

game reviews

game development

interviews

code

tutorials



first look

"Zaon Code 3025"

Fast 3D Shooter and Adventure game get blended in this new Morphosis Game!
Sneak Peek inside!

Game Maker 5

1st Look at the New Game Maker 5
You got to see and read it!

interview

In-Depth interview with the creator of Game Meker
"Mark Oyermars"

reviews

Crafts - Crazytown - Stardust - Fort Defender V2.2 - Sonic Zone

GMDM Issue #1

C. 2003 Morphosis Enter-Actice

March 2003





Morphosis Enter-Active
Morphosis Games
c. John Hempstead
Contact Me

3. *Creator Words*

This is the first mega issue of the new online Game Makers data magazine from the one and the only Morphosis.

4. *"Zaon Code 3025"*

Sneek Peek at the most recent project I am working on. A cool combination using Game Maker and the 3d Game Maker.

6. *Development Tutorials*

6. Good Programming Part 1.

7. Isometric tiles made easy in photoshop.

12. *Game Design: Players Needs.*

Read up on what players look for and need in a good game.

14. *Mark Overmars*

An interview with the creator of Game Maker Mr. Mark Overmars.

16. *Reviews*

Cool and Un-cool games make it in the first issue, read up and play them if you don't take my word.

19. *Game Maker 5 "First Look"*

GM 5. Mark Overmars has done it again!

25. *GAME MAKER vs FLASH MX*

What one got the other one does not?



Some people play games
Some people make games

download gm 4.3

To keep Game Maker free in the future we ask you, if possible, to make a donation of \$ 5 or 5 Euro (or more) using PayPal or your credit card, by [CLICKING HERE!](#) ●



*are you
ready to morph?*

morphosis enter-active

Creator Words

Welcome to this first issue of Game Makers Data Magazine. I hope you enjoy this and look forward in reading this and future issues. I'll sum up what this magazine is. It's a Game Makers mag, a Multimedia Makers Mag, a Programmers Mag, an Artist Mag, and a Game Players Mag. There really is not one topic I plan to cover or just one Game Making program. I plan to cover many areas of "Interactive Development".

I will focus on Game Development and swing around in other multimedia as well, such as web page design and other program development.

As of now this GMDM is all done by me, I am looking for writers, reviewers, and even artist. So if you are interested please contact me. Please enjoy.

Morphosis
John

News Flash

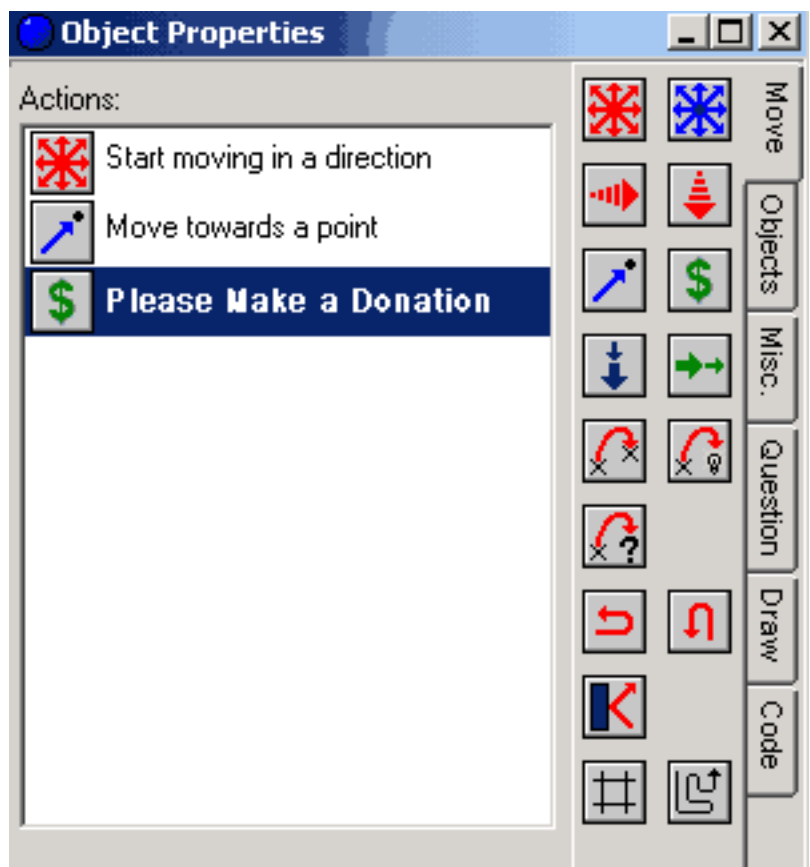
Mark Overmars made some imaginations of others run free by creating Game Maker. A fully FREE program that is worth much much more!

Now Mark is asking that the users of Game Maker to donate a few dollars so it remains FREE. I do understand that GM could go commercial and good money could be made, but a possible loss of the community GM has may be there. So a simple donation is needed to keep GM free. And if the donations are not up to par then GM may need to be registered, in future versions, but the ones who did and plan to donate will get a registration code.

Now I think Mark does deserve something for all his hard work and think we should donate just a little. Even if you are making games for the heck of it to past time, donate and donate especially if you plan to sell your games.

Upset? Mad? Feel like Mark planned this "Scam"? Well he did not plan a scam and you should not be mad!

As of January 18 2003, Mark got in 4 weeks only 12 contributions, that is really not good with a GM community of over 1000 people.



New feature in GM 5?

Also just take note Game Makers site is now:
<http://www.gamemaker.nl/>

Sneak Peek

ZAON CODE 3025

c. 2003 morphosis enter-active



Game Maker: 1 Map area in the world.



Game Maker: Your base, make your move.



Game Maker: Got Code? Hack In!

The Creator:

Zaon Code is a project I started in the start of winter. Since the colds snow made me feel isolated to some point, it shows in this game. I wanted to make an Erie mystery like game with a cold and dark feeling. Zaon code 3025 was started. Also I did not want to make a completely Game Maker game and wanted to mix some other game tools in which would give the game more diversity. I decided to use The 3d Game Maker with Game Maker, and even Flash for interesting cut scenes. I think I can make this game rather interesting, even though The 3d Game Maker lacks so many options, I developed the game around that and fit it in to what it was able to let me create.

Now the 3d part of the game is rather appealing, but I think it's balanced with the overall story.

The Story:

In the years before the power of the computer on earth, visitors came to this planet to bring terror to the species of that planet as they used what they could for power, they are known as "The Zaons".

At this time when they first came, nothing was found but large animals know to us now as dinosaurs. There was nothing they could use from these worthless beast, so they decided after years of testing and research to destroy them with a massive explosion that would destroy them all, and they did succeed in that bringing them into total extinction.

Hundreds of years later, they decided to come again, and the location was Roswell, USA. There was a mess up on their part and that was when they were discovered, but the US govt. doesn't admit to it yet.

The US covered up this occurrence, and the many more to follow. The people of the world thought it was what the US and many other world leaders said it was, an air balloon, an experimental rocket, and comets. The people of the earth soon believed these lies and soon forgot about it as they all enjoyed their new super advanced televisions, cars, and machines. What they all did not know, was the visitors became our neighbors. Not neighbors on another planet but here.....



3d Game Maker: Creator try to stop me!!

here on earth, next door to them, in the same school as them, attending the same church as them, and breeding like them.

Even though they lived among the people of earth for years, their intentions were still the same as they were with the dinosaurs. But they thought they would let the human people evolve which would provide more energy which means more power. When that moment is right, the final attack will come.

In 3015, the final war of the worlds occurs. Countries battle and turn the planet earth into almost nothing. So many lives and resources are lost because of this war and population decreased. In 3020, the battle continues, death and disaster surround us all. The Zaons that still roamed the planet, knew this must be the time to act before the entire population was gone. They had to "collect" the energy from us.

There plan was to activate an explosion that would gather the energy from the humans. They have been developing this for many, many years. Did we know? Yes, but it was very hush, hush. Until now!

The Game:

You play as "Eve Eston" a female opp who had trained for this day. This day when you have to stop this disaster the Zaons intend to bring on us. The only way is to deactivate this death machine through "Zaon Code". This is code that is scattered all thought the world, and you must retrieve it and hack into the mainframe of the Zaon computer and attempt to deactivate it.

Why did the Zaons need us? Why planet earth, are there other planets at risk? Why did the govt. cover this up?----these are questions you ask only to find the true answers.

You do most of you communications and hacking at your base. There you also get access to areas of the world with zaon facilities, that you must enter and get a part of the code. The game feels like "Myst" which is a point and click adventure, also you obtain the code in dark 3d environments, and there are some puzzles to solve!!

Information:

Release Date: Spring 2003

Price: FREE! Mail Only, it's going to be 200megs+

Developer: **Morphosis Enter-Active**

complete



3d Game Maker: Strange, strange, strange.



3d Game Maker: "Ahead, part of the code"

Tutorials

Guide to Good Programming and Game Making Practices with Game Maker. Part 1

By Allen Cheung

Object Orientation

What is Object Orientation? What is Object-Oriented Programming (OOP)? GM is founded upon this system of programming, but what is it really?

OOP is a *style* of programming popularized with C++ and such languages of the past; it has become widespread with the emergence of a wave of visual development languages, such as Visual Basic, Visual C++, and Delphi. Basically, the concept behind OOP is that independent objects make up the program. Each object knows how to take care of itself, and nothing more; it holds all the procedures, routines, and functions needed to run its own purposes. This is seen in GM by the various sections of an object – it takes care of its own creation, destruction, movement, counter, alarm, and input.

Event-Driven

Yet another buzzword. This simply describes how OOP is run, which is by events – the program acts only when an event is detected, and is idle otherwise. For example, GM has events for keyboard commands; the system sits idle until a key is pressed, then looks to see whether that key triggers some snippet of code. Of course, the programmer can choose to have a **continuous** event so that the game is always moving – this is embedded in GM's Step events.

Inheritance

A second feature of OOP is the power of inheritance. Because of the nature of object orientation, it is easy to build a hierarchy of objects, a tree of parent-children couplets. The purpose of such hierarchies is *inheritance* – that is, the ability for a child object to inherit the properties of its parent. For example, suppose that I have an **electronics** object, which has the property that it is run by electricity. Now, a child of this object could be a **computer** object, which not only has the property inherited from its parent (that it uses electricity), but also has the property of having a monitor and keyboard. We can go even further and build a **Mac** object, with all the properties of a **computer** object (and thus all the properties of an **electronics** object) with the addition of having the ability to produce outstanding graphics.

In any case, inheritance is a powerful concept that will potentially save a lot of programming time and possibility programming errors from duplication of code. More on this later.

Variables

Variables are the meat of any useful program; they are placeholders for pretty much everything that can change in a program. There are a number of variable types that GML uses:

Real: Real numbers, that is numbers with decimal places, such as 1.2345

String: A string of characters, such as "this is a string"

Boolean: Either 0 or 1, meaning false and true respectively

And truly, for making 2D games with such computing power, it is not necessary to include any more variable types (unlike the programming of yesteryear, where everything had to be conserved...**byte** types anyone?).

Scripts

Scripts are the equivalent of functions or procedures in other programming languages; they serve to separate chunks of code so that they can be reused or that the general program is simplified. Sadly, while the latter is sometimes utilized, I almost never see the former used. The power of functions cannot be ignored, for they provide the means to repeatedly do a task with different parameters – taking a number and squaring it, for example. As a matter of fact, they are powerful enough to invoke their own style of programming, known as functional programming and seen in languages such as LISP.

Loops

I think that this topic is general enough to warrant some discussion here. As an old CS teacher told me, computers are only good at doing boring things repeatedly, and I tend to agree with him. The existence of loops (and its manifestations in GM, as **while**, **for** and **repeat** statements) greatly simplifies a lot of code with some simple math, and they almost always work with variables to accomplish their tasks. Smart programmers will take time to learn these well – they make case-checking and otherwise tedious code much more bearable (I'm thinking of a specific code sequence found in a "Code Book" that will remain unnamed here).

Well, enough about abstract concepts. Let's get down to the nitty-gritty stuff. The first topic, before any other, is the discussion of the style of programming. This is the most important factor to good programming practice.

Indentation

Indentation is the process of making code more readable by indenting chunks of code by spaces. By doing so, the programmer can easily check for incorrect bracket placements, bad function calls, and overall program intent. Compare two sample programs:

```
for (x = 1; x <= 10; x += 1) {
y = 5;
for (z = 5; z >= 1; z -= 1) {
if (z == 3) {
y += 1; s = "Big Bertha";
}
else
{
name = 'Bill';
}
}
y -= 1;
}
```

and

```
for (x = 1; x <= 10; x += 1) {
  y = 5;
  for (z = 5; z >= 1; z -= 1) {
    if (z == 3) {
      y += 1;
      s = "Big Bertha";
    } else {
      name = 'Bill';
    }
  }
  y -= 1;
}
```

The first one is left plain, while the second is properly indented; by simple inspection alone, we can conclude that the second one is much easier to read at least – we can easily tell which block (that is, the chunks of code that are bounded by { }) corresponds to which statement. I can even attest to the fact that it took me a while to actually check that all the brackets were in the right place.

I can't stress this point enough! I've seen enough GML code with no indentation whatsoever, and it is simply a mess to read. Even with small snippets of code, this can easily create programs that are confusing. (try, for example, organizing a triple for loop with nested if statements) It does not take that much effort to properly indent in the first place, and the program and programmer will be much better off for it.

Comments

Comments were included in GML for a reason, and not just because they looked particularly nice either (although I hope that Mark will someday implement the often-used /* */ notation). Once again, they are utilized to make code readable, but for whom that really benefits is the reader. There will be times when you will stumble across code that you have written ages ago, or code written by others, and you will have

have *no idea* what it does. So, rather than frustrating your future self and readers, why not just comment obscure code? This is especially important in Scripts, where the name of the Script is usually not enough to tell what is happening, and it's rather hard to find where the game calls that Script. Just a `//` may save a lot of searching in the long run.

Naming

While not strictly a GML quirk, it's still a major topic that concerns a style of programming, and I will quickly note my point here. Simply put, make your names mean something. Calling sprites `spr<name>` is a good mnemonic to allow yourself to remember that it is indeed a sprite (especially within programs, where you're not given an easy menu that sorts out everything else), as is calling backgrounds `back<name>`, objects `obj<name>`, etc. If you are using four sprites in four directions, try calling them `sprHeroLeft`, `sprHeroRight`, `sprHeroUp`, and `sprHeroDown`...it certainly sounds more intuitive than `s1`, `s2`, `s3`, and `s4`.

Scripts

*Scripts, the way that they are represented in GM, are roughly analogous to procedures and functions. Pointless trivia: in the programming language Pascal, there are both functions and procedures; in most other languages, however (C, C++, Java, etc.), there are only functions. The difference? Simply that a procedure does not return a value, whereas a function does. C and its family of languages get procedures from defining functions with no return type (*void*). Since GM emphasizes Pascal and Delphi, we will make the distinction as procedures and functions. The importance of scripts, of course, is already highlighted in the section on general topics.*

One notational point: since scripts are closely related to functions in other languages, I will abide by the standard convention that scripts have a pair of brackets at the end, regardless of how many arguments are actually in the script. Hence, `useObjects` is a variable while `useObjects()` is a script with some undetermined number of arguments.

Scripts as Procedures

Scripts as procedures are simply scripts with no return value, so that they are ideal to perform visual operations such as displaying text and drawing graphics, as well as manipulating objects. One calls scripts analogous to procedures in code as follows:

```
Some_script (arg0, arg1, ... , arg9);
```

And declaring that script:

```
{
  for (x = argument0; x = argument1; x += argument2) { ... }
}
```

Note that scripts can have up to ten arguments when called within GML code, and that the script itself strives to only use its arguments and not anything else. Furthermore, there is no return statement at the end; the script finishes its operations and goes back to the main program.

The use of script-procedures can be further simplified into two categories: one-time procedures and multiple-use procedures. One-time procedures are created to make coding easier to understand; `set_global_variables()` and `display_graphics()` is certainly reasonable. On the other hand, multiple-use procedures are usually more general; examples include `moveObjectRight()`.

Scripts as Functions

Scripts as functions have an important difference, in that they require a return value. Because of this, most uses of functional scripts are to calculate and return some value – for example, some mathematical function. One usually seeks not to change the arguments themselves in a function, only to use them for calculations (although the key here is *usually*, for there are uses of functions that manipulate their arguments and still return a useful value). They are used in very much the same way:


```
Ans = some_other_script (arg0, arg1, ... , arg9);
```

And in the script itself:

```
{
  x = argument0 * argument1;
  ...
  return x;
}
```

The important part is that there is a return value, although one is not obligated to use that value. For example, the provided library function `instance_create()` returns the id number of the instance, but the user certainly does not have to use that number if, say, the user is merely creating some graphical menu.

As with procedural scripts, function scripts can be largely separated into two categories. The first is already mentioned, and is mostly for calculating without actually manipulating the arguments given. Hence, good examples would be `cube()` and `findSmallestNumber()`. The other kind of functional script is less used, but still very useful...scripts like the above-mentioned `instance_create()` still manages to return useful information.

Portability

Once a script is written, if it is indeed written well, it can be applied to any program or game. Collections of such functions in other programming languages are known as **libraries**, and GM has its own library of useful scripts. Unfortunately, a lot of scripts that are being written are not truly adapted for other programs right off the bat, for they are hard-coded to a specific program and require some editing on the user's part to make work. This, however, defeats the purpose of scripts – ideally, one only needs to read the description of a script to figure out its usage.

To achieve this, the script-writer is advised to write his scripts using the arguments given, rather than hard-code variables that are used in the original program that the script is written for. For example, a line of code such as `{ y = ball.x; return y; }` will throw errors in any program that does not contain a ball instance; one must edit this script to fit with whatever the script is used for. However, if the script required one argument, such that `{ y = argument0.x; return y; }`, then as long as the user passes an argument, the script will work. This is known as “adding a level of redirection” by using a variable in place of actual objects.

Part 2. will continue in the next issue.



MATCHER CATCHER

Blast 'em in da balls!

"FUN, FUN, FUN"-GRANDMA GAMES MAG.
 "ONE OF THE BEST GAMES SINCE PONG"
 "REALLY FASTER THAN TURTLES"

DOWNLOAD NOW!

Logo: EGT
 Logo: MS (MicroSoft Games)

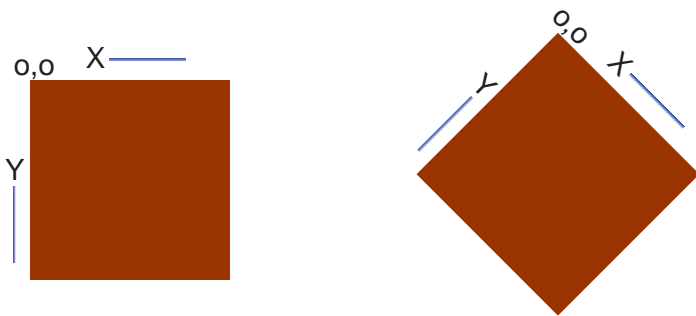
Tutorials

Isometric tiles made easy in photoshop.

By: Morphosis

An Isometric game can help a games world become almost real and make the player feel godlike since the player looks over the world and almost sees all. First of all, tiles are used for saving memory space in 2-D games. They are just square images, each storing a part of an image, and are almost always power-of-two in size (e.g. 32 by 32 pixels).

Isometric tiles plug together to form a diamond map. With 2D maps, the result is a square or rectangle. Isometric maps look like a 2D map rotated 45 degrees clockwise. Here are a couple of pictures for illustration:



The isometric view has been popularized by its "3D" representation of levels in games such as Ant Attack on the ZX Spectrum (the first isometric platformer), Zaxxon, and Atari's Marble Madness game. Since 1982, many game developers have continued to use the isometric view.

It should be stressed that the isometric view should be fully understood before deciding to use it in a game or application. Isometrics should only be used when illustrating depth or providing a non-standard way of looking at something. Isometric views tend to be visually complex, and unless used correctly, can take away from game play.

Most non-isometric views are oriented in a Cartesian straight on fashion. Because the screen is square and the x coordinate increases as you move left and y increases as you move down, this is very easy to draw contents to the screen. Most platformer games like Super Mario Bros. use this view. Unfortunately, there is no perceptual depth in most of these games. Overlapping, scaling, and parallax scrolling are tools most often used to illustrate depth.

Parallax scrolling basically means that something behind and in front of the object of focus is being scrolled in parallel with the object of focus. Objects in the foreground will scroll quickly; background objects will scroll slower. Parallax scrolling does a nice job of illustrating the depth comparison. However, parallax rarely plays a meaningful role in the game play. It is easy to think of an isometric view as the view from a camera, flying in any direction horizontally, over a landscape looking downward at a constant, unchanged angle.

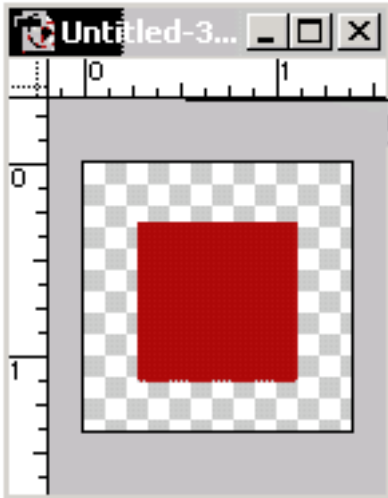
Ok now let's begin. Here you need photoshop or paintshop pro and even other programs might have functions like the ones I will use here.



Fallout

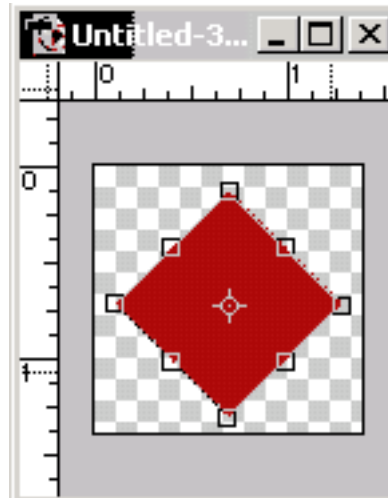


Sim Golf

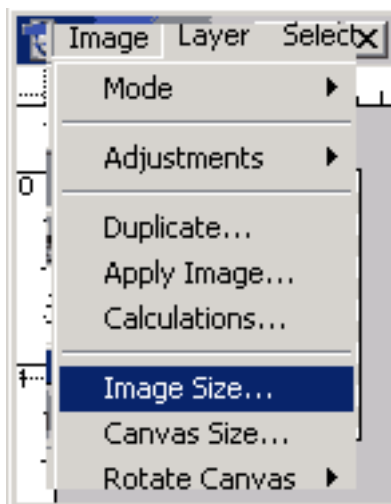


It's simple! Make a new file that is 100x100. In the center make an equal sided square. Hold Shift in photoshop when making a square mask.

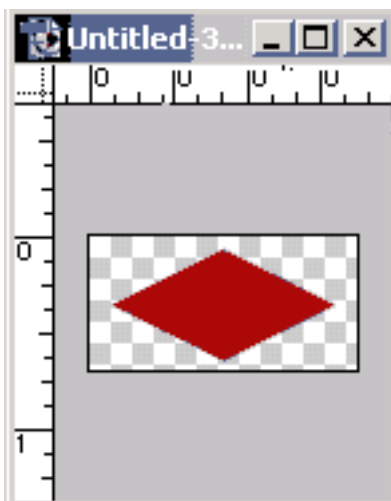
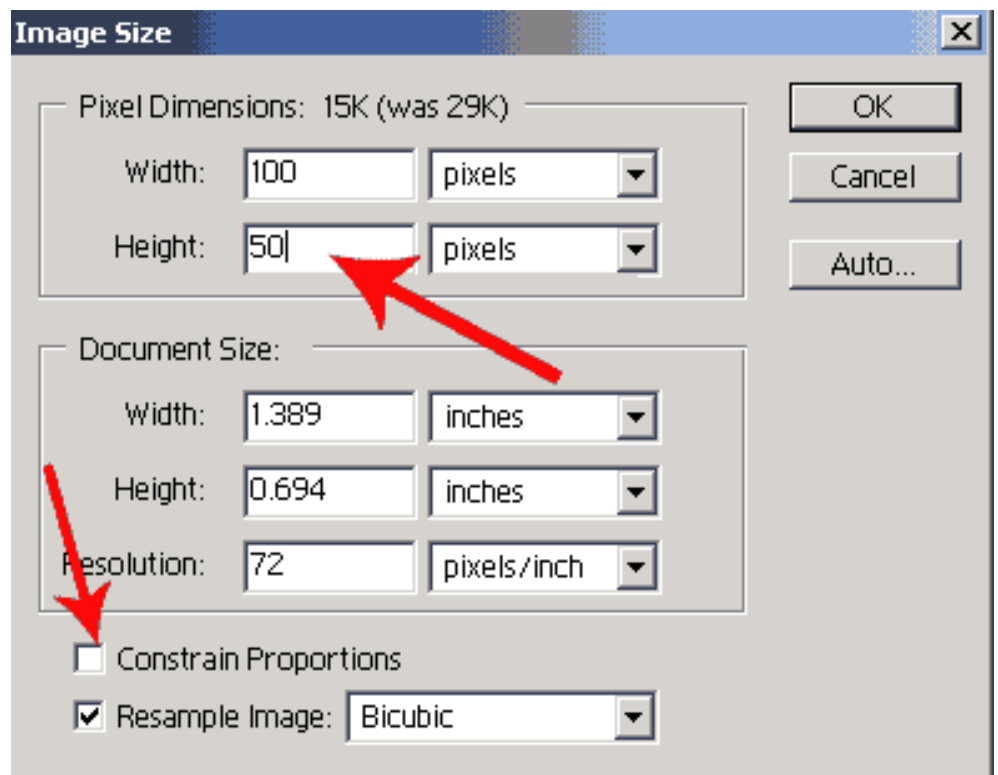
Fill it with the color you want.



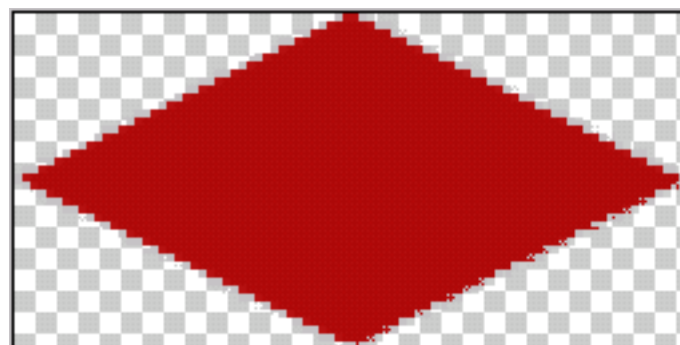
Now, rotate your square 90 degrees.



Go to "Image"->"Image Size". Uncheck "Constrain Proportions" and make the height 50. So now your file is 100x50.



Make sure you zoom in and crop the corners, that way the tiles will fit well and look tight.



And that's it.

Development

What the player needs. Game Design Thoughts.

By Morphosis

These are my thoughts of what a player may look for in a game.

When designing a game, spend a lot of time thinking about what game players are looking for in a game. Think about what can you do that has not been done before and will excite players? Some people may think of other games that are great in the past and also in the present and design their game around that with hopes the player will like it as much as its big brother. If they liked it last year, they will like it this year. But, a lot of players do not want to play a game that is only a clone of another game, a “new” game that only offers old ideas and brings nothing original to the table. As game designers, look at the games that people like and find out what it was about that game that made them want to play it over and over again. It may be the intense epic story, the memorable game characters, or even the cool music.

So, when you begin to make your game, yes you should look at what you would like you play and what seems cool to you, but ask yourself why do players play games in the first place? What makes the game player turn on the computer and attempt to play the game instead of going to a movie? What is different about computer games compared to going to a movie? What do games have that the movie doesn't? The main thing is “Interactively”. Rather than a movie where the one watching the movie can interact with their mind, putting themselves in the movie acting like one of its characters, they still can't interact and they do not have that many choices that will bring out an outcome. Imagine creating a game like “The Matrix”. With the movie, watching it I was feeling like I was in the movie running to the nearest phone booth, as I sat there and watched it. But now imagine, in a game, if I was feeling like I could battle more before running to the phone booth. I then have this choice, control, and challenge.

As for challenge, players do want a challenge. A challenge makes all people either happy, mad, more aggressive, goal oriented (looking for success), and tested. With a challenge, if it's completed in a game the player also has a right to brag, become confident, and get a little ego over the ones who may not succeed in the challenge. This challenge may make the player emotional or develop an emotional connection to the game.

An emotional connection to the game is a sense of belonging to and in the game, a part of the game from the beginning to the end. As with a movie, watching one the viewer may feel as if they are experiencing the exact situations a character is feeling, but with no interaction or control. I am not saying that a Tetris game is a bad game because the player may not develop an emotional connection, but I do find it interesting if the Tetris game had some theme or goal that brings the player closer to the game, for example, let's say in this “New Tetris Game” you are a community leader and need to construct living quarters for people to live in. And in the game, the blocks are the building blocks for homes, and the score is the number of homes developed. Even a timer that consists of the population would spice things up. But here, one may feel as if they are “Needed” to get these blocks together to save a population. That's what I would do to make a great game different and maybe better.

So what does a player expect when they get ready to begin the game? Well, to get to this point some type of information about the game would help. For instance, when I go out to get a new PC or Playstation game, I look at the cover and back of the case it's in. If this seems interesting then I may get that game. If your games are for play from downloads on the web, when trying to get people to play it talk about it before giving the link. Even a web site with a few screenshots would be great. This first gives the players expectations of the game. Then they may decide to play it and new expectations appear. First they expect some type of title screen with a menu, and if possible some type of introduction to the game. This may be from simple text to an intense cinematic scene. A title screen/introduction is important because it is the first thing they see and the feeling of the game and wanting to play may be increased, for example: I remember the first time I played “Tomb Raider”, the opening scene blew me away and I could not wait to play it.

A menu system is needed and expected rather than jumping into the game with no control at the beginning. This shows the player that they will have some lack of control of the game, which is not good. Even a simple “Press Space To Begin” is better than nothing. But options like: Start, Exit, Help, and Options are four good choices to include. Having these options in the game is also a good idea as well as save and load, if needed.

Laws of the game are very important, what I mean by laws is rather simple, and know to us all. For example, if you game has a hero and two green enemies in it, and when you jump on one your hero dies, but when they jump on the other they don't die, that seems odd to the player unless there is prior explanation why one destroys the enemy and the other doesn't. And as for buttons in a game, especially if it's a "Myst" like point and click game, you must show the player what is clickable and what is not clickable. But I am not saying to completely guide the player throughout the game by telling them what to click, but don't have a "Green Button" that means "OK" in the first 3 levels, then have a "Green Button" in the fourth that means something different. Also laws are the games limitations, places they can and can't move to, things they can and can't click on, and so on, should be summed up in the first level or area of the game. Even these laws can be explained more if in your menu you had some type of "Training" which would consist of a small area with elements in the game that shows the player what they can and cannot do in the game. Keeping it consistent throughout the game seems to make the game play much better and making the players experience better rather than making them say "What do I press, what can I press, where can I jump".

Players do need direction in the game but it can not make the challenge very, very simple. The player needs information throughout the game that will keep them inform of their level they are at, what goals they have to get to, reasons to solutions, solutions to reasons, and feedback from accomplished goals. After the player plays the game for a little, the player doesn't want to sit there and pause and wonder "What do I do". This question should be answered from the beginning of the game. It may be telling the player to "Click on this and get points", "Get this item to do this", or "Destroy these to survive". With this direction, the player does expect to fail, and this creates goals and a challenge. So don't make it too easy for the player to complete the game with simple puzzles or easy to defeat enemies.

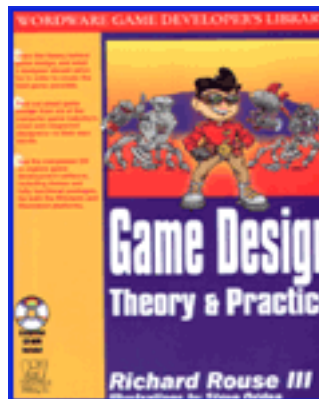
But be fair to the player, mastering a technique at the beginning of a game is hard. The ability to master a technique towards the end of the game is expected. For example, Tetris. What if you never played it before, and as soon as you began the game played as if it was at level 15. That is pretty hard, and may upset the player and lead them into exiting your game. It makes more sense to begin slow and the speed up in increments. The player may then master the technique and have a fair game with goals, a challenge, rewards, and failure.

In conclusion to my thoughts, what a player looks for varies from player to player and the list may be short, as in a solitaire game or long like in a RPG game. These are just my thoughts and really the game must be "FUN" that is important, but what is fun? Again this varies from player to player. I think I'll never know what a player needs.

Reading up: Books For Game Design



Game Design: The art and Business of Creating Games.
Price: US \$29.99



Game Design: Theory and Practice
Price: US \$49.95



Game Design: Secrets of the Sages
Price: US \$24.95

Here are some books that I have come into reading and finding really great information. They are pretty expensive, as most books are, but they should be looked at at a book store or at a library.

Interview

Mark Overmars

By Morphosis

Mark overmars, the creator of Game Maker.

GMDM:

When was the first time you knew you wanted to use a computer?

Mark:

I first encountered a computer when I was in high school, I think in 1975. The father of a friend of mine worked at the computer center of a big bank. I was allowed to see the computer and I was rather impressed. In 1976 I went to university. I planned to study mathematics. I took computer science as minor subject (you could not study computer science in the Netherlands in those days).

GMDM:

When did you decide to get into programming?

Mark:

As indicated above, in 1976 when I started studying. B.t.w., I am not really into programming. As part of my work I hardly ever program anything. I teach courses, write scientific papers, manage a group of researches, etc.



GMDM:

What was the first program you wrote?

Mark:

Probably one of the exercises for the programming class. In that first half year I also wrote a compiler and a little bit later a program to generate and print music scores.

GMDM:

What inspired you to create Game Maker?

Mark:

Already for a number of years I was writing software for children. It started with **Drawing for Children**. Then I wanted to make something to teach children programming and I wrote **Drape** as a kind of visual version of Logo, already containing the mechanism for drag-and-drop to create programs. I took it one step further and tried to create an easy to use animation system, called **Animo** in the summer of 1999. But soon I shifted focus and focused on game creation. The first version was still mainly directed towards children but soon I added more advanced features and focused on adult users. The next version will have even more advanced features.

GMDM:

Did you ever imagine Game Maker would be as popular as it is today?

Mark:

No. I realized it could be popular but the current popularity is amazing. 2500 downloads per day. Millions of page views on the forum. And it is all still increasing. I especially like the fact that people are using the program in schools for teaching programming.

GMDM:

Do you play games and if so what kind do you like?

Mark:

I play very few games. I of course test some of the game created with Game Maker. For the rest I mainly watch my kids play games. I do like strategy and RPG games.

GMDM:

What do you think are some important elements to have a “Good game”?

Mark:

This is not an easy question to answer. I wrote a complete tutorial about good games. Better read that.

GMDM:

And last, I think you like to fish and/or scuba dive. Any other hobbies out there that you like?

Mark:

No fishing but diving is something I like to do. Both in the cold water in the Netherlands and in the Mediterranean and the Caribbean. I also play a bit of pool. For the rest I spend my spare free time with my family.

History of Game Maker

1999

I started working on *Game Maker* (originally called Animo because my first idea was to make a simple system for creating 2-d animations) in the summer of 1999. The first public release was version 1.1. It was release on November 15, 1999. This version did already have a built-in programming language but lacking sophistication. It did not use DirectX and it did not have a separate runner nor could it create executables. So games were simply played in the main window of the program. Since then I quickly create public version 1.2, 1.2a, 1.3. It was not very much know or downloaded. For example in Dec 1999 *Game Maker* was downloaded a total of 366 times.

2000

Version 1.4 was sort of a major release. It got some more attention. Downloads went up: (1000 in Feb 2000, 1500 in Mar 2000, 2000 in Jun 2000, 8000 in Aug 2000 (after some special attention)). Version 2.0 was releases around September 2000. In the whole year 2000 the program was downloaded from my website a total of about 40.000 times.

2001

Then I release version 3.0 which was the first to use DirectX for drawing. Additional version 3.1, 3.2, and 3.3 (april 2001) appeared. The popularity quickly increased. In Mar 2001 the program was downloaded about 14.000 times and in Jun 2001 it had increased to 21.000. In Jan 2001 I decided that I had to redo *Game Maker* I rewrote it completely from scratch, using all that I learned over the years. The interface was completely changed. The new version was also not compatible with the old version. The new red color of the icon symbolised that. Version 4.0 was released in July 2001. It was followed by 4.1 in Dec 2001 (which in particular added multiplayer support). Popularity increased further. In Dec 2001 there where 32.000 downloads. In the whole of 2001 the program was downloaded about 270.000 times.

2002

Version 4.2 appeared in Apr 2002. The main feature was that stability had increased a lot. Version 4.3 appeared in Nov 2002. Popularity is larger than ever. Currently the program is downloaded about 2.000 times each DAY. The number of page views of the main page of the website grew from 2000 in Oct 2000 to 70.000 in Oct 2002.

2003

How will it continue? We will see. I keep on working on the program and hopefully it will get even more popular.

Reviews

CRAFTS

by Mikko Kulmala "Eaglesoft"



"Another Game Maker Space Shooter" that's what I thought when I went to play this game. But as soon as the title screen showed up with the electro-music pumping in the background, I thought this might be cool. I played it and now will speak of my killer crafts.

It's a space deathmatch game that is two player. With "future" internet play, which is not yet functioning. You can select a variety of ships which all have different attributes. This is a nice feature of the game, as well as the options menu, which gives you some control over the game. You can affect the amount of powerups, whether to enable or disable asteroids / ufo's, music / sound. Also before you play, you can choose a time limit to play, this was a good idea as well.

The gameplay of the game consist of **Durability:** Maximum durability of a player's ship is 100. half). **Kills:** When you manage to kill your opponent, you get a +1 to your total kills. At the end of the round, the player with the most kills will win. **Time:** The time is entered before gameplay starts, when you select the Hotseat mode.

There are a number of cool power-ups throughout the game. Some are for new weapons and health items. The game is fast and the music powered by FMOD technology and sound great.

The graphics are done well and the animation is also good. This game does look and play like one of the better space games made with game maker. Although the fun factor is not there for me, I lost interest in playing the game, but don't get me wrong it's still worth a shot.



I give this game a 6/10. It's done really well, good graphics, options, music, and game play. But it's lacking in concept, environments, and fun. My wish list for this game would be different levels(backgrounds), split screen option, computer player, and more emphasis on the ships and it's "personality" such as sounds and behaviors. This game, once again, is really good and I do recommend you to download it. Good job Mikko!

Data.....

Crafts written by Mikko Kulmala
5.2 Mb
Stand-Alone
Program Tool: Game Maker
http://www.gamemaker.nl/games_12.html



5/10

Downtown In Crazy Town By: gzsoft



And just when I was about to head downtown to shop, people become demons and crazy bikers are out there!

Downtown in Carzy Town is an interesting fun and violent game, violence seems to sell these days with GTA and Driver, so that's not a bad thing. Here in Crazy Town you drive an ok car in a downtown area getting commands from someone telling you what to do next in order to move on to the next task. Missions like destroy the crates on the city streets and get rid of the people on the streets before the turn into monsters. Wacky crazy missions that is rather fun.



Some nice features of this game is the ability you have to change controls and music. Even though changing music seems to be buggy and unclear. The graphics are just enough for the game, it makes it comical and funny which makes it fun. The in game options are not there, I wish I could change my controls in game if needed, see some help, and hear music. Also a head up display would be nice, with health, may be a score, or even a radar. Overall it's not that bad. Give it a look!

Data.....

Downtown in Crazy Town by gzsoft
Stand-Alone
Program Tool: Game Maker
<http://members.shaw.ca/gzsoft>



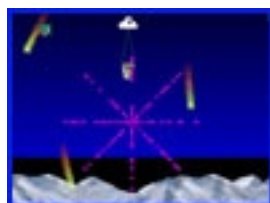
4.5/10

Stardust: By Janthya



An nice game in which you have to collect the stars that fall and avoid your enemies. You play as a cloud like character that holds onto a star catching bucket. The movement of the bucket is wonderful.

The concept is simple yet fun. I became addicted to this game for about 15 minutes. The graphics are just enough to be ok, but it would be interesting to have levels with different environments. There are two ingame meters, one tells you how full the bucket is and the others tells you when a new enemy will appear.



As for the the audio, the song fits the game. It's soft, smooth, and a tad relaxing.

I would say check out this game, but this is a game I would love to be expanded on. Like I said, different levels with different environments. Also maybe the enemies could be something that deals with the skys rather than a green monster. Like asteroid monsters. It's a cool game. Good job!

Data.....

Stardust by Janthya
Stand-Alone
Program Tool: Game Maker
<http://www.gamemaker.nl/games.html>



5.5/10

Fort Defender: By Tanka Flammable Werks



This game is what it's called, Fort Defender. It's a pretty fresh and well done idea and I had fun playing this game. You can play one player or two player and have to defend the parameters of your fort from these balls that try to rip it apart as they bounce back and forth. Your player deflects the balls with hopes it will smash up the others fort.

The game play is cool but the controls of the character may seem odd at first, but it's easy to get use to. The graphics are simple, and do work with the game. As for the music and audio, they are full of energy to make you move and defend your fort!



This fast game will interest you and I do recommend you download it. Good job twanka.

Data.....

Fort Defender by Twanka
Stand-Alone
Program Tool: Game Maker
[Click Here for the link.](#)



An Effeser Project



7/10



That little running rut we love is back in action in "Sonic Zone" I think is one of the best platform games developed with Game Maker that I have played. If you enjoyed the other sonic games you will like this one.

Good graphics and character animation. Not sure if they are original, but still good use of them. The scrolling backgrounds add great depth to the game. The snow falling gives a nice and atmospheric effect. The game play and control of the character are also great. Smooth and nice precise collision detection with objects. The sounds are pretty good also, actually just right in my book.



This is a Sonic remake, so it's not that original, but that's not a bad thing. This is one of the small things that held me back from upping the score, and other things like the font size for the score and level sees to be too big and they seem like they need to be a little more exciting, they need some colors and graphics added. And just when I was almost at the end of a level and made a mistake, I started over at the beginning of the level. It may be nice to have "Checkpoints" in the level as places to continue at. Really though, this is a good game! GOOD JOB!

Data.....

Sonic Zone by Effeser
Stand-Alone
Program Tool: Game Maker
<http://www.cs.uu.nl/people/markov/gmaker/games.htm>



First Look

Game Maker 5

By Mark Overmars

Built-in GML programming language:

The interpreter for the GML language has been completely rewritten. This resulted in a speed improvement for executing code of a factor of about 5. (Realize though that often this is not the limiting factor in your games so actual speed increase might be a lot less.) Also there are new language constructs such as a case statement, break and continue statements, the use of local variables, and multiline comments.

Action mechanism:

The mechanism for actions has been changed completely. Actions are now read from library files. This makes it easy to add actions to the program. A library builder will be provided some time in the future such that people can create their own collections of actions. A number of new actions were already added, in particular actions dealing with health and lives.

Better file size:

Game files and stand alone games are now smaller and load faster.

Time line resource:

A new time line resource has been added. This resource makes it possible to let certain actions happen at predefined moments in time. You can use this to define complicated movements or other behavior of entities, to control the way enemies appear, etc.

Data file resource:

The new data file resource makes it possible to include any type of file in your game. It can also be used to use your own fonts.

Advanced and simple mode:

You can select the mode in which to use the program. In simple mode many resource types and options are made invisible, making it easier to start using the program. In advanced mode it is easier to access the advanced options.

Improved visuals:

Almost all images and form layouts have been improved.

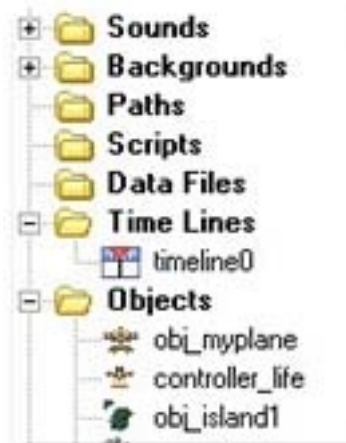
Many smaller improvements:

Such as a better code editor, the use of sprite based fonts, better debug options, mouse enter and leave events, etc.

Release plan:

End of march: public beta

End of april: official release



Tools



Review by: Morphosis

What is Elefont: (It's just cool !)

Elefont is a free tool to create 3D text objects from TTF fonts (ELEvated-FONTSs). They are generated into DXF-files which you can import for example into Anim8or 3d software. It is equipped with a funny non standard GUI. Developed it to create nice landscape pictures together with texts. You can also use it to create free scaleable texts from TTF fonts in a CAD system.

Some use it together with Lightwave. ("This does a much better job than the text plugin included with Lightwave", Dave Edens)



This program is so simple to generate 3d text. It is at first a little because the user interface, but it's rather easy to get use to so don't let that scare you. Also it's only around 350k in file size!

There are a lot of options to manipulate your text before export, nice different bevels and even solo letter manipulation. Check it out!

<http://www.armanisoft.ch/webdesign/Elefont.htm>

Slowview

Review by: Morphosis



Awesome media tool that lets you view images and sound files. It's free and worth more than that! You can open and save over 50 file formats including the ones you might have never heard of, like .dds image files!

Also a few cool image effects can be applied to your images with ease! There are many more options and think you should check into this.

<http://www.slowview.at/>

Development Misc.

by Chris Spicer (Alfly)

Teh Noob

Hello there, and welcome to my development diary! I am Teh Noob, and as you'll soon see, I'm destined to be the World's Greatest Programmer!

I have decided to become a Great Programmer using Game Maker, a program I found on the internet. It sounds great! Apparently you can use it to make games without doing any coding whatsoever! I am already a L33t H@x0r, so I should be able to figure it out really easily!

I haven't had Game Maker very long, but I've already started working on my first project! At the beginning I couldn't decide whether to make an RPG, a shoot-em-up or a platform game, so I decided to be sensible - I'm going to make a game that's all three at once! It'll be an epic RPG with platform-based puzzle levels and massive shoot-em-up sub games! I'm also going to have an incredible real-time combat engine, and in total the game will have over 100 hours of gameplay!! All of this will be done in the revolutionary 3D engine I'm making specially for the game - and all without any coding whatsoever!!! I'm going to make all my own music and graphics too. Apparently some people have unrealistic expectations of what they can do with Game Maker, but I'm determined not to make the same mistake.

I've already almost completed the first and most important stage of the game-creation process (I read that in a computer magazine - see, I'm just like Shigeru Miyamoto) - the name. I can't make up my mind between Blood Dagger: Son Of The Deathlord, or Dagger Blood: Lord of the Death Son. Or maybe Blood Lord: Son Of The Death Dagger. Anyway, it's definitely going to have Death and Blood in it somewhere. Or possibly Night.

Anyway, I've already made a detailed plan for the production schedule of Night Blade: Vengeance Of The Sword Demon (the title's a work-in-progress at the moment). I've worked out how long everything will take so that I don't fall behind schedule:

Name game: ½ Day

Learn all of Game Maker: 1 Day

Develop cutting-edge 3D engine: 2 days

Write intricate, far-reaching plot building cohesive game world, detailing characters, creating back history, developing sub-plots and weaving a rich tapestry to act as a backdrop to the game while still leaving enough open-ended scope for a possible sequel: 1 Day

Scripting and Level Design: 1 Day

Creating all graphics and sound from scratch: 1 Day

Testing: 1 Day

So, as you can see, even allowing for slippage (another phrase I read in a computer magazine - see, I'm just like Peter Molyneux!), the entire game should be ready in less than two weeks!! Am I great or what?!?!

I played some other games that people have made with Game Maker, and they're rubbish! Noone seems to have made a proper 3D game yet, which can mean only one thing: they're all too lazy to figure out how to do it. When I've finished Dark King: Lord Of The Devil Warriors (I still haven't picked a title) I'm going to make a first-person shooter, although I haven't decided whether it'll be like Unreal Tournament 2003 or just a regular shoot-em-up like Half-Life, and then I'm going to make a real-time strategy. After that I might make a Tomb Raider game, or maybe a Gran Turismo game. And all without coding! I can't believe that noone else who uses Game Maker has done it yet! They must all be useless!!

Actually, this diary isn't the first place I announced my game. I also announced it in the official Game Maker Community, and let me tell you, it didn't go down too well!! Here's a copy of what I told everyone:

--

Subject: BEST GM GAEM EVER!!!!1!

.....continue on next page.

Development Misc.

Okay everyone, stop coding now! because a game is coming that's so good that you'll never be able to beat it!!!1!

I'm making a RPG with built-in shootemup and platform levels! Here are a few features!!!

- * Best grap[hics EVER!!!1!
 - * Really cool music WOW WICKED!!
 - * Over 200 hours of gameplay!!!!
 - * Platform sections where you fight against zombies and robots... and robot zombies!!
 - * Shoot-em-up sections where you pilot your ship The Death-Mobile against Hordes of enemies!
 - * Hidden bonus games!!!
- & LOTS MORE!

I have just started this game, so it won't be finished for at least a week! BUT that'll give you time to prepare yourself for the BEST GAMEPLAY EVER! Tie rocks to your ankles, because this is going to BLOW YOU AWAY!!!1!

--

Cool message, eh? Anyway, someone started complaining about me posting in the wrong forum - I wrote it in the Questions and Answers section, because I want people to answer the question "Is this game cool or what?!?" (the obvious answer is "Yeah, it's the coolest game EVERRR!"), but then they complained that that's the wrong type of question and anyway I hadn't actually asked it! Then loads more started saying that I shouldn't say my game's the best ever, since I've never made a game before! They're all just jealous of my skills - HATERZ!!!

Anyway, I started defending myself, and some guy called DT came along and threatened to ban me! All I'd done was defend myself! It's not my fault that people can't take a few little tiny insults about their mothers! In the end he did something I couldn't believe - he closed my topic! I know!! Unbelievable!!! Well DT, if you're reading this, you just made my Revenge List. When I'm a rich and famous programmer I'll get you for that! You'll pay, all right. YOU'LL PAY!!!!

Anyway, I gotta finish choosing a name for my game, otherwise I'll never get it finished! Just think - by the time the next issue comes out, I'll have finished my first game! In fact, by then I'll probably have done about twenty!!!

Seeya next issue!

Noob Out.

Teh Noob's development diary was transcribed by Chris Spicer.

Story Driven F-P-S

by Reiji

What is an FPS?

In the videogaming world, an FPS is a First Person Shooter ie. A game where you view the world through the character's eyes, gun in hand.

FPS: Different Styles

The First Person Shooter (FPS) takes many forms. I'll list the basic types here (in no particular order).

1. Deathmatch (online): A game in which your objective is to 'frag' the other players. 'Callsigns' (nicknames) are commonplace and a chat facility is usually integrated to allow communication. However, voice communication is being pioneered on the game consoles eg. SOCOM: Navy SEALs has a headset and Xbox Live comes with a voice changer-equipped headset. Within a few years, the Broadband revolution will begin and voice communication online will take off.

2. Deathmatch (offline botmatch): A 'botmatch' is like an online Deathmatch, except the human players are replaced with AI opponents, whose qualities can often be altered at the game setup screen. Bots are usually programmed with variables and if commands- a lot of this is done in Delphi.

3. Squad-based FPS (online): A variation on the online Deathmatch, the online squad-based shooter can either be included as a mode in the game, or as a downloadable 'mod' (modification). Built-in modes include Star Trek Voyager: Elite Force's Team Battle or Delta Force's Team Deathmatch. Mods include Half-Life's Team Fortress and Counter-Strike.

4. Squad-based FPS (offline): Any of the Tom Clancy games fall into this category, as well as SOCOM with its intuitive headset commands. Basically, an offline squad-based FPS is a game in which you are part of a team who will work independently and follow your orders.

5. Objective-driven FPS: An example of the objective-driven FPS in the Nintendo 64 game GoldenEye. The player is required to find switches and complete tasks, shooting down anyone who gets in their way. Sometimes, the shooting comes second for switch-orientated tasks, but in protection tasks, the shooting once again takes precedence.

Story-driven FPS: Ah, now we're getting somewhere! Half-Life is a prime example of this, an adventure game viewed from the first person. However, whereas Half-Life's story advanced through in-game action, story-driven FPS games with animated cutscenes can be used to follow a more intricate storyline. Another plus is, FPSs driven by cutscenes are easy to make in Game Maker, usually not requiring any coding in GML.

Step One: Story Outline

First things first- you need an outline for the story. Use the Five W's:

1. Who: Who are your characters?
2. Where: Where are your characters during the game?
3. What: What is the goal of the game- what are your characters aiming to achieve?
4. When: When does all this take place- night, day, past, present, future...?
5. Why: Why are your characters doing this? This is where you make people care about your characters- use negative emotions such as anger (vengeance) or sadness (right a wrong).

Step Two: In-Depth Script

Right, the next step is to write the actual story.

First, you need to write the introduction montage scene- this is where the story begins. Use dramatic and harsh imagery- echoed voices and poignant music. For example, in Sakatoshi vs Ikira, the imaginary game I've thought up for the purposes of this tutorial, the intro could be of Sakatoshi's family being murdered- maybe explain Ikira's motives.

Next, write the opening movie. Use mysterious imagery, constantly fool the player. A good source of inspiration here is Metal Gear Solid 2, where Snake is concealed first under his coat, then the Stealth camo. In fact, it almost seems like it was an accident revealing his Sneaking Suit when he hit the wall. Eventually, reveal the character's identity.

Let's go back to Sakatoshi Ikira, the imaginary game used for tutorial purposes:

Step Three: The Levels

We'll use a city example here because it covers all the basic techniques.

Background layers: Have a looping road animation- eg the lines on the road move towards camera once, then loop forevermore. The far background, the sky, doesn't move- so this is usually the most detailed part because the artist hasn't had to animate it. As for buildings, use the looping trick you used when you did the roads.

Foreground layers: The weapon should be fixed in the bottom-right of the screen. The object you're chasing should have a left-right movement (use paths) and a mouse-click event that goes to the next room, an interlude movie.

Step Four: Interlude Movies

Okay, now the player is involved, keep the tension up in the interlude scenes between levels. Over the course of the game, the character could slowly become borderline psychotic as he relishes the action, or he could become a nervous wreck: "Why did I do that?" etc.

It usually helps to have a team of friends in contact with the character who can give him advice. It is a useful way of getting the story across without having to animate complicated scenes. Also, they can make the game more emotionally gripping by having in-depth conversations with the character.

Step Five: That's A Wrap!

In my opinion, the things a story-driven FPS needs to be successful are: 15% marketing and hype, 20% action, 25% emotion and 40% Hollywood action.

Here are my tips for creating one of these games:

1. Write the story late at night, listening to music. At night, emotions usually flow more freely and there aren't many outside distractions.

2. Write using a PDA whilst somewhere comfortable: Your writing is likely to be much more focused, whereas it wouldn't be if you were hunched over a keyboard and staring at a headache-inducing monitor.

3. Use the Game Maker Community Forums. Start a website (preferably with a short domain name), release screenshots and demos throughout production. That way, if people think your game looks below-par at an early stage, you can take the advice and improve your game. And later, when your game's finished, impart the advice on other people who request help.

4. Most importantly, have fun producing your game! You never know, if you want, you could send it off to a publisher's and try to sell it!

-end-

Power To The Programs

GAME MAKER vs FLASH Part 1.

by: Morphosis

I have a nice connection with GM and other multimedia programs. Here is how I see GM vs Flash MX.



Yes it has this

Power

GM = Game Maker
MX = Flash MX

Features	Game Maker	Flash MX	Power	Comments
Programming	Y	Y	GM/MX	They both have almost the same power.
Script Editor	Y	Y	GM	Larger window, easy to copy and paste, cleaner.
Import/Export Scripts	Y		GM	Import and Export scripts. Import into library.
Own Language	Y	Y	GM	GML is for GM. Action Scripting is for flash, alike Y
Drag and Drop Behaviors	Y		GM	No programming is needed for some games.
Debugger	Y	Y	GM/MX	Inform you almost the same with errors
Graphics	Y	Y	MX	Easier to handle, animate, and create. Vector
Image Creation	Y	Y	GM	Pixels do bring out more detail with image editor.
Import Graphics	Y	Y	GM	Many more file formats to import.
Export Graphics	Y	Y	MX	More file formats to export to.
File Format Imports	Y	Y	GM	Many more file formats to import.
Vector Image Usage	Y	Y	MX	Create and Animate them.
3d (Real Time)				Can be faked though.
Image Effects	Y	Y	MX	Easier and faster to create. Run smooth
Programmed Drawing	Y	Y	MX	Seems vector shapes in flash run better.
Animation	Y	Y	MX	It's Flash!
Audio	Y	Y	MX	You in more control over sound and compression.
Audio Import	Y	Y	GM	It's the midi.
Audio Export	Y	Y	GM	Easier and faster to create export.
Audio Formats	Y	Y	GM	
Audio Control	Y	Y	GM	Easy play, stop and check for sound.
Audio Effects	Y	Y	MX	No programming is needed.
FPS Control	Y	Y	GM	Each room/level can be different fps.
Levels	Y	Y	MX	GM has "Rooms" Flash has "Scenes" and "Movie Clips"
Organization of objects	Y	Y	GM/MX	
Organization of scripts	Y		GM	
Organization of levels	Y	Y	GM/MX	
Organization of audio	Y	Y	GM/MX	
Create non-game media	Y	Y	GM/MX	Easier and Faster in MX.
Web Publishing		Y	MX	
Multi-player	Y	Y	GM/MX	MX games can be multi-player web games though.
Compatibility with machines	Y	Y	GM/MX	MX creations for PC and Mac.
Compatibility with other software		Y	MX	.swf files work in many other programs: Director, IE....
Compression	Y	Y	MX	***I must test more*** but MX compression is good.
Produce .exe files	Y	Y	GM/MX	

Game Makers Data Mag

Created by Morphosis Enter-Active
John Hempstead

<http://galileo.spaceports.com/~morphea/>

March Issue #1
March 1st 2003