Issue 10

MarkUp

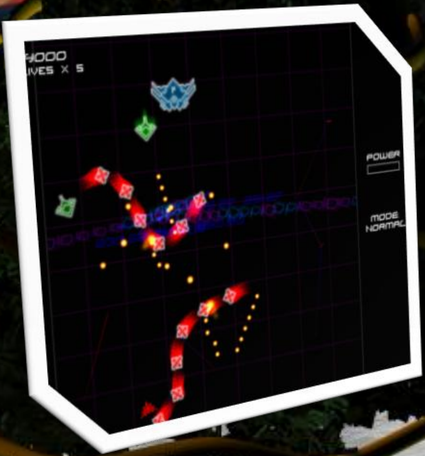
The Holiday Special!

The Making of Varia & Pollen

Detailed Effective Journals!

Detailed Step-By-Step Tutorial!

Real Time Anti Aliasing



Game Development Journals

God of Rock & Falcon Wars

Game Maker Insight

Speed Drops and Game Freezes

And Much More!

Resources Previews Reviews Tutorials

Table of Contents

Editorials

Tutorials

Development Entries

Monthly Specials

Book Reviews

Game Reviews

Exclusives

Welcome!

Contributors

Eyas Sharaiha	Sr. Editor
Robin Monks	Sr. Editor
Andris Belinskis	Editor
Philip Gamble	Writer
Bart Teunis	Writer
Erthgy	Writer
Leslie Bicker Caarten	Writer
Julian Abelar	Writer
Alec Smith	Writer
Kairos	Writer
Jonah Turnquist	Writer
Tarik Abbara	Writer
Jono Alderson	Writer
Mathew Malone	Writer
Leif Greenman	Writer
Veeti Paananen	Writer
Merrie Schonbach	Writer

Special Thanks

This issue includes commentaries made by the guest judges of The Game Maker Race on the participating game. A special thanks to all guest judges:

KC LC	Rhys Andrews
Snabela	TGMG

About MarkUp

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

Contribution

For more information about contributing to MarkUp Magazine, visit our forums here.

Editor's Desk

The World's Latest Trend: Parallel Programming

Sorry for taking over Robin's spot, but I'm only doing so this issue; he's been really busy! Anyways, I bet many of you would remember a couple of years ago when all CPU chip-makers were concentrating on higher clock speeds rather than anything else. You might look back and be impressed at the developments we have now, hyper-threading is taken for granted, dual-core is in almost every computer; and multi-core CPUs are on the rise!

While I do share the satisfaction and astonishment of our latest achievements, I cannot but long to the 'old' and 'simple' times! Back when a single-core CPU was the dominant in the market, all that we as programmers needed to do is write a 'linear' set of instructions, and the CPU would execute them in that order… pretty simple, pretty easy.

Now, we can still do that, but we wouldn't be taking advantage of the multiple cores in a computer. Most programmers are still used to the old ways: linear programming. That's not a big issue now, but in the future when multi-core computers and CPUs become more dominant, these programmers are going to need to adapt their programming techniques to take advantage of the new incredible speeds, the 'slowness' of programs we put up with now might become unacceptable in the future.

It is important to take advantage of multiple CPU cores, but this is a very difficult task. It is really difficult to take full advantage of this in programming languages like C++, but that's not where the problem lies! The problem lies with platforms like Game Maker!

With Game Maker, we cannot, no matter how good we are, take advantage of parallel programming, unless we use DLLs to draw and execute certain instructions on the Game Maker window. GML and GM actions are incapable of being executed at the same time, since Game Maker itself and its runner was not adapted to parallel programming. Rewriting Game Maker to become adapted to parallel programming impractical for two main reasons: first, its really tough to do so, but secondly, it'd be a challenge to keep Game Maker as easy to use as it is while being adapted to parallel programming. I'm looking forward to see if YoYo Games is planning to do anything about that. We might not need to worry now, but in the future, we certainly do.

See You Next Month!

Eyas Sharaiha■

Holiday Memories

Three years ago in December I was diagnosed with cancer. I went through chemotherapy and radiation. The following year, I was diagnosed with it again, it had spread. So once more I went through chemotherapy and radiation. I am happy to say as of this year, I am two years in remission.

Though this story starts sadly it has a silver lining. My family who lives in Phoenix Arizona wanted me with them for the Christmas holidays so they all pitched in and bought me a plane ticket. I live in Linfield, PA.

When I arrived in Arizona I was picked up by my favourite sister, Veronica. (I am the baby of the family.) She took me to her home, which is a very nice townhouse. The front porch was sparkling with white lights and snowflakes. What a contrast to the desert landscaping in the yards.

The living room was like stepping into another world. A beautiful tree was in one corner, the fireplace was brightly decorated. They had a complete country house, snow scene on the top of the entertainment center. But what really caught my eye was a stack over 5 feet tall of boxes, from a large box to a very small box on the top with my name on it.

Christmas morning my other sister, Marge, came over from her house about a block away and I started to open those boxes. Each box was filled with tissue so I had to dig for the gifts inside. There was laughter and joy in each box that I pulled off the stack and opened.

This was one of my happiest memories and the one that touched my heart the most. We are closer now then we have ever been and so much love has come out of my illness, and I'm grateful for every day I have on this earth.

Merry Christmas and Happy Holidays!
Warm wishes,
Merrie Schonbach

Graphics or Gameplay

The topic has been going on for a long time and after the review of the topic (and counting the money that 'Yourself' gave me from PayPal for influence on this topic!), I have concluded that game play is more important than graphics. Sure graphics may be fun to look at, draw people into your game, create cool effects, add a sufficient future for those who invest in it a lot (art) etc. But without game play, graphics just don't cut it!

What Graphics Do

Now some of you may of stopped reading this article as of right now, because that paragraph up there may have actually completely gone against what you believe, or perhaps have simply reassured an obvious fact. That's fine; just make sure you listen here:

There is definitely a ton of info that should not be questionable about the amount of value added by graphics to the game, I mean seriously graphics aren't useless; they can (at the very least):

1. Traps [*attracts*, actually!] the user into playing your game.
2. Create cool effects, which just about anyone can't deny that they want.
3. Add to the fun tone of the game; good graphics stand out boldly declaring "Hey you! Play this game, its fun, plus the visuals COMMUNICATE well with you to make them further play this game!"
4. Provide an excellent future for those interested in arts. Even pixel art I swear, you would just laugh at how many different forms of graphical jobs you could get.

I am not denying that I believe graphics contribute a huge part to a game, I am just merely stating they aren't the most important.

Be Brave

"The brave men profit their city more than the finest, most cleverest speakers." –Aristotle

Game play is the brave men, and graphics are the clever speakers, if you have a game with horrible graphics, yet amazing game play, well guess what? People will play your game. However if you have a game where graphics overrule game play, the game is only fun for a short period of time, then once they start playing it its complete crap.

Game play, if developed correctly, will overrule correctly developed graphics.

Please use this topic to help you develop your games further, focus on the number 1 importance if a game, which by now you should understand what that is. However by now you should also understand that by looking at the above list, you can use graphics to your advantage.

Link to topic (http://xrl.us/GFXorGP)

erthgy■

Solving the Problems

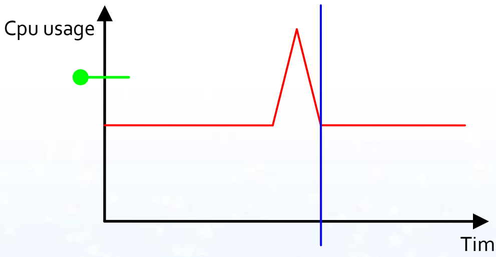
Of a Game Freezing and Speed Drops

A problem that occurs is quite often speed drops and in the worst case the game freezes. These are annoying situations because the player or user is not aware of what's happening. Fortunately, in most cases, you can rewrite your code a bit to avoid all this. Let's have a look at how this can be achieved.

Causes of speed drops and the game freezing

Because a computer only has limited processing power (it is not infinite), it can only do a certain amount of calculations at a time. This is the reason why speed drops can occur. If too many calculations need to be made in a short time, the processor is not able to handle them in that amount of time. It is then that speed drops will occur.

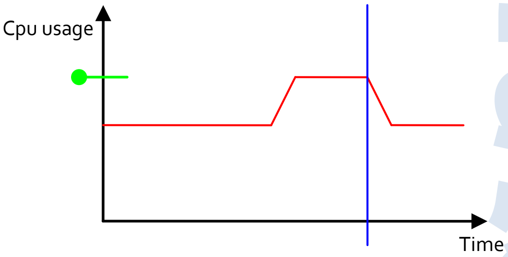
In Game Maker, speed drops also occur because of that same reason. Usually, the game speed will drop significantly when many calculations are made in a short time. This is e.g. when resources are loaded or when a loop is executed many times.



There is a peak in CPU usage:

Let's say that the green line represents the CPU usage where speed drops don't occur yet. Then in this case there definitely will be a speed drop. So it is actually the high peak that we want to lower, as shown in the figure on the top left.

The blue line shows the time when the action is complete. You can clearly see that it takes longer to execute the code in the second case than in the first case. This is a disadvantage when you want to avoid speed drops. The average speed will be lower than before. First we'll have a look at the problem specific to GM, and then we'll attempt to solve it.



Note: the graphs do not show data from real measurements. They are only intended to make the reason clearer.

Speed drops in Game Maker

In Game Maker, `room_speed` defines the speed at which the

Quick Review

Image Data Structure Scripts

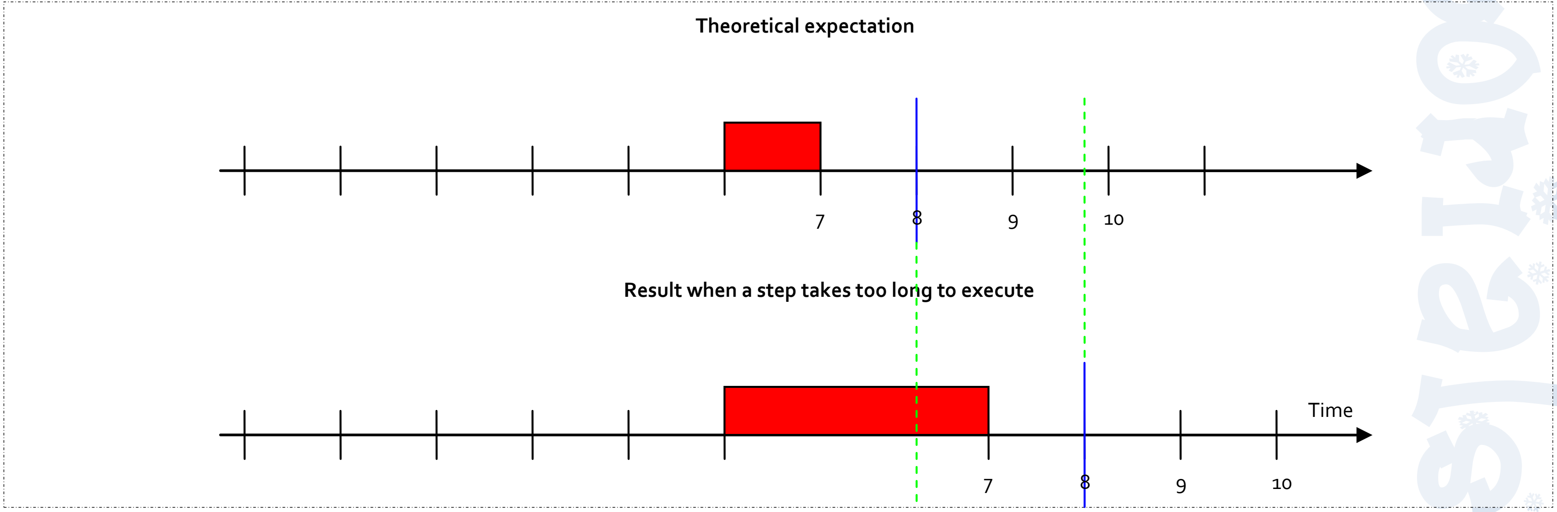
The image data structure script (ds_image scripts) allow users to convert regular images (whether they are sprites, backgrounds, etc) to data structures storing the color information for each pixel.

While this appears to be memory and resource consuming, it is done for a reason: it makes mathematical manipulation of color values, like brighten, darken, invert, greyscale, and color blending much easier!

gmc.yoyogames.com/?showtopic=342952

Solving the Problems

Of a Game Freezing and Speed Drops



game is run. It is expressed in steps per second (theoretically). The default setting is 30, so this will also be used in the explanation.

Theoretically, a step in GM would be: $\frac{1}{30}s = \pm 33.3\ ms$. Now that's the theoretical value. If not all code can be executed during that amount of time, GM will still execute all code in that step. The consequence is that the step that follows the current step needs to wait until the current step is finished. And then the game slows down or freezes. It could look like shown on the opposite page.

As you can see, the step behind the slow step starts much later than theoretically expected. Suppose a sprite is drawn. When the 10th frame should be drawn, the 8th frame is drawn.

Now that we know what the problem is, let's solve it.

The solution to the problem

If it can't be done without a speed drop in a single step, then why not do it in a few steps? It's a very simple solution. Unfortunately, it can't be done in all cases. Suppose a very large sprite needs to be loaded into memory. This can be done with the `sprite_add` function. The problem here is that when `sprite_add` cannot be executed in a single step, there is no way to do this in more steps. The speed drop here cannot be avoided. The most important cases where it can be done are when loops are used. Suppose an object needs to execute a huge **`for`** loop many times when it is created. This will cause a speed drop when an instance of the object is created. To avoid this, the 'for' loop can be removed and instead the step event can be used for the same purpose. Everything that was originally in the loop now comes in the step event. An example will make this clearer.

Quick Review

D3D Distance Formula

The 'D3D Distance Formula' is a nice little script that calculates the distance between two points on the 3D Cartesian space.

'D3D' is a bit misleading since this little script actually calculates the distances between any two points in any 3D system and not just Direct3D.

The script can be particularly useful in a multitude of occasions, especially drawing certain effects of interaction between two objects, determining AI in 3d-worlds, etc.

Few lines of code hold much power, indeed. This script demonstrates that.

gmc.yoyogames.com/?showtopic=344960

Solving the Problems

Of a Game Freezing and Speed Drops

Original

Create Event

```
var i, total;
total = 1000;
for(i = 0;i < total;i +=1)
{
        //execute a lot of code
}
```

Step Event

➔ nothing special here

Loop in step event

Create Event

```
i = 0;
total = 1000;
```

Step Event

```
if (i < total)
{
        //execute a lot of code
        i +=1;
}
```

As you can see, the ‘for loop’ has just been divided over the ‘create’ and ‘step’ event. Now there will be no more speed drops, assuming that the piece of code in the ‘for loop’ can be executed in a single step. But the code as it is now only executes the piece of code once per step. Often, you’ll be able to do this more than once per step and then it’d be better to use a repeat loop in the step event. How many times does it need to be repeated then? After some testing, I found out that a good repeat value is a factor times the fps. This is logical. If too much iteration is done per step, the fps will drop. Because the number of iterations per step is determined by the fps, this number of iterations will also lower immediately and then the fps can go up again. The step event could then be something like the following:

Step Event

```
repeat (factor * fps) //factor should be defined in
the create event
{
if (i < total)
{
        //execute a lot of code
        i +=1;
}
}
```

And that’s the best it will get. You can play with the factor a bit to get the best results.

You can also add a progress bar if you want. This isn’t too hard to do. The percentage completed is of course $\frac{i}{total}$. The following piece of code can be used for a simple loading bar:

```
draw_rectangle_color(x1,y1,x1+i/total*width,y1+height
,back_color,back_color,back_color,back_color,0);
draw_rectangle_color(x1+margin_hor,y1+margin_vert,x1+
width-margin_hor,y1+height-
margin_vert,front_color,front_color,front_color,front
_color,0);
```

Just declare the used variables with a decent value and you’re ready to go.

Conclusion

Most speed drops in a game or program can be avoided. The disadvantage is that it takes longer to execute code than it normally would. To solve the problem, you can divide a loop over the ‘create’ and ‘step’ event and set the number of iterations per step. And for the code that really takes a long time to execute, you can display a progress bar to keep the user informed. And that’s all there is to say about it. I hope this article has been of use to you.

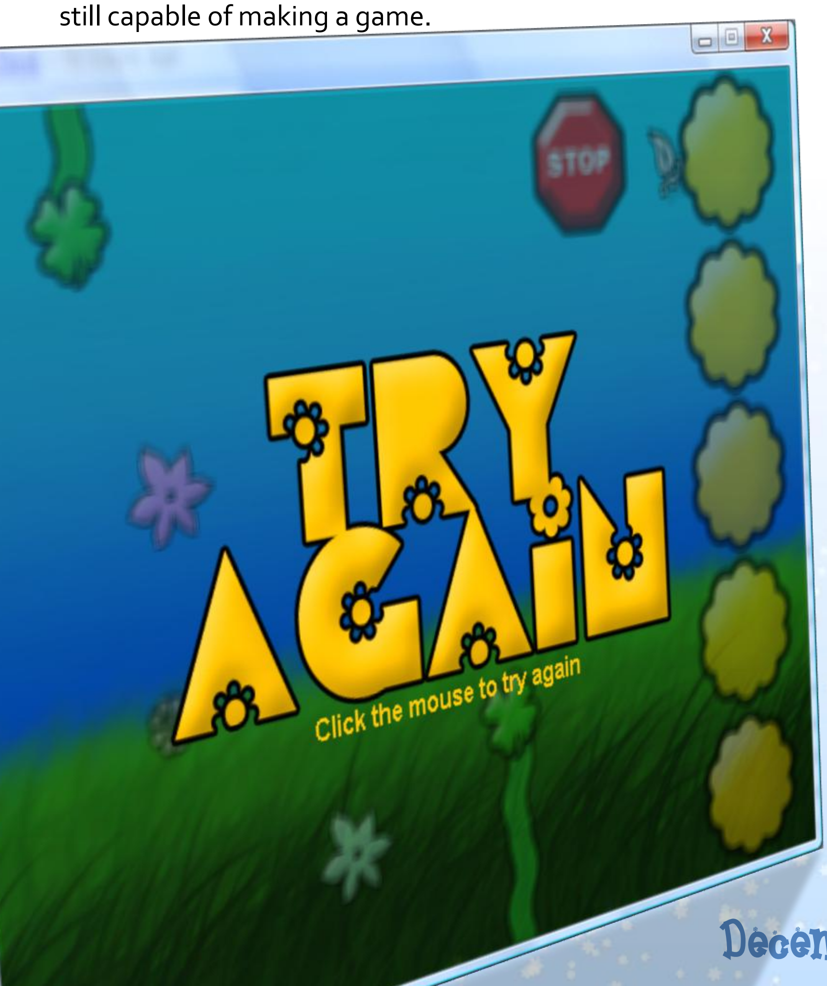
Bart Teunis■

The Making of... Pollen

Hello, my name is Alec Smith (cbdman25), I created the game Pollen. I have been asked to write a short article on how the game was made, however I'm terrible at writing anything longer than a sentence but I'll do my best.

Beginnings

It all began way back in November 2006 when I decided to post one last game on the GMC and leave for good. Now, most people who "leave" the GMC soon return as another user or return as the same user making a topic on how they've made a triumphant return. However, I left because I found the GMC boring. All the members acted the same way and in all honesty I was growing tired of making games. I wanted to focus my attention on animation. That was my real goal in life; to become an animator, not a game maker. So I left the GMC and stopped making games for about 5 months. By the time it was March 2007 I realised I hadn't used Game Maker in a very long time and I was scared that some of my GM knowledge had committed suicide over the months. I had to prove I was still capable of making a game.



Basically, Pollen was a test to make sure I still had my game making abilities.

The Idea

When it came to sitting down and starting Pollen the long gap in time had affected the way I made my games. First, I sat down on my chair and starred at my computer screen trying to think of a game idea. I knew I wanted to do a puzzle game because they have a certain sophisticated element to them. I was blank. I had no idea what to do. Sure, there are loads of puzzle games out there I could have cloned. Like that box pushing game! **OH BOY I JUST LOVE THAT BOX PUSHING GAME!** My source of inspiration lies in Lemmings. Indeed, it is the greatest game ever to be witnessed by my irises. However, Lemmings is too difficult to do on Game Maker and too ambitious for a catch-up game. I was stumped and that meant drastic measures, I had to think of the name first and work backwards.

WHAT?!

Yeah, so, it was summer and I had hay fever. BINGO! Pollen! What a pretty name! I bet that would make a nice game! With still no clue what happens in the game I found an appropriate font and made a menu background in Photoshop.

Okay so what do you actually DO in the game? Hang on....flowers need to get pollen from one flower to another! **THAT'S RIGHT!** Apply the Lemmings factor and **ALAKABOOM** the concept of Pollen was born. I made one attempt at drawing a flower and it turned out something that looked like a paperclip that's been bent too many times and then attempted to bend back into its original

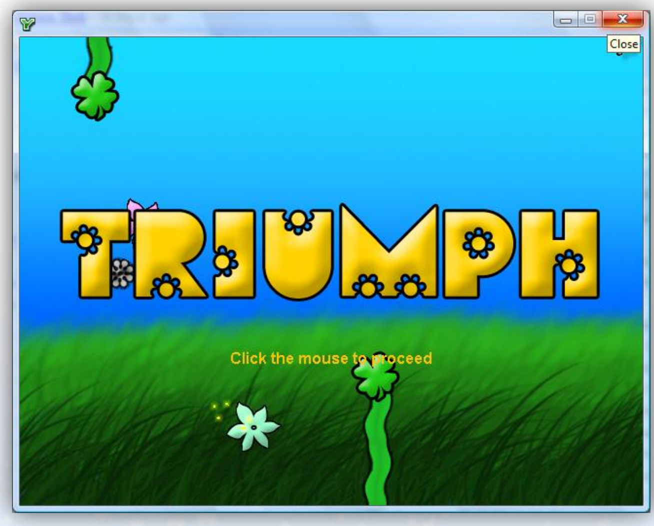
The Making of… Pollen cont.

form. Luckily, Photoshop was open at the time and I just used a default flower shaped object. However, Photoshop has anti aliasing applied and using that as a sprite would be me a horrible white outline.

'The Making of…'

I was forced to use the alpha channels to create the illusion of smooth edges using the good old, `draw_set_blend_mode(bm_subtract)` and then plonking on a sprite of some sort. It turned out nicely!

Next were the spinning fans. Now, GM doesn't have motion blur options and that means if you have an object spinning at 360' a frame, it won't look as if it's moving at all. So I attempted some weird methods of getting a look that the fan was actually spinning very fast rather than just rotating weirdly. I can't remember how I did it so just pretend you're amazed.



Click the mouse to proceed

I cannot remember what happened after that except that by day one I had the engine finished. Day two and three would consist of me designing levels. That's one of my game making philosophies. PERFECT THE ENGINE! It's so tempting to just go ahead and make levels but you really, really need to get that engine done! All the small awkward bits too like HUD effects, you need to do them. And don't just get it out the way, take your time! When I do an engine, I always try to work it around myself. Creating a system that's easy to edit yet is comfortable to use and affective too. I cannot describe how important it is to get the engine done before you make levels. **IT'S VERY IMPORTANT!**

Another philosophy of mine is to try and make THE PLAYER CONTROLS COMFORTABLE. Let's say you have a game that uses the arrow keys and after playing the game for 5 minutes there's a button on the screen that you can ONLY press with the mouse. That's uncomfortable. I try to stick to one input device. Pollen is solely a mouse based game.

I don't really know what else say except I was very lucky that Pollen turned out the way it did. I'm not a game maker, I'm an animator. My Youtube account is full of cartoons not Naruto dub overs.

Conclusion & Advice

If you're feeling really bummed with Game Maker at the moment, take a break from it, learn another program then return in a month or two. You'll have a fresh approach and your mind won't be filled with what the GMC want you to put in a game. Staying away from the GMC and any other GM game refreshes your perception on design.

Oh, and some game design tips: don't make text suddenly appear on screen, have it slid on! Don't have the player sit through hours of pointless intros to fake companies. **I DON'T WANT IRON MONKEY STUDIOS I JUST WANNA PLAY THE GAME!** Stuff like that, ya' know?

That's one of the main reasons I hate the GMC. Look, if you're posting a game there is no need to have a ridiculous topic title. If the game is good it will get its own publicity. Otherwise you better hope that your long title saying "ONLINE COMING SOON" gets the attention you want mate. But hey! What do I know?

GMC = Bad

Thanks for reading this by the way.

From the guy who lives in the UK,

Alec Smith■

The Making of... Varia

Varia was originally a game that was started out of fun for a competition. After some toying around, however, I landed on a solid concept and decided to take it from there. It was going to be my second game, and considering the popularity of my first one, I decided I'd step it up a notch and try to fix what was wrong in the first game with my second one.

Eventually I landed on an even better concept, and the game was no longer being made *just* for fun. I had an idea that could possibly win the competition, so I decided to put all my effort into making it. Deadlines help a lot with game development, especially with rewards, thus causing Varia's gameplay duration to be at least triple of my first game.

Concept

The theme of the competition was "Options", or satellites that trailed after your ships or orbited them. We were to make use of the theme in some way. I decided that I'd have the trailing options. If you've ever played a Gradius game, you know what they were. They're the little ghost ships that follow after the location of the player, and in recent games you can freeze them in their positions relative to the location of the ship. I thought the freezing part was quite interesting, so I wanted to think about what I could do with it. I thought up of having them absorb enemy bullets – then sending them back when you shoot. Thus the number of enemy attacks was decided at 10 different types of attacks, which you could touch with one of your options and then immediately gain the ability to use that attack with that "option" or however you want to call it. To get other types of abilities, you'd want to release the option power at some point, so you could select an individual option with the up and down arrow keys. I restricted the buttons to Z,X,C and Arrow keys simply because I did not want too many buttons for a shoot-em-up game, since it's an arcade-type game. This leads to a large degree of multitasking, but in the end it worked out well. Pressing X would let you go into slow mode and freeze your options at the same time, which worked perfectly, so it was two things in one.

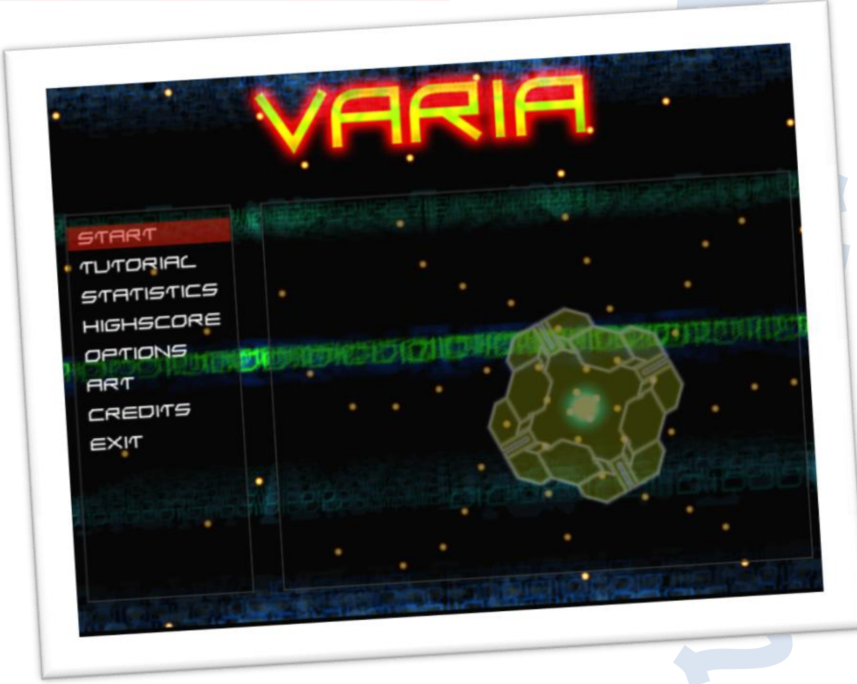
I decided that enemies should be mostly made of a single part, while bosses and mini-bosses should be made out of multiple parts. This inspiration came from Warning Forever, a shoot-em-up game based on evolving bosses and destructible parts (as did some of the graphical style). This lead to interesting fights with bosses, where you could decide what to destroy first instead of just blowing the entire thing up after who knows how many shots. Attack patterns were going to be fairly mediocre, though the speed of the bullets would be fast for a good feel of a high paced game.

Levels

I planned to have 4 levels, which was a lot for the time period I was working in because of the enemy wave planning and balance, along each individual boss and mini-boss. I spent hours playing over and over certain stages to check if the waves felt just right, and then I let some beta testers play the game and they would report if it was too hard or not. I treated the project as professionally as possible. I made some concept art on enemies, and carefully thought out the design of the levels. Since I went for a cyberspace sort of feel, my levels were designed to give the player the impression that he or she was inside some sort of computer world, destroying virtual enemies and bosses. The backgrounds were all thought up as I approached the development of each specific level, and were modified several times to make the game look right.

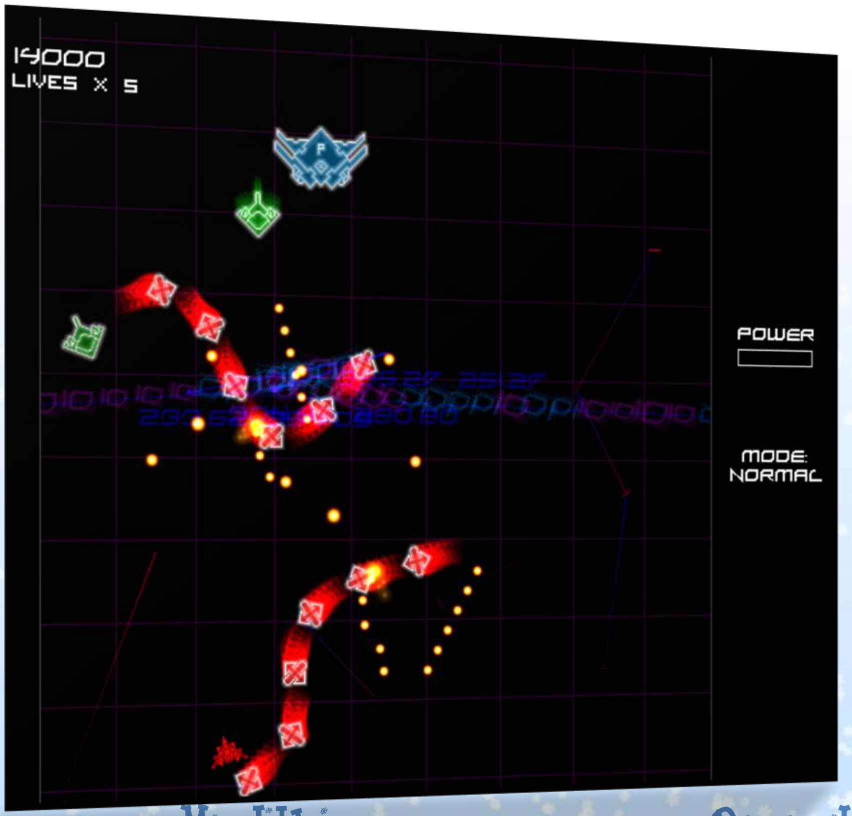
Varia Cont.

Bosses were the hardest part of each level, as each was individual and unique with their own attacks and abilities. The bosses took a few days to make and perfect with their many attack patterns, destructible parts, behaviour, and balance. I also had to make sure that they weren't going to kill the framerate due to their large usage of sprites, parts deciding what to use and when to attack, and the number of bullets on the screen.



Balance

The abilities were going to be available for the player to absorb in increments – the first level had four absorbable weapons, second level had six, third had eight, and fourth had all ten. This was done so that there wasn't a huge amount of different enemies thrown right at the player with the first stage, and so that the player could get used to the weapons before trying new ones. The abilities were pretty much balanced as each weapon had an advantage and a disadvantage. You could charge up the charge shot and it would take a long time to accumulate a powerful attack, but in the end it is worth it for using it on a boss or similar. If you wanted to eliminate enemies and avoid getting killed as much as possible, you could get the blast ability to cause some area damage or get the homing missile to not worry about some enemies at all (the side effect of that is you could easily break the chaining system that gives you a good score). The nuclear missile is not too useful while fighting a swarm of small enemies, but using it on a boss would be perfect since the bosses are huge and the missile does a great deal of damage. Each ability had an advantage and disadvantage that was carefully balanced throughout the game with a great deal of testing. Balance is definitely crucial for a game to be successful.



Polish

Finishing up the last few things of the game was probably the part that was the hardest. After finishing the four levels, I had a load of work to do for customization (difficulty levels and game options), a decent menu screen, getting music to go along with the levels, adding certain features, additional eye candy, and similar. These were the finishing touches that were being made when I started not wanting to work on the game, but I forced myself to work

Varia Cont.

on them anyway, because I kept looking at the probable outcome and success of the game. With the music, a composer contacted me and I showed him the game, which lead to the custom made music for each of the level, which would fit the game perfectly. Generally that doesn't happen unless you have a good game to show off, so I was lucky there. I finally got everything touched up, and added a comprehensive readme (even though there was a tutorial in the menu) and released the game. I wanted to make the game look as professional as possible, and it paid off in the end.

Tips

There were many things done in Varia to make the eye candy look great. The explosions were mostly lines and translucent shapes that all appeared to be different, yet they were one or two sprites changed with scaling, image angle and alpha values. Don't use the built-in particle types for your explosions or effects; you're better off making your own type with sprites. Try stuff you haven't seen before for effects and graphics; the type of graphics I stuck with became my style. Most of the explosions are therefore unique because they fit the graphic style. I've also never seen a distortion effect used in a particularly great way, so the bubble distortion along with the massive explosion afterwards, for those who have seen it, was something most people did not see before with a game made in GM.

As for the game development in general, as mentioned before, give yourself a deadline. Plan ahead of time. Perhaps draw sketches of what you are planning to make, and think it out thoroughly before starting on the game. That is, if you're serious about making a good length game and if you are planning to be professional about development.

My coding skills were not that fantastic at the time I made this. Don't let stuff like that put you off if you aren't that great at programming. If you think you can make something, then go and try to do it. You'll probably learn something new each time you try making a more complex game. Once you look back, you'd be surprised at how much you've progressed.

Kairos■

Quick Review

Instant Nearest Extended

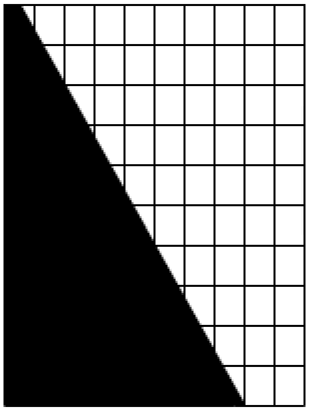
The 'Instant Nearest Extended' Game Maker Extension Package allows members to use several functions (or scripts) that are similar to but more extensible than the built-in Game Maker function: `instant_nearest`.

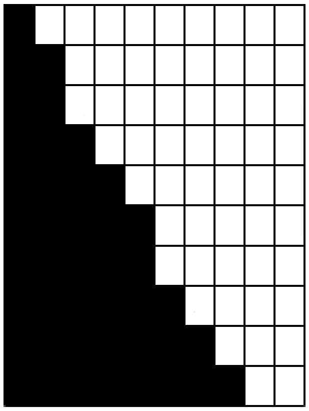
It can find the nth nearest instant, the nth furthest instant, instances with certain criteria and variables also according to their distances from a certain position, etc.

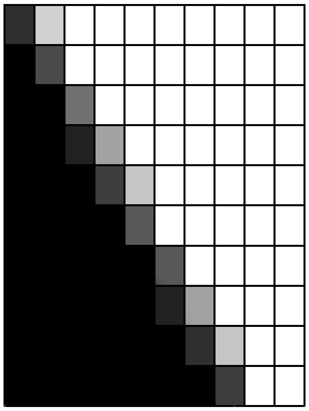
gmc.yoyogames.com/?showtopic=284288

Real Time Anti Aliasing

As many of you may know, anti-aliasing is the technique of smoothing the jagged edges known as aliasing when representing a high-resolution image or shape at a lower resolution. Aliasing occurs because when you try to draw a shape or image with a higher resolution than that of the screen; the pixels need to be approximated, which causes the jagged effect.


the "ideal" image


the image, aliased

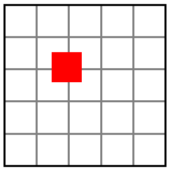

the image, anti-aliased

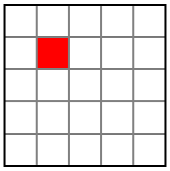
Imagine drawing a square (a pixel, in this case), but it needs to be snapped to a certain resolution. Usually the point is simply rounded to the nearest pixel, but with anti-aliasing we can create a smoother effect by approximating how much of a pixel an "ideal" shape would cover. For example, if an ideal line segment one pixel thick crosses a pixel and covers 60% of that pixel, we can fill in that pixel with 60% alpha instead of the 100% we would get by default.

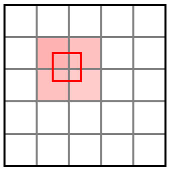
Game Maker Implementation

So we have a basic idea of what we need to do, but how do we do it? A good idea would be to consider how we can approximate a single pixel that *isn't* on our pixel grid, interpolate it between multiple pixels, as it were. Game Maker has an option for this, but it doesn't quite fulfill our requirements, as it only works for images, not shapes. Very well; let's look for a way to do this ourselves.

If we have a point at position (x, y), and x and y aren't integers (round numbers), then drawing it normally will cause aliasing. So we need to change our drawing alpha based on the coverage that all the pixels around x, y get. How do we do this? Let's start with an example. Say we want to draw a pixel at the point (1.45, 1.45). You can see how this would look by looking at the example image.


ideal


nearest pixel


interpolated

If we were to draw a pixel normally, the result would be the second image, because Game Maker rounds pixels to the nearest point on the pixel grid. The third image is our interpolated image: as you can see, the top-left red pixel has slightly more than a quarter of the coverage, the top right image has a bit less, the bottom left has the same coverage as the top-right pixel, and the bottom-right has the least coverage.

Calculating 'Coverage'

So how much coverage is this *precisely?* This is actually quite easy to calculate. We know that coverage of the first pixel starts at (1.45, 1.45) and ends at (2, 2). So, our coverage is (1 - relative x) * (1 - relative y), where the relative x and y are the coordinates from the top-left part

Real Time Anti Aliasing

Cont.

of the pixel. In this case the relative x is 0.45, and the relative y is too. In fact, we can now state that the relative coordinates are the *fractional part*s of our original coordinates. In GML: (`frac(1.45)`, `frac(1.45)`). So, the alpha we should use for the top-left pixel is `(1 - frac(x)) * (1 - frac(y))`. We can deduce the coverage of each other pixel in a similar way:

```
top_left_alpha = (1 - frac(x)) * (1 - frac(y));
top_right_alpha = frac(x)*(1-frac(y));/*look at the image! the
surface area of what needs to be covered is frac(x) by (1-frac(y))*/
botton_left_alpha = (1 - frac(x)) * frac(y);
bottom_right_alpha = frac(x) * frac(y);
```

Very well! We now have our alpha values in code, and we can proceed to draw a pixel outside of our standard pixel grid! A simple game maker implementation to draw an interpolated pixel:

```
//argument0: x
//argument1: y
var current_alpha, ptl, ptr, pbl, pbr, floor_x,
floor_y, frac_x, frac_y;
//first determine on which pixels this point lies
floor_x = floor(argument0);
floor_y = floor(argument1);
frac_x = frac(argument0);
frac_y = frac(argument1);
a_tl = (1 - frac_x) * (1 - frac_y);
a_tr = frac_x * (1 - frac_y);
a_bl = (1 - frac_x) * frac_y;
a_br = frac_x * frac_y;
//draw it.
draw_set_alpha(a_tl);
draw_point(floor_x, floor_y);
draw_set_alpha(a_tr);
draw_point(floor_x + 1, floor_y);
draw_set_alpha(a_bl);
draw_point(floor_x, floor_y + 1);
draw_set_alpha(a_br);
draw_point(floor_x + 1, floor_y + 1);
```

Now we have this, we can proceed to draw all sorts of shapes, with correct anti-aliasing! I'll provide you with two examples to help you on your way, but you should be able to figure out the rest by yourself. Note that these aren't the fastest possible algorithms for this kind of thing, but more complicated examples like Xiaolin Wu's line drawing algorithm would defeat the purpose of a simple explanation, in my opinion.

```
//draws an anti-aliased line from x1, y1, to x2, y2
//arguments: x1, y1, x2, y2 respectively
x1 = argument0;
y1 = argument1;
x2 = argument2;
y2 = argument3;
dx = x2 - x1;
dy = y2 - y1;
m = max(abs(dx), abs(dy));
for (i = 0; i < m; i += 1)
{
   draw_pixel_interpolated(x1 +(dx/m)*i,y1+(dy/m)*i);
}
```

And:

```
//draws an anti-aliased circle at x, y with a given radius.
//argument0: x
//argument1: y
//argument2: radius
var circ,i,rad,_x,_y,frac_x,frac_y,floor_x,floor_y;
circ = pi*argument2*2;
draw_set_alpha(1);
draw_text(0, 0, string(round(circ)));
for(i = 0; i < circ; i += 1;)
{
    rad = (i / circ) * (2*pi);
    _x = argument0 + cos(rad) * argument2;
    _y = argument1 + sin(rad) * argument2;
    draw_pixel_interpolated(_x, _y);
}
```

Conclusion

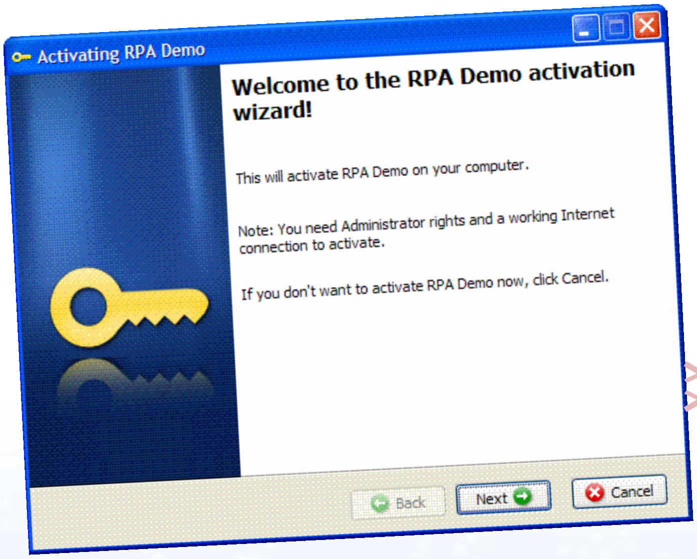
Anti-aliasing is a great way to overcome the existing limitations in computing and graphics, and I've tried to present methods which allow real time anti-aliasing of shapes to occur within Game Maker, rather than 'fake' anti-aliasing.

Leslie Bicker Caarten■

Ruby's Product Activation

Want to make money off your Game Maker games and applications? Don't know how to keep your products secure so only those who paid can use them? We'll, you've found the right extension. Ruby's Product Activation, or RPA, allows you to let your users activate your application online.

This extension uses product keys of 25 characters length, divided in five groups: XXXXX-XXXXX-XXXXX-XXXXX. Every product key can only be activated on one computer. If a user wants to transfer an activation from one computer to another, he can simply deactivate his or her product key, and then re-activate it on the new computer!



Welcome to the RPA Demo activation wizard!

This will activate RPA Demo on your computer.

Note: You need Administrator rights and a working Internet connection to activate.

If you don't want to activate RPA Demo now, click Cancel.

Before you start, you need to register your application and its Game ID at http://rpa.rubysoftware.nl/. Once you register a product to the site, you will be able to keep track of how many people activated it. You can also generate activation codes there. You can take these activation codes and even slap them on CDs holding your application. Also, since the rubysoftware website generates the activation codes, people hacking your application will not be able to generate activation codes.

The download package includes the GEX extensions, a DLL, a GM5 and GM6 example, a GM7 example and a Delphi example.

Good luck!

Download (gmbase.cubedwater.com/?page=extension&id=163)

Jonah Turnquist■

Quick Review

Smooth Pathfinding

This Game Maker Extension package contains a simple function that allows smooth pathfinding to occur within the game. It uses the Motion Planning functions of Game Maker, but is slightly more customized to include more 'stuff'. For example, motion could be set to become 'smoother' by changing the function's 'smooth factor', which decreases the precision and accuracy of pathfinding but makes object motion much smoother, depending on its level.

The script is also 'smart' to an extent, as to optimize for speed and save some of the computer's processing power. For example, it stops processing and calculating movements when the path to the object becomes clear and straight and switches to using the built in Game Maker functions for motion planning instead of any edits, such as path smoothing, etc.

The extension is only available as a GEX, no scripts or anything, so there's no hope of getting it if you don't have Game Maker 7. However, it does come with a well-explained example for those with GM7.

gmc.yoyogames.com/?showtopic=338426

Map Loading & Saving

One of the important Data Structures in GM is the map data structure. Maps allow certain variables to be assigned to certain positions/regions throughout the screen. The script of the month allows the users to save maps to files and load them.

Map Data Structure Loading

```
/*  Usage:
**      ds_map_load(file [,delim])
**  Arguments:
**      file        file name, string
**      delim       delimiter used between elements, string
**  Returns:
**      the id of the loaded ds_map data structure
**  GMLscripts.com
*/
{
    var FileIn, MapItems, CurrLine, NumItems, i, TempKey, TempValue;
    // Open the file that contains the map
    FileIn = file_text_open_read(argument0);  // Open the Map File
    MapItems = ds_map_create();               // Create the Map to be read into
    CurrLine = '';                            // Default to an empty string
    // Read the entire file into a variable
    while (!file_text_eof(FileIn)) {
        CurrLine += file_text_read_string(FileIn);
        // Technically we should not have a return and a comma
        // But just incase lets make sure that we do not add an extra one.
        if (string_char_at(CurrLine,string_length(CurrLine)) != ',') {
            if (argument1 != 0) {
        if (string_char_at(CurrLine,string_length(CurrLine)) != string(argument1)) {
                    CurrLine += ',';
                }
            }else{
                CurrLine += ',';
            }
        }
          file_text_readln(FileIn);
     }
     // Remove the final comma
     CurrLine = string_delete(CurrLine,string_length(CurrLine),1);
     // Add Support for other separators.
     CurrLine = string_replace_all(CurrLine,';',',');
     if ( argument1 != 0 ) {
 CurrLine = string_replace_all(CurrLine,string(argument1),',');
     }
     CurrLine = string_replace_all(CurrLine,', ',",");
     CurrLine = string_replace_all(CurrLine,' ,',",");
//Get the # of items to be placed into the map, if there is an
     // odd number, round up and we will default to 0.
     NumItems = ceil((string_count(',',CurrLine) + 1)/2);
     // Read The Values into the Map
     for( i = 0; i < NumItems; i += 1 ) {
          if ( string_count(',',CurrLine) > 1 ) {
    TempKey = string_copy(CurrLine,1,string_pos(',',CurrLine)-1);
    CurrLine = string_delete(CurrLine,1,string_pos(',',CurrLine));
    TempValue = string_copy(CurrLine,1,string_pos(',',CurrLine)-1);
    CurrLine = string_delete(CurrLine,1,string_pos(',',CurrLine));
        }else{
            if ( string_count(',',CurrLine) = 1 ) {
                TempKey = string_copy(CurrLine,1,string_pos(',',CurrLine)-1);
                CurrLine = string_delete(CurrLine,1,string_pos(',',CurrLine));
                TempValue = CurrLine;
                CurrLine = '';
            }else{
                TempKey = CurrLine;
                CurrLine = '';
                TempValue = 0;
            }
        }
        ds_map_add(MapItems,TempKey,TempValue);
    }
    return (MapItems);
}
```

Map Data Structure Saving

```
/*
**  Usage:
**      ds_map_save(map, file [,delim])
**
**  Arguments:
**      map         ds_map data structure to save
**      file        file name, string
**      delim       delimiter used between elements, optional
**                  0 = Carriage Return (default)
**                  1 = Comma (for CSV files)
**                  2 = Semicolon
**              other = a string of your choice
**  Returns:
**      nothing
**  GMLscripts.com
*/
{
    var FileOut, NextKey, i;
    FileOut = file_text_open_write(argument1);
    if ( ds_map_size(argument0) > 0 ) {
        NextKey = ds_map_find_first(argument0);
file_text_write_string(FileOut,string(NextKey)+', ');
file_text_write_string(FileOut,string(ds_map_find_value(argument0,NextKey)));
        NextKey = ds_map_find_next(argument0,NextKey);
        // Define the format to use
        switch (argument2) {
            case 0: file_text_writeln(FileOut);break;
            case 1: file_text_write_string(FileOut,',');break;
            case 2: file_text_write_string(FileOut,';');break;
            default:file_text_write_string(FileOut,string(argument2));break;
        }
        if ( ds_map_size(argument0) > 1 ) {
            for ( i = 1; i < ds_map_size(argument0); i += 1 ) {
file_text_write_string(FileOut,string(NextKey)+', ');
file_text_write_string(FileOut,string(ds_map_find_value(argument0,NextKey)));
     NextKey = ds_map_find_next(argument0,NextKey);
                // Define the format to use
                switch (argument2) {
                    case 0:
                    file_text_writeln(FileOut);
                    break;
                    case 1:
                    file_text_write_string(FileOut,',');
                    break;
                    case 2:
                    file_text_write_string(FileOut,';');
                    break;
                    default:
                    file_text_write_string(FileOut,string(argument2));
                    break;
                }
            }
        }
    }

    file_text_close(FileOut);
}
```

Conclusion

Loading and saving maps gives us the ability to confidently use them in games to help in complex situations.

Eyas Sharaiha■

The Game Maker Race

The Game Maker Race is a competition created by collaboration between MarkUp Magazine and GMTech Magazine. We've announced it and discussed it in previous issues, and now, the 20 games have become 5 after a tough couple of rounds of 'Head-to-Head'!

We'll go through the processes which lead the top 20 games to become five.

Round 1

Basco's Portals vs. Forest Planet

Basco's Portals: Nice concept - but there's not much to help the player out. Having a player construct portals and move does seem like a nice idea, but it had some problems: first of all, the only thing related to fire and ice was the portals themselves -- they had no power other than to distinguish 'two types' of portals. There was no background music, which kept the game feel like its missing something; a single sound effect was there, which is the player's shooting of fire and ice blocks, though I wasn't too annoyed by this. The difficulty curve was too steep; I was unable to complete level 2 for a long time. I don't mind the actual difficulty, but the player isn't provided with any sort of tips for the first couple of levels. We are never introduced to the game elements and are left to try them out by ourselves. There was no menu, and the game starts abruptly after seeing the 'Show Information' dialog for Game Maker. My biggest problem was that also: you never lose; there is no such thing as lives: if you fall in a hole, you keep trying again.

4/10

Forest Planet: Cool game overall. Fire and Ice is integrated as a concept of 'temperature' in the game. A too high or too low temperature would cause destruction -- I like that, not too much integration but certainly okay. The game had a cool cartoonish comical feel overall, sortof 'cute'. I found the game graphically above average (if not very good), and it had great style and elegance. However, I *was* disappointed by the lack of background music -- although at times of 'action', the sound effects of cutting and burning trees, or attacks does give the player a feeling that "something's going on". I was disappointed to see little help or guidance, but 'tooltips' here and there were still very helpful. The player is welcomed to the game with a nice comical introduction to the events of the game, before having the real action start. Over all, I was really impressed by the game, although I missed a couple of small features which would have made the game more meaningful.

7/10

Eyas

Eternal Dawn vs. Hero of Ankston

Eternal Dawn: First look at the screenshots, made me excited to see what sort of things I should expect, nice intro to game, with a good text engine, great looking GFX, not much to do with Fire and Ice (red clothes and blue clothes) a whole load of walking, sad music and not much else, however I think just because of the look and feel of the game, its slightly ahead.

5/10

Hero of Ankston: I think more work could of gone into making the game more playable, as I just ran around amilessly, until I found some enenys, whom I could easilly run passed (as there AI was very simple), it does get extra marks for having a good storyline, more work could of gone into the programming of the main character, as there was a few qlitches, here and there (like jumping amazingly high)

4/10

gamez93

Element Mine vs. Elements of Escape

Element Mine: This game has a nice look and feel. Basically, it's an arcade-style game where you have to keep your player alive as long as possible. You lose health each time you collide with a variety of moving obstacles. As far as I could tell, the winning strategy just relies on good reflexes and a bit of luck. There wasn't much in the way of the fire and ice theme. The player and obstacles periodically change from red (fire?) to blue (ice?). But that's about it. So I can't rate it highly on "interpretation of theme". But the graphics look nice and there are some nice effects when the player moves (glowing tail) and when obstacles are hit. Overall, it's pleasant to look at, and the game play is OK - although at certain times, the game lacked action. But it losses points for lack of 'creative integration' of the fire and ice theme.

5/10

Elements of Escape: This game provided a very good interpretation of the fire and ice theme. The game is a series of challenges where the goal is to get two characters (fire and ice) to the exit door of each level. The player does this by switching from one character to the other. The "fire character" can burn down obstacles, and the "ice character" can place temporary ice blocks which can be climbed. Finishing each level requires the two characters to cooperate in order to reach the exit. There is a short tutorial at the beginning to show how the game works. Everything is well-explained and the game play is fairly intuitive. The graphics are very basic and rather unappealing. But the game play is pretty good and the challenges are interesting. Most important, this is a very good interpretation of the contest theme.

7/10

KC LC

Orbiter vs. Fire and Ice

Orbiter: This was a fairly standard platform / exploration game. Your character lands, and you must explore the game world. The instructions are rather vague. They explain how to dash and shoot... but not how to move. They also refer to a "save station" which I could not find. But the usual arrow controls worked, so I suppose the author figured it would be obvious. Overall, the game was playable and it ran smoothly. I was able to move about without any glitches. The graphics were well made and it was clear the author put some effort into it. There were a few things to explore, but not much to really draw my attention. Sadly, the use of the fire and ice theme seems missing altogether. So although the game was reasonably well made, I can't score it very highly for this competition.

4/10

Fire and Ice: This is an old-school asteroid-type game. The goal is the kill the attacking ships while staying alive. The enemies come in two forms: red and blue (presumably fire and ice...). The player can change forms as well, which changes his vulnerability to attacking ships. The graphics are like those old arcade games. This is done intentionally, but it still makes for a rather boring appearance. The game had a very well-explained intro screen that described the movement and scoring system. Unfortunately the actual scoring in the game didn't seem to follow the rules. I was able to score several thousand points by just crashing my ship into anything that moved without ever firing a shot or changing my color. Worse, the game is flawed. On two occasions it generated the error message:

```
ERROR in action number 2 of Draw Event for object obj_enemy_ice:
Error in code at line 1: if drawmine=1 at position 4: Unknown variable drawmine
```

The game was able to recover and continue working, but errors are errors.... The only use of the fire and ice theme was red and blue colored enemies, and a player that could shift back and forth. Not very imaginative, and the error hurts.

2/10

KC LC

Save the Ice Caps vs. Freeze, Melt, Vaporize

Save the Ice Caps: A very nice game and a great concept. However I think it is very difficult. The sprites of the tracks do not differ very much thus it is difficult to remember where to find the next track you need. So either the sprites should be changed or you should have more time.

7/10

Freeze, Melt, Vaporize: A damn cool... no hot... umm whatever. A GREAT game! The documentation is a bit, but when you get to understand the concepts and the controls of the game, it is fun and entertaining. You need to keep your head cool while you are melting or freezing you from the beginning to the end of each level. So as I said: A great game with an original concept. The only minus is the documentation which should be a little more structured.

8/10

Snabela

Triton vs. Ball in Hall

Triton: WOW. If I am not able to spell the words correctly, it is most likely because I am a bit giddy. It's an excellent and original game... At least I have not seen anything like this before (maybe except Kula World, but that's a bit different). The first time I played this game, I used YYG Instant Play and were disappointed to hear no sound or music even though there are options to turn it on and off. I downloaded the game and bingo – now with sound and music. I am not sure what this "Super Sound System DLL" does but maybe it is possible to make it use the built-in sound functions in GM if the DLL is not useable. I think the levels get harder to fast. The first levels was piece of cake while the second one was much more tough. These "turn 180 degrees"-moves are pretty difficult, but maybe I am the one to blame as I played it on my laptop with the laptop mouse.

8/10

Ball in Hall: A game with nice graphics, and lots of effects (sometimes too many - it slowed down the game)... But not much new thinking in this game. The theme was fire and ice, and this game is a classic breakout game where the ball is the ice and the bricks are the fire.

5/10

Mount Contradiction vs. Jiang

Mount Contradiction: I like the way it explains how to play the game. But pressing F1 for help would have been useful if I forgot something. it gets increasingly harder and introduces new elements to the gameplay as time goes on, which is very good. The gameplay is great, addictive but I would say a bit too easy. You can get more than half way though the game with limited skill. I got stuck on the first level at the flag as I didn't realise you had to press enter... strange! I like the game idea/story. It would have been better with the use of views. Graphics are quite nice, but the flag at end should have had transparent background. They would have looked nicer if they were bigger! The music add a nice background but would be better if it changed from time to time and if there was a way to turn it off.

7/10

Jiang: Music is really good almost worth downloading the 21mb for it. Help file completely useless, it didn't say how to save the game, only to use the arrow keys to guide the ship which is the only thing I figured out to do on my own! Very easy to die, good games start off easy and get progressivly harder. I HAD to use this game using instant play as the exe wasn't compatible with vista, wasn't this a rule or something? As for the graphics, they are average, could have been more detailed, could have been worse. Nothing really new gameplay wise, is it just me or have I played this type of GM game thousands of times, also not really related to fire and ice apart from the gems. Menu's are quite professional, but it should say press enter and use arrow key as other wise the user couldn't even use help. Initially I was using the mouse to click, you could have hidden the mouse on that screen as it is useless.

5/10

TGMG

The Meltdown vs. Acre City Madness

The Meltdown: Spelling mistakes (live instead of life for example) but I will ignore that. I like it how it walks you through the levels with messages. Music is very good for this game. I also like that you can turn music on and off and it changes track when you do. It doesn't get annoying so it is perfect for this game. The graphics are also very good although the background could be a little more interesting. Also if the blocks could have been improved. The particle effects are very well done. I like the fact that you can set enemies on fire and that you have to switch to solve puzzles. I love the game idea and it works very well. It gets progressively harder and is actually quite challenging at parts. Also very glad that you included a help file! Sadly I found a very annoying bug, when you die (game over) and click the start button again it says game over again as you have not reset the lives. I enjoy playing this game and could get addicted. A game I could play for hours. This is a potential winner. Also it is very related to the ice/fire part of this comp.

8/10

Acre City Madness: Music is ok but can get annoying, no obvious way of turning sound off. No help file so I had to guess the buttons. No story. the graphics are average. The game play can be fun until you die for the third time then it just gets repetitive. Too easy to die, no lifespan. It should start easy and get harder as time goes on but it is hard right from the start. the scenery goes past too fast, makes it hard to play the game for a long amount of time. Sorry but I couldn't play this game for longer than 10 minutes.

4/10

TGMG

Fire and Ice vs. Galexion Invaders

Fire and Ice: A very boring concept. Bad quality images, easy gameplay, one level in a concept that needs to be level-based.

1/10

Galexion Invaders: Interesting alteration on a popular classic. A bit confusing and over-the-top at times, but well designed and clear that much time has been spent on it.

4.5/10

Rhys Andrews

Freeze vs. Firepong 3000

Freeze: A very original little game where you must freeze the intruders squashing your civilisations buildings and throwing fireballs at you (raising your core temperature – this is what will ultimately kill you and submit your highscore). The voices weren't that great, and other than more intruders there wasn't much change in the gameplay, but it had cute little graphics, and a simple but sweet storyline and gameplay.

7/10

Firepong 3000: I'm afraid I didn't see anything special in this game. It was a simple game of pong, with a blue and a red bat. Nice little sound effects, but that's about it – other than that I would've been happier playing Game Maker's tutorial pong (if it's still around).

2/10

The Game Maker Race

Rhys Andrews

Round 2

Forest Planet vs. Triton

Forest Planet: Help your bears to take over the opponent's cave... And how to do that? Keep them warm. A great idea with great graphics in a great game.

7.5/10

Triton: Control the room to get the ball the finish. A nice idea and again with great graphics. These two games are very close.

8/10

Snabela

Eternal Dawn vs. the Meltdown

The Meltdown: The better game of the two, kept me hooked and playing for a good ten minutes.

6/10

Eternal Dawn: Not as good, it didn't keep me hooked to it, and I didn't really like just roaming around

4/10

gamez93

Elements of Escape vs. Mount Contradiction

Elements Of Escape: Nice game, enjoyed the goal of changing your character and using their skills.

6/10

Mount Contradiction: Another platformer game, it's a shame both games were put against each other, as they both deserve to go through, nice GFX, and nice progression of levels.

7/10

gamez93

Galexion Invaders vs. Orbiter

Orbiter: There's an overall lack of help, instructions, and indication -- we don't know what we have to do. Nevertheless, it's a smooth and cool game. Fire and ice didn't fit that well, although it was minorly integrated within the storyline. That's fine.

6/10

Galexion Invaders: The game tries to achieve a retro look -- but the graphics hurt my eyes and the sound hurt my ears. The game was fine, a cool and smart idea, but certainly not up to level with some of the great games we expect to moving to the final voting round. I just hoped the game was less cluttered overall.

3.5/10

Eyas

Freeze vs. Freeze, Melt, Vaporize

Freeze: The "fire and ice" theme was captured fairly well here. I liked the graphics and sounds. They were simple, but effective for the game. One disappointment was the lack of variation, so the game play got repetitious fairly quickly. Still, there was some re-play value to get a higher score in less time. One complaint would be the controls. I found the use of diagonal movement on an isometric grid a bit frustrating. Also, on one occasion the game appeared to lock up when I froze one of the enemies near the boundary. But I couldn't repeat this, so I won't consider this an error. Overall, the game was quite good. Good use of the theme, and fun to play.

7/10

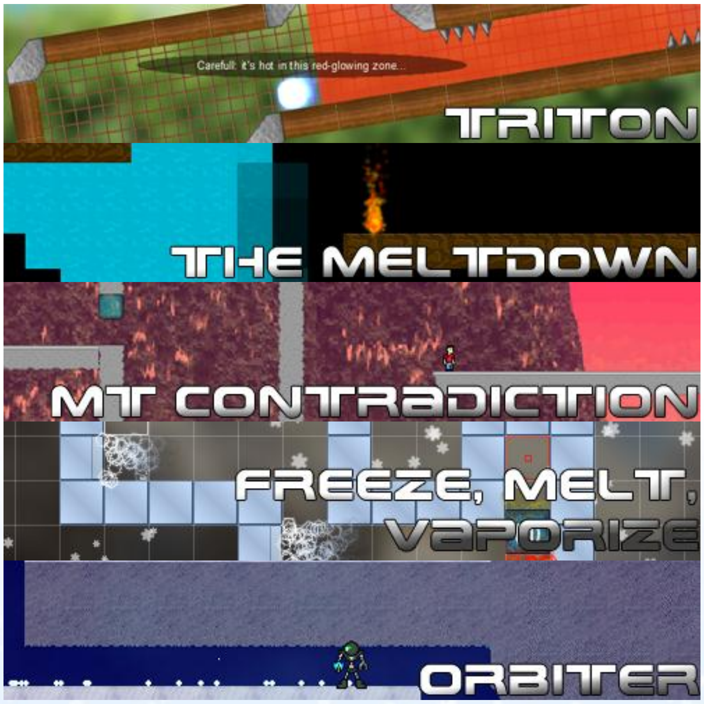
Freeze, Melt, Vaporize: I thought this was a very good game. It made excellent use of the contest theme. I thought the idea behind it was very original, and the puzzles were cleverly constructed. There is a bit of learning curve at first, because the controls are somewhat complex. But the author provided a very good tutorial (along with a YouTube video). So that made it easy to understand, and I was playing it within a few minutes. The graphics were basic, but they were very well suited to the game. Overall, I thought it was an excellent entry. The best I've seen so far.

9/10

KC LC

The Top 5

After two tough rounds of Head-To-Head, we finally were able to narrow down the amount of games we have from 20 to 5! The top 5 games are as follows (in no particular order):



Final Voting

You can vote to one of the games above and help it become one step closer to the final prize by navigating to the main site for The Game Maker Race:

http://www.thegmrace.com/

Eyas Sharaiha■

Auto Zoom

Today I want to share with you a use of the views feature in Game Maker that you may find interesting.

I came up with this after pondering the game Super Smash Bros® from Nintendo, which my 6-year old son plays almost non-stop. In that game, the view will automatically zoom in and out to keep as many of the four characters in view at a time as possible. (Note that Super Smash Bros uses 3d graphics but the action only happens in 2d, so this is really a 2d method and I'm using it for pure 2d game.

So how can you do this in Game Maker? It would seem to be a simple matter of just checking all the positions of the characters and then changing the view size and position to fit. However, it is not that simple because: a) you need to stop the view from going outside the boundaries of the room, and b) you need to keep the view proportions scaled the same as the port proportions to ensure no distortion.

Background Topics

While this is a novice topic, there are some intermediate concepts used: views, parent objects, and the *with* statement.

Views

To use views, you need to enable them. That is done in the Views tab in the room-editing window. Views can be confusing at first because there is a view and also a view port. This is the way I keep them straight:

> The view is the area of the room to be displayed, and the view port is the area on the screen in which the view is displayed.

There are number of built-in variables (`view_xview`, `view_yview`, `view_hview`, `view_wview`) that control the position and size of the view. Similarly for the view port there are built-in variables (`view_xport`, `view_yport`, `view_hport`, `view_wport`) that control the position and size of the view port. We'll naturally be using these variables a lot.

Parent Objects

Most people use parent objects to help propagate common events to children objects. However, parent objects without any events can be very useful for simply grouping dissimilar objects for processing. For this example I wanted to make it generic (i.e. to allow any type of to objects stay in view, not just player characters), so I use parent objects for this.

The *with* Statement

This is a key concept for Game Maker. In fact did you know that all the "drag & drop" actions invoke the `with` statement? (When you select the self, other, or object option at the top of the drag & drop action you are actually setting up the corresponding `with` statement.)

The `with` statement will repeat the following code block for all instances of the specified object. We need this, since we will be scanning the positions of all the instances to keep in the view.

Okay, Now The Real Work

The View Controller

A "controller" is an object that you may not see but that executes events and code necessary to control the game. It is a good idea use controllers in your games, rather than trying to stuff all the game code into the main player object.

First of all, you will need to create an object called `objViewController`. It should have no sprite, and probably should not be solid or visible. Don't forget to place it in your room!

There will only be two events for the view controller, the Create event and the Step event. The Create event will

Auto Zoom cont.

just call a script called `scriptGlobalVars` and the Step event will call a script called `scriptKeepInView`.

The Parent Object

Next, create an object that will be used to tag all the objects you want to keep in view. Simply make this object the parent of any other object and that other object will be kept in view.

The object should be named `objKeepInView`. It should have no sprite, and no Events. It should probably not be solid or visible.

Objects You Want Kept In View

For any object you want kept in view, in the object editor window just select the parent to be `objKeepInView`. (If you already have a parent, then make a parent of the parent.)

The *scriptGlobalVars* Script

This script simply sets the value of a couple global variables.

The first global variable, `view_border`, indicates the number of pixels outside the outermost object you want the view to show.

The second global variable, `height_over_width`, is the ratio of height to width of the view port that needs to be maintained in the view to prevent distortion.

The *scriptKeepInView* Script

I'll discuss this script in sections.

The first thing the script needs to do is figure out the outermost positions (`min_x`, `max_x`, `min_y`, `max_y`) that need to be displayed. It does that by looking at the *x*, *y* position of all objects of type `objKeepInView` (and its children). This will require the use of the `with` statement.

```
// this script will adjust the view[0] such that the
object that called it is in the view
```

```
// assumes that the sprite origins are at top left,
so need to adjust if different
// initialize local variables to first child object
of objKeepInView
min_x = objKeepInView.x ;
max_x = objKeepInView.x ;
min_y = objKeepInView.y ;
max_y = objKeepInView.y ;
// loop to find actual minimum and maximum positions
with (objKeepInView)
{
if (x - global.view_border) < other.min_x then
other.min_x = x - global.view_border ;
   if (x + abs(sprite_width) + global.view_border) >
other.max_x then other.max_x = x + abs(sprite_width)
+ global.view_border ;
if (y - global.view_border) < other.min_y then
other.min_y = y -global.view_border;
if (y + abs(sprite_height) + global.view_border) >
other.max_y then other.max_y = y + abs(sprite_height)
+ global.view_border ;
   }
```

This code above will loop through all instances of `objKeepInView` (and its children) and figure out the minimum and maximum points in the room (including the extra `view_border` space) that need to be displayed in order to keep all the objects in view.

Note that I used the absolute value function, *abs()*, when using the built-in `sprite_width` and `sprite_height` variables. That is because the width and height will return negative values if the image scale (`image_xscale` and `image_yscale`, respectively) are negative.

Anyway, you might think the script would be finished now, but there remain two problems: the `view_border` might have caused the view to go outside the room, and the proportions of the view may not match the view port.

So the next step is to ensure the top left corner and bottom right corners are in the room:

Auto Zoom cont.

```
// keep top left of view within room
if min_x < 0 then min_x = 0 ;
if min_y < 0 then min_y = 0 ;
// keep bottom right of view within room
if max_x > room_width then max_x = room_width ;
if max_y > room_height then max_y = room_height ;
```

Now we need to adjust the proportions, and this will be "forced" by either the horizontal or vertical spacing of the objects that are to be kept in view.

```
zoom_horizontal = max_x - min_x ;
zoom_vertical = max_y - min_y ;
// need to determine whether the vertical spacing or
the horizontal spacing is forcing the zoom amount
// because proportion of height to width must match
view or the view will get distorted
if zoom_vertical > global.height_over_width *
zoom_horizontal // vertical is forcing zoom
    {
    zoom_horizontal = zoom_vertical /
global.height_over_width ;
    }
else // horizontal is forcing zoom
    {
    zoom_vertical = global.height_over_width *
zoom_horizontal ;
    }
```

Now due to the previous adjustment it is possible that the zoom is larger than the room, in which case the zoom should be scaled down to the room size.

```
// if either zoom is bigger than room, set view to
full room
if zoom_horizontal > room_width or zoom_vertical >
room_height then
    {
    zoom_horizontal = room_width ;
    zoom_vertical = room_height ;
    min_x = 0 ;
    min_y = 0 ;
    }
```

Lastly, we need to check again to keep the view from going outside the bottom right corner of the room:

```
// keep bottom right of view within room
if min_x + zoom_horizontal > room_width then min_x =
room_width - zoom_horizontal ;
if min_y + zoom_vertical > room_height then min_y =
room_height - zoom_vertical ;
```

Finally, we can update the view!

```
// adjust the view
view_xview[0] = min_x ;
view_yview[0] = min_y ;
view_wview[0] = zoom_horizontal ;
view_hview[0] = zoom_vertical ;
```

Applying It to Your Game

After creating the controller object, parent object, and scripts described above, you can apply this to your game by simply:

Enable Views

In any room you want this effect, enable `View[0]` in the room editing window. Set the "Enable The Use Of Views"

Quick Review

Room Import & Export Scripts

This issue has an amazing record of quick review for Kyle_Solo, and this is just one of the many quick reviews you might've spotted all over the magazines dedicated to a creation of his. The Room import and Export Scripts allow anyone to 'save' a room, with all of its objects and contents, for them to be loaded later. This is incredibly useful for level editors and level packs for games. The scripts export almost all elements of a room, however, as of this issue's release date, the scripts do not support tiles.

gmc.yoyogames.com/?showtopic=341944

Auto Zoom cont.

checkbox, select View 0 and set the Visible When Room Starts checkbox.

For the View In Room, set the X, Y, W, H to any value. It doesn't matter because these will be updated in the first step of the game by the `objViewController`.

For the "Port On Screen", set the "X", "Y", "W", "H" to fill up the whole screen. For example, at 600 x 800 resolution you should set X=0, Y=0, W=800, and H=600.

No not enable object following here; for Object Following select <no object>. Our script would do that.

Set Parent Object For Objects To Keep In View

For every object you want to keep in the view, simply set the parent to be `objKeepInView`.

A Known Limitation

One known limitation is that the room has to have the same proportions as the view port. Why? because otherwise it is impossible to ensure all objects will be in view at the same time the view is kept within the room.

In Super Smash Bros, they solve this by allowing the characters to sometimes go out of the view. In that case, they place an indicator on the edge of the view to point to the out-of-view object. Maybe someday I'll update this script for that functionality, or better yet: you should try it!

Closing thoughts

I've applied this auto-zooming to a 2d fighting game and I like the effect. Try this on different types of games – it can provide a lot of visual interest in any action game.

Remember it is possible to change the parent object for an object during the game, so you can change which objects are kept in view during the game.

Anyway, I hope this inspires you to try other auto adjustment effects using views!

Julian Abelar■

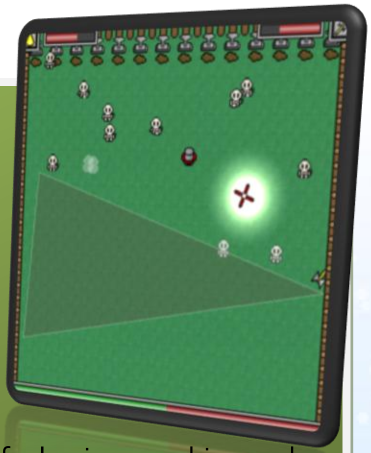
Game in a Month

Tied Winner #1

For this month's 'Game in a Month' competition by Reflect Games, two games managed to tie for first place. The first of them being 'Zombeez' by 'Kenon'.

Zombeez has been described my Martin Crownover, who is organizing the competition, as "space invaders gone zombified!"

The game is indeed similar to Space Invaders; zombies are invading the earth and you have to stop them, this time, with your mouse. Stopping zombies with the mouse seems like this is even easier than the original space invaders, but the increased speed, in addition to the time delay between shooting the zombies, as well as the new monsters that require multiple shots before dying.



The game has a nice overall feel, nice graphics and gameplay. But gives no orientation to the player and provides nothing other than 'gameplay'; certain "additions" are needed to complement the game.

Eyas Sharaiha■

C++ Classes and Data Structures

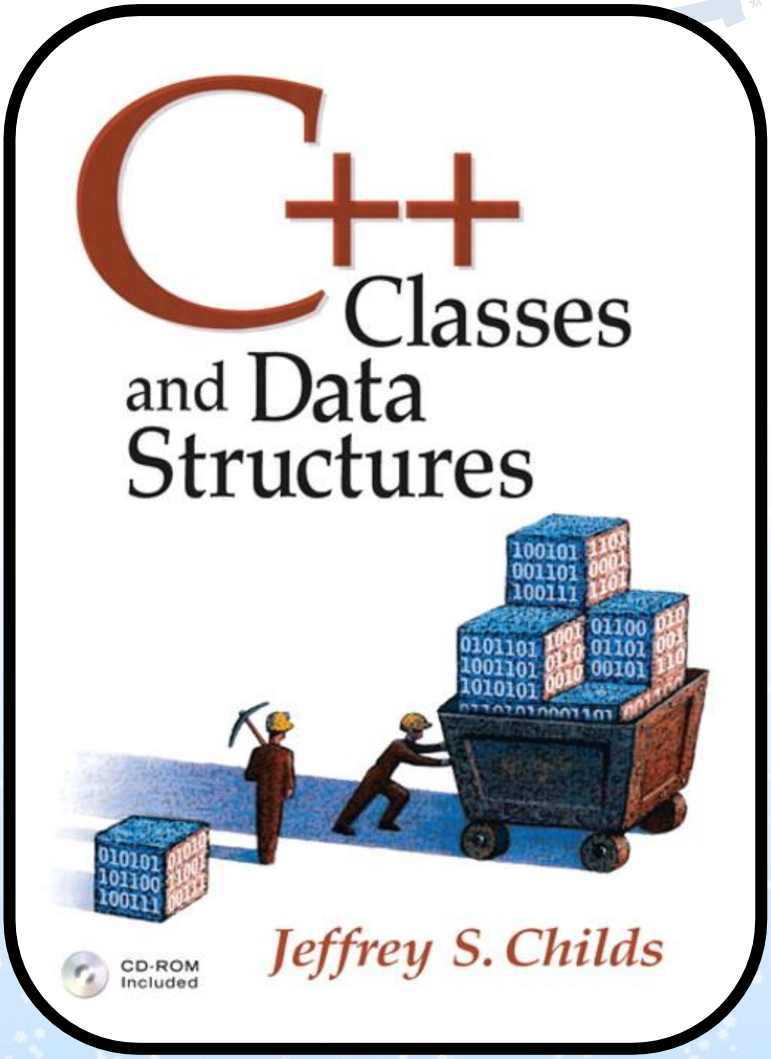
Book Review

C++ Classes and Data Structures is a first rate introduction to data structures and how to program them in C++. Of course, everything in the book can easily be adapted for use in Game Maker but to do so you will need some basic knowledge of C++ and/or object oriented programming in general; most of the time just knowing GML will suffice.

Written by Dr. Jeffrey S. Childs for his Clarion University of Pennsylvania classes and published by Prentice Hall (www.prenhall.com), it is a strait forward approach to data structure design and application that starts at the beginning and covers a great deal of subject matter including structs and classes, operator overloading, object oriented programming, recursion, and many types of data structure and the means and methods for constructing them.

Contents

I unboxed the book to find a small, neat, textbook of about 400 pages (including an appendix) and a CD-ROM containing a copy of Microsoft Visual C++ 2005 Express Edition. The book appeared well structured, starting with an introduction to classes and ending with chapters on recursion and sorting algorithms. The tone of the book turned out to be casual, enjoyable, and easy to comprehend. After browsing through the table of contents and reading an excerpt from the first chapter I was already very pleased. The book begins at the beginning, assuming that you know nothing about data structures or classes but that you have at least had an introduction to C++. It begins by teaching the building block of data structures, classes and arrays, and by introducing the reader to abstraction and operator overloading. It then gives a brief introduction to object oriented programming including inheritance and polymorphism (a must read for all GM users) and then begins on methods for creating data structures. After that it moves on to discussion of the standard, well defined, structures; many of which are already available in GM or in the C++ STL. It covers basic time complexities, Big-Oh Notation, logarithmic time complexity, amortized analysis of array expansion and contraction, etc. and wraps up with a chapter on recursion followed by an introduction to data sorting using several common sorting algorithms including heapsort, insertion sort, quicksort and counting sort, several of which are quite easy to implement using only basic GML. I was also pleased to note that it contains an appendix on compiling code in multi-file programs. While this chapter is not applicable to GM specifically it helped make the book a singular reference which needed no other



C++ Classes and Data Structures

complimentary texts.

Audience

It would be pertinent to note that *C++ Classes and Data Structures* is written as a text book, albeit a very casual one, and is therefore geared towards students, not the casual game developer. While it is an extraordinarily useful text from which both beginners and experts can glean a few tips, if you are not competent in GML or C++ this book is probably not for you.

Pros

- Casual and easy to read
- Assumes very little knowledge of the topic and covers a great deal of content.

Cont.

- Easily adaptable to any object oriented language including scripted languages such as GML.
- Uses real world abstractions and examples
- 7x9x1 inches; easy to carry if required for a class.
- Very complete; Can be read from cover to cover as a singular reference without the need for supplemental texts.

Cons

- Assumes knowledge of pointers and memory allocation. However this is not relevant to GML.
- Short, could cover more sorting algorithms and data structures.
- Often does not go into a great deal of depth. (Geared towards less advanced users)
- CD Does not contain code samples (may be downloaded from a website).
- Could go into more theory.
- Needs more material on real-world abstraction.

Quick Review

Title	*C++ Classes and Data Structures*
Author	Jeffrey S. Childs
Publisher	Pearson (Prentice Hall)
ISBN-13	9780131580510
ISBN-10	0131580515
Year	2007
Pages	~400
CD included	Yes
Overall score	7/10

Leif Greenman■

INI Opening Script

While Game Maker's built in INI opening and reading mechanism **is** good, it does not allow you to open files unless that file is from the same directory as the game itself.

This script allows INI files from other directories to be opened, and is faster than other scripts that only copy the file then open it. It's almost as fast as the built-in GM script.

gmc.yoyogames.com/?showtopic=291349

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

Beginning Game Development

with Python and Pygame

Beginning Game Development *with Python and Pygame: From Novice to Professional* is a book for beginners to begin their game development with Python and Pygame. The book is written by Will McGugan, a Scottish 31 year-old at the time of writing, who is a game developer working for Evolution Studios, worked on the PlayStation 3 title *Motorstorm*. The book is published by Apress.

Book Review

My first impressions on the book were quite positive: it seemed like a decent introduction to game programming from a quick glance, and includes enough explanations and introductions to all programming concepts.

Content

The book has 316 pages and has 12 chapters, which first introduce Python basics, then Pygame basics and 3D functions. Appendix A and B discuss distribution of your completed game.

Who's this book for?

This book is meant for beginning Python and Pygame developers, who should find it very useful for learning the basics, with some drawbacks. Programmers that are already familiar with Python will also feel that the book is useful.

Relevance to free software

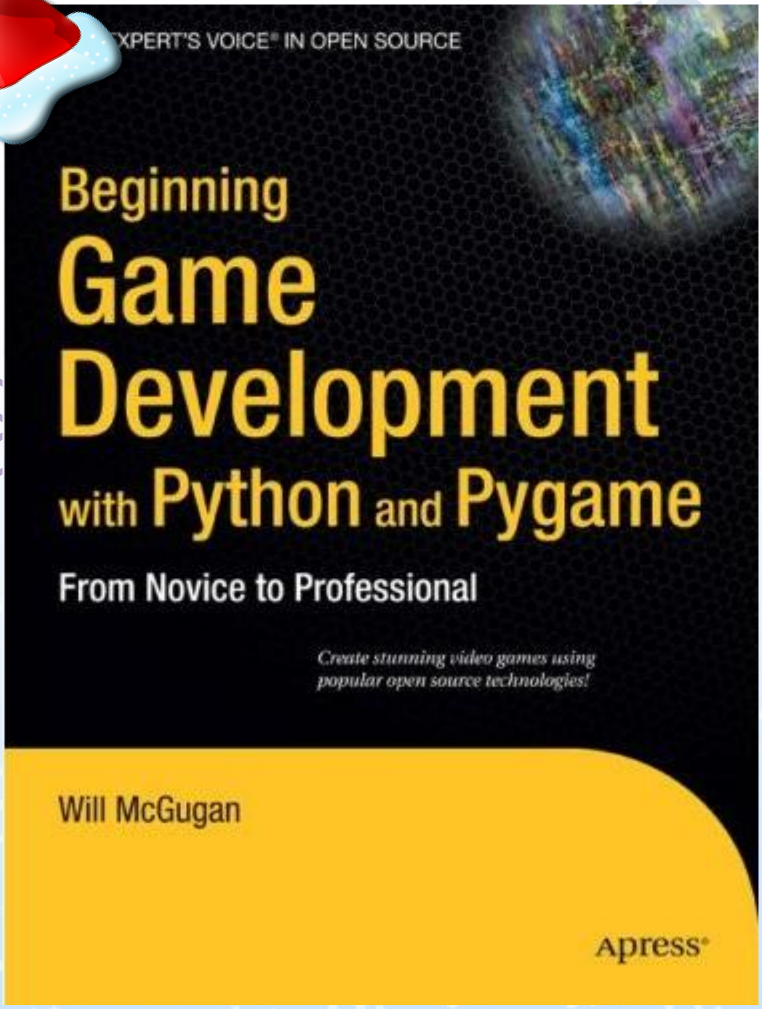
The book is weighted towards free software, due to the free software nature of Python and Pygame. Many free applications are introduced in the course of the book, and the user is encouraged to use these as development tools.

Review

This book is certainly a useful introduction to the world of game development: anybody with some experience with computers should be able to understand the language used in the book. Python and Pygame in my opinion both form an excellent option to start developing games, and the book introduces all the basic concepts and advanced features, including AI and 3D programming.

The first two chapters focus on introducing the reader to the basics of Python: chapter 3 then introduces Pygame basics, with further chapters introducing the reader to more advanced Pygame functions and features.

I am disappointed that no actual game is made during the course of the book: the book simply explains the


EXPERT'S VOICE® IN OPEN SOURCE
Beginning
Game
Development
with Python and Pygame
From Novice to Professional
Create stunning video games using popular open source technologies!
Will McGugan
Apress®

Beginning Game Development

with Python and Pygame Cont.

basics and some few advanced features, while not guiding you through a complete game. The title "*Beginning Game Development with Python and Pygame: From Novice to Professional*" implied that, however, the book might guide the reader through a complete game. I was certainly expecting that, for example, a simple Pac-Man or Pong clone would be made, but instead only tech demos showcasing basic & advanced functions are done. While this may be sufficient for an experienced programmer looking to start game development with Python and Pygame, a beginner will easily feel lost, and won't be able to make his/her own games without outside help.

The book also does not include any media (CD/DVD) at all; instead, this must be downloaded off the publisher's website. This is also a minus, because a CD in most cases would be more convenient. The publisher also has a limited offer lasting until April 2008 that allows a book owner to buy a discounted e-book version by answering a question related to the book (such as the last word on a page) in a limited amount of time.

In overall, the book is a slight disappointment for my type of people, but was at the same time an asset to learn certain technical features of Python and Pygame. Having no examples of actual games is a big minus, however. I suggest that people planning buying the book consider on looking at other books first as an introduction, unless they are experienced programmers.

Pros

- Introduces basics
- Goes from basic to advanced
- Clearly explains everything

Cons

- No CD included
- Does not actually guide the reader through the process of making a game

Title	Beginning Game Development with Python and Pygame: From Novice to Professional
Author	Will McGugan
Publisher	Apress
ISBN	1-59059-872-5, 978-1-59059-872-6
Year	2007
Pages	316
CD included	No
Overall score	7

Veeti Paananen■

Game in a Month

Tied Winner #2

The other winner, tied for first place, is 'Nightmare Manor' by Tahnok100.

Nightmare manor has a simple storyline: your little sister is lost in a haunted mansion and you need to find her.

To find your sister, you need to go through the haunted mansion and all sorts of disasters hidden in it.

The game's graphics were indeed really well done. Tahnok100 indeed did pay attention for the details, both in the graphics themselves and their animations.

The major problem with the game is that it gets hard too fast. That's not a big problem, cause it usually challenges the player to try again and again. I wouldn't mind, though, if the game had more 'introductory' levels to techniques so that the player fully gets used to the controls.

Eyas Sharaiha■

Faction Wars

Development Journal

With Christmas on the horizon, the Faction Wars team has knuckled down to meet our deadlines, finalise and tick off our outstanding tasks, and to get ready to have something special to show to our fans in time for the New Year.

In fact, as it happens, we've already got an impressive array of new content running happily in our current release that's yet to be seen by the general public. So where is it, and why haven't we bundled it up with this article to show off?

This month's article looks at Structure, Goals, and Releases.

Faction Wars is a long-term project – one that's been underway for a number of years.

Though our team began the project as inexperienced developers, learned as we've created, and we're still improving today. Faction Wars has evolved, and it's only through the use of Structure, Goals and Releases that we've got to where we have.

Right now, our current release features special effects, combat, GUI components and even the beginnings of our AI system. However, we're not going to release the game as it is, as none of the content in it is polished or finalised. In fact, *almost everything in the current demo is likely to change completely and beyond recognition over the next few weeks*.

So why have we put all this work into creating functionality and effects that we're getting rid of?

Faction Wars is built on an underlying structure. Our attention is focussed developing the core game engine, which allows us to create and edit content quickly and easily. Though obvious visual progress is slow to emerge, the more work that we put in to making the game engine adaptable and systemised, the less work will be involved in adding and changing content, weaponry, menu systems, AI, missions, and more, later down the line. This also allows for a consistent, steady pace of development - rather than sporadic bouts of new content and burnout periods.

Structured Development

Along the way we've tried, thrown away, retried and eventually settled on several key principles of good development strategy. The most important of these is 'Mapping'.

When developing engine-based models, (that is to say, when your core development work involves building

Quick Review

Professional Menu Scripts

The 'Professional Menu Scripts' by Mordi is, without question, an excellent set of scripts that allows a user to create professional looking, functional game menus.

The scripts include different types of buttons, for input, scrolling, etc. The scripts also are capable of drawing different boxes with cool effects, including shadows and glows.

gmc.yoyogames.com/?showtopic=318527

Faction Wars cont.

infrastructure for further development, such as with Faction Wars) it's imperative to know exactly what you're developing before you start. Begin by establishing all of the conceivable variables involved in your system's functionality, and break these up into individual sections. Do the same for your interface, visuals and any other resources.

The Faction Wars 'In Space' GUI – a working example

The first stage of developing Faction Wars' GUI was to break it down into individual components. Every button, every piece of functionality, every variable was identified, then built into a functional group. One of our key points of discovery during this process – which we may well have overlooked entirely otherwise, thus causing a large amount of extra work and revisions down the line – is that Faction Wars has two distinct GUIs.

Though they are both similarly presented and visually styled, it occurred to us that during gameplay, the player's interaction with the GUI would be drastically different whilst 'in space' and 'docked'.

When in space, flying around and engaging in mining and combat, almost all of the GUI would be 'passive' – the player would view a large amount of structured information, but there would be very little interaction through the GUI that would affect its contents. The 'docked' GUI, however, involves trade, ship configuration, missions, and much more. Suddenly we discovered that what seemed like a mountainous task could be split easily into two distinct parts, and those could be further subdivided into functional areas.

The example here shows a screenshot of our 'In Space' GUI modelled in Microsoft Visio (which we strongly recommend – or use www.gliffy.com for a free alternative). In the diagram, each of the boxes links to an in depth description which includes variable listings and link trees for that section of the GUI. With this document finalised, the GUI can be built quickly and efficiently, without needing to constantly revise and reassess content and structure.

Conclusion

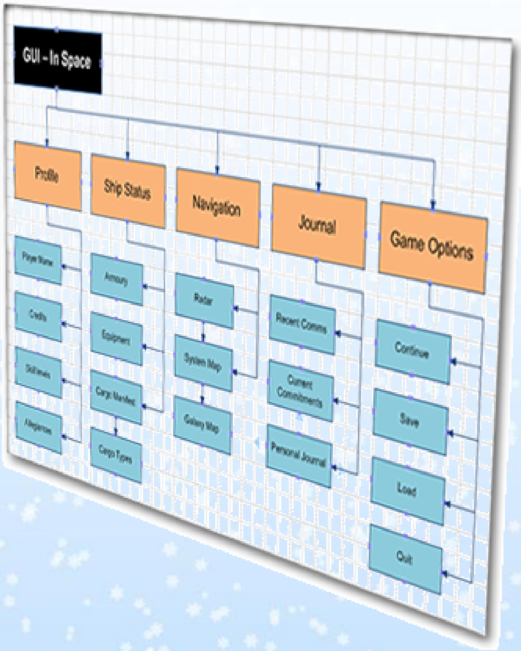
So where are we at?

The Faction Wars team is frantically churning out maps, documents and variable lists, which are rapidly being integrated into the core engine.

With these core systems in place we're generating massive amounts of 'placeholder content' – weapons, ships, AI, GUI components and more. Later down the line, we'll simply plug in finalised templates for functionality and visuals, and we'll have a finished product.

In the meantime, look forward to the New Year, when we're confident that we'll have enough of the core components completed to produce some placeholder gameplay that guarantees to tantalise you all…

Tarik Abbara & Jono Alderson■



God of Rock

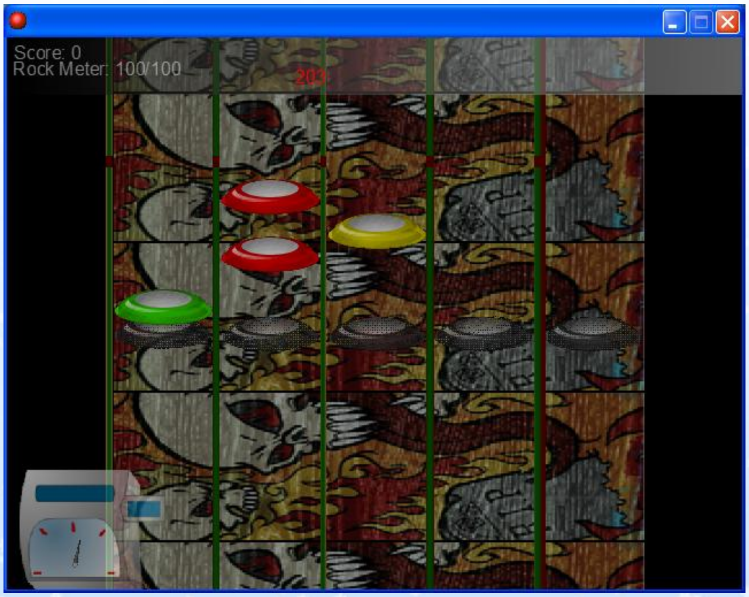
Development Journal

I have begun to code the online in God of Rock. So far it is extremely laggy. The main problem I have been having with this issue is the band system. (It's basically like the Guild thing in Gaia Online but with a different name.) Well anyways the band system has a function called 'Switch Members' where you can change the bassist or something like that.

The problem with that function is that you can switch a band member and that doesn't register on the other players machines. So to fix this I'm thinking maybe setup a server or something. Well the next thing im having problems with is deciding how I'm going to make the Song Editor, for the user. But I'll decide that on the last day before release.

Circulations per Second

Well, the next problem I started correcting is the calculations per second. What are calculations per second (CPS)? CPS is how many notes are made and initiated a second. The reason counting this is required is because the game automatically, adjusts its priority depending on the FPS and this setting. Interesting isn't it?



Score: 0
Rock Meter: 100/100
203

The next problem I had was calculating the score into money. Using the function '`sqrt`' just seems kind of cheep to me and sometimes you might sort of overpay the player. The score actually gets kind of high anyways and sometimes will result in the player getting large amounts of money like, '$3,548.10' that's a heck of a lot of money, in the game and sometimes you get decimals or 'cents' in this case. So I wrote a system depending on the amount of notes you get, which divides the percentage from the total amount of notes and rounds it up.

Pause Screen Disasters

After I fixed this pretty much all of the issues with the engine where then solved... Except one, and that's the issue of the mysterious bug of the pause screen crashing your computer when you restart a song. I have yet to figure this one out. I'm guessing it has to do with the DLLs I'm using. Those are all the issues and they have now been corrected, except that last one.

Conclusion

And that was the December Development Journal for my ultra-cool game titled "God of Rock". So this is the end of the Development Journal until the January issue (Which is also the day a new version of God of Rock will be expected to come out but I'm not sure if it will happen).

Mathew Malone■

Christmas Games

I have never really understood the purpose of Christmas themed games. Christmas accounts for just one out of 365 days a year (or 60 if you believe the retail industry, a month if you're MarkUp) so the effort put into creating a game for this one, albeit reoccurring event, has always mystified me.

Larger successful games may choose to offer Christmas themed add-ons to spread "seasonal spirit" and attempt to rekindle interest in a product perhaps several months old, but these extras will only have a lifespan of around 15 days a year. It may have been a better move to have instead invested the time into creating a full sequel which will have year round appeal or fixing the inevitable bugs which will be frustrating gamers.

Many games produced for the Christmas market also tend to fit into the category of mini-games as they have often been put together in a rush in order to meet a pre-Christmas release date. However once you've made a game, you can always use it the following year and the following year can't you? Many games are, take for example the hundreds of websites offering free flash games. Although some games retain popularity over time many games age quickly due to the poor quality graphics and game-play which lags behind the most recent breakthroughs used in recent releases. Also once someone has played, and probably become bored of, a Christmas themed game is it really fair to inflect it on them again the next year? It's supposed to be the season of goodwill after all!

Types of Christmas Games

There are two kinds of Christmas games, the first fit into the category of mini-games with festive decoration. These are the lazy-mans way of producing a Christmas game, for this reason they are often some of the worse. All that is required is the substitution of a few snow-covered sprites, however most people insist on going over the top adding Christmas music (cue annoying jingle bells) and distracting snow effect animations which add nothing to the game. In fact these cheesy add-ons can ruin a game and put people off the rest of your releases for good. Typically knocked up in a December afternoon without much prior planning these tend to be low quality and tacky, resulting in an additional release with no real purpose except the time of year. With YoYo Games' first competition currently underway and games requiring a seasonal theme I expect a good number of the entries will fit into this category, after all it isn't hard to make one.

The second type of Christmas games are those designed from scratch with a seasonal theme. These are usually of a much higher quality than games which have just been modified, for the simple fact that they have been planned. An example is JassInc's Randolf The Reindeer game which rather interestingly came out in two versions for different Christmases. Instead of being a standard game with added festivities these games have been designed specifically for Christmas releases. However, these can easily be of low quality if they are rushed as an entire game takes a lot longer to plan and code than changing a few sprites does. These don't have to be long games, but a unique feature or Christmas concept central to the game such as landing a sleigh are often characteristics seen in these types of game.

Short Reviews of Christmas games

There are currently very few Christmas themed games on the YoYo Games website, mainly because there hasn't yet been a Christmas since the site came online. It appears that some people however have uploaded their back catalogue of games, or have planned well ahead in making their Christmas themed game. As most of the seasonal games on YoYo are mini-games I will only take a brief look at each offering.

Now Reviewing: Santa's Sleigh

What they say

A fun little game where you have to land Santa's sleigh on the roof of the house. It's rather difficult at first, but after a few runs you get the hang of it.

Controls

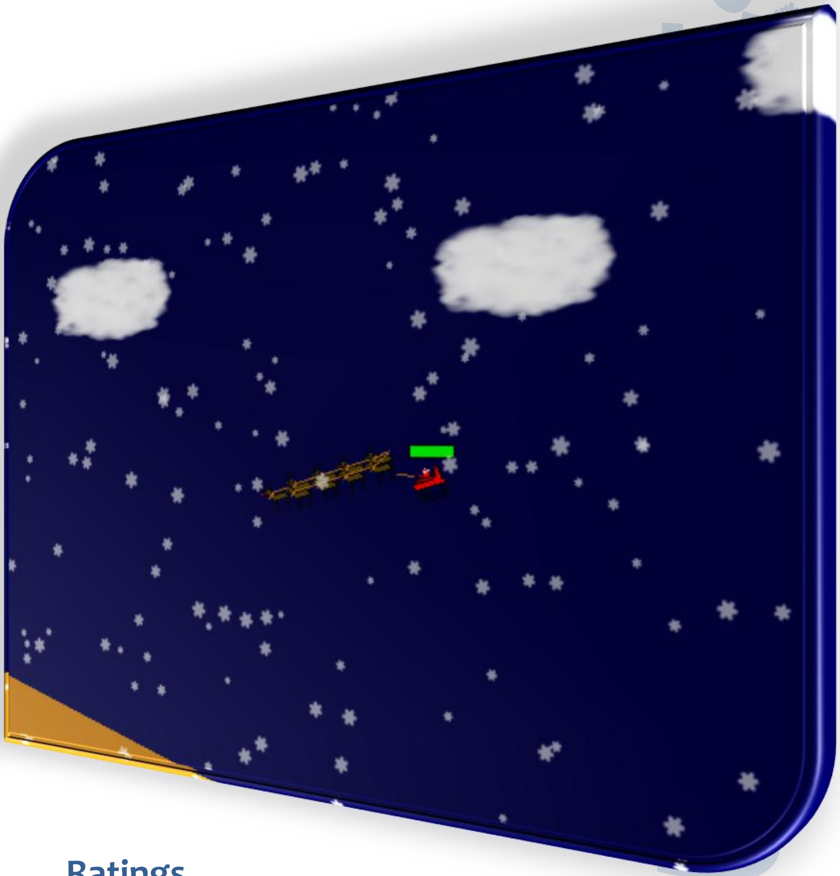
Left and right arrows accelerate and decelerate the sleigh, up and down arrows pitch it up and down. The score for each landing is based on how smooth the landing was, (i.e. higher scores for more parallel to the roof, lower for nosedives) and it's also based on how fast you're going when you land. (i.e. slower landings are more points than fast landings).

My opinion

Fractangle was certainly right to say that the game is difficult at first, but after a few attempts I did finally become accustomed to the controls and succeeded in landing the sleigh successfully. Once you have done this there is not much else to do, there is only one level of the game. More variety in the form of landing on increasingly odd-shaped and smaller roofs would have been a big improvement and would have extended the game length past the 2 or 3 attempt novelty factor. Flying off the top of the screen results in you emerging at the roofline, a perfect way to save time, but otherwise serves little purpose. There are also seemingly random occasions on which the sleigh becomes stranded far behind the reindeer, rendering your controls as good as useless.

Thankfully the graphics are good, the roof is about as basic as it comes but the snow animation and clouds do not distract from the gameplay. The absence of Christmas music is also a blessing as people will not be scared off after one attempt!

This is a fun game, but in its current state it is only a novelty and would not keep anyone entertained for more than a couple of minutes.



Ratings

Graphics: 9

Sound: 10

Gameplay: 4

Design: 6

Effectiveness of Christmas theme: 8

Get it Now

Christmas Games

Now Reviewing: Santa's Helper

What they say:

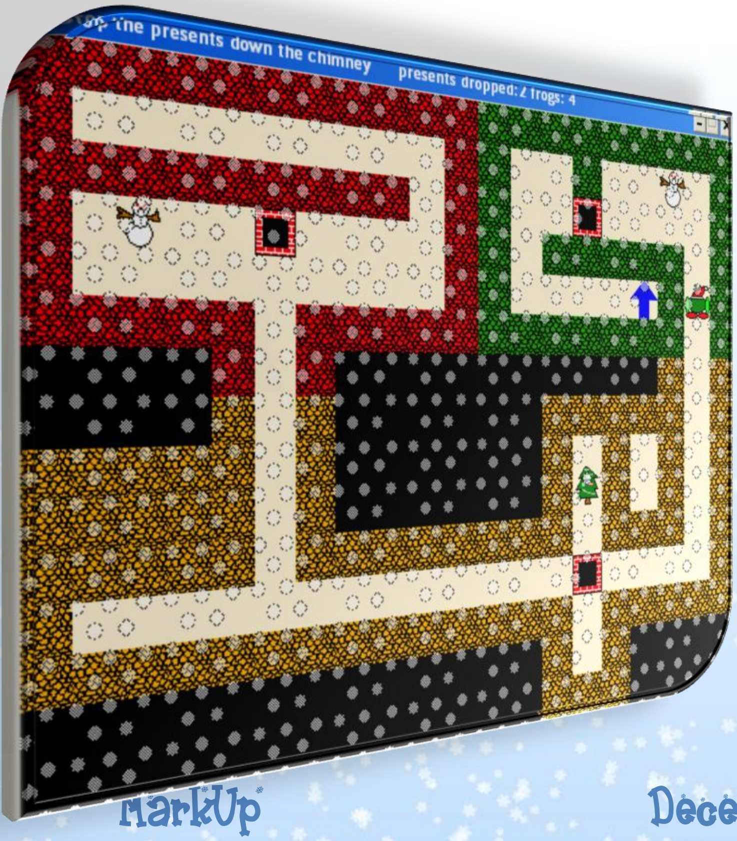
Roggy is back to his old tricks, and he made a frog cannon made to shoot Froggy into outer space! Roggy, however, pointed the cannon in the wrong direction and launched Froggy to the North Pole!

After running into Santa Claus, he tells Froggy that he will take him home in a magic sleigh with one condition. Froggy now must deliver presents to get back home!

Are you ready to play as Santa's helper?

My opinion

The aim of this game is to deliver presents down chimney points whilst avoiding the moving Christmas items including snowmen and Christmas trees. The controls are simple, like in any maze game, and thankfully the path to be negotiated is never too easy or too hard. The addition of terrain such as ice, and cracked ice adds variety to the game and adds extra problems you must overcome to finish each level. Sadly the snow effect which covers the entire screen is a major annoyance; it doesn't even look like snow, just large white circles moving diagonally across the screen. There are no sound effects or background music, and I think in a game like this which isn't over in just a matter of seconds, sound would add an extra dimension to the game.



Santa's helper is a fun seasonal game with the variety in the levels providing a logical challenge.

Ratings

Graphics: 7

Sound: 5

Gameplay: 9

Design: 9

Effectiveness of Christmas theme: 8

Get it Now

Quick Review

MReg Registration System

While we have featured the 'Product Activation' system as this issue's extension of the month, I think it's also important to mention the MReg Registration system. "The MReg System program allows you to create registration keys for users in your program or game.

Each user can have an unlimited number of registration keys. Each key also has options, like the amount of times it can be used, and IPs to ban from it." The registration system features a powerful, clean UI, as well as some cool features, including IP Banning, general banning, key expiry, and more. Highly recommended!

gmc.yoyogames.com/?showtopic=318752

Christmas Games

Now Reviewing: Randolf the Reindeer

What they say

This is the biggest ever upgrade of Randolf the Reindeer with tonnes of changes, additions and bug fixes. Randolf the Reindeer: Christmas Edition is an adventure platformer with 45 different levels for you to work your way through. The first release (released in 2003) won 3 separate competitions and had over 2000 downloads.

The evil Phanta Claws continues his mission to destroy Christmas by stealing Santa and his elves. It's up to Randolf to save Christmas. Luckily Phanta Claws didn't realise he left a trail of elves to his hide out. Follow the elves and you'll soon find Phanta Claws.

In each level there are 5 elves waiting for you to find them. Find and collect all 5 elves in the level take them to the sled and the finish will be unlocked. Be careful though, there are evil toys all over the lands. It only gets harder as you get closer to Phanta Claws.

My opinion

Cute Christmas music and seasonal menus. This game takes Christmas to the heart. Randolf is a well designed platformer with a good variety of gameplay provided by the increasingly harder levels. The graphics which appear to have been drawn as pixel art are well shaded and nothing looks out of place.

To start with I found restarting each level from the beginning on death annoying, particularly when I was exploring a level for the first time. But I don't think I should blame my inability to dodge the obstacles on the game. This is an excellent game, and thankfully attention has also been paid to the in-game documentation and menus as well ensuring that the Christmas theme is spread throughout.

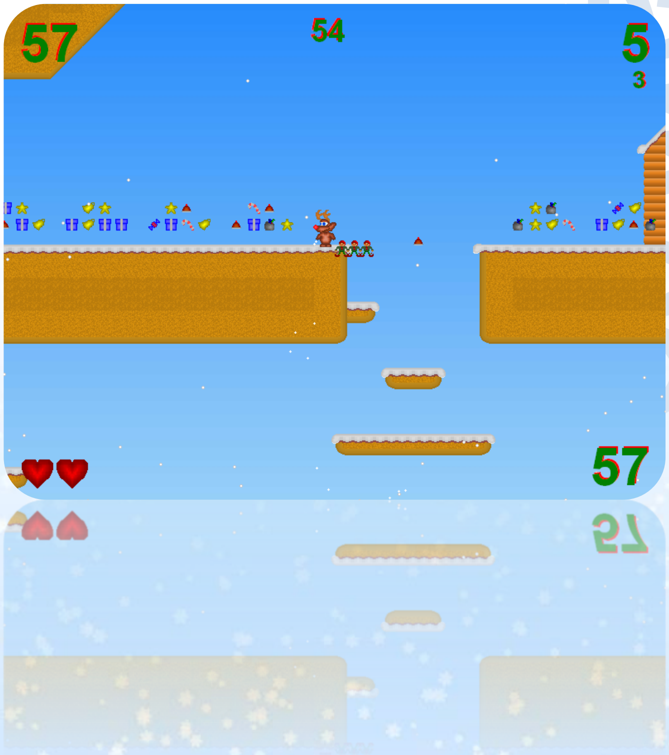
Randolf the Reindeer is a prime example of a game designed entirely around the concept of Christmas which works well and is fun thanks to the well thought-out setup of the 45 levels. I don't think there could be a more seasonal game.

Ratings

Graphics: 9
Sound: 7
Gameplay: 9
Design: 9
Effectiveness of Christmas theme: 10

Get it Now

Philip Gamble■



Lux

What they say

“Lux by ryguydavis is an amazing game that you should be playing right now. If you want to read this review first, fine, but then go and play!” says Elmernite at YoYo Games (although I feel that you shouldn’t go play before reading the whole article).

Jim of YoYo Games says “*The graphics are clear and simple, with a lot going on in a way that's not overly confusing (unlike that sentence). The concept is great, pretty original and actually works very well - Not too frantic, but not slow or boring either.”*

Description

Lux is a minigame, which mostly seems to be designed to serve as a time killer when one is needed. In the game, the player has to click balls, which will then explode. Causing a chain reaction (when a ball explodes, the explosion hits another ball, for example) is the main objective, with many kinds of power-up balls “spicing” the experience and adding to the points.

The screen quickly gets filled of them and will make the game a very hard, yet addicting experience. But that’s not all: if you’re not quick enough, balls will blow up and cause you to **lose points**. This combined with fast-paced action is all that is needed to make an addicting mini-game that will keep you *“glued to the edge of your seat for a long time”*, as the author of the game says.

The graphics are simple and fit the game, although more eye candy wouldn’t have hurt. Everything’s simple shapes with only a few colors (with a few exceptions), that may not look too appealing to the player, no matter if graphics matter to him/her or not (as in my case they didn’t).

The sound is fairly decent, although more sound effects would add to the experience. There are a few sounds you will hear in the game, and there is only one background tune, that will grow annoying in a short while.

There is no storyline. Gameplay works out good with no bugs which is a big plus: games like these often are annoying due to even the tiniest bug. However, I did not find myself *glued to the seat for a long time*: instead, the gameplay starts to get more and more boring after a while of playing.

Pros and Cons

Lux is an addicting minigame that anybody should play as their number #1 time-killing aid.

Pros

- Clean game engine
- Good design

Cons

- Graphics too plain
- Music gets annoying
- May not entertain for as long as supposed to

Conclusion

This is a game anybody should check out.

Get it Now

Veeti Paananen■

Info…

Developer: ryoudavis
Version 1.0

Graphics:	6
Sound:	6
Gameplay:	8
Storyline:	-
Design:	8
Overall:	7

Artex 2

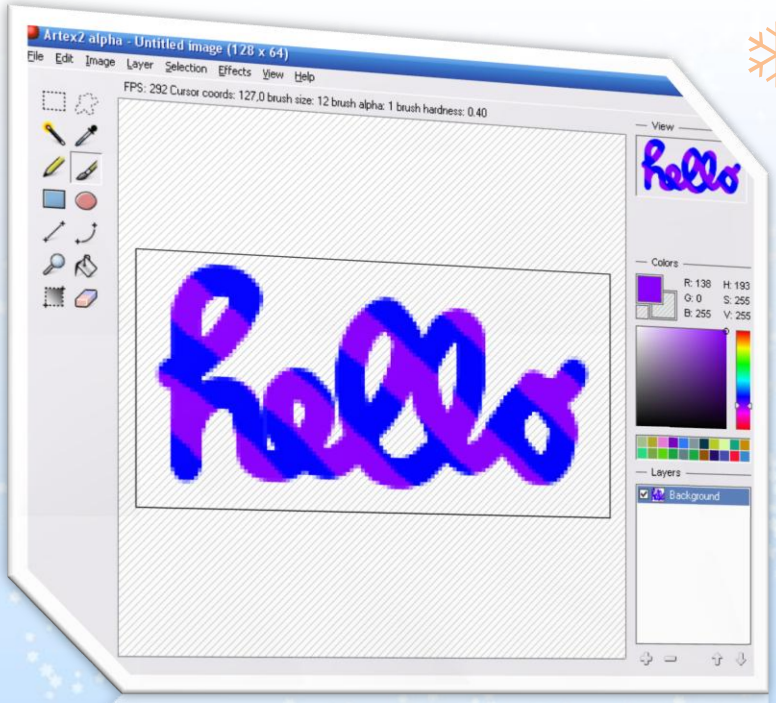
'Lelo707' (Leslie Bicker Caarten) was most famous in the GMC for a long while (since back in 2005), for her amazing work on the GWindows Scripts, and further went on to develop the GWindows 2 scripts, which were definitely revolutionary in many ways for Game Maker.

GWindows allowed standard GML to be used with no DLLs to create a windows-like environment of textboxes, buttons, controls, windows, scroll bars, list boxes, and more. This all seems normal now to the GMC, many other scripts soon came out to create similar functionality, but to me and many others: none has been as well rounded.

Lelo707 went on to create an image and icon creating and editing program: Artex, and released its first version to the GMC, which became really popular. Artex was famous of its seemingly native controls and interface.

Recently, though Artex 2 has been in development and its new additions seem to be indeed revolutionary in many ways.

New techniques such as awesome anti-aliasing, layers, transparency, and more are all coming to Artex's new version. That, as well as an all-new interface and tools.



Artex2 alpha - Untitled image (128 x 64)

FPS: 292 Cursor coords: 127,0 brush size: 12 brush alpha: 1 brush hardness: 0.40

R: 138 G: 0 B: 255 H: 193 S: 255 V: 255

UI Tools

From all the figures , we can see all the planned UI tools for Artex. Not all are currently functional, but most are. The layout has been designed to resemble the concept of Adobe Photoshop and the logic it uses.

There'll be tool sliders where you see the FPS, cursor coordinates, etc and under the tools will be special controls for icons. An .ico file can contain multiple formats to allow for different situations and computers, for example most professional icons have at least a 16x16 icon for in the quick bar, a 32x32 icon for on the desktop and a 48x48 icon for in windows explorer.

Windows Vista Supports 256x256 icons, however these will not be supported into Artex 2, as they exceed the 'pixel' count limit, which is (96x96).

Alpha

An interesting feature which seems to be unique to Artex is the alpha lock functionality. You can lock your current alpha map to whatever layer you like, and anything you draw will never get more alpha than on the map. This is an extremely useful tool to preserve anti-aliasing, an essential part of all modern-day graphics.

Artex 2 does, however, also include native support for Anti-Aliased icons, i.e. icons with 32-bits for all pixels, to allow for alpha blended images.

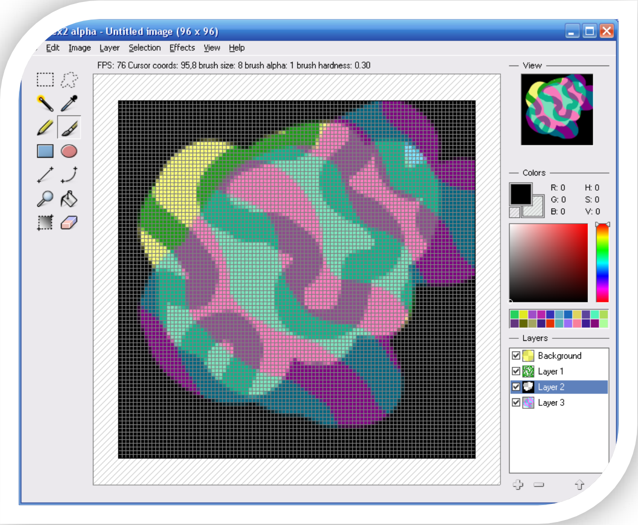
A demonstration of the Alpha lock feature can be seen in the first image, where the alpha settings of the "hello" text is preserved, and the purple stripes have been drawn locked to the hello's alpha. Therefore, the shape of the 'text' remains intact.

Artex 2

Frame Rate

Unlike the original Artex, which never ran above 30 FPS even when idle, Artex 2 delivers a much higher FPS, ranging from 300% to 1000% as much. Artex 2 is capable of running at 300 FPS when idle, and 100 FPS when constantly drawing alpha-blended images.

Obviously on CPU-intensive actions, the FPS might drop below the average, however these numbers seem very encouraging and impressive.



FPS: 76 Cursor coords: 95,8 brush size: 8 brush alpha: 1 brush hardness: 0.30

Layers

As you can see layers are already working well, they're easily accessible and you can add, remove, and move them around easily. There are still some missing controls, notably for layer opacity and possibly blend modes, and there are some additional options when you right-click a layer.

Also, if you look at the little preview images to the left of the layer names, you can see how the image is composed.

The screenshot also demonstrates other features, such as opacity of an image, and the enabling and disabling of the grid to be shown.

'Little Features'

According to Leslie, 'little features that make your product different' are the things that make a 'good application' become a 'great application'. An important example would be the 'smart grid' that only draws itself around the mouse, with a cool fading effect.

She argues: sure it's a subtle effect that might not be noticed, but "It's just a nice little touch, something most people probably won't notice, but I think those are the things that make a great application."

More 'Little Features'

There'll be tons of options to customize how the program works, including full hotkey mapping and multiple speed preferences. Another thing you don't see often in GM apps is that the window is resizable, and it actually retains the formatting and it adapts the room size, which is ridiculously annoying to do in game maker.

Conclusion

I'm not sure how this appeals to newer GMC members, but as a relatively 'older' one, I'm really excited to see such application come together. Not only because of its usefulness (which, indeed, is great), but also the techniques it uses; it might create new benchmarks for game development and application development in Game Maker – something for future GM developers to consider.

I'm seriously positive that such project will materialize to become even more impressive than it already sounds.

Thanks Leslie for the information!

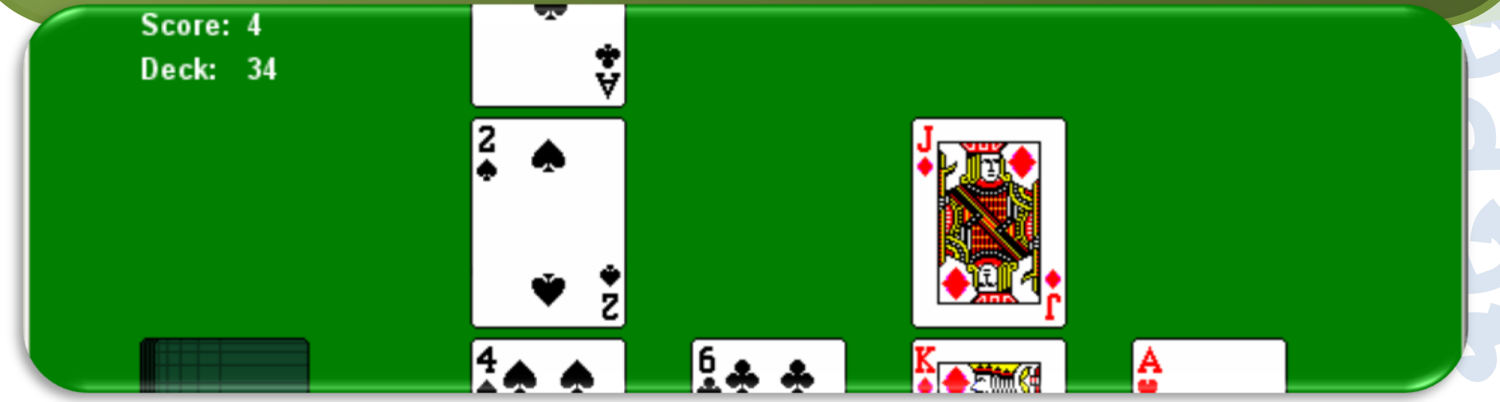
Eyas Sharaiha ■

Bureaucracy

Bureaucracy is an original card game developed with Game Maker. The concept of Bureaucracy is similar to that of solitaire games, except that (strictly speaking) it cannot be categorized as a solitaire games since the ordering of cards is not done by suit and value.

The object of the game is to clear the rows in the window from cards. You start with four rows with cards, and you need to promote and demote cards, rows can be 'bumped up' (or, promoted) by placing a card with a higher value than the bottommost card of the row in the space below the row, thus forcing all the cards up one space. Similarly, rows can be 'bumped down' (or, demoted) by placing a card with a lower value than the topmost card of the row.

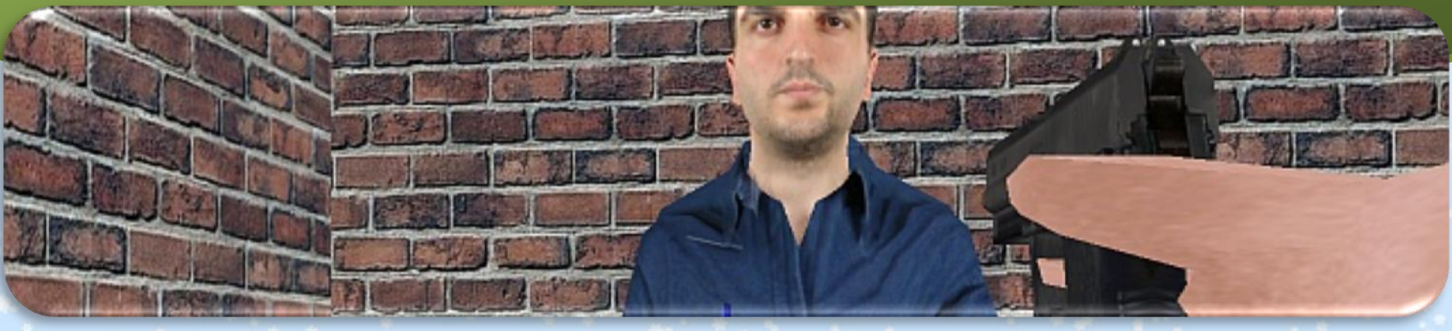
The concept of the game can be weird and confusing, but it grows on you. It's really nice and refreshing to see a Game Maker card game, while these often appear to be simple, in reality, they are not; card games require a lot of effort in programming good engines, whether it comes to shuffling, placement, responsiveness, etc.



Playback Hour

I've mostly been attracted to the technology in this Game more than anything. The 3D in the game is rendered using DirectX 9, has cool animated models, a skybox, good collision, excellent pathfinding, etc!

The story's a bit short and creative: you are being followed! At the beginning of the story, you –as a player – would not know why you are being followed but will only need to escape from whoever is following you in order to, later, find out why.

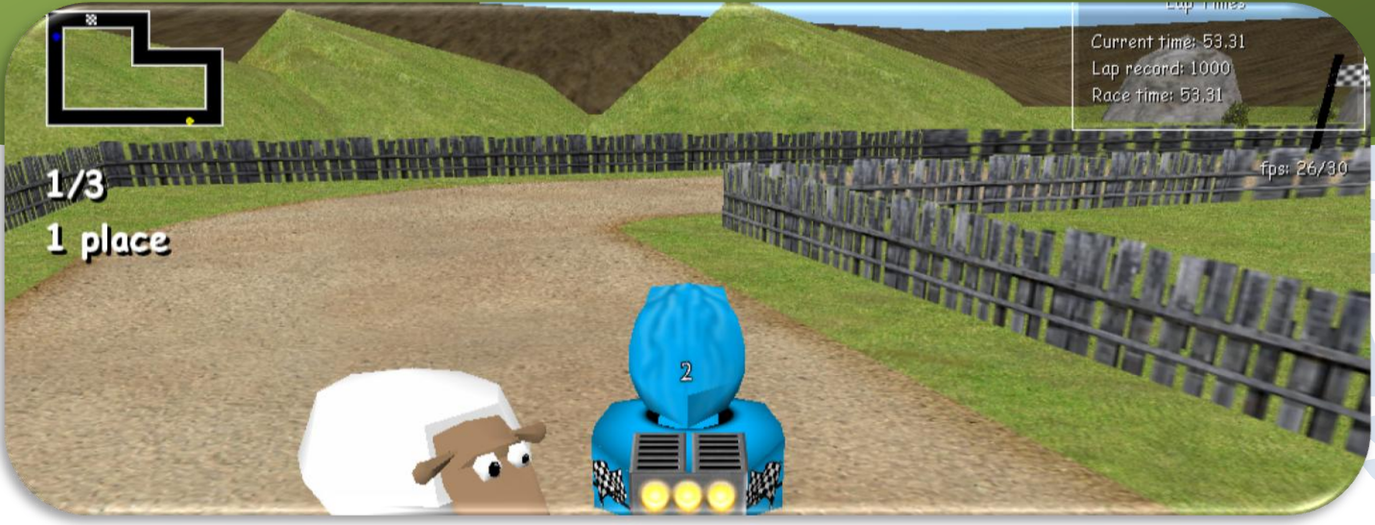


3D Racer

A new project that has appeared in the WIP is '3D Racer'. The game is, as the name implies, a 3-dimensional racing game in Game Maker. In the game, you should race to 'shoot sheep' with a click of a button – space!

The game's supposedly a funny, light-hearted one, although dmitko, the game's developer, assures us that the game will be much more interesting and hilarious once sounds and music are added to the game.

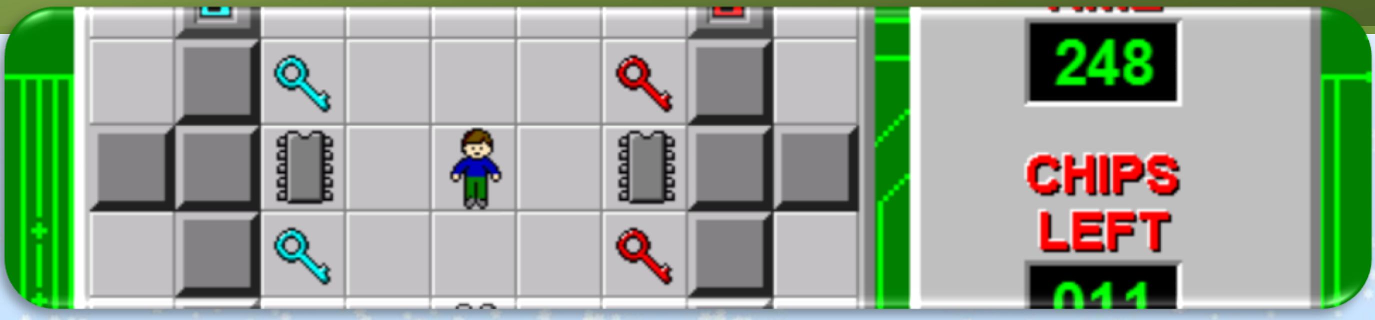
Overall, the game does feel comical and nice. While 'shooting sheep' doesn't seem to have any point, or some sort of deeper-level justification, it is a fun-to-play game to waste some time whenever you have any to waste. An impressing thing about the game, I thought was that it had sufficient enemy AI, something that is generally not achieved in Game Maker games.



Chip's Challenge

Cool game, nice concept and revival of the original Chip's Challenge game a decade ago. The graphics are adequate and the colors are fine, although I must confess I would've expected more from a 'remake' – something to improve from the times of Windows 95 is certainly the graphics!

The game seems to have an imbalance between 'imitation' of the original Chip's Challenge at some places such as the UI, and some improvements in other areas. It has been suggested that if the game was to be made more unique, or alternatively, completely identical, it would be better.



Goodbye!

Thank you for reading the December 15th Issue: Issue 10 of MarkUp Magazine. We hope you've enjoyed our new holiday-based theme as well as the new text layouts, etc. Obviously future issues will look much more enhanced than this one, but the overall whiteness is just added to emphasize the nature of the holiday season – no I'm not talking about purity of the heart, I'm talking about snow!

As you have seen, we came back, this time with two 'The Making of…' articles to substitute for the missing article in Issue 9. We're 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.

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

Happy holidays!

The MarkUp Staff■■

Check Out…

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

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

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!

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 reliable, GMking.org makes no guarantee as to, and assumes no responsibility for, the correctness, sufficiency, or completeness of such information or recommendations.