

markup

I S S U E 1 5

Introducing MarkUp's New Logo

Plus: the All-New MarkUp Puzzle!

The Making of Visit

Character Development

Drag and Drop to GML

Tutorials

Repainting the Screen

Calculating a Hamming Distance

Plus... Interview with Coffee

... and much more!

WELCOME!

Editor's Desk

FiveOne Reasons Why Things That Come Out Later Are Better

Wow, we've been way too long getting this issue ready! I guess we've been so busy time move by a bit faster than usual. But that's not to say this isn't my favorite MarkUp issue to date! Of course, being a contributing editor, I'd have to say that anyways.

GMTV is still chugging along too, it's always great to get to see a sneak peak of things you know no one else is going to see for weeks. And having had the opportunity to do that with GMTV (and a number of other projects) I can say you'll be impressed!

While I'm on the topic of other projects, I've recently been trying to find open source Christmas music to little or no avail. Archive.org is becoming an online cess-pool of copyrighted content, and none of the Creative Commons libraries seems to contain anything remotely seasonal. Why hasn't open source music caught on generically yet?

Before I let you move on to the rest of this issue, I'd like to take a moment to thank everyone that had a role in preparing this issue, you're all an amazing group of people!

Until next time,
Robin Monks

New to MarkUp!

In Issue 15 of MarkUp Magazine, we introduce you to an all new section and type of content: MarkUp Puzzles! The MarkUp Puzzle will be a monthly code question or challenge in which you can test your knowledge of Game Maker and the Game Maker Language. The MarkUp Puzzle is much different than other 'competing' puzzles and focuses on advanced Game Maker Knowledge. Make sure to check this month's puzzle at page 1.

Eyas Sharaiha
Robin Monks
Bart Teunis
Blake
Erthgy
Halfmillennium
Philip Gamble
Robert-Jan Bruintjes
Sam Whited
Shadow Master
Sokota
Lasse Klarlund
Ted Lauterbach
Suhaib Al-Dari

Sr. Editor
Editor
Writer
Writer
Writer
Writer
Writer
Writer
Writer
Contributor
Contributor
Graphics Design

Table of Contents

The Making of...

The Making of Visit	4
Journal: Cities at War	8
Journal: World in War	10

Editorials

Drag and Drop to GML	12
Character Development	15

Tutorials

Advanced RTS: How it Works	18
Repainting the Screen	19
Calculating a Hamming Distance	25

Monthly Specials

The MarkUp Puzzle	27
Script of the Month	29
Extension of the Month	30

Reviews

Senseless War	31
Angels	34
Mubbly Tower	36
Seven Minutes	38
Silent Dawn	42
D-A-N-C-E	47

Interviews

Interview with Coffee	33
Interview with Virtanen	40
Interview with OCGaming	44
Interview with Nightmare	49

Gaming milestone for YoYo

On November 3rd YoYoGames.com surpassed the 25,000 game mark 21 months after Mark Overmars announced his partnership with the site. The Game Maker creator claims that the site "has by far the most games of all game sites in the world" and with games currently being uploaded to the site at a rate of more than 50 a day it appears this will continue.

Improved Game Maker Wiki

Last month YoYo Games and GMking announced that they were merging their two rival wiki services to create a single resource for developers using Game Maker. Approximately 600 articles from GMking.org's GMPedia were migrated over to the YoYo Games service and in exchange all adverts on the YoYo wiki were removed and its contents licensed thorough a Creative Commons License, in keeping with GMking.org's not-for-profit ethos.

Game Maker Competitions

A \$100 first prize is up for grabs in the TIA Productions [limited graphics competition](#). Submitted games may only include shades of two colors and all graphics must be made from basic shapes. The deadline is November 10th from when 10 days of voting commences. At the time of writing there have only been 5 submissions so although the deadline is not far away you may want to consider entering for a shot at \$100. The odds of winning are certainly much better than in any of YoYo Games' competitions.

Save the Planet is the theme of the latest official YoYo Games competition. \$1750 is once again up for grabs with

cash prizes for three games deemed best by the sites employees. The late deadline of January 4th 2009 ensures that people can take advantage of their winter breaks to finish their games and should mean fierce competition for the top prizes.

YoYoGames.com finally out of Beta

In mid-October the YoYo Games website finally moved out of its so-called 'Beta' status with the introduction of some new features. The homepage now includes site-wide statistics mentioning the number of guests and registered users online at any time, and the number of games and game plays on the site.

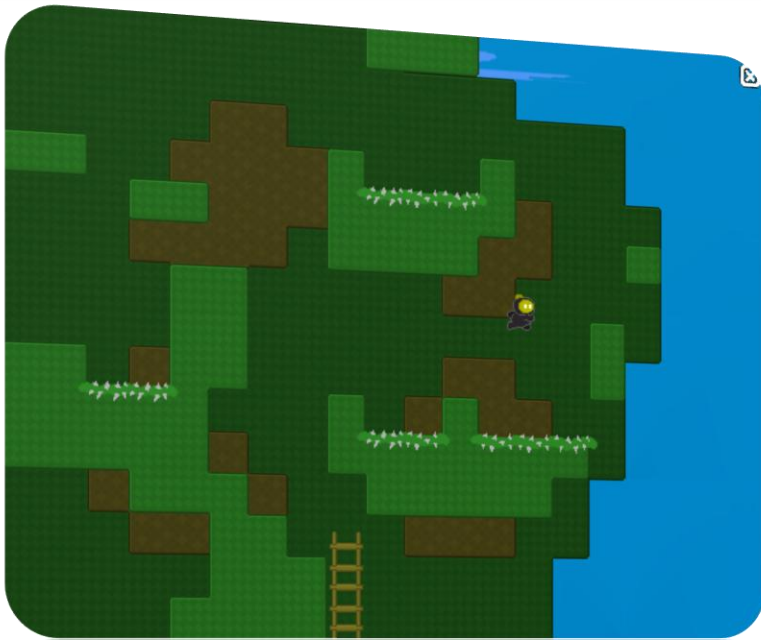
The homepage now includes direct links to far more games than it did previously and games are ranked by various methods including all-time plays as well as featured games from each category. This should ensure that more prominence is given to higher quality games, whilst less-slick games are buried further in the site and require a user to choose to view all games or search for them directly.

There were also some slight visual changes to the site which have had the effect of making the site look smoother, and the bizarre 6-star rating system was replaced with a more traditional 5-star method which pleased many community members.



The making of Visit

By Ted Lauterbach (rotten_tater)



Introduction

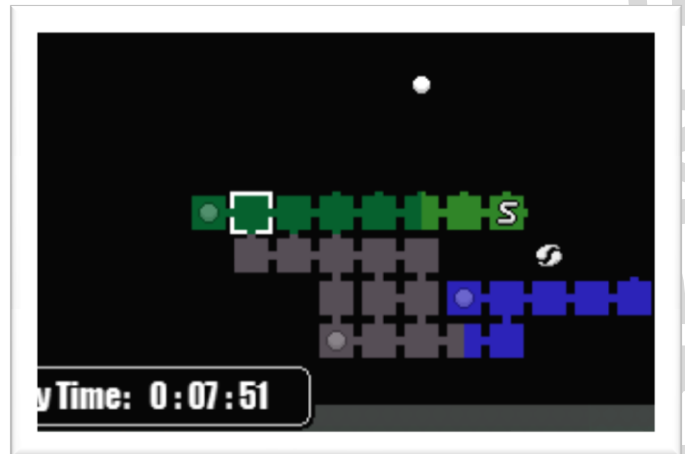
If you haven't played Visit, the concept is simple. Visit is an adventure platformer with tons of puzzles, and you must locate 8 keys and 15 coins to arrive at the final challenge. I originally designed the game for the YoYo Games Ancient Civilization Competition. Visit is visually simplistic and cute, and *everything* was built from scratch. Some of my other games include I.C.E. and Wave Shock.

Map

In Visit, I wanted the world that the gamer would play in to be very large and complex. However, the requirements of retrieving a series of keys and coins for a full completion would force the player to journey virtually everywhere! I felt that one of the simplest solutions to making this huge landscape less daunting was to create a map.

To give Visit the element of adventure, I needed to make

the map reveal itself as the player journeys from one room to the next. I decided to represent the rooms on the map as interconnected squares. I charted the location of every room and drew the map as a sprite. I blocked out unvisited sections of the map by using Game Maker's rectangle drawing functions.



To start the system, I created a global two-dimensional array.

```
global.map[30,15] = 0;
```

The first index of the array denotes the horizontal position of a room while the second index represents the vertical position. This means that the array is prepared to handle a map that is 31 by 16 room units. The purpose of the array is to chart whether or not the player has visited a room. To handle this, I needed to do a check every time the gamer entered a new room.

```
//inside the room creation code
r_x = global.v_roomX;
r_y = global.v_roomY;
if global.map[r_x,r_y] = false then
global.map[r_x,r_y] = true;
```

The variables `global.v_roomX` and `global.v_roomY` represent the player's current position on the map. These variables must be changed every time the player leaves a

The making of Visit

By Ted Lauterbach (rotten_tater)

room. If the player proceeds right, the `global.v_roomX` variable increases by one. If the player leaves through the bottom of the room, `global.v_roomY` will increase by one. The opposites apply when you leave through the left or top of the room.

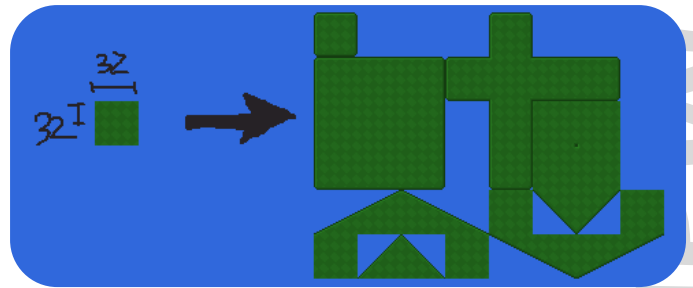
To cover up areas of the map, I used the `draw_rectangle(x1,y1,x2,y2,outline)` function within a nested for loop.

```
//inside Draw event of the Heads Up Display object
s = 20; //square size
for (i=0;i<=30;i+=1)
{
  for (j=0;j<=15;j+=1)
  {
    if global.map[i,j] = false then
      draw_rectangle((i+1)*s, (j+1)*s, (i+1)*s+s, (j+1)*s+s,
false);
  }
}
```

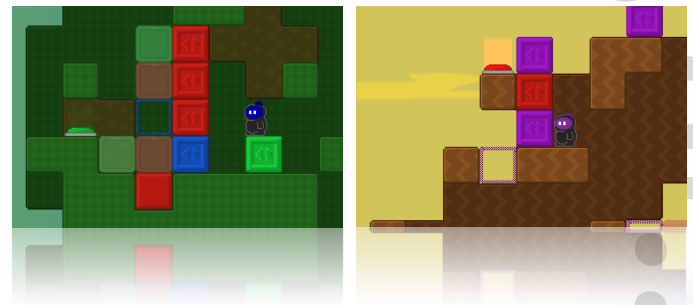
This segment of code will draw a series of squares that cover up locations that the player hasn't been to yet. I felt that it was an efficient and simple way to draw many individual squares using nested for loops.

Tiles

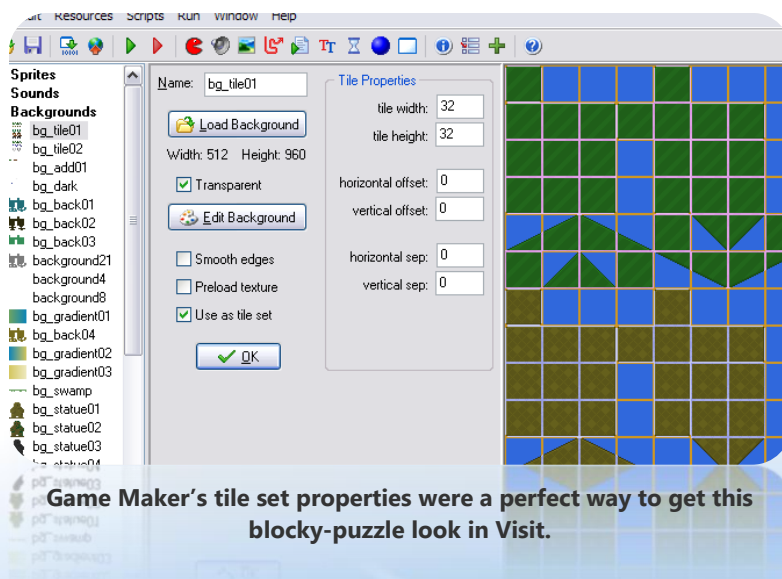
I wanted Visit to look something like a children's storybook the moment I started. I wanted everything to appear simplistic and colorful yet still be able to visually represent specific types of locations. The most basic things that assisted me in presenting this type of imagery were the tile sets I created.



I always kept simplicity in mind when I drew the tile sets. I restricted myself to a small 256-color palette and a tile size of 32 pixels by 32 pixels. When I started drawing the actual tile sets, I would start with a simple 32x32 tessellated design. To make the tile sets fit an area's theme, I would tweak this small design until I felt it colorfully represented the area it was going to be placed in.



After making a design, finishing the tile set would only be a matter of taking the square and modifying its edges to work in a number of situations and have crisp definition. After finishing these images, the only thing left to do was place them in the game!



The making of Visit

By Ted Lauterbach (rotten_tater)

Sound & Music

The audio was one of the most enjoyable things to work on for Visit. All of the sound effects in Visit were done with a plain \$10 microphone that was purchased in a department store, and the music was made in a music program I had purchased a while ago called FL Studio (\$100). I always have the most fun developing things quickly and cheaply. I never let the lack of equipment stop me from making something awesome.

To make the actual sound effects, I had to get creative! The noise for pushing blocks was the sound of a ceramic coin jar being slid across my desk, while the main character's walk noise was me tapping my fingers on my laptop. Some of the sounds had to be computer generated. When the player collects a coin or a key in the game, the jingle that plays is a sequence of midi sounds in FL Studio.

One final form of sound in Visit was the voice acting. I had tried several times to get my voice to a wonderful squeaky pitch, but I could never get my voice to sound happy or cute enough. Fortunately, my sister, Susan, was so excited about this cute little game that she volunteered her voice. The result was the color names that the main character shouts when he changes colors.

I wanted the music in Visit to be very simplistic and happy music. The best way to make happy music is to use major keys and chords. Making simplistic songs in the game required that I use only a small amount of instruments when sequencing. To also help make the midis cute, I had to use the cute instrument patches as well. These included the Music Box, Marimba, and Recorder. Using midis in Visit was a great way to reduce the file size, but I felt the songs didn't sound bold enough. It was a simple fix to enrich the sound using Game Maker's built in reverb and chorus effects.

Final Thoughts

I enjoyed every minute of creating Visit, and I don't think I would change much. The only thing that hurt me while making Visit was how spontaneous its development was. I planned out very little of what was made in Visit. I simply added rooms and interconnected them through code. I have mixed feelings about that, but maybe that's what helped give Visit its own distinct flavor.

Get it now:

yoyogames.com/games/show/34069

Quick Review

ds_box

Do you have multiple data structures stored? Sometimes you might be employing too much data structures of different types (lists, maps, queues...) that you end up losing track.

Perhaps the ds_box extension is the solution for you; the extension provides an advanced data structure for storing... other data structures! The box has functions that let you easily save, load, and destroy multiple structures. Multiple types of data structures are stored together in one box.

The entire box can be freed and saved to a string to be loaded later, or have the string saved to a file to be loaded later as well. By having the ability of handling a bulk of related data structures at once, you can manage your game more efficiently. All data structures pertaining to a certain section of the game are only loaded when needed and the memory used is freed otherwise.

Get it now: gmc.yoyogames.com/?showtopic=358690

REVEALING MARKUP'S NEW LOGO...

See Page 11

JOURNAL: Cities At War

By Robert-Jan Brintjes (rjbruin)

I am back again with the development journal about **Cities At War**. The journal has been gone for a while because the project was paused, but Xsoft Games has resumed progress on Cities At War, so I'll be writing the development journals again from now on.

We've got some great plans for the game in the making, and I'm going to tell you about them. But first of all, what is CAW again?

Cities At War

Cities At War

The game Cities At War is made using Game Maker, and it is a Real-Time Strategy game. Think of Age of Empires, Rise of Nations, those kind of games. The game will be freeware. So that's the original plan for Cities At War: just a normal RTS like any other. But we from Xsoft Games have thought about some more options for the game that would make the game more interesting.

Flexible story

For example, we've recently come up with the idea to let the game have a flexible story. That means that in Campaign mode, the player will follow a story, and that story will depend on the choices the player makes. He can choose war or friendship, attack or defend. This way, the Campaign mode will be interesting even after playing it through once.

Just make a different choice, and the whole story will change with you!

Multiplayer mode

Most RTS games have two different modes, one for a single player on one computer, and one for multiple players, who

Quick Review

GMcap

GMcap is an excellent Game Maker extension by Ravotus. GMcap is an extension which allows you to record from any window, or the entire desktop, into an AVI file. All codecs are supported via a friendly compression dialog interface. GMcap is multi-threaded and therefore captures the screen elements of the game at a regular speed without slowing it down. The frames can be added and converted in a video either to cover the entire window or certain regions and coordinates.

You also can decide to draw the cursor or not include it in the captured version depending on what the interface and video requires in your own opinion.

The extension is compatible with GM versions 5.3A up to 7, and has a comprehensive error recording mechanism should anything go wrong.

Get it now: gmc.yoyogames.com/?showtopic=280053

JOURNAL: Cities At War

By Robert-Jan Brintjes (rjbruin)

can play using a network or the Internet. Cities At War will also have these two modes, at least, that's what we are planning now. But the special thing about the multiplayer mode of Cities At War will be the online ranking system. We are not sure yet how we will exactly do this, but I am quite sure we will make a ranking system based on points. You can earn points by winning from your opponents, and this way to can earn a high ranking. More details on this mode will come later, I expect.

Cutscenes

The Campaign mode of Cities At War focuses on a complex story. To make this story more attractive, we have decided to add cutscenes in between the mission. These cutscenes will explain the story, and they will include the choices discussed under 'Flexible story'. I will explain how these cutscenes are programmed, in the next issue of MarkUp.

Current version

The current version of Cities At War can be found at YoYoGames.com in the 'incomplete' section, but that version is not even near completion. There are many bugs

in that version, which we are now trying to fix. The whole development of Cities At War will probably take quite a while still, although nearly everyone at Xsoft Games is working on the project. We've got some really good people working on it, so I trust it will be really good when it's

Quick Review

ClipCursor DLL

The ClipCursor DLL by Daniel-Dane allows you to confine a cursor's motion to a particular region.

The DLL is particularly useful for 3D first person shooters that take advantage of mouselook for instance, or other games that depend on extensive and quick mouse motion where you wouldn't want the mouse to go out of the window by mistake and therefore let it lose focus by clicking outside.

The DLL works for GM7 and GM6. The DLL stops confining the cursor's motion once the window loses focus, so you can continue to use your computer without closing a game/program that uses the dll.

Get it now: gmc.yoyogames.com/?showtopic=271473

Cities At War

Blood shall be shed, few will survive, friendships broken,
treason and evil on the rise... Let the war begin.

XSOFT GAMES

To begin our series of our Markup development journal, to which we are thankful and honored to have been given, I think it fit that we begin with **where Word in War began.**

The Early Days

In the early days (Pre-GMC release), WIW wasn't even really named. The start of the idea was actually a World War III scenario I made up in school because I was bored. School is a great place for thinking, you know. This WWIII idea was spurned by the infamous WIW sprite of the infrared-goggled soldier, which I made while practicing up on spriting. I was trying more cel-shaded graphics after I just starting working on my Winter Competition Entry, Frozyn. I wanted it to be small, so it would be easier to edit, and yet I wanted it to be cartoony, since that was my style. And so, the USBC had a soldier, and from there, I worked on giving each faction a different look and color, as I wanted the gameplay of WIW, both offline and online, to be very factional. Of course we have not gotten to the online side of World in War, so don't be disappointed waiting for it.

Development

World in War really began in December 2007, when I met c_raethke on the GMC. He and I met because he answered a lot of my programming questions I had while working on Frozyn. I asked if he wanted to join my team, which would have been him and I at that time. Now Jay (Jay Da Master), Seth (Seth52), and Trevor (trevuar25) have joined, making up the entity we know as Skyfire. All that was relevant because Skyfire is the team responsible for the development of World in War. Anyways, WIW began when, Cody (c_raethke), began the programming for WIW. I still have the very first engine he sent me, from Christmas Eve, and if I get a good response from this, I'll post it up on the

WIW topic on the GMC, just for kicks/memories.

From then on, we have usually spent a fair time on WIW's progress weekly, to what it is now: 5 (going on 6, hint!) vehicles, 'plenty?' of weapons already, a few variations of RF and USBC soldiers, and a few buildings to boot. Did I mention all the gorgeous particle effects, such as missile and sniper trails, airstreams, and dirt flying everywhere? And our own ragdolls, made from scratch, which are still being worked on?

Yet be wary fellow game developers, because we haven't had an easy road. We at Skyfire have had to learn many new things, and we aren't always motivated to work on WIW, which can turn into weeks of belated updates. It's especially tough if no one responds to your game, which can kill a project in little time. But don't fret. World in War is a mainstay. It had humble, very humble, beginnings, but we plan to finish it to the end, and perhaps make a bit of money off of it. With the right touch of originality, unlockables, online, and cash shop, why not? Hope you enjoy WIW, so try it today!





The new MarkUp Magazine Logo attempts to embody what we believe ultimately represents this project – MarkUp Magazine: the Science of Game Development. Hence the game development gear, atom features, and astronomical 'effects' on the central gear.

This logo was designed for us by our graphics artist (Suhaib), who also worked on the inner layout revamp of Issue 14 and this issue. This logo will be used to represent MarkUp Magazine, hopefully in places like the cover, site, and other similar locations and occasions.

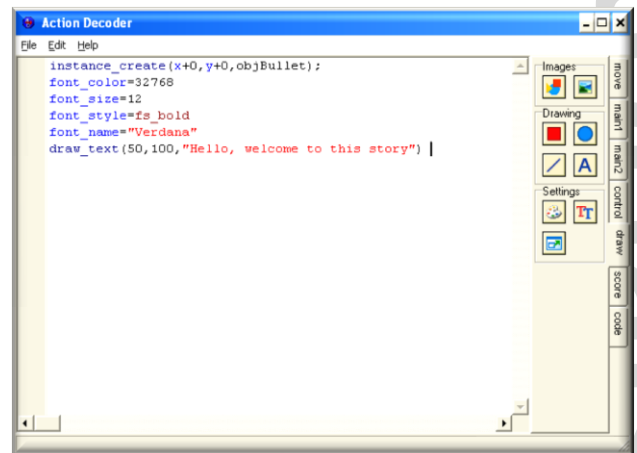
D&D TO GML

By Philip Gamble

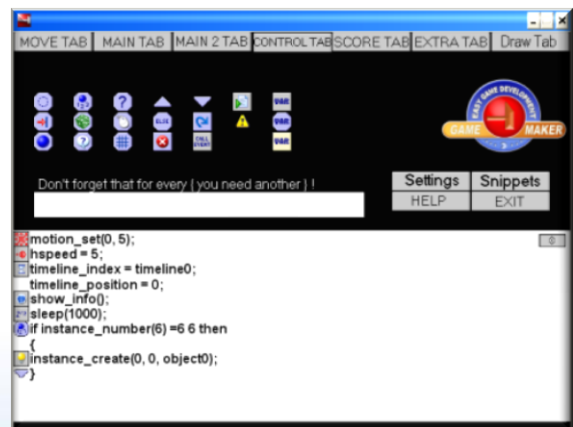
One of the reasons for the widespread appeal of Game Maker is that it can be picked up by children at a very early age who have no prior knowledge of programming and then later, through the use of GML, can introduce them to more advanced programming using GML and making use of external resources such as DLLs.

All of the drag and drop actions have GML equivalents, and as such it is useful to know how to convert an graphical action, say move left at a speed of 5, to the GML - `motion_set(180,5)`. Thankfully there are a number of different programs, themselves created with Game Maker, which hope to take the pain out of swapping Drag and Drop for code by enabling you to quickly convert D&D actions to their GML equivalent.

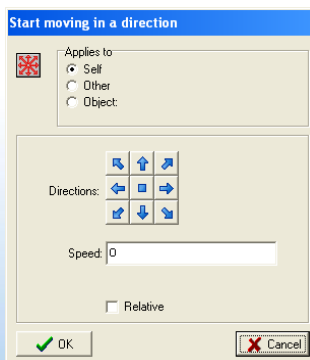
Xception's Action Decoder 1.0 is the earliest D&D to GML program I have come across – it was designed for use with Game Maker 5. Selecting an action from the tabbed bank on the right brings up a window that looks exactly the same as it would if you were adding the action in version 5 of Game Maker. Once you have setup a Drag and Drop action its color-coded GML equivalent is added to the code area on the left which can then be exported as a GML file to be imported into Game Maker. There is also the option to open pre-existing GML files so you can add to code made on a previous occasion. For the limited number of actions that were included in Game Maker 5 this is a very efficient and easy-to-use tool which should make converting D&D to GML a breeze.



ChipChamp's D&D to GML Convertor (version 0.98 is compatible with GM7 actions) blares out background music each time you start it up. Not exactly the best environment to concentrate in when you are working hard on a game.



Selecting an action brings up several task prompts asking for details about the action – e.g. for the "Start moving in a direction" action you are asked for the direction, speed and whether or not the action is relative (to set an action as



D&D TO GML

By Philip Gamble

relative enter "1"). The latest section of GML code that has been generated is automatically copied to your clipboard so that you can paste it within your Game Maker creation.

Sadly this program lacks the gloss and ease of use of Xception's and appears as if it has been put together in a haphazard way as different fonts and capitalization schemes are used throughout the program.

melvin2001's Action Decoder 2006 is billed as a demo, but seeing as it was designed for use with Game Maker 6 I can only conclude that the project was abandoned before its completion. This program has the largest file size of all the D&D to GML convertors reviewed so as you would expect it takes a little longer to load.

Like Xception's Action Decoder the action tabs are a faithful reproduction of those found within the Game Maker software which ensures you can find actions here as easily as you can when making your game.

None of the Draw actions have been included, but from a brief look around everything else appears to be fully functioning. As with ChipChamp's D&D to GML Convertor prompt boxes are displayed after you have chosen an action for you to input the required attributes.

I am thankful that this was branded a demo as an animated background image and the fact that only a single piece of code can be shown at any one time mean that Action Decoder 2006 is not as helpful as it could be.

The most recent addition to this bunch of tools which aid moving towards using GML is cswenson1's GML Helper Lite. This is the most up-to-date program, but currently only includes the actions included in the free version of Game Maker 7.

GML Helper Lite uses the same tabbed action sidebar as Action Decoder 1.0 and Action Decoder 2006, in keeping with the design of Game Maker. A combination of drop-

down menus and text prompts are used to input action attributes and code is output in a similar box from where it can be copied and pasted into Game Maker.



Unlike D&D to GML Convertor 0.98 there is no automatic copy to clipboard feature here, and the program would really benefit from one since there is no window to display

Quick Review

PNG DLL

For the four of you who still use GM6: you can now have PNG file support for your game! This means that you will be able to have PNG and alpha blending in your game, and thus deliver professional sprites.

The DLL of course works on GM7 as well, though you probably would prefer to use the built-in function for that. Using this DLL you can convert a PNG file to a sprite, a sprite or surface to a PNG, or you can create a .png file from two BMP files: one with the colors and another with the alpha in the form of greyscale black/white colors.

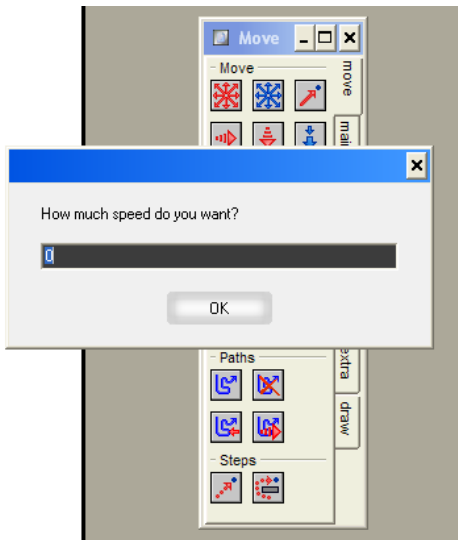
You can also decompose a 32-bit PNG file into an alpha index and a color BMP file.

Get it now: gmc.yoyogames.com/?showtopic=397947

D&D to GML

By Philip Gamble

previously generated code. A version including the Pro features of Game Maker is promised in the future – after the programs creator has registered their copy of GM that is!



Verdict

The version of Game Maker you use is key to your decision on which of the above programs to use. Users who do not require particles and other advanced features added in the upgraded versions of later editions of Game Maker can happily use Xception's slick Action Decoder 1.0.

For Game Maker 7 I would recommend using GML Helper Lite if you can do without particles and advanced drawing features or Action Decoder 2006 if you wish to see the benefits of the pro version of Game Maker.

D&D to GML Convertor

By ChimpChamp. For Game Maker 7.0 Beta (DandDToGML13.exe)

Get it now: gmc.yoyogames.com/?showtopic=254272

GML Helper Lite

By cswenson1. For Game Maker 7 (GML Helper Lite.exe)

Get it now: gmc.yoyogames.com/?showtopic=391153

Action Decoder 1.0

By Xception. Built for Game Maker 5.0 (ActionDecoder.exe)

Get it now: home.tiscalinet.de/xception/programs.htm

Action Decoder 2006

By melvin2001. For Game Maker 6 (actiondecode.exe)

Get it now: gmc.yoyogames.com/?showtopic=119354

Character Development

By halfmillennium

For some game genres, the backbone of the entire project is the story. A story can't work without being written well and the same applies to its characters. So, this should teach you the basics of getting them to work effectively.

What You Need

Generally, a story can be written in any text program. You can even write it on paper if you want to. Good stories aren't dreamt up in seconds unless you've somehow sped up time. The same applies to characters. It's a good idea to keep something with you which you can use to record ideas. They can strike when you're least expecting them and often, this is when they do. However, there are a few programs which are recommended for story writing.

Which Comes First?

One of the biggest questions is perhaps whether the story or characters should be developed first. There is no universal answer to this. The characters only work within a story, but the story only works with characters. Without one, there is no other. Constructing a good story requires both. Characters will be covered first simply because we need to cover them separately.

The Basics of Characters

Generally, a game has a central protagonist. This is a character that the player generally controls and, at the very least, is the most significant playable character. There are many ways this character can be designed. Some games feature very talkative protagonists; others say very little and have no self personality. Decide whether you want to tell the player a story about this character, or whether you want

them to make one up for themselves.

Equally important to a good story is a central antagonist. The antagonist is usually fought as the final or penultimate boss. In the latter type the final boss is usually a formerly unknown enemy. As well as the central antagonist, most games have lesser antagonists for the player to fight.



Now, I am not going to list any further character types here. The reason is because there is no set formula for characters. To list any would detract from the originality of characters.

The Basic Story

Before you start writing a story, you have to determine

Character Development

By halfmillennium

where you want it to go. You should consider things like the theme, the intended players, the complexity of the story you want to write and other factors you can think of. Most stories tend to start with a basic idea and then expand into bigger ones.

For now, all you need is a general overview of your story. It doesn't have to be Dickens quality, just write the basic story.

Making the Character

When you've identified a need for a character in your story, or you want to get ideas recorded for future use, you need a simple, quick and effective way to get those ideas filed. One idea is to open a text document and write several categories which you want filling in. Think of it as a basic application form without the formatting. General categories would often be "Name", "Gender" and "Age", as well as a few relating to their physical appearance - "Hair", "Eye Color", etc. Also consider making a section to fill in some

background details of the character. This doesn't have to be completed straight away, nor do any of the other categories, but creating them now should allow you to alter them when you get the ideas needed.

Now, to actually design the character... This is where it gets difficult. The characters you need or who will work in your story will vary depending on the story. There are few guidelines which apply to all character design, but there is one. What you should be very careful about is using stereotypes. Not only are they likely to make your game look shallow and thoughtless, you also run the risk of offending someone playing it. If used in a humorous context they are in some contexts accepted, but be careful.

Developing Your Characters

Once you've made a character and given him a story role, you need to develop him. This is where character and story work together to achieve it.

>>>Future Dev

Physics Cards

Processors and GPU's are becoming more advanced by the day, and yet, games are still outpacing them in terms of the sheer amount of processing power that is needed to seamlessly play the most advanced games which often have near photo realistic graphics and next to lifelike physics. With such a need for power, it is no wonder that a recent trend in game design is in the use of dedicated physics cards containing a physics processing unit (PPU) among other hardware tailored specifically to the complex calculations that are needed for realistic physics. However, the technology is still new and, mostly, un-standardized. To

write a videogame which utilizes a physics card the game designer must utilize a specific cards APIs and also provide a way for the same calculations to be performed on the CPU (since most users still do not have PPU's). For the industry to take off, this needs to change. Game graphics APIs such as DirectX and OpenGL interface with graphics cards which support them, and, since they are industry standards, most graphics cards do. If APIs like these are created and physics cards begin to support them directly, and, if these cards become commonplace in higher end computers, computer gaming will become both more fluid, and more common.

Sam Whited■

Character Development

By halfmillennium



As all stories are different, there is no way to guide you on what should and should not be done with regards to certain characters in certain stories. There are, however, a few tricks you can use.

One way of emphasizing certain parts of a character's self is to show accent. There is, however, a problem with it. Because few people speak the "neutral" accent (RP) in reality, there is often no specific focal point to work around. As a general rule, use standard spellings unless it would clearly detract from the character's accent. While not everyone speaks RP, many accents can be represented without having to alter the spelling of words. Occasionally a character may be required have a stronger accent; for these characters making alterations to the spellings of words may be necessary to show the accent.

Another good technique involves being subtle about the character's development. Most players would rather see gradual character development than a monologue which breaks up the game. This requires you to be careful with where you put certain parts of the story, but if used correctly can enhance the story greatly.

Making the Story

From this point in development, you're effectively on your own. You have a basic story and a few basic characters. Feel free to add, remove, multiply or divide as you continue writing. Story development often takes it in new directions from the original plan.

Thanks for reading this article! I hope it will be helpful next time you go about developing your game's story and characters.



Quick Review

KBinput

This is a simple extension that improves the input capabilities of Game Maker by adding action mapping features to them.

This means that the user will be able to select, modify, and customize the controls that he/she would want to use for the game, and it would all work the same.

For instance, a user can change it so that the space bar produces a jump instead of the "Up" key, etc. The extension provides a "transparent" interface with the developer due to its simple use of variable naming.

Get it now: gmc.yoyogames.com/?showtopic=398995

Advanced RTS: HOW it WORKS

By Lasse Klarlund

In several posts of my [Advanced RTS Example topic](#) on the GMC, I've found that many of the novice users didn't really want all that flashy stuff, but only the source code for the selecting and movement. Therefore, I made this simpler example, which is much easier to understand. To those of you who don't understand, here's a small walkthrough on how this works:

The General Idea

We have two objects: the control object and the unit object. The control object does all the "drawing rectangle" and "choosing the selected unit's position", and the unit object is simply a slave, moving towards (And choosing) it's go destination.

Let's look at the control object, first. In its Global Left Pressed Event, it starts selecting units. In the Global Left Released Event, it stops selecting, resets every unit's "selected state", and then makes all units, which are colliding with the drawn rectangle, selected. Note, that the rectangle's starting point (x1,y1) is declared in the Global Left Pressed Event, too.

Now, let's look at the unit object. This one is a little simpler. It has two coordinates, gox and goy. That's the go destination. It is declared as the unit's x,y in order for it not to run away when the game starts. Why? Well, in its Step Event, it continuously moves towards that point no matter

what. So if we for example declared gox,goy as 0,0 instead, it would move towards the upper-left corner of the room, when the game starts, which looks stupid. Declaring that as its x,y solves that. Anyway, in the Step Event, it moves towards that point, and makes sure the image blending is set to blue, when the unit is selected. And in the Global Right Pressed Event, it sets the unit's go destination to the mouse's position.

Conclusion

Hope that walkthrough sort of helped. I only included the important stuff, so you will have to figure out the other details yourself, which is a good learning experience.

Become a writer at MarkUp Magazine

CLICK HERE

Repainting The Screen

By Bart Teunis

There are a few topics in the GM manual that are particularly interesting. One of them is "Repainting the screen". At first sight it might not be too useful, but it actually explains quite a lot about how GM handles drawing and how this can be influenced through GML.

This tutorial will explain GM's drawing in detail, based on the topic in the GM manual. We'll have a look at how the functions can be used in a useful way and whether and when they're worth using. But first of all, let's start with some basic GM concepts that should be explained to understand GM's drawing.

Basic concepts

An important concept to understand is the concept of a step and together with that, event execution order. The basic time unit in Game Maker is a step. Depending on the `room_speed`, a step can be long or short. `room_speed` sets the number of steps per second (theoretically). So when `room_speed` is 30, then a step theoretically takes 33ms. Within each step, events are executed for all instances of each object. These events are executed in a certain order, as explained in the manual:

- Begin step events
- Alarm events
- Keyboard, Key press, and Key release events
- Mouse events
- Normal step events
- (now all instances are set to their new positions)
- Collision events
- End step events
- Drawing events

Important to know here is that the draw event comes after all other events. It will be important later on when we'll have a look at drawing. So each step, all events that you, the programmer, have defined for your objects are executed. Each step, events are executed in the above order. So at a normal room speed of 30, this happens 30 times per second (theoretically). The following image will make the above explanation a bit more clear, it is a representation of a **single step**:

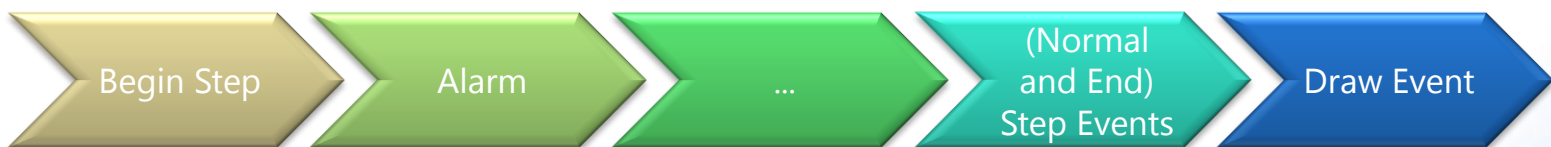


Figure 1

Where can you keep up-to-date with the **latest GameMaker news?**

with all the latest GameMaker news, comment and opinion

plus have your say on community matters

GameMaker Tech - MarkUp - GMTV - GMKing
YoYo Games - GMArcade - Snowmoons
Game Glyph - GMScope - GameMaker Podcasts

GameMaker Blog.com



Repainting The Screen

By Bart Teunis

GM's drawing

To understand this, it's important to know that GM uses an image on which all drawing happens. First of all, we'll have a look at what happens when the default values are set (or in other words; when no drawing options are changed). The drawing can be compared to a painter that paints on a canvas. A painter might paint using layers, where parts of an underlying layer may no longer be visible when he paints a new layer on top of that layer. GM's drawing works in a similar way. At the start of a game, the internal image is black. As soon as a drawing command (`draw_...`) is encountered, the requested image (clear, text, sprite, surface, ...) is drawn on the internal image, regardless of what is already on it (compare this to the layers). You might think: how come the room displays a background color, backgrounds and all object sprites (and tiles, ...), while you haven't given any draw command to do so? That's what GM's automatic draw does: it draws some things automatically (obviously). By default, it is set to `true`. Now when does this drawing take place? Well, theoretically, it happens each step. Each step, backgrounds and object sprites (and tiles, ...) are redrawn. To be more precise: the previous internal image is erased with a clear color (this can

be achieved with `draw_clear`); the background color and then all backgrounds and other images are redrawn. This all happens in the draw event. Now you should also understand why draw commands in the step event give no result. The draw event comes after the step event (as explained above) and since a clear (= clear everything) is drawn in the draw event, everything that's been drawn on the internal image during the step event, is erased. This wouldn't be a problem if the draw event would be executed once, but it's executed every step. Let's have a look at an example that shows that the draw event comes after the step event and consequently, nothing that's been drawn in the step event is visible on the screen (Figure 2).

The figure shows the events in a single step, including the drawing command that is executed during that event (the `draw_clear` command is executed automatically, remember: automatic draw is enabled), a view of the room and a side view that shows the "layers". The arrow to the right of the images shows the viewing direction. Important to know is that the final image that appears on screen is the result of all the drawing commands executed during the series of events in the current step. Only the final image is drawn on

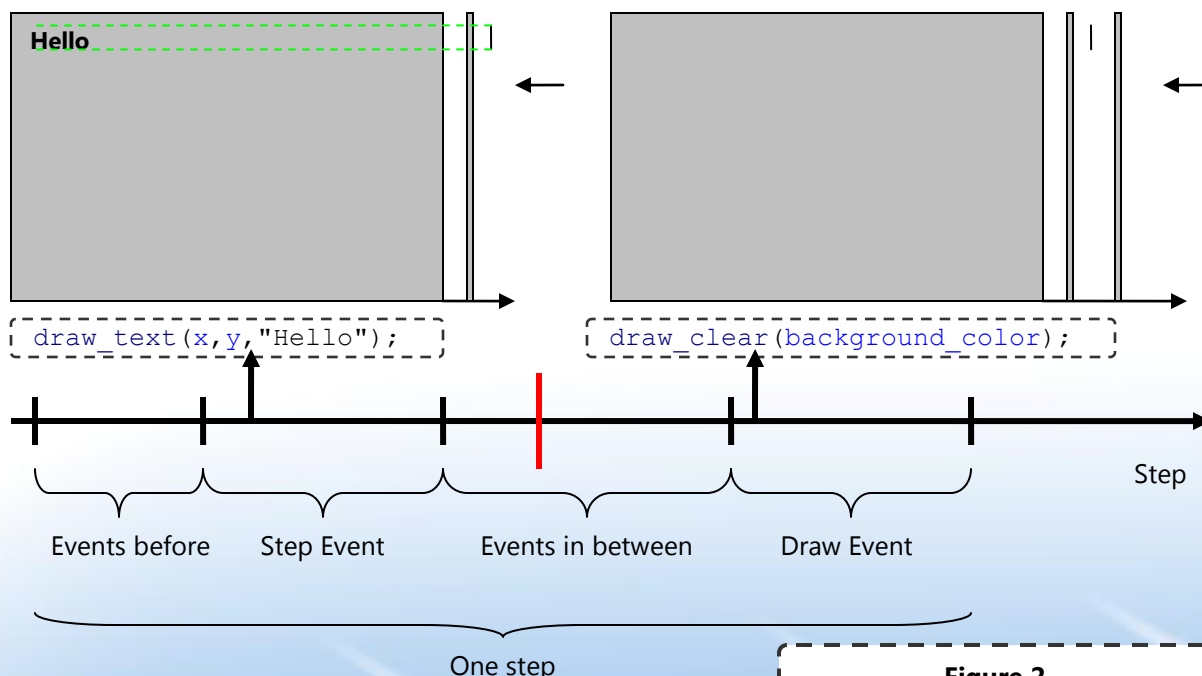


Figure 2

REPAINTING THE SCREEN

By Bart Teunis

this can be done. But first, there's another thing to know about GM's drawing. The default room speed is 30 steps per second and, when automatic draw is turned on, draw events are executed 30 times per second. But the refresh frequency of the computer monitor is most likely to be quite a bit higher, about 60 to 100Hz (depending on the type and settings of the monitor). This means that 60 to 100 times per second, the image on the monitor is refreshed. But draw events are only executed 30 times per second. To synchronize the drawing with the refresh rate of the monitor, GM has a built-in function:

```
set_synchronization(value);
```

The value can either be `true` or `false`. Turning synchronization on means smoother graphics, but with that, you get a significant increase in CPU usage.

And finally, GM has a function to wait for the next vertical synchronization of the monitor:

```
screen_wait_vsync();
```

This function simply makes the game or program wait until a new refresh of the monitor occurs.

The following two figures (Figure 3, 4) show a few steps

with synchronization turned on and a few steps with synchronization turned off (but with a vertical synchronization). Assume that the refresh frequency of the monitor is higher than the room speed.

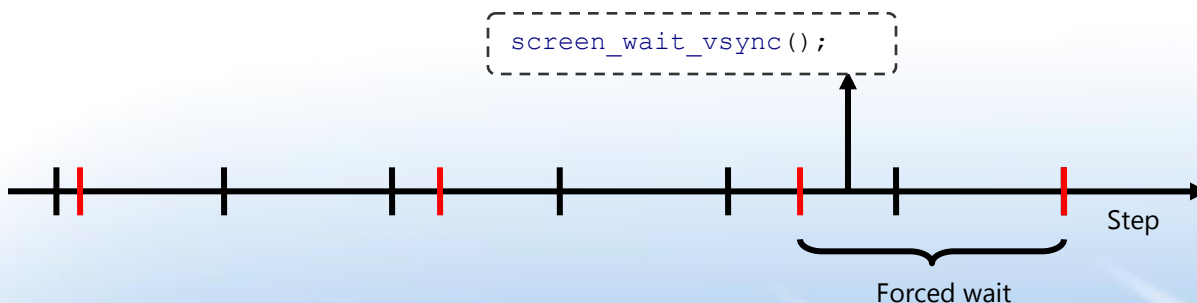
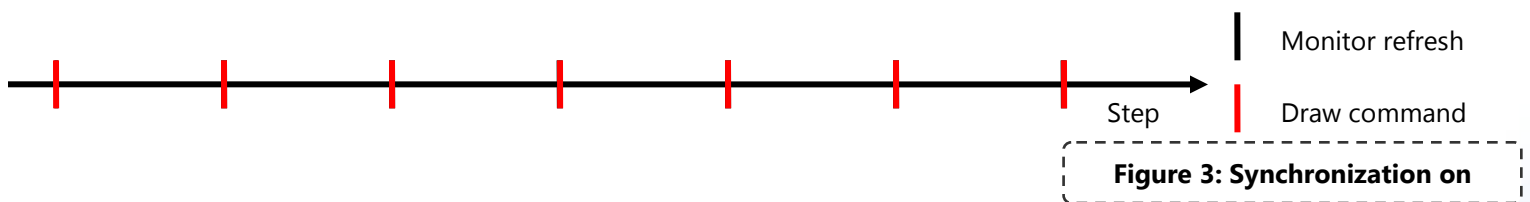
And that's the most important there is to know about GM's drawing. It's now time to take full control of drawing.

Take over control: redrawing and refreshing the screen

So far we've only discussed what happens when automatic draw is on. To see for yourself how GM draws everything, it's interesting to turn automatic draw off. Time to do some testing. Create a new GM file and add a new object and room, and place the object in the room. Add the following piece of code to its Create Event:

```
set_automatic_draw(false);
```

Now run the game and see what happens. You'll notice that the background is black, while the background color as defined in the room editor is gray by default. Why is this? After turning off automatic draw, nothing is drawn



REPAINTING THE SCREEN

By Bart Teunis

automatically, so no background colors either. And when nothing is drawn, the background is black. Another thing you can notice is that the CPU usage has lowered (you can check this in task manager). This is a very obvious thing, since the `draw_clear` command that is normally executed each step (by GM itself), isn't executed at all now. So we have a potential speed gain here (possibly). More about speed gain can be read in the last chapter of this tutorial.

Next let's try to find a way to get the same room image (with the gray background) when automatic draw is turned off. Actually, the only required draw command that hasn't been executed is `draw_clear`. Add this to the object's "Create Event" after the previous line of code:

```
draw_clear(background_color);  
screen_refresh();
```

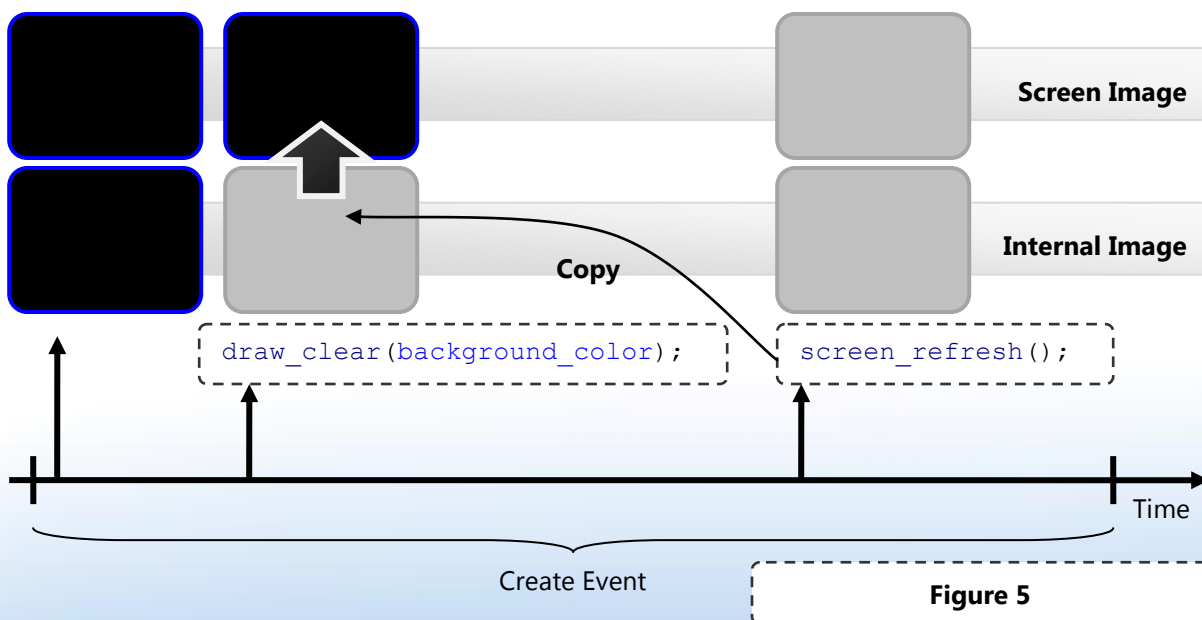
Now run the game again and see what happens. You'll see that the room looks exactly the same as when automatic draw is on. What doesn't happen here is that each step, the internal image is fully redrawn. You might start to think that it's actually a serious waste of time to redraw the room each time when it's actually not necessary. That's true, but we'll

see later on that the potential speed gain (with `screen_refresh()`) is less than expected. More important here is the function `screen_refresh()`. What does it do? According to the GM manual:

`screen_refresh()` Refreshes the screen using the current room image (not performing drawing events).

Ah, so it "refreshes" the screen. That might not be so clear, therefore: a detailed explanation. First of all, remember that automatic draw is turned off. So no drawing happens as long as you, the programmer, don't give a command to do so. But even if you do give a drawing command, the result won't be visible on screen and this is why: when the drawing command is executed, something is drawn on the internal image, but this image isn't copied to the screen. All this is, again, done automatically when automatic draw is on, but now it isn't. Figure 5 should make it clearer.

The figure shows the create event of the object and the code that is executed. For each situation, the internal image is the lower image and the screen image is the above image. You can see that only after the `screen_refresh()`



Repainting The Screen

By Bart Teunis

command, the internal image is copied to the screen image. In the GM manual, this is called double buffering; you're actually working with two "buffers" (the internal image and the screen image). So simply said, the `screen_refresh()` command does the following:

`screen_refresh()` copies the internal image to the screen image. The screen image is the image that is visible on screen. One could say that the screen is "refreshed".

Important is that, also when automatic draw is on, you can refresh the screen image at any time. You then see the content of the internal image on screen, while no drawing events have been executed yet. So you get a "snapshot" of the internal image between two draw events. Let's have a look at such an example. Create a new GM file and add a single object and room and place an instance of the object somewhere in the room. Add the following pieces of code to the object's step and draw events:

Step Event

```
draw_text(x,y,"Step Event");
screen_refresh();
```

Draw Event

```
draw_text(x,y,"Draw Event");
```

Suppose the `screen_refresh()` command wouldn't be there. Then it's logical that the text "Step Event" will never be visible, since a clear is drawn when the draw event is executed (and draw comes after step). By using `screen_refresh()` in the step event, something completely different happens. Run the executable and see what happens. Now that's an unexpected result: both pieces of text are visible. How can this be explained? It's actually quite tricky. The first time the step event is executed, the text "Step Event" is drawn on the internal image. Then the screen image is refreshed, so this image is drawn on screen. However, this is such a short time period that it's barely visible (read: not). Next, the draw event is executed: a clear

is drawn and the text "Draw Event" is drawn. Then, the screen image is, again, refreshed. So this image also appears on screen (again for a very short time). But since the next step is already there (a step only takes 33 ms), this drawing is about to be erased. The internal image now contains a gray background and the text "Draw Event". Then, the next step event comes, where the text "Step Event" is drawn at the exact same position on the internal image. Aha, so the text "Draw Event" is overwritten. That's right. And then the screen is refreshed, which results in the image you see when you run the executable. This is done every step, so that image remains the same. You will, however, notice that, at high room speeds, the text "Step Event" blinks: the time that the text "Draw Event" is visible, becomes longer than the time that the screen refresh from the step event is visible.

And finally, there's one other function that's related to drawing: `screen_redraw()`. It actually does almost the same as `screen_refresh()`, but with one difference: `screen_redraw()` first executes all draw events before it does a screen refresh. So with this function, there's no way get weird results as shown in the previous example. You just get the normal room image as you would when automatic draw is on. The only difference is that `screen_redraw()` is executed only once, and when automatic draw is on, `screen_redraw()` is executed every step (by GM itself). If you want to mimic automatic draw when it is actually turned off, you can do a screen redraw each step and you get the exact same result.

And now that you've got all that, we can go on to the last chapter of this tutorial: speed considerations.

Speed considerations

Well, yes, they're all great things to know and they do sound useful, but what is the practical use of all these

Repainting The Screen

By Bart Teunis

functions? There are two important advantages: speed gain and keeping the user informed.

Let's start with the first advantage: speed gain. When automatic draw is turned off, you have complete control over drawing. You might think that a screen refresh from time to time might do the trick, but it's not really practical to work with (you'll need to redraw all things yourself then) and the speed gain is negligible. Real speed gains can be achieved by redrawing at a lower speed than the actual room speed. Let's say the room speed is 60 steps per second. But you'd like the room to be redrawn at a speed of 30 times per second. You can use an object that sets an alarm to 2 steps. Then each 2 steps, this object executes `screen_redraw()`. The result is that drawing happens 30 times per second. You can imagine the speed gain here: drawing only happens half of the time.

`set_synchronization()` is not a good way to gain speed. It can however be useful when you want smoother graphics. But as explained before in this tutorial, it will cost a lot of CPU time.

Using `screen_wait_vsync()` also won't result in a speed gain, since the redraw frequency will then most likely be higher than the room speed.

The second advantage is that you can keep the user informed when many calculations need to be done. Normally, when code takes too long to execute in a single step, this step is executed and the next steps need to wait. Then you get the black screen you've probably already seen when the game or program freezes. This is not a good way of keeping the user informed. However, when you do a screen refresh right before the calculations, the screen image will not be black and you can draw some text to let the user know what's happening. Suppose you need to e.g. load a very large sprite (`sprite_add`). This sprite can't be

loaded in the time of a step normally takes (33 ms at default room speed). Therefore, you can add a screen refresh before this function:

```
var message; message = "Loading, please wait ... ";
draw_clear(background_color);
draw_set_halign(fa_center);
draw_text(room_width/2, room_height/2 -
string_height(message)/2, message);
screen_refresh();
sprite_index = sprite_add(/*...*/);
```

And you'll see that this looks very professional. Remember: the program will still not respond, but at least it shows what it's doing. This piece of code clears the internal image, but you can also keep the internal image and draw some text on it at e.g. (`x,y`).

Conclusion

This tutorial has explained GM's drawing in detail, including important functions like `screen_redraw()` and `screen_refresh()`. GM's drawing is automatic by default, however, this can be changed by using the function `set_automatic_draw()`. It's interesting to know how GM's drawing works because manual drawing can improve performance quite a bit. And it also allows you to keep the user informed when functions that take longer to execute than a step are called.

Voilà, that's all for this tutorial. I hope you found it useful.



Calculating a Hamming Distance

By Sam Whited

The *hamming distance* between two values of equal length is the number of characters or digits in the given value which are different. For instance, given the string "Hello World" and "Hello Workd" the hamming distance would be one (1 character, the 'l' and the 'k' differ), or, given the two binary numbers 1110 and 1011 the hamming distance would be two because the digit in the two's place and the eight's place are different. This concept is useful for approximating the error between two copies of the same data, or for comparing data. For instance, if we have two recordings of a song and we want to know if they are the same song (even though they are different recordings) we could calculate an acoustic hamming distance between the two raw streams and see how large it is, the smaller the hamming distance the more likely it is that the two recordings are in fact same song. The hamming distance is also often used to compare hashes, and most commonly, is used for error checking. Below I will present several ways for calculating the hamming distance between various data types.

Binary Hamming Distance

Normally, when someone mentions the hamming distance between two numbers, they are referring to the binary hamming distance. That is, the hamming distance between the binary representations of two numbers. The binary hamming distance is trivial to calculate using only a few simple operations. The hamming distance between two binary numbers is simply the number of ones in the resultant of **xoring** the two input parameters. For instance, **101 xor 110** would give **011** which has two ones, therefore the hamming distance is two. This means we can calculate the hamming distance between two numbers using the following:

```
var dist, val;
dist = 0;
val = argument0 ^ argument1;

while (val)
{
    dist += 1;
    val &= val - 1;
}

return ( dist );
```

Decimal Hamming Distance

The hamming distance between two decimal numbers is a little more complicated problem (since we now have ten

Quick Review

Text Drawing Functions

This set of scripts created by grumpymonkey is simply fantastic. It offers new functions to draw strings, many of which are very useful.

You can draw underlined "hyperlinks" that execute code when they are clicked, normal underlined text, text in a box, striked, bulleted, and more! The great thing about these functions is that they take line breaks into consideration and therefore render text in a flawless way.

The bulleted list feature imports the wanted text from a data stack and writes a proper list. Highly recommended! This script might end up saving you a LOT of time!

Get it now: gmc.yoyogames.com/?showtopic=286021

calculating a hamming distance

By Sam Whited

digits to worry about, whereas with binary we only had two). Just like the binary hamming distance it is easiest to loop through the number, checking each digit individually. This means that we need a function to get the length of the decimal digit, and a way to mathematically isolate each digit. Getting the length of the number proves to be quite easy, we can simply get the magnitude by taking the floor of the log of the number and then add one. After that, we can simply loop through the number, isolate each digit, and compare. Isolating individual digits turns out to be rather difficult (at least, without using multiple loops). Isolating an individual digit requires that we find the floor of the number divided by the $10^{\text{length} - i}$ where length is the length of the number calculated earlier and "i" is the location of the digit in the number (the index of the loop). We then must take the floor of the quantity $\text{num} - \text{floor}(\text{num})$ where num is the new number we have just calculated. We can then multiply by 10 to get the digit we were looking for. To do all of this, the following script is presented:

```
var len, dist, i, a, b;
len = ( floor(log10(argument0)) + 1 );
dist = 0;
```

```
for(i = 0; i < len; i += 1)
{
    a = floor(argument0 / ( power(10, (len - i)) )) * 0.1;
    a = floor((a - floor(a)) * 10);

    b = floor(vec2 / ( power(10, (len - i)) )) * 0.1;
    b = floor((b - floor(b)) * 10);

    dist = dist + (a != b);
}
```

This method is mathematically interesting, but as you can see it is rather inefficient. It would be easier and more efficient to simply convert the number to a string and compare the characters in the string using GM's built in string functions. Another plus to using strings is that you can enter binary numbers directly, making the string function type in-specific (it will work for any character representation, strings, decimal numbers, binary numbers, hexadecimal, etc.).

>>>Future Dev

Ray Tracing

Ray tracing is by far the most exciting development in the world of graphics and 3D rendering – at least, in my point of view. It is a technique that produces extremely photorealistic images by basically ejecting a dense stream of "virtual photons", using CPU calculations and physical equations of light to simulate what happens in real life to produce images with detailed diffraction, light effects, reflections, blooming, etc.

The problem with such technique is that it is very "expensive" resource wise and is seldom used in games and

other applications that require real time rendering. Instead, ray tracing is currently used to produce static raster images in a time-consuming one-time rendering process. This is because it is practically impossible – with today's hardware – to manage to render scenes in real-time while maintaining an acceptable speed for games. However, lately, a lot of developments have been occurring, such as the new OpenRT library, an experimental RTU (Ray Tracing Unit) hardware, and improved CPUs with ray tracing support. Will we be seeing such thing in games commonly in the near future?

Eyas Sharaiha

The Markup Puzzle

By Bart Teunis

Markup Puzzle: Ids and the “with” statement

Question

A single object and room are added to a new, empty GM file. One instance of this object is placed somewhere in the room. The object executes the following piece of code in its Create Event:

```
var _obj; _obj = -floor(random(5));
with (_obj)
{
    value = 0;
}
```

In this situation, four values of `_obj` will produce the same result. Which are these four values and what will be the result in this case? What will happen for other values of `_obj`? Now replace `_obj` with `_obj.id` in the “with” statement. For one value of `_obj`, an error will now occur. What is this error and why does it occur?

>>>Think...

Solution & Explanation

To fully understand what happens, we'll have a look at each line of code.

```
var _obj; _obj = -floor(random(5));
```

This line of code sets a temporary variable `_obj` that only exists in the current piece of code. Which are the values that can be assigned to `_obj`? `random(5)` returns a random real (no integer) number between 0 and 5. So the number returned by this function will never be exactly 0 or 5. `floor` rounds the number down to an integer. So the values that can be assigned to `_obj` are 0, -1, -2, -3 and -4. Right now, the value stored in `_obj` has no meaning; it's just a random number. But then the “with” statement is executed:

```
with (_obj)
{
    value = 0;
}
```

What do all these values mean when used within the “with” statement? A with statement can be used with the keywords `all`, `self`, `other` and `noone`, with object indexes and with instance ID's. The trick here is that `all`, `self`, `other` and `noone` (and also `global` and `local`) are in fact negative integer constants:

- **self** = -1
- **other** = -2
- **all** = -3
- **noone** = -4

This explains 4 of the 5 possibilities. When `_obj` is -1, the instance of the object will assign 0 to a local variable called value. The same will happen for -3, since `all` means all instances that exist and there's only one instance, which is the instance itself. -2 is a special one. There are no other instances in the room and no collision occurs. It appears

The Markup Puzzle

By Bart Teunis

that `other` is then the same as `self`. So these are 3 values in which case the same happens: 0 is assigned to a new local variable. Which one is the fourth: -4 or 0? It's obvious that it won't be -4, `noone` means no instance at all so nothing will happen. So it must be 0. But what does 0 mean in this situation? Is it a special constant? No, all special constants are negative as mentioned before. Is it an instance ID? No, instance ID's start numbering at 1000001 (you can see this in the room editor). So it must be the ID of an object. Object numbering starts at 0 and since there's only one object in the file, it's the object of which there is an instance in the room. When `with` is executed with an object ID, the piece of code is executed for all instances of this object. In this case, the code is executed for that one instance. And that's the fourth value that gives a same result.

Now there's a tricky thing you could do. You duplicate the object, keep the copy and delete the original one. You then put the copy in the room. Will something still happen when `_obj` is equal to 0? No, because there is no longer an object with that ID.

In the second question, the situation is slightly different. Instead of `_obj`, `_obj.id` is used in the "with" statement. This simply means that we're no longer working with object ID's, but with instance ID's. The full piece of code looks like this:

```
var _obj; _obj = -floor(random(5));
with (_obj.id)
{
    value = 0;
}
```

We now already know what -1, -2, -3, -4 and 0 actually are. The question is: for which value can an error occur?

`_obj.id` returns the ID of the first instance of object `_obj`. There won't be a problem for 0, since 0 is the object index of the instance. The instance in the room is the first (and the last) instance of object `_obj`. -1 won't cause a problem; it returns the ID of the instance itself. The same is true for -2. -3 (`all`) also gives the same result. The same rule applies here: `_obj.id` returns the ID of the first instance of object

`_obj`. `_obj` is now `all`. So `_obj.id` returns the ID of the first of all instances, which is again the instance itself.

The value that causes the problem here is -4 (`noone`). It's quite logical to understand, since no instance can't have an ID (when there's no instance, there's no ID either). You're asking for the ID of no instance. The error that occurs then: "Unknown variable id".

If you like to test a bit yourself, you can do so in [the included .gmk file](#).

Quick Review

Bullnet

This multiplayer script prides itself in having "The Ease Of Mplay + The Speed Of 39". Other than its speed and use of use, Bullnet has some useful features, such as an auto-updating server list, support for chat in the lobby, moods for LAN and Internet, as well as a debugging "loopback" mode.

Joining, hosting, and leaving sessions occurs immediately without slowdowns, and, like the auto-updating server list, the player list is also real time for each session.

Messages can be sent between players too. The general interface is very similar to mplay scripts and is therefore easy to use.

Get it now: gmc.yoyogames.com/?showtopic=238358

SCRIPT OF the MONTH

By Eyas Sharaiha

Thanks to Sam Whited for pointing this script out for me. I had come across it several times but I hadn't had a chance to look at it closely. The script is interesting, and though few might find the script itself useful, it can be pretty educating.

The Script: speakable_password

```
/*
** Usage:
** speakable_password(length)
**
** Arguments:
** Length      the length of the generated password
**
** Returns:
** a speakable random password of the given length
**
** GMLscripts.com
*/
{
    var result,i;
    result = "";
    for (i=0; i<argument0; i+=1) {
        if (i mod 2) result +=
string_char_at("aeiou",ceil(random(5)));
        else result +=
string_char_at("bcd fghjklmnp rstwyz",ceil(random(18)));
    }
    return result;
}
```

gmlscripts.com/script/speakable_password

The script returns a random yet speakable and pronounceable string of characters that might be used as an easy-to-remember password, etc. The way this is achieved is by making sure the random character alternates between a consonant and a vowel in the pattern c-v-c-v-....

Explanation

The interesting thing about this script is not its function; it is how it goes about doing it: the way the script is written and executed, making it pretty efficient and "straightforward", something which is becoming increasingly rare in Game Maker programming.

To achieve the effect of alternating returned consonant-vowel characters, a typical Game Maker script might create a third variable and use the "!" operator to switch between 0 and 1 for instance, each leading to a certain course of action. E.g.:

```
var vowel;
vowel=0;
... //inside for statement:
if (vowel) { ... } else { ... }
vowel=!vowel;
```

Others would perhaps make the if-statement more complicated by checking to see if the number is odd or even, using an expression similar to (in the if-statement):

```
Floor(i/2)==i/2
```

The statement uses the much more straightforward **mod** to find the remainder of *i* divided by 2. There are two possible values for the remainder of that operation: 0 and 1. A return of 0 means that the current location at which the character will be added is even (starting 0) and thus a vowel would not be added but a consonant instead, a value of 1 returned means that a vowel should be added, thus ensuring an alternating c-v-c-v-... pattern.

Using **mod** in loops is quite helpful. Other than in alternative occasions, you can also use it when you want special code to be executed every **n-times**, or –alternatively – not be executed every n-times!

Using `if(i mod 3)` for instance is used, the statement will fail at 0 and execute at 1 and 2, and go on in that manner (since `if(n)` always passes if `n>0`). Similarly, you can get **mod 3** to repeat only every n time by adding `if((i mod 3)==1)`. The "1" could of course be substituted by 0 or 2 as well. This way the event will be executed in a loop only once every 3 times, the significance of "1" (or 0 or 2 alternatively) is when will the first execution of this "special piece of code" occur with respect to the entire loop.

Special thanks for GMLscripts.com for powering the Markup Magazine script of the month!

extension of the month

By Eyas Sharaiha

GM Tween

GM Tween introduces one of the simple and awesome feature of Adobe Flash (and earlier versions of Macromedia Flash for that matter) to Game Maker: tweening.

“Tweening” is a short for ‘in-betweening’ and refers to the process of a smooth transition between an initial state and a final state.

GM Tween functionally is capable of tweening colors and numbers. Tweening numbers is a simple process of smoothly incrementing values to a number until the end value is reached. However, GM Tween manages this process behind the scenes so that all that you really need to worry about is the variable’s initial position, final position, and the duration of the tweening process.

However, the process of tweening number has some beneficial implications on the graphics side of things as well. Since image x-scaling and y-scaling values are actually numbers, you can smoothly stretch and compress a sprite from one or both directions using tweening.

Other applications of numerical tweening could be motion of a non-playable object, the rotation of a background object, etc. While it is pretty simple to manage things like gradual rotation and scaling, having to use just a few functions instead is much easier, especially when background objects like fading stars or slowly rotating clouds are the ones being managed.

Numerical tweening can also be directly useful by – for instance – managing the number of objects created per step, like enemies, etc., depending on the time, to get a certain difficulty curve.

But the real impressive feature of GM Tween isn’t its numerical tweening but its color tweening capabilities.

Color tweening changes the value of a certain color from one to another smoothly. The change of colors numerically in GM is difficult if done straightforward, since different numbers of variables do not form a continuous color

spectrum in the way that most would require when fading in and out of one color.

GM Color tweening can be used to change the background color of a certain room or the color of a certain shape that fills the room. This is particularly useful when drawing rain particles, smooth clouds, or a dynamic sky that has a sense of time passing.

The feature could also be used with the color blending property of object images to make the object’s sprite have a gradually changing tint.

The GM Tween extension is written by Tyler Gill. The extension also includes a .gml file for all of those with Game Maker 6 and 5.

Tween operations can be started, stopped, paused, and resumed. The progress of a tween can also be changed. Tweens can be set to be repeated as much times as required.

The script can also be used to call a certain function when the tween is complete, which is particularly useful when an object is supposed to wait for an animation/tween to continue before executing more actions.

Other features of GM Tween including the ability to change the *ease* of the tween, such that a “gradual change” in a certain value would not be necessarily linear. This means that if you want to change from a certain value to another, you can choose to have it start fast then slow down, or slow down in the beginning then speed up, or slow down somewhere in the end, etc.

Overall, GM Tween is a very beneficial extension, particularly useful in the graphics side when changing scaling, rotation, sizes, colors, tints, and other visual elements drawn on the screen.

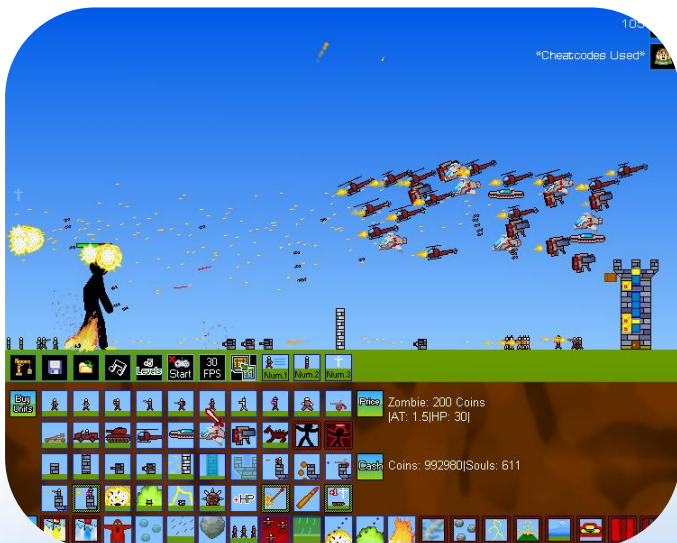
senseless war

By Shadow Master

The Review

After 17 months and 2 days in the making, "Senseless War" has finally been officially released. Long awaited by many fans since the release of its first demo Senseless War has gathered a large band of loyal followers. From the opinions of fans on the GMC it is just one of those games you could play forever.

From the very outset of the game you take command of your castle with high hopes to lead your army on to glory by advancing your military and technology as you climb through level after level, all the while keeping an eye on your budget. The longer you hold out the more difficult the game gets until you finally complete the goals highlighted out by your chosen game mode. Either "Arcade Mode", where you fight wave after wave of enemies for as long as you can with the ultimate goal of achieving and submitting the online top score upon your imminent defeat, or "Conquest Mode" where you attack the enemy castle until it's destruction, or you can play a classic game of even Capture-the-Flag, where you must break behind enemy lines to capture the flag and take it home to score.



As far as the forces you command go, if you can think of it,



it's probably there. Units range from the basic front line militia to gruesome zombies, to tanks, dragons and even the ultra-modern UFOs capable of demolishing opposing military, saving up for the more expensive ones however can be a problem for the impulse buyers like me. But power doesn't stop at military units, in fact that's only the tip of the iceberg, provided you gain enough money and even souls from killing off enemies, you can activate the most powerful weaponry imaginable: The God Powers. They strike fear into the heart of the enemy as you command the earth to do your bidding by raising hell, striking lightning or bringing total Armageddon through the "Wrath of God" or "Downfall" and so much more. But again, this game isn't all about attacking; no player will survive long without investing in the defensive capabilities, such as a variety of land mines, a selection of turrets, or the mighty cannons, with a great deal more options available too.

SENSELESS WAR

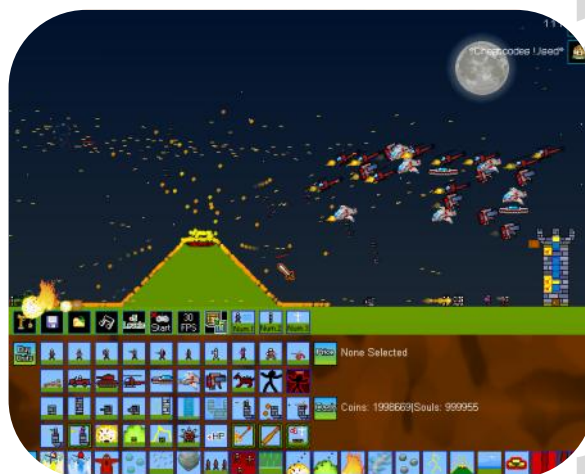
By Shadow Master

While gunning down villains and making acid rain fall from the skies, one doesn't tend to stop and think about the AI of this game, but for a brief moment here, I have nothing but praise for that area. The AI doesn't give in and is consistently mixing attacking strategies with defensive ones to make for quite a challenging foe. Despite holding no prisoners I've yet to find a single real fault or error with this intelligence, which makes the player's experience far more rewarding when you turn the tides of the war or 'pull a rabbit out of your hat!'

If there were one area of this game that I would have to be critical about, it'd be the reasonably poor graphics and lack of sounds, apart from the occasional yelp on the battlefield. The graphics are a bit too cartoonish for my liking and a little too similar to each other also. But the visual saviour is the well planned out menu and layout, which helps you to easily negotiate your choices during the war, and see exactly who's winning.

Overall, the game has gone down a treat at the GMC and I'm certain it will continue to do so in other communities. In

conclusion this is one game you won't want to miss out on if you're a big strategy fan.



>>> Scores

Ratings

Gameplay



Graphics



Sound & Music



Realism & Viscerality



Design & Controls



Overall Score



Developer:

CoffeeStained Games

Version:

Beta 20

Game Maker:

Started with 6.1, ended with 7.0

File Size:

3.17 MB

Get it now!

willhostforfood.com/access.php?fileid=30054

Interview with COFFEE

By Shadow Master

Coffee of the GMC, creator of "Senseless War" is interviewed by MarkUp's Shadow Master with regards to his game.

The Interview

Have you any plans for a sequel? Or updates?

Updates? Perhaps, since the highscore system is still a bit weird, and sequel, I've been thinking about it.

If so what extra what would you put into it?

I would probably add new units, god powers, game modes, and of course better graphics.

Can you give me a breakdown of how long you spent doing each of the Game Making areas. Such as spriting, coding etc.

Well, I can't really remember it all perfectly, but I think I spent most of my time on the Coding part, even though a lot that I made you'll never even see. I had to remove a lot of features that people either didn't like or that I didn't like, one of these features took a whole month to add. Then came the graphics, they used to be stick figures for the longest! How awful, lol. As well as a lot of other things were kind of badly made, like the lightning, the fireballs, and the menu. Thank god they've improved.

Is there anything you aren't pleased with about the game?

The Highscore Table. I'm annoyed because Vex K, the host for the highscores, is currently glitchy, giving Senseless War some unexplained glitches, of course the actual gameplay isn't affected, you'll just have to go on the website now to see the scores.

What was the hardest part of making the game?

I would say having to polish it, making everything better and faster.

How much planning did you do for this game?

I never planned anything actually. I'd just open up Senseless War and add whatever I felt like adding at the time, of course while listening to the wonderful suggestions given to me by other members who played and loved my game.

Are CoffeeStained Games working on any other titles?

Yes, we're currently working on several other titles, the biggest one being a Zombie game. Sounds genetic, but this one is going to be different, with different types of zombies and weapons you can have, you can even set up trip wires against the zombies. The name is Victus Mortuus and a demo should be coming out soon.

What inspired you to make the game, and then urged you to keep going at it?

Actually, I was inspired to make a tower defense game the day I played Middle Earth by Michael Beckers and I just loved the game. Then I decided to create my own, and from that came Senseless War. I almost stopped a few times, but with all the posts and awesome compliments I got from it, I just had to finish it.



The Review

In a land filled with Dragons and mythical creatures, you take on the role of an arrogant young man with special powers; Arron. Arron is an Angel, with the power to cast away dragons, and save lives of thousands. Although this game seems like it has *some* originality to it's cause, and could make for a reasonable gaming experience (and give hope to the stereotypically poor "story driven" Game Maker genre) I fear that it was not executed properly.

The game itself is, in want of a better word, irritating. First off the mark, we discover that there is no instructions guide in the game. Controls can be picked up bit by bit throughout the course of the game. As if this didn't bother me enough, I was faced with a long opening scene (which I couldn't skip) every time I had to restart the game (before reaching a save point). But in between my efforts to play through this title, I was consistently interrupted by cutscenes everywhere. After less than 10 minutes of play I discovered something else about this game. The creator must have forgotten about masking sprites, as the ghosts were regularly getting trapped in the main character, quickly killing either one of us. Usually me. On multiple occasions of this glitch forcing me to revert to my last save, I almost threw the computer out of the window.



Alas, the above annoyances were not the last of my troubles. Perhaps the worst of all of the problems with this game was the story line itself. It was too complicated and around every twist and bend, there seemed to be very unsystematic parts and battles, which almost forced the player to lose touch with the game story. Added to this is the final major problem. The learning curve, this game seems to suggest it's more like a learning spike or two, or three. Many parts were too simple while others far too difficult, and when saving opportunities are limited, this can be infuriating.

However, this game isn't all that bad. I managed to pick out a couple of good features, that I did enjoy. For one, the artwork (although dodgy at parts) is quite impressive. Especially that on the main menu. Another article I'd like to mention is the music, it was very well suited to the atmosphere of the game, and was very consistent, even attempting to excite you at boss battles.

As a game, it's not the worst I've played. Far from it, actually. But to really see that this game isn't that bad, one would need to look under the covers a little. This game seems to

lack some basic polishing off, which is disappointing, really, because such a simple thing could've added a whole lot more to it. For example, there were many typos which could have been avoided easily. This leads to poor Viscerality (see Markup #14) which had a 'knock-on' effect leading to a distinct failure to absorb the player into the game world, making the game seem a lot worse than it actually is.



Community opinions, like my own consider the game as something which "with a little work someone would believe it really was a game for the GBA" Several gamers, who seem to overlook the bugs and other problems, do hold this game in quite high regard, going so far as to say:

"Nice game 6/6. Lots of chapters and bosses! sweet"

Despite my opinions, and those of others, some gamers were very impressed.

Not the best game out there, but: I do feel that with just a little more effort, this game could have been much better.

>>> Scores

Ratings

Gameplay



Graphics



Sound & Music



Realism & Viscerality



Design & Controls



Overall Score



Developer:

The Zro

Version:

8

Game Maker:

7

File Size:

6 MB

Get it now!

yoyogames.com/games/show/5191

Mubbly Tower

By Blake

The Review

Coming from our good old friend 2Dcube is yet another amazing little game by the name of Mubbly Tower. Mubbly Tower is a tower-building arcade game, as you might expect from the title, in which you have to stack an assortment of differently shaped blocks to build a tower higher than a given level. Anyone who has played 2Dcube's other game titled Frozzd will instantly recognize the resemblance between the floating characters in that game, and the earth-bound ones in Mubbly Tower. In fact, Mubbly Tower is set 5000 years before Frozzd. The story follows the lovable Mubblys as they try to build up their decidedly undersized city. It is up to you to not only build the city, but defend your creation from the evil Wobblies that want to take it down to ground-level.



The gameplay is simple and intuitive. Building is as simple as giving a right click to rotate the shaped blocks, and left clicking to drop them in place. Defending is just as simple. You pick from the four different classes of Mubblys (arrow shooters, spear chucks, boomerang throwers, and the classic spitters), each having its own set of pros and cons

that make it right for one job and wrong for the other, and click to put them in action. With that said, the game is a lot harder than you would expect. The physics make sure your tower is likely to crumble as all sorts of things, from rammers to blowers, attempt to ally with gravity to bring it down. The music consists of a little looping jingle, which isn't to say it gets annoying. To be honest, you probably won't pay much attention to it anyways since the gameplay sucks you in and holds your attention all the way through. As for sounds, each of the Mubbly and Wobbly classes make their own special noise when they use their weapon, and some of them are quite funny when you aren't expecting them. There's nothing like seeing a giant yellow ball fall out of the sky, followed by a "wheel!" to make you giggle. The graphics are just as you'd expect from 2Dcube; vibrantly colored and eye-catching, with a lot of yellow. If you're looking for a deep storyline, however, you'll want to look elsewhere. There is a short comic sequence at the beginning of the game that offers a tiny back story, but besides that, there is no story to be found.



After a while of playing, you begin to feel like you're playing a mix between Tetris and Defend Your Castle. Apparently that is exactly the kind of game people want to play, as

MUBBLY TOWER

By Blake

community response to this game has been really high. People are calling it "one of the best arcade games ever." After playing the game for three days straight, I would have to agree. The game is addictive, plain and simple. I would tell you about some of the quirks, but I really couldn't find anything serious. Sometimes a Mubbly or two will get fused into a block of your tower, but they can't move on their own anyways. I would have liked to see a larger variety of Mubblys, the option to play the game full-screen instead of the forced window mode, and a larger variety of music. Hopefully we will get to see a sequel with even more content from 2Dcube's imaginative head.

Pros and Cons

Pros

- Easy-on-the-eyes graphics, though 2Dcube's favorite color is apparently yellow
- Online highscore system
- Easy to pick up, not so easy to put down

- Interesting physics engine that makes keeping your tower balanced pretty challenging

Cons

- Little to no story
- Even though there are five other modes, gameplay gets a little stale
- Bigger music variety would be nice

Conclusion

No matter how you slice it, this is an awesome game. Definitely give it a try.

>>> Scores

Ratings

Gameplay



Graphics



Sound & Music



Realism & Viscerality



Design & Controls



Overall Score



Developer:

2Dcube

Version:

6

File Size:

11.6 MB

Get it now!

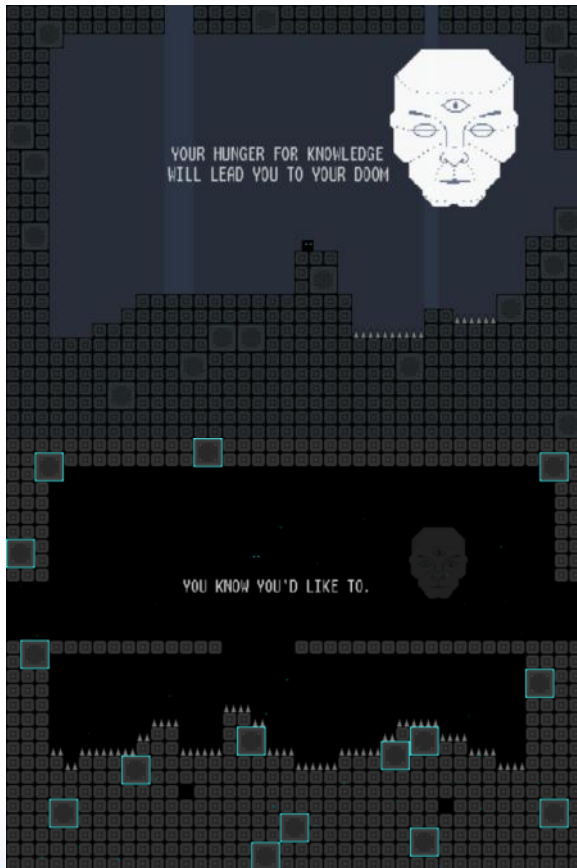
yoyogames.com/games/show/34470

seven minutes

By erthgy

The Review

Seven Minutes is a very divine and interactive game. You play as a square with blue eyes and a green outline on the outside of him. (In case if you were wondering,) the game is indeed a 2 Dimensional platformer, designed to suffice all gamers, even when the player loses. A spooky atmosphere in gameplay and resource development may scare even the toughest gamers out there. (At least I thought so!) The scary skull character in the beginning of the game made the hairs on my back shoot up for a moment... (Something that a Game Maker game hasn't done to me since... well; let's not get into that.)



Above (both): The only reason I connected level one and level two together had been because they looked like they would flow together and they do!

Seven Minutes is indeed a short game, but the most important part of Seven Minutes is to understand that you only have **seven minutes to complete the entire game**. Is this enough time to beat it? For me, no, for others (actually I'm not sure if this game ever ends) probably not. The fact that the game is a (seemingly) endless maze, and you are just one, lost soul trying to find its way through the darkness to achieve eternal life may make one wonder just how (in the plotline) the game ends. The Viscerality in this game is outstanding. I couldn't ask for better storyline, dialogue, text, sound effects, music, and overall enclosure of the player in the game. What does this game lack in? Graphical smoothness is the biggest problem. I think that if Virtanen were to round the head of the mummy a bit more, and perhaps even the small green platforms, then the game would feel just a bit (yet importantly) more smooth. The second problem I thought that this game had was not long enough play time. Perhaps, (instead of giving you seven minutes, and counting down per second... Why not give the player seven minutes, yet the counter only goes down a "second" when you die (as you do tend to die quite a bit in Seven Minutes). This isn't exactly the best solution, as the game might be too easy, or Virtanen would have to add many more temporary traps (such as falling bricks) to make the game more challenging and hard to the gamer.

The graphics are fitting; I thought that all of the graphics had a ton of work put into them. I just hope that the few (in my opinion) flaws here and there are taken care of.

As for the gameplay, it is pretty fun at first; I just think that Virtanen could vary it up a bit to make it better.

One of my problems was with the same background music throughout the whole game; that can discourage some.

Conclusion

I did have to say that this game got boring after the second or third set of "Seven Minutes" playing through this game, as every gameplay experience you have to go through (seemingly) the same levels over, and over again. Maybe Virtanen could add a point system, and when you collect 10 or so points, (or collect (what looks like) a big circular clock), then the game adds a minute or so time to the gameplay. Another thing to probably add is a level tracker, perhaps by integrating more story into the gameplay, and then some form of save files could be added in.

In conclusion to my review on Seven Minutes, this game is indeed short or small as the game is (it may look big), but let's all still all still hope that Virtanen is planning for a sequel and fixes the few afore mentioned problems... I just can't wait for it to come out (as well as many of you (after reading this article, and hopefully playing this game)).



Quick Review

Game Widgets

This is a script project by B&B_Gaming to replace existing Windows API DLLs with a pure GML-powered set of scripts for drawing and utilizing window controls.

Already, there are menus, textboxes (both single and multi-line), buttons, radio buttons, checkboxes, and list boxes... and more should be coming soon, too.

As a script, of course, it is open source, so others can have give suggestions or submit improvements as to make this set of scripts better-rounded with time!

Get it now: gmc.yoyogames.com/?showtopic=367834

>>> Scores

Ratings

Gameplay

Graphics

Sound & Music

Realism & Viscerality

Design & Controls



Overall Score



Developer:

Virtanen

Get it now!

yoyogames.com/games/show/26339



Interview with Virtanen

By erthgy

We got an exclusive interview with Virtanen on his game, 7 minutes. This game is an extraordinarily (means uncommon) fun puzzle game. These interviews are exclusive, so all of you that don't normally pay attention to these listen up.

The Interview

What inspired you to create 7 minutes?

I wanted to do something different for a change and when I found Cactus' winter competition on Game Maker Games forum I thought it was the perfect excuse for me to make an untypical game. Cactus has made some really odd and awesome games and I wanted to make something like that too.

Interesting, what gave you the idea for the 7 minutes function in the game?

I wanted the game to have more scary and serious tone and so I went through some ideas that were related to death and dying. I was thinking about the time we have on this planet and how short it was and then it just hit me. I thought it would be interesting to see how people would react when you tell them right from the start that this is the end, you are going to die in seven minutes.

Would this have anything to do with the Egyptian religion? I am just wondering because the guy's head in the background looks like an Egyptian maybe going into afterlife or something.

No, not really. I don't really know anything about Egyptians or their religion so all possible similarities are just a coincidence. The floating head guy is just generic god-like figure with an added third eye.

How long (hour wise, and day wise, as in, how many hours a day) did you work on your game total?

This game came to together amazingly fast. Everything went easily in the production and there were hardly any bugs that bothered me longer than a couple of minutes. I started working on the game on Monday and the whole game was done before the end of the week. On some days I worked on the game for as long as three hours and on some days less than thirty minutes. I guess the total amount of time spent would be somewhere around 10 hours.

So, will you ever release an open-source version of the game, or keep it together as an Executable version to keep it more directed for gamers, and not for game makers?

I don't think that I'm going to release the source. Seven Minutes doesn't really have any special/secret coding anyway. It just has the basic platform physics that everyone can learn from the basic Game Maker tutorials. I'm not that good of a coder anyway, so people wouldn't probably understand my code.

What do you do a lot around the community?

I haven't lately been that active here on GMC but when I go here I usually check the Game Maker Creations and the WIP forums. If I see a game that looks promising I give it a try and maybe drop a line or two. I also check the Community section from time to time to keep up with the latest Game Maker news.

What are you best at: Graphics, Implementing Game play, Sound/Music? Could you give us a few tips?

I'm terrible in making music, so I'd say that my strength lies in the graphics department (pixel art) although I still have a

Interview with Virtanen

By erthgy

3DX 1.5

For those of you who are new to the 3D capabilities in Game Maker and not so experienced with them, this new GM Extension adds 3D library to your system that is much easier to use than the default 3D libraries. The extension uses very simple drag and drop actions to create great 3D games. For the more advanced users, there are actions such as lighting and importing objects from .obj, .vtx, .c, .asc, .x and .mod formats! It also has effects such as weather and rain which can also be used to boost the quality of the game.

Get it now: gmc.yoyogames.com/?showtopic=300273

Quick Review

lot to learn. When I started making games 5 years ago, my graphics were very rude and unpolished but I kept trying and eventually got them to work. So I'd say that the key to success is to practice, practice and practice.

Was there any set style to the game that you chose? Did you get a base sprite and then base every other sprite off of it or what..?

I knew I wanted the game to look simplistic and retro(ish) and also the graphics had to be easy to do, but I still wasn't going for any specific style and it just kind of evolved during the production. I tried different approaches and then chose the best one.

Did you or have you ever done a planning sheet out for any of your games? Plus, have you ever wished you would've done a planning sheet in the past?

I write lots of things down on paper and then the big pile of

paper I have lying on my desk gets bigger and bigger every day. I usually try to plan my bigger projects more carefully, because they quickly become hard to manage if you don't keep track of your progress. Seven Minutes was a small project, so I didn't really plan it that much.

When you're planning for your game, does it ever seem easier if you plan object for object, code line by code line, or just list out features and tell what this object does... What that object does... etc..?

I don't usually write down any code, just the ideas and things for the game. I think it's easier to write the code in Game Maker because that way you can easily test if it works.

What advice do you have for people that are depressed at Game Maker, and feel as if whatever they do doesn't work or go for good?

If you feel like you don't have the energy to finish a game, I'd recommend working on projects of smaller scale. Sometimes it helps to take a break from game creation and come back later with fresh ideas and full of energy.

Do you ever plan to sell a game commercially or even start a company? If so, would you mind giving us all the details like what ESRB rating, what publisher, what console etc.

I don't really have any plans for going commercial right now. I'm pretty happy just making freeware games and letting people play them for free. Ok, it would be awesome to create games and make a living out of it, but at this point, I don't think my games are up to it. Maybe things change in the future, who knows? But for now, this is just a hobby for me and a hobby should be fun.

Silent Dawn

By erthgy

Silent Dawn is really an amazing title for any Sci-Fi, survival horror fan. The game starts you off on the earth, on November 14th (year unknown) with a Glock behind a barricade. You have to use your mouse and flashlight, and be on the ready for the waves of oncoming zombies.

Players must use their flash light to see the oncoming zombies that are seeking to kill them (shown below). However, if you want to survive, you'll have to buy the night vision goggles (also shown below, notice the comparison. Osborne did an amazingly good job on illustrating the importance of backgrounds, look at that grassy background ☺. I'd also like to commend him for his interesting layout on the road tiles and the barrier.



The weapon selection screen works as follows: 1) Players may buy Night vision goggles, a survivor (up to 6), and health for their barricade (once the barricade gets destroyed, so do you), and lastly, up to 5 guns. 2) Players left click on the primary gun to shoot at the enemies (default gun, or if you switched your gun, you could easily press the letter "1" to switch back), and right click on the secondary gun (you can also easily get this by pressing the number "2", or press space).

You have to admit, Lewis did a pretty good job on the death screen.

The plot is amazingly scary and short enough for even the most hardcore, non-English loving players to read through (you can speed through type-ins like this by holding down the space bar), understand, and adore every aspect of it.

In survival mode, players will find themselves with a flashlight, an AK-47, full health, and a bunch of **angry** zombies.

Now, when I was interviewing Lewis (guy that created Silent Dawn, also known as OCGaming), he asked me "How in the world did you kill that many zombies! I can't even do it!" So, like all of you Markup fans, I just had to keep it a secret until now (to tell you the truth, I didn't even tell Lewis yet!). In these next (short) paragraphs, I will reveal my level by level gameplay tactic for beating Silent Dawn!

Ok, so, in the beginning of the game from day one, just shoot all the zombies with the Glock, and save your money up to buy the night vision goggles, (that will help you tremendously). Once you have the night vision goggles, buy all of the guns up to the Uzi. Then what you do is gradually buy 3 of the survivors, and then max out your gun power by gradually buying more guns (once you get the money to do so, of course). Then buy all of the survivors, and by now, you should be up to day 25. As your primary weapon use the M4A1 and fight the zombies when they are at a long distance, then pull out your AK-47 once they get up close. This tactic works perfectly for large hordes of enemies as it subsides them when they are far away (as the M4A1 is more powerful than the AK-47, yet the AK-47 takes less time to reload) and then annihilates them once they are close up.

Some Improvements: The allied AI was a bit stupid as they **only** used Glockes, and **only** shot at such a close distance,

Silent Dawn

By erthgy

another problem was that there was only one type of enemy. Sure, it was fun and all, but it gets slightly repetitive towards (oddly) the beginning of the game, as there's really not much of a challenge to it with only a few zombies coming at you. I also wish that there were more guns and perhaps a vehicle or two into the game; maybe on the last day you have to fight to the helicopter instead of it all just going to the next day. Perhaps another game mode or two on Multiplayer like a 4v4: 4 human players, and 4 zombie players, (except the four zombie players get to control more than one zombie).

Conclusion

This game gets top grades for gameplay, perhaps more variance would add more flavor to the game. The graphics in the game are stunning; no matter how hard you try, you just can't get better graphics than this.

The music and sounds were quite spectacular and spooky, and to improve on this would mean to vary up the

background music according to how many zombies there are. Perhaps when there is a large horde, it slowly transmits from one background music to faster paced, more epic background music.

The controls were very casual and simple to pick up, the weapon controlling is a little edgy to get used to, but you'll get used to it "right as rain". As to the replay value, its fun to feel the accomplishment you get when you beat the game and compete online in survival mode, but Lewis needs to add a few more gaming types to suffice fans.

I think that Viscerality is a huge, and powerful part of this game, there really aren't that many improvements Lewis could make to the game at all!

An astoundingly great game made by OCGaming may actually convince some users who don't use Game Maker to start using it, if the game had enough publishing. So far, Lewis should consider going commercial, especially if he can make the core parts of the game in 5 days!

>>> Scores

Ratings

Gameplay

Graphics

Sound & Music

Realism & Viscerality

Design & Controls



Overall Score



Developer:

OCGaming

Get it now!

gmc.yoyogames.com/?showtopic=377843



Interview With OCGaming

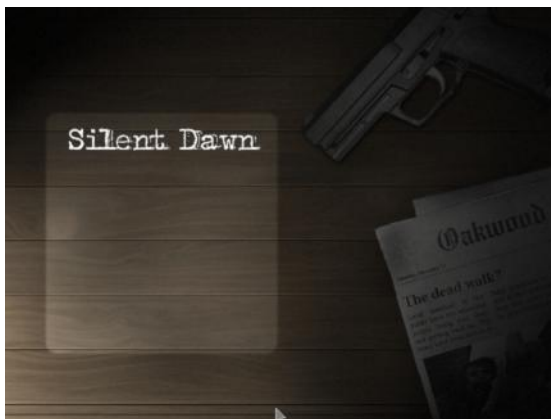
By erthgy

A while ago, I had this interview with Lewis (OCGaming). Now, I asked him a few questions as to how he made and got into the development of his game Silent Dawn. I always wondered a few things about the game, and here I hope to answer others (as well as my own :D) questions as to how Lewis made Silent Dawn. When reading this, please note that it was an AIM discussion, and I took out some AIM aspects to the names, such as date, time etc...

The Interview

What inspired you to create Silent Dawn?

Well, my friends and I were having a chat about a thing called a 'Zombie Plan'. This is where we talk about what we would do if Zombies took over the country. My friends came up with stupid plans, like getting a chain machine gun and going on a rampage. I decided that I would build a barricade and try and contact help from a radio, because it would be a life situation. Then the idea hit me. Why not make a game just like that? I'd put myself in the characters position, fighting the zombies off until help arrived.



Did it really take 5 days to create the game? Or was that just a rumor..?

Well, the main game was indeed coded in 5 days, but of course I had to go back and tweak minor things in the

engine, like Zombie spawning, weapon settings, prices of certain things and the like. But yes, all graphics, engine code and all that was all done in under a week.

Wow, so, in that week, how many hours did you work on the game? (Also, overall how many days and hours did you work on the game?)

I reckon I must have spent at least 3 hours on the game each night, correcting bugs, making more graphics and so on. I reckon so far I've spend at least 48 hours in total on the game so far.

Wow, so, what updates are you working on for the game, and, what ideas have you considered?

Well, I've been trying hard to add what people want into the game, but as you know, once a game engine is coded, it's pretty hard to go back into it and add a completely new thing without disturbing the game engine already, so adding things like traps and that would have been good if I decided to do that from the start. Things like multiplayer and co-op are hard to do when you have everything coded all ready.

I've been tweaking the Zombie count more, as well as the weapon settings, in terms or accuracy and power in the weapons. I've made the survivors able to see longer distances, so they can hold back the zombies later on in the game when things get a little hectic. I'm also planning on redesigning the menu too for the new update.

Are you planning for a sequel?

I'm not sure. I have a lot of projects out of the Game design area as well. If I have enough time, then I would love to make a second. I was thinking of making the second game in a forest area.

Interview with oegaming

By erthgy

Are there any ideas as to how the plot could flow together? Like, maybe the helicopter could break down?

I was planning on having the second game have the refuge point already taken by the Zombies, which makes them have to change course and head into the unknown. They run out of fuel and have to hard land in a forest, where they have to fight other creatures that have been infected too. I was planning on taking away the 'Day' genre of it and adding a free roam type feel to it.

Nice, also, about the other projects you mentioned earlier, what type of games or programs are you planning to release soon?

Well, the thing is, I do a lot of other things other than game design. I like to produce machinima, and I am currently working on a series called 'Fatum', being filmed in Halo 3.

(Side question, what is machinima?)

Machinima is a collection of associated production techniques whereby computer-generated imagery (CGI) is rendered using real-time, interactive 3-D engines, such as those of games like halo, half life, and counter strike.

Are you a paid developer, or is it just an independent (for fun) type thing?

It's simply a hobby. Most things that I do (Voice Acting, Game Design, Graphics, and Producer work) are all because I enjoy doing them. I don't make any profit from what I do.

Last stand?

There is a game on new grounds called 'The Last Stand' <http://www.freewebarcade.com/game/the-last-stand/> which people think I copied to make Silent Dawn.

When you started developing Silent Dawn, did you even know about Last Stand? (To me, they

look nothing alike, and besides, doesn't that discourage you towards development?)

I didn't even know that the game existed despite such claims that it looks so similar. I only found out about it when I was looking on new grounds after a post said that there was a game on there, so I searched for the keywords 'Zombie Survival Game' and the game popped up. Although now that I *do* know of the game, I guess I could take a look at it to see if there is anything cool I could add if I make a second game. After all, why not, it would only make the game better.

I have a few questions pertaining to how you created the game, how did you do the night vision goggles?

For the night vision goggles, I had a global variable telling me if the goggles were on or not, let's say that this would be called `global.nightvision`. When the user presses N, it would set the global variable to true. Now, I disabled the surface drawing when the variable was true, and I drew a green rectangle over the view with an alpha of 0.5. I also drew a sprite with an alpha of 0.2 (the tiny lines you see).

OK! What about the slightly inaccurate guns?

For the guns being inaccurate, I used this code for the accuracy of the bullet:

```
direction=angle+choose (random (3), random (-3))
```

this would choose either to make the direction a random of -3 or a random of 3.

Nice, what about the lighting?

For the lighting, I made a surface which was 16 times smaller than the room, and I drew it 16 times bigger. This is because it uses less ram and keeps the game running fast. I drew blended sprites (the torch and lights) to the surface.

Interview with OEGaming

By erthgy

There are many lighting engines on the GMC that can do this for you.

Lastly, can you tell us about the enemy's zombie generation technique(s?)

For the zombies spawning, I made a global variable that went down each day. I made the Zombies spawn on a timer using the global variable. As the global variable goes down, the timer gets faster, thus creating more Zombies. I also made a check to make sure that there wasn't too many being made.

Nice, also, on some of the later levels (day 28 on) for me, it was a bit laggy to me, now, I'd like for you to know that it wasn't that laggy, and my PC isn't THAT new, but all of those objects are quite a stretch to the PC run time, perhaps you could simply increase the health of the zombies instead of making many more to therefore lower CPU usage.

I designed the game for people who had 1GB or maybe 2GB of ram. I wanted to give the player the feel of being swamped with zombies. I'm not sure how I can make the game go any faster in terms of CPU usage. Changing the zombies would affect gameplay, and that would be unfair to those who have a score on the rankings.

Have any topics on the GMC stood out for you?

There are quite a few to really list actually. I treat all topics as equal. I'm more of a browser than a poster really.

Have you ever thought about entering your game into a competition (particularly the ones on Yoyo Games?) I think your game could get a bit of success

I think the current theme isn't applicable for Silent Dawn, but when the theme for survival comes along, I'll be sure to

enter it.

(Side question) Do you read Markup Magazine?

I've read a number of issues when I have had to chance to, and I think that the layout is brilliant and easy on the eyes. It has useful information that is handy to any game maker user.

Are you ever going to go commercial?

I don't think that people should pay for the games that I make right now, as they are not quite commercial quality. Once I start getting into 3D and so on, I'll work on some games that will be high quality work and I will be able to sell them. But for now, enjoy free games from me!

What about ads, they generate some pocket change, and still keep the game free...?

That's a good idea, although I'm not one for clogging up games with advertisements. I would provide a splash screen at the start of the game, but I wouldn't want to have ads in game. I don't like the idea of ads getting in the way of game play.

So, what are you working on at the moment?

I'm currently working on the latest update for Silent Dawn, and another game called 'Arrow Tunez', which is a clone of Dance Revolution and Guitar Hero mixed into one game. It's a short and simple game, but fun to play. I'm trying to perfect Silent Dawn, and all that I am capable of until I let it be the final build.

Conclusion

Indeed, I was intrigued as to how Silent Dawn could handle so many zombies without lagging (except slightly when on humongous numbers such as 100 by 100). Well, now we know!

D~A~N~C~E

By erthgy

From the time of the download, to my entrance upon the main menu (as solemn as I had been); I was not impressed.

I opened up the song list; I still, was not impressed, but something caught my eye.

and I would want them to be better, but what can I say? Load your own music :D!)



After finishing the default song Frozen Fire, I looked at the second option of how to play the game: Load MP3.

I was surprised! What does this mean? After finding out about this option, my mind was blown! Ever want to import your own song files into a game and play along with them without "hacking" (I put hacking in quotes because it's really not that hard to do with open folder games, such as a few externally loaded Game Maker games to save loading time.)

The way that this works is as follows: 1) Select the difficulty of the game 2) The user must go to the option "Import song from Computer", 3) The user must browse through their files to get to a .WAV or .MP3 song file (note that if you have a few songs in iTunes or some other music player you could always search your PC for the .MP3 version) and 4) The player then must open that particular file, and then you're off to go! (The arrows to the music beat are a bit off,



Above: I would recommend true beginners to start with the difficulty of "beginner" but it's truly up to them (I started with light and I was fine).

The game DANCE is just what it sounds like, an awesome, PC version of DDR that allows you to play through your own music.

The only few problems I have with this is that 1) It can be hard for me, (personally) to find my own music files, as I have a quite cluttered computer memory storage drive... 2) The game is a bit hard for the 4 difficulty modes that it has (Easy, Normal, Medium, and Hard). 3) The menu can be hard to navigate through for the first time, but once you get used to it, it's not that much of a problem. 4) Not enough default songs. (I'd say that there should be at the very least 10 or more.)

D~A~N~C~E

By erthgy



After all of that, DANCE is an intuitive game. Perhaps Nightmare could start a little fan club and users could start make songs for his game on his website forums? The possibilities are seemingly endless...

With the many promises that Nightmare has for his fans in the interview I did with him; I think we all would be surprised not to see DANCE and Nightmare's upcoming

projects on ever Game Maker gamers' top 10 "games to play" list.

Conclusion

Overall, the graphics are not the best that I've seen, but they are fitting, and they fit their purpose. As for gameplay, Nightmare's arrow directions and timing for custom loaded songs is off tune, and sometimes can be unfitting to this game.

The very few sound effects that I've heard in the main menu so far were interesting, sort of like a pop culture kind of style and the music is a very subjective thing, of course, but Nightmare's preloaded song isn't that great.

As far as controls go, WASD would be a good default, double variable checking, and allowing the user to change one of those two variables which is the key they press for a direction. The PC checks the other variable; the default control sketch is the arrow keys.

>>> Scores

Ratings

Gameplay

Graphics

Sound

Music

Design & Controls



Overall Score



Developer:

Nightmare/Version

Get it now!

gmc.yoyogames.com/?showtopic=345615

Interview with Nightmare

By erthgy

Editor's Note: DANCE is a project of Verision, which is comprised of Nightmare and ad134. This interview is with Nightmare, representing Verision Games.

The Interview

Could you give us a little (ok, maybe big) preview of your upcoming game DANCE REVENGE?

My next game titled "D.A.N.C.E REVENGE" is a major update to DANCE, if you would like to, we can talk about that during the interview a little and I will send you a few screenshots of it.

The difficulty modes are getting tweaked in DANCE REVENGE. Also, the creator of a song file can create their own difficulty. (This can also be done with ROCK...)

The menu system in DANCE REVENGE is getting a complete overhaul. It will also be fully skin able.

The song selection list is also getting changed. There will be more options like the ability to delete songs from the list, add new songs to the list (Hotkey rather than option on the menu) and there will also be more details on the song selection screen such as the tempo of the song, and may also display high scores for the songs.

Verision Games will take all suggestions on board, if anybody has any suggestions please post them on the D-A-N-C-E, the GMC topic, or the Verision Forums. (www.verision.co.nr).

Are you doing anything about the new YYG competition? Have you posted DANCE or are you going to post DANCE REVENGE there?

That wasn't part of the original plan since I never knew about that competition, but now that you have mentioned it I will look into it.

If you have used/seen the site yet, then how do you think YYG could improve on their site?

Well first of all I think they should include support for the Opera web browser (Which is a very good browser by the way, you should try it).

I also think there should be some sort of filter for games which aren't very good; I'm not sure how they could do that... maybe a rating system that places lower rated games at the bottom of the page. (I haven't been in the YoYo Games' games section recently so they might have that already.)

Did you go through the Game Maker's Apprentice or any books/tutorials of Mark's?

I didn't go through the Game Maker's Apprentice since that wasn't out when I started, but I do remember doing the Capture the Clown tutorial many years ago, apart from that everything I've learned is mainly from experimenting or browsing the manual.

Do you have any ideas that Mark Overmars (and YoYo Games for that matter) could include in Game Maker version 8.0?

I think the main thing for Game Maker to improve on is Portability and Speed, and as far as I know Yoyo games' games' are working on these already (e.g.: GM for Mac). With the C++ rewrite, GM may (or may not be) faster depending on how optimized the runner is. With GM games getting bigger and bigger all the time, I think this is a good area to focus on. (GM will never be as fast as C++ itself, but the faster the better.)

A small handy thing I can think of for the code editor would be, when there is an error detected in the code the editor will change the font color of that text to be bright red, rather than just moving the cursor nearby. This will speed up error locating by quite a bit.

Interview with Nightmare

By erthgy

What else would you like to talk about?

I think a worthy topic of discussion would be Verision's eXteNsity Media Player and The Grav Warper.

EXteNsity media player is designed to communicate with D-A-N-C-E REVENGE and ROCK, there will be options in the player to load the currently playing song (or the entire playlist) with either game. EXteNsity also supports the loading of .pls files, so with eXteNsity you will be able to load your Windows Media Player/Win amp playlists with DANCE REVENGE or ROCK.

Another one of Verision Games' project is The Grav Warper, it's an original platform game that allows you to warp gravity to get around obstacles and puzzles, and there is a Beta version available right now on the Verision Website. (www.verision.co.nr)

Have you been on the Game Maker community lately? If so, did any topics stand out?

There aren't really any topics on the GMC that have stood out with helping me create D.A.N.C.E, however the SXMS DII topic has helped me a lot in creating the eXteNsity media player which will have a test version released soon.

Sweet, so when will you be releasing DANCE REVENGE?

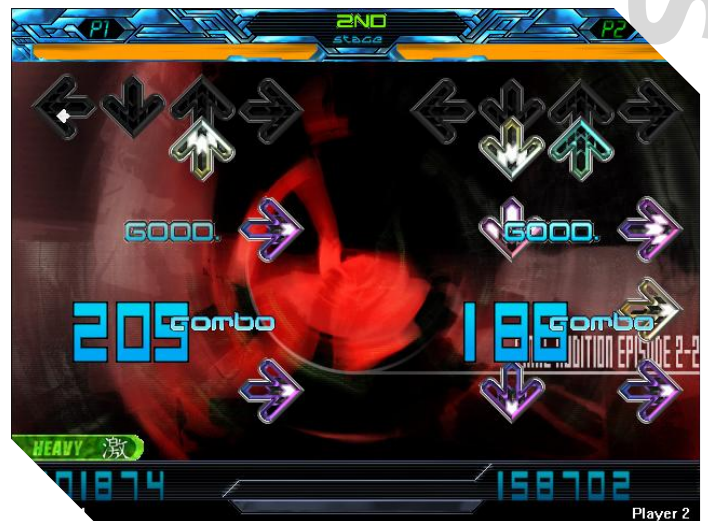
There's no solid date yet, but it will be sometime in autumn.

Also, I'm planning to keep track of development progress on the main website; it will have detailed info on how far DANCE REVENGE is getting in development, so people can look that up on there.

Dance Revenge Preview



Single Player Gameplay



Two player gameplay



GOODBYE!

Thanks for Reading Issue 15 of MarkUp Magazine! We realize this one took quite a while, but we certainly hope the wait was well worth it.

In this issue, we introduced the new MarkUp Puzzle monthly special, and hopefully in the future issues we will have more puzzles with different degrees of difficulty and hopefully one big challenge for our savvy readers as well. The Issue also includes a new color scheme and a slightly updated look and feel graphically. We also introduced our new logo, which we hope shall inspire our readers and contributors and tell them what MarkUp Magazine really is about: the science of Game Development.

GMking.org in general went through some interesting mergers as well. GMking.org's wiki service – Gmpedia.org – has merged with the YoYo Games wiki to provide a much bigger more centralized area for Game Maker content. GMking.org also merged with Dan Egger's Game Maker TV (GMTV) to provide that service under the GMking.org umbrella, stay tuned for the upcoming episode of Game Maker TV.

MarkUp Magazine is made, maintained, and supported by people like you! So, if you like us, make sure you support us in a multitude of ways; you can [contribute](#) to MarkUp Magazine and submit content such as tutorials or editorials that you have written, or you can [apply to staff](#) and become a fulltime writer to the magazine. Please share all your opinions and feedback regarding this issue, MarkUp Magazine in general, and anything else on our [GMC topic](#) or contact us by [mail](#)! Thanks a lot for supporting MarkUp Magazine and making it what it is today!

The MarkUp Staff■

[GMking.org](#) is the parent network for **MarkUp Magazine**. It is constructed as to behave like a centralized portal that links to the main aspects of GMking.org and its projects. These include MarkUp Magazine, Game Maker Television, the Game Maker Audcast, and the GMking.org Forum. GMking.org also assists in the maintenance of the YoYo Games Wiki: make sure to visit the wiki and contribute there!

GMking.org *Let them make games!*

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

©2008 MarkUp, a GMking.org project, and its contributors. This work is licensed under the Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA. Additionally, permission to use figures, tables and brief excerpts from this work in scientific and educational works is hereby granted, provided the source is acknowledged. As well, any use of the material in this work that is determined to be "fair use" under Section 107 or that satisfies the conditions specified in Section 108 of the U.S. Copyright Law (17 USC, as revised by P.L. 94-553) does not require the author's permission.

The names, trademarks, service marks, and logos appearing in this magazine are property of their respective owners, and are not to be used in any advertising or publicity, or otherwise to indicate sponsorship of or affiliation with any product or service. While the information contained in this magazine has been compiled from sources believed to be reliable, GMking.org makes no guarantee as to, and assumes no responsibility for, the correctness, sufficiency, or completeness of such information or recommendations.