock Jr.rc". The results of the RC command are shown below.

Linking to a DLL

To link a RES file to a DLL, multiple ways exist. Though for a professional C++ user, the .res file could be merged to an existing DLL s/he is using, the method we'll approach is much simpler: merge the .res file into an "empty" dll with no code. DLLs like that are known as "resource-only DLLs", this means that this DLL will not include any sort of executable code. To do that, we'll use **LINK.EXE**, a piece of software that also comes with the **Windows SDK**.

To use Link.exe, open the **CMD Shell** and enter the following command (write it as a single line):

```
link /nodefaultlib /dll /machine:ix86 /noentry  
/out:"path/to/the/dll.dll"  
"path/to/the/existing/resource.res"
```

All commands coming after "link" such as "/nodefaultlib", "/dll", etc. reflect the nature of the DLL we want to create. All these switches need to be included, but we'll explain important ones:

- "/dll" – informs the linker that the resource file will be linked to a DLL
- "/machine:ix86" – informs the linker that the compilation needs to be x86-compatible (32-bit)
- "/noentry" – informs the linker that no executable code shall exist in the DLL
- "/out" – specifies the output file

```
E:\Windows SDK>RC "C:\Users\Bashar Sharaiha\Documents\Investigator\Sherlock Jr.rc"  
Microsoft (R) Windows (R) Resource Compiler Version 6.0.5724.0  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
E:\Windows SDK>
```

Vista's Game Explorer: Revisited Cont.

```
E:\Windows SDK>link /nodefaultlib /dll /machine:ix86 /noentry /out:"C:\Users\Bashar Sharaiha\Documents\Investigator\gdf.dll" "C:\Users\Bashar Sharaiha\Documents\Investigator\Sherlock Jr.RES"
Microsoft (R) Incremental Linker Version 8.00.50727.762
Copyright (C) Microsoft Corporation. All rights reserved.
```

```
E:\Windows SDK>
```

The result of linking my own RES file into a dll is shown above.

As you can see, what I did was merge "Sherlock Jr.RES" into a **new dll** that didn't exist before; the dll I decided to create had the name "gdf.dll". Give any name you want to the DLL to be created. Remember to always include full paths to the RES and DLL files.

Adding to an Installer

Now comes the fun part: adding the actual GDF dll to the installer that contains your game! This could be done in multiple ways (only follow one of the choices):

- Adding it to an **MSI Package**. MSI Packages can be created with Install Shield, Orca, and other software, MSI Packages create installations that are installed by the "Windows Installer". Recommended.
- Adding it to an **InstallScript Installation**. InstallScript Installations are powerful installations that are created only using Install Shield. If you've got the money to buy a version of Install Shield, then why not!

Adding the DLL to MSI Packages

To add the DLL to the MSI package, we will use a piece of software called **Orca** that comes with the **Windows SDK** as an optional component. Orca is an MSI-editing piece of software.

Installing Orca

To install orca, run the **CMD Shell**, and type the following:

```
orca.msi
```

This command will launch the Orca installer. This MSI file already exists in default Windows SDK installations, so simply type the command, and an installer will appear so that you can install Orca. Full instructions on how to install Orca are available in GMpedia.org.



For More information on the installation of Orca, view the [Installing Orca](#) article on GMpedia.org

Creating a simple MSI

After successfully installing orca, you need to use **Installation Creation Software** to create an **MSI Package**. It is really difficult to use Orca to create the installation package from scratch, and therefore using an installation tool in conjunction with Orca would be best.

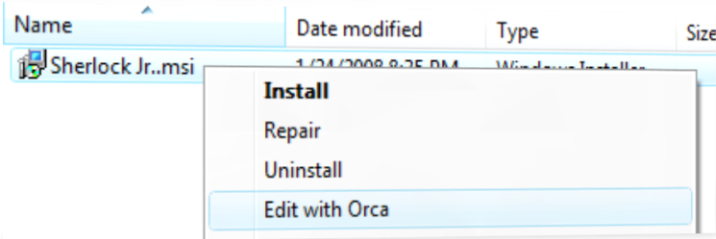
You could find a list of tools that create MSI Packages [here](#). Choose a tool, download it, and install.

The MSI Package created by the tool needs to include the **all of your Game Files that are needed to run the game**, in addition to the **DLL that contains the GDF Resource** and information regarding your game.

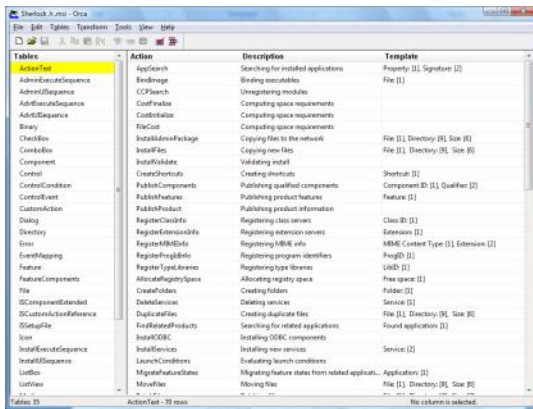
Editing the MSI

Afterwards, build the installation **as an MSI file** and edit it with Orca.

Vista's Game Explorer: Revisited Cont.



Now, after the MSI Package is open, you can navigate to different tables (different tables can be selected from the column on the left) and add different rows to these tables. To add a row to a table, selected the wanted table, then scroll down to the bottom and double click the empty row in the end of the table. This will present a dialogue that guides you through adding the row.



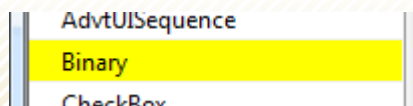
A total of **20 rows** needs to be added to the MSI package. These will all be explained in detail [here](#) (or the full article, [here](#)). We will only go through the process of inserting two fields (which are the only ones that need explanation).



For More information on the rows to be added, view the [corresponding article](#) article on GMpedia.org

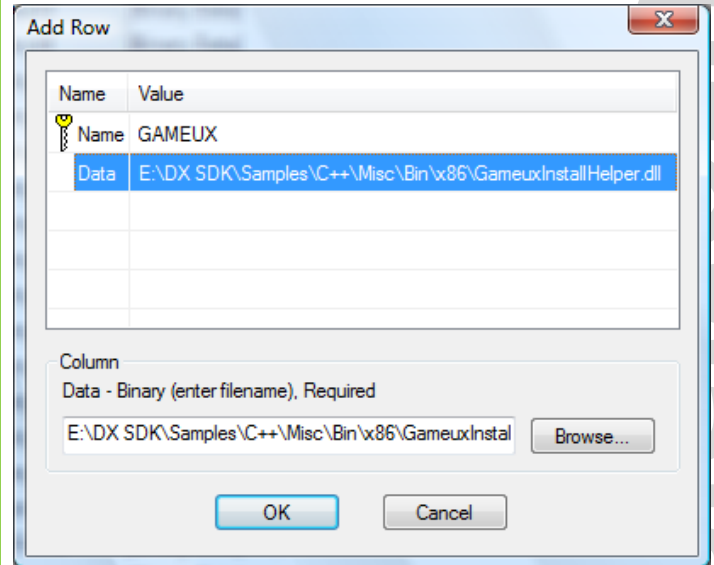
Adding the Rows

The full process of adding the 20 rows is described in the GMpedia.org article, I'll describe the process of adding the 2 most important rows here.



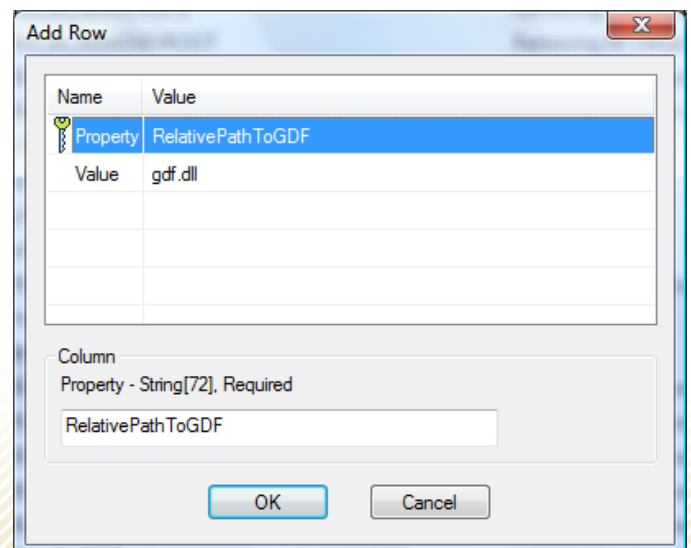
For the first one, the row **GAMEUX** needs to be added to

the **Binary** table. Select the Binary table, double click the last empty line, and enter following information:



The **GAMEUX** field describes the path to the "GameuxInstallHelper.dll" DLL file. This is included in the **DirectX SDK** in the directory "Samples\C++\Misc\Bin\x86".

Now, we will add the row **RelativePathToGDF** in the **Property** table. The value of this row should be the *relative* path to the GDF dll you used. This means, that it is the path to the DLL in respect to the installation destination chosen.



Vista's Game Explorer: Revisited Cont.

After all rows are added, the MSI package must then be saved, and... **that's it!** Your game is now fully compatible with Windows Vista's Game Explorer!

Adding the DLL to an InstallScript Installation

An InstallScript project is only editable in **Install Shield**, which is commercial software. If you're lucky enough to have a copy of Install Shield for some reason, then you could add the DLL to an InstallScript installation.



After creating the project, add GameuxInstallHelper.dll to your **Installation Destination**. This is included in the **DirectX SDK** in the directory "Samples\C++\Misc\Bin\x86".

Also, add **the GDF File you created** to the installation destination. The GDF File you created could be a DLL or an EXE.

Afterwards go to the **Installation Designer**, to "Install Script" under "Behavior and Logic", and then double click "Setup.rul".

Under `/* export prototype MyFunction(HWND); */`,

paste the code found [here](#). Make sure to update all of the data that is in Green to reflect the status of your game.



For the script to be added to the installation, view the [corresponding article](#) article on GMpedia.org

Next, simply create the installation, and you're done!

Resources

- [Game Definition File Editor](#), Microsoft Developer Network
- [Windows Game Explorer for Game Developers](#), Microsoft Developer Network
- [Orca.exe \(Windows\)](#), Microsoft Developer Network
- [Building Resource DLLs](#), Dolphin Map

Conclusion

I've tried to make this article as clear and detailed as possible, yet brief, straightforward, and most importantly, easy to understand. I hope the article opens many doors to commercial Game Maker developers, and I'm really, really sorry I made you download so many SDKs! In the end, we must agree: for a professional developer, it's worth it.

■

Write for MarkUp Magazine

[CLICK HERE](#)

IDs in Game Maker

By: Bart Teunis

All resources in Game Maker are referred to by their id. Not only the typical resources as shown in the resource tree e.g. sprites, sounds, objects, etc. but also data structures, files, surfaces, tiles and particle systems have an id.

This is not really useful when all resource IDs are known at the game start, but it gets more interesting when resources need to be added or removed during the game. And IDs have some other advantages. First of all, let's have a look at resource IDs in GM.

Resource IDs

When a resource is added in GM, this resource gets an id. Such an id is always a real number. This id is not unique. Suppose that in an empty game, a sprite is added. This sprite gets id 0. Next, an object is added. This object also gets id 0. Why? For each resource of a certain resource type, GM assigns id's starting from 0. So the first sprite gets id 0, the first object also gets id 0. An image of how the resource structure could look is displayed on the bottom of the page.

You can also see this in the resource names. `sprite0` has id 0, `sprite1` has id 1 ... And why is `sprite2` missing? That's simple. It has been removed from the list of sprites. But as

you can see, there's no longer a sprite with id 2. That's right, GM keeps incrementing the id counter, and also when resources have been removed. This doesn't cause any problems, since you'll (usually) refer to a resource by using its "name". A resource name can be seen as a constant. It is a variable of type real and it cannot be modified. So an expression like this is wrong and will give an error:

```
sound0 = 4;
```

You cannot assign a new value to a constant. You can assign a resource id to a variable, though. Actually that's what you've probably done before already:

```
/*assign the constant value sound0 (= 4) to the local  
variable mysound*/  
mysound = sound0;  
/*assign the value 4 to the local variable mysound*/  
mysound = 4;
```

Both pieces of code give the same result. Note that for sprites and backgrounds, `sprite_assign` and `background_assign` can be used to assign a different

Sprites		Sounds		Objects		...
name	id	name	id	name	id	
sprite0	0	sound0	0	object0	0	
sprite1	1	sound1	1	object1	1	
sprite3	3	sound2	2	object2	2	
sprite4	4	sound3	3	object3	3	
new_sprite	5					

IDs in Game Maker Cont.

variable to a sprite/background resource constant.

Now what happens when we want to add a new sprite with the change resource functions? `sprite_add` returns the id of the newly added sprite. As you can see in the above image, that sprite has id 5. So when you add a resource through gml, the id counter is incremented as well. The main difference here is this:

```
new_sprite = sprite_add("sprite.png",1,1,1,1,0,0);
```

`new_sprite` is not a constant and thus can be changed freely. So `new_sprite` now contains the value returned by `sprite_add("sprite.png",1,1,1,1,0,0)`, but after a new call to `sprite_add`, it will contain the id of another new sprite. This makes it very easy to add some resources and quickly perform some actions on each of them.

Nice uses

IDs can be used for nice things. We'll have a look at a few of those here, but there are definitely other uses. You can e.g. determine the number of items of a certain resource. Each resource has an id larger than zero and most likely smaller than 1000 (or maybe, in rare cases, 10000). Combine this with the `sprite_exists` function (or any `<resource>_exists` function) and you have a nice loop to check for the number of items of a certain resource. Let's have a look at the code that does this for objects:

```
var n, total;
n = 0;
total = 0;
repeat (1000)
{
    if (object_exists(n)==1) total+= 1;
    n+= 1;
}
return total;
```

After the loop the variable `total` will contain the number of objects in your game, assuming that there are less than 1000 of them (remember that GM doesn't decrement the counter when you remove a resource).

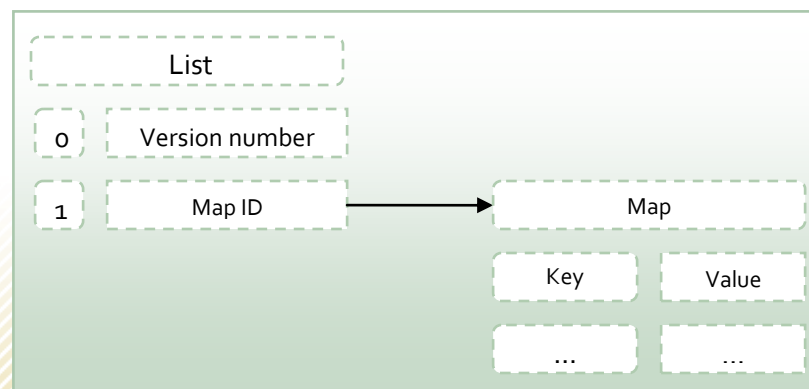
You can also combine IDs with GM's built-in data structures. That way, it is possible to create nested structures: data structures that contain data structures (and other things that have an id). Let's say you need to read data from a saved file. This will be done by passing a filename to a script. The file contains a version number and some attributes (with name and value). All this info cannot be returned in a single variable. Well, theoretically, it can be put in a long string, but there is an easier and neater way to do this. The solution is to create a data structure (that might even consist of other data structures), put all the information in it and return its id. The easiest way is to use a list that contains all the values. You should determine beforehand which list id will hold which value. For the above example, we'll need the following data structures:

- A list: this list's id will be returned by the script
- The first value in the list will be the version number
- The second value will be an id of a map structure. This map contains all attributes with their name and value.

So the new structure we just created is shown on the bottom of this page.

To create the list and map, the following code could be used:

```
data = ds_list_create();
ds_list_add(data,1);
ds_list_add(data,ds_map_create());
```



IDs in Game Maker Cont.

Assuming that the data has been put into the list, it can be retrieved as follows:

Version number

```
ds_list_find_value(data,0);
```

Any attribute

```
var map;  
map = ds_list_find_value(data,1);  
ds_map_find_value(map,key); //key can be any  
existing attribute name
```

And voilà, you have the data structure and some simple pieces of code to get all values. That's all you need.

Unfortunately, this can give a problem when many of these data structures are created. It'd be best to destroy the data structures when they're no longer needed. In the above example, it's very important that the map is destroyed first. That's quite logical to understand. Suppose the list is destroyed first. Okay then, time to destroy the map. But where is the map? By destroying the list, we lost the id of the map. So the map still exists, but you won't be able to find it back. That's just a waste of memory. So the map should be destroyed first. The code

looks like this:

```
var map;  
map = ds_list_find_value(data,1);  
ds_map_destroy(map);  
ds_list_destroy(data);
```

That's all. Everything's gone now.

GM has a nice built-in system for resources, but when you want to use the advanced resource functions (add, change, replace, delete), you'll have to do the resource management yourself.

The example shown above is just a simple example of how to nest data structures. It doesn't have any practical use. When you build your own data structures, you should see for yourself which data structures and variables you need.

Conclusion

So that's all for resource id's. I hope this article has taught you a bit more about them.

■

You ask, MarkUp Answers: Looping MP3s

Question

I often times find MP3s I would like to use in my games, but Game Maker doesn't seem to be able to play MP3s in a perfect loop. Is there any way around this, such as an extension? Your help, or anybody's for that matter, would be greatly appreciated.

Aaron

Answer

Eyas: Game Maker uses DirectX for sound playing, and it sometimes has problems with detecting the true length

of the sound track. If `sound_loop` doesn't work, then you could always try one of the following DLLs for playing sound, they have loop features which could be handy:

- [SXMS](#), by Shaltif
- [Audio](#), by Andewmc
- [Super Sound System DLL](#), by tsg1zzn

For your case, I'd recommend using **Saudio**, since it is relatively lightweight and easy to use. Saudio supports MP3 and can play them looped. Perfect timing remains difficult but is better achieved with such DLLs.

Eyas Sharaiha ■

Extension of the Month

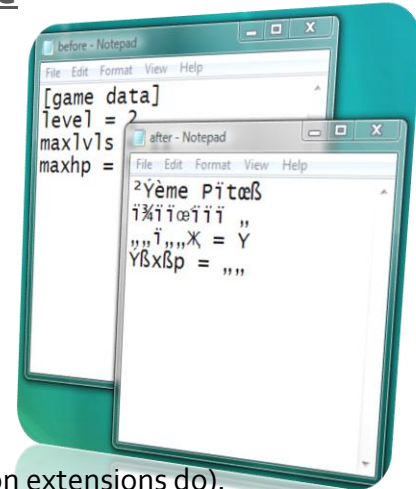
By: Schyler

Ultima Crypt Encryption Powered By GMBase

For those who don't know what 'Encryption' is, encryption is changing the contents of a file so that it is harder to understand (and then later, when it comes time to read the contents of the file, decrypting it). Usually people would encrypt their work, maybe a save file to a game, and images, sounds or data ext. so that other people can play the game (however not extract the data from it easily, or edit their progress through the game). Encrypting will usually require having to go through to each individual character and changing it accordingly from a set database of characters (i.e. change 'a' to 'e', change 'o' to 'g') however this is a very weak encryption technique.

The smart part of this extension is that it uses a technique similar to the NC4 technique (where it takes a key, or a string and uses it to encrypt the file). This means that even if the person wanting to decrypt the file has the script they must also know the key that the game uses.

Now this extension sets itself apart from others for many reasons, it's easy to use, it has passwords but I simply like it because it's compact and doesn't increase the size of the file (like some other



encryption extensions do).

In the short time I had playing with this extension I found that amongst another extension it seemed to compile the final product of the encrypted file in about ½ the time. This would make it very efficient to encrypt loops of data at the same time.

Personally, this extension will in future prove good to me and I recommend it to other people who want to keep their game save files or data safe from "off-line hackers". Don't forget if you use this extension to give credit to Strider6282.

gmbase.cubedwater.com/?page=extension&id=111

■

Extensions

Advertisement

XSOFT GAMES

PURE GAMES. PURE POWER

XSOFTGAMES.COM

Advertisement

Script of the Month

By: Eyas Sharaiha

Resource Management Powered By GMLscripts.com

Resource Mapping

For this special issue, we'll be having an entire group of scripts for the issue's "Script of the Month" section. These are resource management scripts that map resource names into a `ds_map` data structure.

The set of scripts was created by **paul23** for GMLscripts.com. xot explains the importance of this set of scripts by saying:

"Many times Game Maker users want to access a sprite or some other resource by its name (as a string) rather than its index. Normally this is not possible (or at least not economical) without resorting to `execute_string()`, which is something that should generally be avoided."

Script: `map_object`

```
/*
** Usage:
**     map_objects(ds_map)
**
** Arguments:
**     ds_map     map to which all objects are
Loaded
**
** Returns:
**     nothing, but fills the map with {key=name,
val=index} pairs
**
** GMLscripts.com
**
*/
{
    var no,i,ds_map;
    ds_map = argument0;
    no = object_add();
    object_delete(no);
    for (i=0; i<no; i+=1) {
        if (object_exists(i)) {
            ds_map_add(ds_map,object_get_name(i),i);
        }
    }
}
```

The following script will immediately map all of the existing objects (whether or not instances of these objects are available) into a `ds_map` of your choice. The `ds_map` must be already created.

Similar Scripts

All similar scripts are available [here](#), these include:

- [map_backgrounds\(ds_map\)](#)
- [map_fonts\(ds_map\)](#)
- [map_objects\(ds_map\)](#)
- [map_paths\(ds_map\)](#)
- [map_rooms\(ds_map\)](#)
- [map_sounds\(ds_map,ex_sound\)](#)
- [map_sprites\(ds_map\)](#)
- [map_timelines\(ds_map\)](#)

Usage

The `ds_map_find_value()` script can be used to import a resource name from the `ds_map`. An example of this would be to use:

```
index = ds_map_find_value(mymap,name);
```

In the script above, `name` of course is a string.

Conclusion

The scripts by **paul23** are certainly very useful. Any resource might need to be referred to as a string, especially in cases where the identity of the resource changes according to user input or other events. This will make referring to objects easier.

YoYo Games Competition Winners

By: Dougie

The **YoYo Games Winter Competition** was the first Game Maker Community specific competition of YoYo Games, and, unquestionably the largest competition ever held for the GMC. The winner of the First Place \$1000 cash prize for the competition was Frozzd, created by Jesse "2Dcube" van Brux. I have conducted an interview with 2Dcube and Ultimortal, creator of the second-place prize winning game, asking them about their achievements.

Interview with 2Dcube

D: What inspired you with the concept of Frozzd?

Jesse: As is obvious, my inspiration came from Super Mario Galaxy. I thought running around on planets in 2D would be fun. It was fun, but I couldn't do much with the concept, and I couldn't think of something else for the character to do that wouldn't be too difficult to make or too frustrating to control. So I thought I should give him (or her of course) a buddy that would follow you and that had special abilities you could use to solve puzzles. Not long after I thought of making more of these creatures so you could build up an army, and the concept came together.

D: Did you expect it to be such a success?

Jesse: I did expect people to like it but I wasn't sure about the difficulty and length of the game. I'm glad everyone seems to enjoy it. I really believe in the originality of the game, the visual style, and the cute Mubbly creatures, but it's difficult to tell beforehand whether other people will think the same way.

D: Will this be the end of the Mubbly creatures or will we see them again in a new game?

Jesse: I don't think we will see them again, unfortunately.



D: What do you think of the other winning entrants?

Jesse: Garden Gnome Carnage was my favorite of the entries because of its quirkiness, originality, and style of control. I think Ultimortal could have added another mode or maybe specific missions, to add more variety. Granny & Snowmen is a very well made game with a lot of content, but it's just not really my type of game. It did very well in the community so I'm sure it deserves a prize.

D: How long did it take you to make Frozzd?

Jesse: About 4 to 5 weeks. Especially in the last week I worked many, many hours on it.

YoYo Games Winter Competition Cont.

D: Who would you choose to win?

Jesse: Apart from myself? Haha. I would choose Garden Gnome Carnage. Besides that, I also thought White Sheet was pretty well done.

D: What are you going to do with the \$1000?

Jesse: I'm not sure yet. This year I will probably buy a PS3, but I think I will wait a few months. So I guess I won't spend it on something immediately.

D: Do you have any advice to be successful game designer?

Jesse: I think it's a good idea to look at other games when you play them, and try to understand why exactly the creators made certain choices. In good games, every single thing is there for a reason. I think every designer has their own ways of creating and solving problems, and especially in the case of game design it's difficult to give a clear answer to this question. Experiment a lot and try things out you've never seen before.

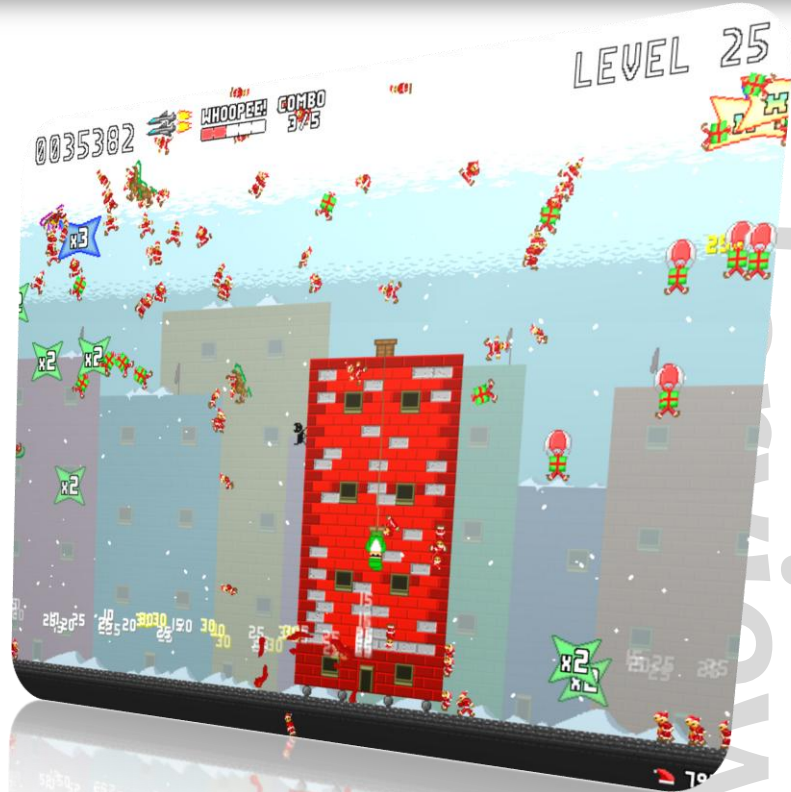
Conclusion

Frozdd is indeed a well done game that deserved the \$1000 prize it received. I truly hope that the interview was informative and beneficial.

Interview with Ultimortal

D: What inspired you with the concept of Garden Gnome Carnage?

Ultimortal: Some two years ago, two other guys and I were going to design a game to be made in five weeks. One of the ideas during the first brainstorm was something involving garden gnomes, and then we added a bungee cord, a house, and the house would also have wheels and be remote-controlled. Then one of us said, no, this is getting ridiculous, let's be serious now. The idea of



the game still lingered in my mind, though, and I later built a prototype in GM6. Some friends and I added a lot of crazy stuff to the game, including the Benny Hill theme for the music. It eventually became something of a small hit in my school.

When we saw the YoYo Games competition, my friend suggested we updated GGC for it. So I remade and improved nearly all the graphics, added a lot of features and polish, and recorded the voices and sounds with the help of some classmates. SoulEye also wrote an in-game song for it, and I made the tutorial video. I also made the game easier, to make it more accessible.

D: Did you expect it to be such a success?

Ultimortal: No, not at all. At first we only did it for our own amusement, and then we were surprised people found it fun to play. When the game was updated and released as an entry for the YoYo Games competition, it got pretty famous and eventually took my server's entire bandwidth with it. It's still surprising it did so well.

D: Will this be the end of the Garden Gnomes or

YoYo Games Winter Competition Cont.



will we see them again in a new game?

Ultimortal: Nah, like most of my games GGC is a one-time concept. Instead of wanting to make a sequel, I'd rather update and tweak it.

D: How long did it take you to make Garden Gnome Carnage?

Ultimortal: It's hard to say. After it was first envisioned, I worked very seldom on it. I'd say it took about three months in total, with most of the focused work going into the version released for the competition. The original game looked pretty basic, so I needed to tighten up the graphics.

D: What are you going to do with the \$500?

Ultimortal: The musician gets his share, but other than that we baked a GGC-themed cake to celebrate. It looked

like the house in the game, with a marzipan gnome tied with a candy rope at the top, with wheels and a chimney of chocolate. It was pretty cheap to make though, and the rest of the money will probably go to paying off my debts.

D: Do you have any advice to be successful developer?

Ultimortal: Well, to begin with, don't be lazy. Once you've decided to become serious about game making, do your own thing and find your own style. I like to take many different game concepts and put them together, and if it works, I start building a new game around that. There's no shame in being inspired by games you've played before, but cloning them completely is not a good idea. Unless you make games for the sole purpose of making money, of course, then cloning the most popular puzzle and platform games may get you a long way...

If you're not confident in your skills, be it coding, drawing or writing, the only way to improve is to practice and listen to feedback. I've made games since '95, and BOY were they terrible back then, but nowadays I strive to improve all the time. It really goes without saying, but you can never expect perfection at first.

Conclusion

Indeed, Ultimortal gives us some good words of advice on how to become a successful developer. It wouldn't hurt following such words of advice, who knows, some of you might be the future winners of these competitions!

Conclusion

So that was it! I hope you enjoyed the casual interviews with the developers of the two winning games in the first ever YoYo Games Competition: the Winter Competition.

Game Maker Experiences

By: Stephen H.

Many of us has went through many experiences that have allowed us to become better game developers, and I'm here now to share some hints and tips that I have learnt from my own Game Maker Experiences. I'll be talking about two things in this article: Releasing your Game, and Managing Teams.

Releasing your Games

Here are some different stages that games and software usually go through are these:

- Alpha Versions
- Beta Versions
- Release Candidates

Now I'll tell you what these mean:

- When you release an Alpha, you should still be actively developing your game/software and implementing many features
- In the Beta stage, you should usually have most features implemented. During Beta's, people should be testing to find any bugs currently in it.
- In any Release Candidate (RC), the game/software is in the form it will be released in. No new features will be added, and all that should be happening is bug-fixing and testing.

I hope that you learned something from this short article on what these mean. Now you can release your games/software under the correct stage names. Good luck!

Managing a Team

Are you the leader of an inactive team? Maybe the members are just lazy, in which case you may want to find new members, or perhaps you could do something that you aren't doing. To work best, teams need to be active,

and members need to be able to communicate with one another. They also need members with good personality, ones that can get along together and not fight.

One great way to communicate is through forums, the best of which are IP.Board. These do cost money, about 200 US Dollars in fact, but they look more professional. If you can't afford that, I suggest going with PHPBB (www.phpbb.com), which is free. You also will need a good host for forums, one that has PHP and MySQL. You can look at my article in this issue titled "Good Webhosts".

Another way to communicate is via Instant Messenger programs such as Windows Live Messenger. If you only have a team of a few people, all of which can be online at about the same time, this may work better than forums as you are talking to them live.

To keep your members active, assign them all different jobs and make sure they know exactly what they should be doing. Remember not to get mad at them if they become inactive, as that will probably only make things worse and they will quit. Instead, send them an email or a PM that is asking for them to be active. Send something like this:

Hello <name here>!

We have noticed that you have been inactive recently, and we want you to be more active. If you have been on vacation or unable to talk to us lately, that is fine, we just want you to be back so we can work on some great games!

Thanks,
<your name here>

Game Maker and Windows Vista

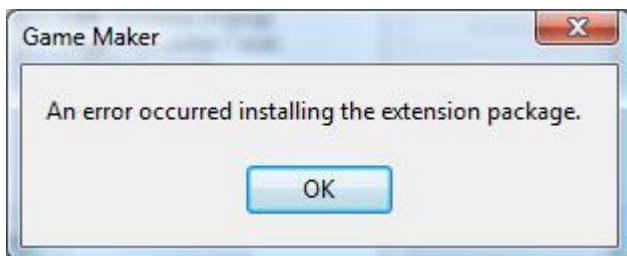
By: Thomas

Running Windows Vista can be a rewarding experience on multiple levels; however it introduces some problems when running a program such as Game Maker. Starting a freshly-installed version of 'Game Maker' would usually show multiple error messages.

When Game Maker is opened, the following dialog is shown:



When installing packages, a similar error might also appear:



The Solution

You can run Game Maker as an administrator and it works fine! First, go to Control Panel (Start>Control Panel). Select add or remove user accounts. You should get a pop-up message, click on 'continue'. This already means you are an administrator but to double check, check under your user name for Windows and it should say 'Administrator'. If not, just get the administrator of your computer/network to grant you access in the next few steps or you can try changing your account to an admin by clicking on it and clicking change account type.

Now once you have confirmed you are an administrator, you need to go to where you launch Game Maker 7 from. It can be the main .exe file that is where GM is installed or it can just be a shortcut on the desktop you use. Either or, this will set it up for the whole lot. Right click the file and click Properties. Click the Compatibility tab.

Now, check the check box that says Run as Administrator next to it. Click Apply then Ok.

The UAC

You know that Administrator Confirmation pop-up thing that always appears for some things? And now that Game Maker requires an admin to run it, you have to wait for the pop-up and click continue before Game Maker opens. You can optionally turn off the UAC by going to the Control Panel, type "UAC" in the search bar and press enter, the 'Turn User Account Control on or off' will be loaded, and you can now uncheck the box, click okay, restart your computer, and you'll be free of the UAC.

The UAC is a good security feature of Windows Vista which might be found to be annoying by more experienced members. It is generally not recommended to turn it off, but you can do so by following the instructions mentioned.

Conclusion

And now you're free! If you ever have problems with other programs, try setting them to run as an administrator first and see if that fixes the problem!

By: Philip Gamble

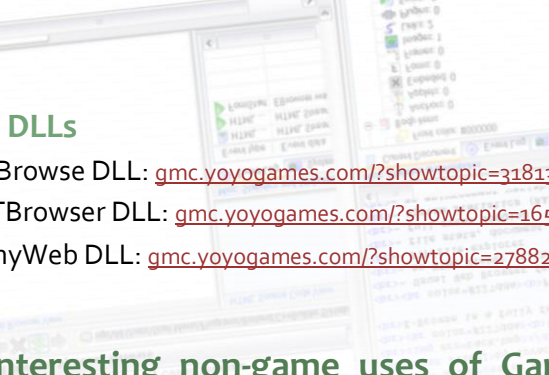
The screenshot shows the E-Browser v3.x application window. The title bar reads "E-Browser v3.x by Covac Software". The main content area is titled "HTML Source Code View" and displays a list of features:

- << Usual Web Browser features
- << An HTML Viewer (with Syntax Highlighting)
- << FTP/File explorer
- << file stats, document object view
- << full event logging
- << system statistics (RAM, Page file, environment variables)

Below the feature list, there is a section titled "Misc. Statistics and Attributes" with tabs for "Event Log" and "System". The "System" tab is active, showing a table with columns "Event type" and "Event data". The table contains the following entries:

Event type	Event data
HTML...	HTML View
HTML...	HTML View
FormStat	EBrowser v3.0

The interface also includes a menu bar at the top with options like "File", "Edit", "View", "Tools", "Help", and "About". A sidebar on the right shows a tree view of the current document's structure, including "Body Items" and "Form Elements".



Browser DLLs

- E-Browse DLL: gmc.yoyogames.com/?showtopic=318136
- DTBrowser DLL: gmc.yoyogames.com/?showtopic=165786
- TinyWeb DLL: gmc.yoyogames.com/?showtopic=278823

er interesting non-game uses of Game
ker

Browser DLLs

- E-Browse DLL: gmc.yoyogames.com/?showtopic=318136
- DTBrowser DLL: gmc.yoyogames.com/?showtopic=165786
- TinyWeb DLL: gmc.yoyogames.com/?showtopic=278823

Other interesting non-game uses of Game Maker

WallShare

This program allows you to share desktop wallpapers and

More than Just Games Cont.

take your pick from those uploaded by other members.

gmc.yoyogames.com/?showtopic=338808



WallShare

Zipped Folder Compression/Decompression

At just 50kb this achiever even supports the creation and extraction of password protected zips.

gmc.yoyogames.com/?showtopic=326729



Graphical Programs

Game Maker and making graphics don't normally go together. But with Textbar Maker 4 they can. This small

utility can be used to effortlessly create the professional looking 'userbars' that adorn the signatures of plenty of forum users. yoyogames.com/games/show/10631



Conclusion

Hopefully this article has shown that Game Maker can be used to create many different types of application, and not just the normal range of games that its name suggests.

Remember: The possibilities of Game Maker far exceed games.



Advertisement

Say goodbye to broken affiliate links

Adiós

Au revoir

Auf Wiedersehen

Ciao

Tot ziens

- Over 950,000 banners shown
- Over 8,500 banner clicks
- 100 members
- 95 banners in rotation

Over 35 Game Maker websites are already using the free

Game Maker Affiliation Service

to exchange banners with other websites

Members include:

MarkUp, GameMakerTech, GMTV,
RPG Revolution, GM Tutorials,
Snowmoon, GM Edge, GM Helper...

Advertisement

Games I Would Like to See

By: Rick Slick

There are **far too many people** attempting Pokémon, Guitar Hero, and Zelda clones with Game Maker. Yes, that's right... you know who you are – get your face out of your Pokédex or whatever and listen up. Sure, maybe it's safe to stick with what you know... but whatever happened to good, old-fashioned creativity?

Here are some awesome game ideas that people should consider for their next project:

Xylophone Hero

Sure, everyone wants to be the lead guitarist in a rock band, churning out the solos and wailing on the whammy bar, but what about the glamorous, fast-paced life of a xylophonist? Play along with xylophone favorites such as "Chopsticks" and your favorite Christmas music as you progress your character through different grade levels of elementary school holiday concerts as the difficulty increases. The rest of your pre-pubescent band will be using hand bells, triangles, and cymbals. Hit too many wrong notes and groups of parents will get up and leave early... completely fail and even your own parents will depart the auditorium, heads hanging in shame. Do you have what it takes to be the Xylophone Hero?!

Counter-Strike: Pokémon

Poké-this, Poké-that... hasn't everyone had enough already? It's time to send in the elite commando soldiers to eradicate those miserable things before they take over your lives. Play this action-packed first person shooter online with your friends as either the Counter-Pokémon forces, armed with silenced M4 Carbines and H&K MP-5s, or play as the Pokémon legion, armed with... uh... whatever Pokémon use to defeat each other, I don't know. Catch some pervert in the act of breeding Pokémon and

take them all out with a frag grenade for extra points!

Grand Theft Zelda



Link is a legend in his own time, but what if he were magically transported into modern times, where he would have to quickly adjust to life on the streets in rundown

CH's Mouselook

This extension is a smooth mouselook system that is meant for 3D applications or games.

When direction of view is changed, it is done very smoothly according to mouse motion, also limiting the upwards and downwards directions of view.

As with most extensions, CH's Mouselook is only available in GEX format and therefore only compatible with Game Maker 7.

The extension uses no rounding functions (floor, round, or ceil) to approximate directions and mouse positions, but instead uses appropriate exact values.

The extension could be used for free, just remember to credit its creator, Carl-Henrik Bernhoft.

Get it now:

gmc.yoyogames.com/?showtopic=308664

Games I Would Like to See Cont.

Liberty City? Violently swinging his sword at innocent pedestrians, Link must obtain a set of wheels, avoid the police, and complete various missions leading up to the rescue of Princess Zelda, who has been forced into prostitution by the city's top crime lord. Let's see how long Link and his sword last against a bunch of thugs with shotguns!

Super SOCOM Bros



If you're going to go after the Princess and face Bowser and his army of turtles and mushrooms, don't you think you'd put together a team of elite commandos to go do this instead of just asking your fat plumber brother Luigi to tag along? Utilize state-of-the-art equipment and weaponry as you send in your team of Navy SEALs to infiltrate Super Mario World, rappel into the castle where the Princess is located, clear the area of hostiles, and rescue the hostage. Don't forget to have a member of your team laser-designate Bowser's castle as a air-strike target after you leave!

Madden '08 Tank Battle



There's probably nothing less exciting than two tanks lobbing shells at each other on some sort of playing field with all sorts of random walls someone constructed, but what if this happened in a stadium, with a roaring audience, instant replays, and an announcer covering the play-by-play? Sure, maybe you won't get John Madden to do the voiceover work, but if you and a friend spend a good weekend recording a few hundred .WAVs for the play-by-play announcements, you could really spice things up. Make sure you state the blatantly obvious, too, like "You know, Matt, those shells can really damage your tank sometimes!" or "That's right, Bob, his turret sure has that 360-degree field of rotation going for it!"

Those are just a few examples of some games I like to see developed with Game Maker, instead of the same old and tired re-makes. Let's get creative, everyone! I wish you the best of luck coding your next Game Maker project – except if it's another Pokémon game, of course.

GiiMote

GiiMote is an open source Game Maker Extension licensed under the [Ms-PL](#) that adds support for the Nintendo Wii Remote (unofficially, "Wiimote") to systems with compatible Bluetooth devices.

The extension supports many aspects of Nintendo's innovative input system including extensions such as the Nunchuck and Classic Controller, the vibration motor, the infrared sensors, buttons and joysticks.

GiiMote was developed using Microsoft's .NET Framework technology and therefore requires the .NET Framework (Preferably version 3.5) to run. You can read more about .NET technologies [here](#).

For those of you interested in C++ and .NET development, you can check out the source code anonymously over http.

For the SVN, you can check out:

<http://giimote.googlecode.com/svn/trunk/> giimote-read-only

Get it now:

code.google.com/p/giimote/downloads/list



GiiMote
The Wii Remote Extension



<http://giimote.greenmangames.vze.com/>

The Default Message Box

By: David Yates

Imagine this: You're playing a GM game, and enjoying it greatly. Then, all of a sudden, the game freezes up, and you are confronted with an ugly grey message box, full of badly spelt words which form poorly-constructed sentences...

After recovering from the shock you received when you first saw this box, you read the text. It hurts your eyes, but bravely, you struggle onwards. And then, when you finally get to the bottom, you find that you have to choose from three options. As the game has been controlled by the keyboard thus far, you try fiddling with the arrow keys. Nothing happens. So, you reach for your mouse. You select an option, and you pray that it will allow you to return to the game.

But alas, you are confronted by yet another message box. Fortunately, this one is an ordinary "OK" message box, so you just press enter after reading it. And then another one comes up. You read it, and close it. Yet another pops up. Once more, you read it and close it. By now you're really getting irritated and just want to go back to the game. So you skip reading the messages and just hold down enter. Message box after message box flashes on your screen. Just as you are on the brink of hopelessness, the last one disappears and is not replaced. Stunned by disbelief, you sit still for a moment. Then you realize the game is back. You reach for the controls, but you are too late. Another message box comes up. This one reads: "GAME OVER".

I hope that the above illustration gives you at least a small taste of the pains of the default message box. Sure, it may be easy to use, but is it worth it? Is your convenience really worth the player's loss of immersion?

The Solution

Below are a few rules that I have figured out for the use of the default message box. Of course, not using it at all is the best option, but you can negate its negative effects somewhat by following these rules:

1. Always customize the box. For those who don't

know, `message_background()` and `message_button()` are the main functions involved in this. For more information, see the GM manual.

2. Never use the box in-game. By this I mean that you should never use the default message box to tell the game's story, show characters' speech, let the player choose in-game options, or anything else that happens in-game. Acceptable uses are in menus. This is because menus are usually static, and having a message box pop up and freeze the game won't interrupt the game's flow.
3. Avoid using "Yes", "No", and "OK". It makes the game look better if you use alter those texts depending on what the message box in question's function is. For instance, you can say `show_message_ext("The moon is made of cheese.", "True", "", "False")` instead of `show_question("Is the moon made of cheese?")`. You can also say `show_message_ext("Get Ready!", "", "I'm Ready", "")` instead of `show_message("Press OK when you're ready.")`.

Conclusion

In conclusion, you should avoid using the default message box, but, if you absolutely have to use it, you should use it in places where it won't interrupt the player, and you should make sure it looks as different as possible to its default grey self.

■

YoYo Games

By: Veeti Paananen

What would make the site much better...

While the YoYo Games website could be seen by many as effective and punctual, it is, in fact, lacking in many, many directions. I am not quite satisfied with the YoYo Games website and have therefore decided to direct some *constructive* criticism at the site.

What's wrong?

I have written a complete list of necessary improvements to YYG.com. I will divide these arguments into multiple categories, "Profile", "Main", and other general criticisms about the site.

Profile

As YYG.com is the official community website for GM-made games, it should be clear that it needs a complete user system. But in this case, it does not: it is lacking in many directions.

1. While this isn't exactly an issue related to the user system directly, the **lack of moderation** is worrying me: there are a lot of flame fests on users' pages and imposter profiles keep on coming. No matter how many reports are sent, nothing seems to happen. Either YYG's staff is taking one hell of a New Year vacation or they just suck at checking the report queue.
2. The **user shout boxes** or **game comments** are not **user-moderation** possible. While reporting is possible, there is no karma system (such as YouTube's new commenting system) and users can't even remove comments on their own profile or games.
3. The user **profiles** cannot be **customized** almost at all: being a "web 2.0 gaming website" (as they claim it) means some sort of customization. But no, you throw in your age, description and some other basic user info and that's it. Let's say that you don't have favorite games in YYG: your profile still shows it. It'd be convenient to be able to hide it from showing up.

Gaming

YYG.com certainly has made a lot of effort on the side of the games, their upload and playing. Their InstantPlay mechanism certainly has helped on the amount of games tested out: no more manual downloading, it does it for you. But deep down, this system is still flawed in many, many ways.

1. As **Windows Vista** came out, users quickly discovered that Game Maker 6.x-made games would not run on their computers. Obviously this caused some kind of outrage, but fortunately after a short time, an official GM6 to Vista "upgrader" was released, allowing GM6 made

executables to be converted to Vista-compatible games. YYG.com lacks any indication if the game will work with Vista. They should auto-detect a GM6 signature, for example, in the game and show a warning to Vista users: or just simply demand game uploaders to do it themselves: GM6 or not (or made Vista-compatible)? **Note:** Instant Play will make GM6-based games compatible on Vista; but some users might prefer to download it instead: they'll be greeted by a nice "game.exe crashed" dialog box.

2. Although this is a minor point, InstantPlay needs some **additional improvement**. For example, users should be allowed to set another temp directory where it should download the games to.
3. The game pages should also allow **maximum customization**: maybe make it more attractive than the typical white background - green/gray content? Custom backgrounds would be nice, so would be custom text boxes (instead of the small Description box). The navigation is flawed. I doubt a user looking for games to play would want to go through hundreds of pages to find a good game. Perhaps a randomizer with some extendable search queries (e.g. I don't want to play Catch the Clown!) would help.

And the rest

1. **Their forums. There's the GMC:** what do we need a pathetic clone of it for? This is like decentralizing the community. There's the "YoYo'ers" and "GMC'ers". It doesn't make sense: there's always space for a YYG section in the GMC.
2. Their **lack of communication** with the community. Yes, we've got a lot of announcements, they even have a "glog", but their communication with "us" just seems so stiff.
3. YYG still hasn't used their full power to **improve the state of GM**: much more could be done.

Conclusion

I hope I've established some good points for you all to consider. YoYo Games is an incredible opportunity for Game Maker as a platform and its community, but it needs to be improved in several ways before we could look at the YoYo Games transition as an improvement.

Space Explorer Deluxe

By: Jelle Straatsma

Hamburgerspel.com

Space Explorer Deluxe is a scrolling shooter game by Erik Leppen. Set in space, you are required to control your space shuttle and destroy all enemies in your way, ranging from small, easy, and insignificant U.F.O.s to larger and more dangerous level bosses.

Description

When you start the game you will begin in a menu. The menu looks nice, but there are only two buttons: Play and Quit. I always read the help file before I play a game, but there isn't one so starting immediately is the only thing you can do. When you press Play the game begins right away.

In this scrolling shooter you control a space shuttle and you have to shoot everything that stands in your way. The basic shooting goes automatically which is a very good thing, because you want to continue shooting the entire game. You control the space shuttle with your mouse, it moves after your cursor. First, your enemies are very little U.F.O's. They don't shoot at you and they are easy to destroy. After a while more different enemies appear, some are stronger, some are faster and some are bigger, there are about 50 different enemies, all with their own beautiful graphics. There are even five bosses which you can shoot piece by piece.

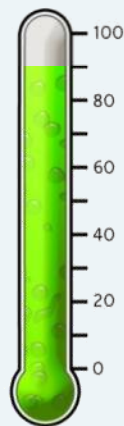
When you complete a level you can use the money that sometimes appear when you kill an enemy to buy upgrades and extra weapons. As you start, you can only improve basic things, like more firepower or more agility. When you advance in the game can buy more things, like side rockets and bombs. This adds a bit RPG stile in your game and you can choose between different tactics.

The music and sounds are very good, it's not distracting and not annoying. It fits perfectly and adds a bit more feeling to the game. The particle effects that are used in the game look beautiful. You can turn them off but you shouldn't do it, unless the game slows down when particles are used.

The complaint very people had is that the game is very difficult. I had the same problem the first try, but when you play it a few more times you keep getting better and better and eventually you will finish it. It will take you some hours, but because of the variation in enemies you won't get bored.



Ratings



Graphics: 5/6

Sound: 5/6

Gameplay: 6/6

Design: 5/6

Developer: Erik Leppen

Version: 1.0

Game Maker: Version 6.1

Instant Play:

Get it now!

Pyramids

By: L4nt0m

As most of us would know by now, the “Ancient Civilisations” competition, the second competition held by YoYo Games so far, is under way and already the games are flying in. One such game is ‘Pyramids’. As the name suggests, the game revolves around the ancient Egyptian civilisation, but in particular building their Pyramids.

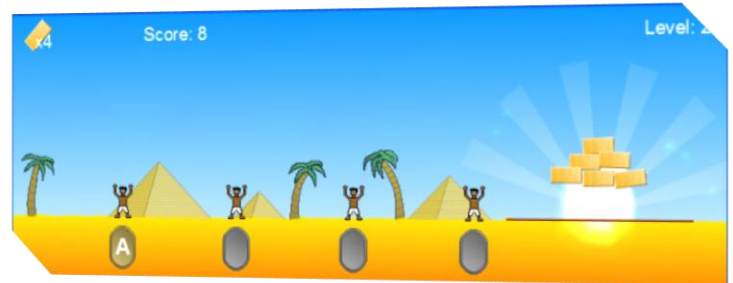
Description

The creator takes a cartoony approach to his latest game, adding smooth and simple graphics, gameplay, and a neat interface. The general consensus, around the community has been reasonable, most people are in agreement that it is a “nice game, but very simple”. Other mixed emotions about it include that it is both “funny and difficult” and even that it will “win da comp”.

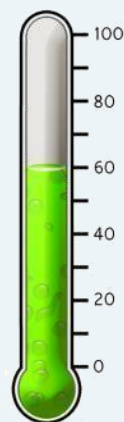
The game opens up with a well-presented menu, the basic instructions on the left, gaming options in the middle and the current high scores on the right (dictated by online, uploaded results) to encourage some competition. When you start the game, you are pitted straight into the action on level one, the game quickly opens out in front of you and becomes steadily more difficult, unlocking new characters to control as you progress, until you inevitably

lose all of your lives and are forced to start again.

The creation is very simple, and neat, and as such it makes for a nice game to enjoy at a break time, but it is let down by a lack of replay value and any chance lengthy gameplay. Overall, it is a simple and somewhat enjoyable game but it loses much of its value after a short while and becomes rather uninteresting and boring.



Ratings



Graphics: 4/6
Sound: 5/6
Gameplay: 3/6
Storyline: 3/6
Design: 3/6

Developer: L4ntom
Version: 1.0
Instant Play:

Get it now!

The Game Maker's Apprentice

By: Darragh

It seems like years since we first decided it would be a smashing idea to create a game by ourselves, thus beginning our quests to provide the world with more and more games to enjoy. We all started at different places, with different ideas but really no way to put them together just right. But that changes for the newer, lucky generation of designers, due to Mark Overmars's *Game Maker*.

Description

Mark then went one step further and has proceeded to improve upon this gift, by bringing us his book, *The Game Maker's Apprentice* written with the help of an equally experienced Jacob Habgood. This book provides a whole new insight to Game Maker and its features, it:

"Provides an excellent first step into game development; Teaching game concepts, terminology and logic"

Or so says Mark Swinhoe, a senior programmer at Sony.

The book itself is divided into five different parts which include:

- Getting Started – A simple opening into Game Maker and how to use it.
- Action Games – A chance to create another couple games and a brief theory lesson.
- Level Design – This part covers: guess what, designing gamer friendly levels.
- Multiplayer Games – And how to insert them into games correctly
- Enemies and Intelligence – And the secrets of the Game Maker Language



The book brings the reader through a journey of learning the practical basics in the art of game creation and of course, the theory behind it. Through the book you will create nine different games, each one better, more advanced and more enjoyable than the last.

This book could definitely prove an investment to anyone looking for a future in game design, and as for the rest of us, the book has a wealth of information to share for "hobby game designers" too, and even if you all ready think you know a lot about game maker, it is sure to give you that extra edge to polish off your games. As for the technical specs, most computers running windows 98 or above with a 32MB DirectX – compatible graphics card should work fine with Game Maker.



MarDar: Creations

By: Eyas Sharaiha & Philip Gamble

Timedefenders

"Timedefenders" is 3-dimensional RPG game with online capabilities by "krejic" of the GMC, better known for his "Expedition China" series. He describes the game and its story by saying:

*When time-traveling is possible, it's time to explore the possibilities.
Space pirates travel through space searching for the time-machine. This would give them the power that no one ever could think about.
You need to save the technology to prevent the space pirates from stealing this technology and from destroying the whole universe!*

Timedefenders is a high quality 3D Game Maker game with standards controls and a challenging gameplay. You are guided through the storyline using multiple levels that generate excitement and often surprises to the player. The levels are numerous and large in size, giving you a large variety of places to play with in the game.

The game also has some numerous useful hidden features that could be unlocked if utter attention is paid to the details of the story. In general, the game is clearly well thought out, and delivers an excellent experience. With excellent graphical effects that are either turned on or off according to the settings to adjust to performance, the game never fails to deliver and excellent atmosphere. Well done!



Boulders Never Die

"Boulders Never Die" is an action-packed, mind-challenging puzzle Game Maker game that requires a lot of strategy and thought. In Boulders Never Die, you need to build a set of blocks in a certain way as to withstand the raging boulders and therefore protect a small red ball. The game is simply controlled and uses real Physics to achieve the breaking, etc. of the blocks and is therefore quite realistic. The game includes multiple levels and has been selected as a staff pick in the YoYo Games site.



MarDar: WIP

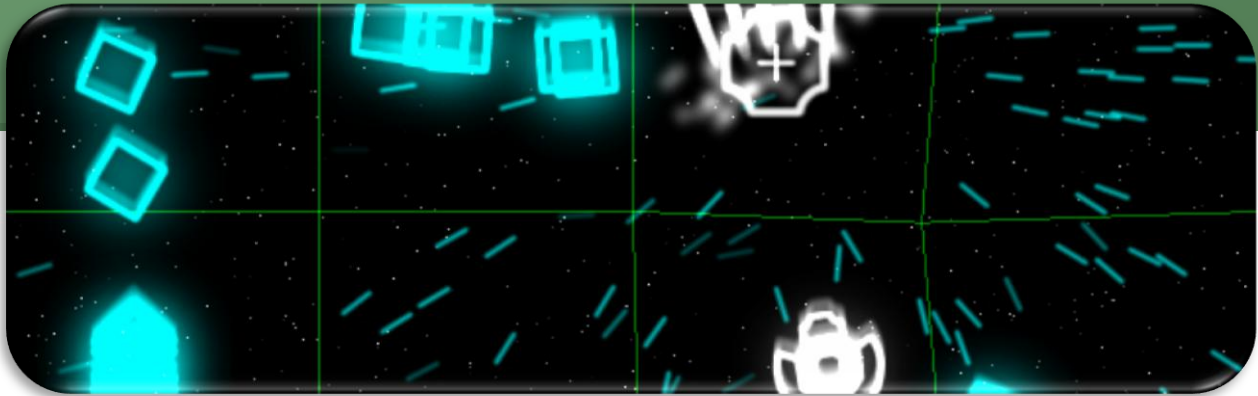
By: Eyas Sharaiha & Philip Gamble

Grid Defender

"Grid Defender" is an excellent recently complete game, similar to Geometry Wars or Asteroids. The game has been created by 'Necromian' of the GMC.

The game has six different modes to play with: arcade (in which you play trying to achieve the highest possible score, the game getting progressively difficult with time), survival (where you start with only a single life), formation mode (where you fight endless formations), sandbox mode (in which you just play around the grid), mini-melee mode (in which you fight enemies locked in a certain area), and the "Run Away!" mode (where you have a single life and just 3 bombs, and achieving a high score by avoiding enemies).

Indeed, the game is excellent and the gameplay experience is rewarding on multiple levels. Excellent job.



Practically Suicide

"Practically Suicide" is an 'annoyingly hard,' very challenging, platform game developed with Game Maker by "huh?" of the GMC. In Practically Suicide, you need to progress through the platform world by simply moving and jumping around the world.

The game is designed in an excellent way, but is really challenging: apart from its difficult nature, the game is all about one hit kills, and so you're expected to die multiple times during a level (and therefore restart it) before being able to complete it. However, you *can* save between levels.



Goodbye!

That's it: our One-year Anniversary Special Issue! Thank you for reading it! We hope you enjoyed this issue, with its excellent content which was delivered by hard working contributors and staff members to deliver an amazing one-year anniversary special that you won't forget!

We've worked hard on the layout, which was finalized for the Issue 11 new-year mini-special issue, and made sure that it is enhanced enough to fit perfectly all future issues of MarkUp. Hopefully there will be less change in the future, and only a modification of the color schemes, etc., so that we can deliver you the upcoming issues of MarkUp Magazine in a layout that is friendly, familiar, and most importantly: practical. We hope this does the job and we can't wait to listen to the feedback from the community!

The Issue 12 of MarkUp Magazine also included some follow-ups to previous MarkUp articles, including the Windows Vista Game Explorer, 65 Million and One BC, and other DLL- and resource-related information.

The issue comes with a new "Development Journal Article" for "Cities at War" to replace "Faction Wars". We hope the new content will be satisfactory to all of you, our readers.

The contributor and reader support for the making of this issue has been overwhelmingly positive – we thank the entire community for their efforts, contributions, comments, and overall support for the project; it certainly could not have been the success it now is without you! Indeed, this issue delivers the largest number of contributing members for a single release for *any* issue of MarkUp Magazine, or any other Game Maker Magazine for that matter. And for that, we say cheers to you!

We're also moving forward with Book Reviews, with more book publishers behind us than ever! Any staff member can receive a free book to review it. Click [here](#) for more information.

We have also overhauled the MarkUp Magazine website and improved it dramatically for this occasion; and, the end result is quite impressive! We hope the all-new improved MarkUp website will make it clearer to users to learn more about MarkUp, read our latest issues, subscribe to our latest news, and contribute to the magazine – all could be found through direct links on the site!

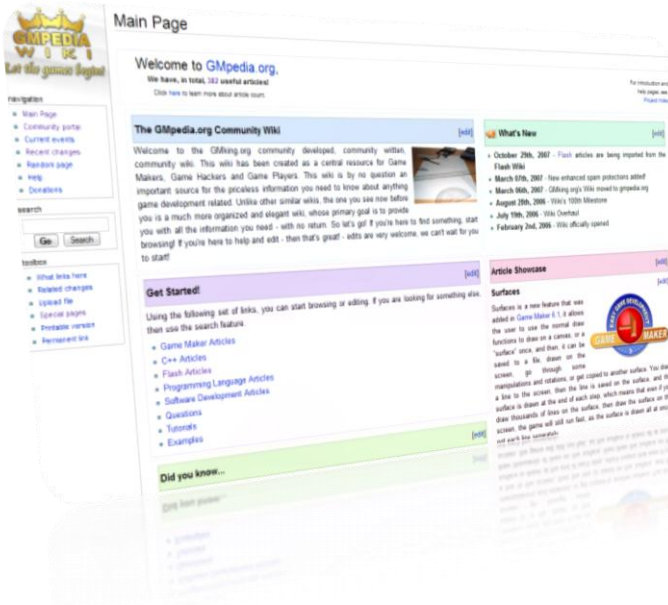
We'll coming back in April with our thirteenth issue of MarkUp Magazine. While many people decide to be superstitious and skip this issue number, we'll be releasing an awesome MarkUp Issue 13 for one main reason: *because we can!* Please send along all your contributions along, details of contribution are found on the MarkUp Magazine website.

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! With feedback, you can help by either joining the MarkUp [forum](#), or e-mailing the MarkUp [staff](#). You can also help us by sending us your own content, to do so, please check out the [contribution page](#) on the MarkUp Magazine website! The contribution page also includes a list of advantages to becoming a contributor, including information on advertising space and free game development books. Remember, 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 MarkUp's entire set of 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.

GMPedia.org has also been tied deeply with MarkUp Magazine. For more information and detail on certain topics discussed in MarkUp Magazine, visit GMPedia.org!

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