

JUNE 2009
ISSUE 16

gmt

GAME MAKER TECHNOLOGY MAGAZINE



EXCLUSIVE

FIVE PAGE INTERVIEW
thatgamecompany



gmt

JUNE 2009
ISSUE 16

WELCOME READERS!

Our first issue of the year received some fantastic feedback from you, so thank you all for that. This is our second issue of the year and based off some of your feedback we hope that we have further improved the quality.

We have continued with the same landscape layout that was positively received, however we did make a few tweaks to it in order to make it just that little bit better. We are always happy to react to any comments and suggestions that you have, so just email them in.

It has become apparent that we seem to regularly switch

between monthly and bi-monthly releases, so expect an issue every couple of months. Just over two months after issue 15, we have some great content that you should be interested in, such as a 5 page interview with Kelle Santiago and John Edwards from 'thatgamecompany' on page 25. An article on how Cut Scenes can improve your games, a feature on GM IN SCHOOLS and a fascinating feature on Identity and the Player / Avatar Relationship.

That is all from me, post your opinions in the usual ways or even email us at gmtech.magazine@gmail.com.

Enjoy the issue.
gamez93 (GMTech Owner)

CONTENTS

FEATURES

- 3** COMMUNITY NEWS / TEAM GMTECH
- 4** MORAL CHOICES
- 6** ENHANCING GAME PLAY THROUGH CUT SCENES
- 9** GM IN SCHOOLS
- 12** IDENTITY AND THE PLAYER / AVATAR RELATIONSHIP
- 15** SURFACES EXPLAINED
- 17** USING DATA STRUCTURES (PART TWO)

REVIEWS

- 19** A DAY IN THE LIFE OF BOB
- 21** DUO WARFARE
- 22** ASTATINE
- 23** GMTECH HIDDEN GEMS

EXTRAS

- 25** INTERVIEW WITH: THATGAMECOMPANY [EXCLUSIVE]
- 31** GM CLASSICS - HOVER TANK 3D
- 33** WORTH A LOOK - GM HIGHSCORES
- 34** CLOSING

COMMUNITY NEWS

A quick round up of all things Game Maker

Game Maker Blog 500th Blog

In the two years since its launch in April 2007 Game Maker Blog has regularly been updated with news on anything from GM publications to interesting GMC topics. It has established itself at the heart of the Game Maker world. Now the Game Maker news site has made its 500th blog post. Phil Gamble owns and operates the site and hopes to ensure Game Maker Blog remains the number one source of Game Maker news and topical articles over the next few years.

Game Maker 8 Progress

A couple of months ago Mark Overmars blogged asking members to contact him if they were interested in creating free-to-use sprites for distribution with Game Maker 8.

Mark received over 25 applications and has now made a decision on which spriters' work he will use. YoYo also plan to also include a new collection of sound effects with Game Maker 8.0 so if you are good at making your own original sound effects visit the Glog and submit them.

Game Maker Community Game

Not many community games that have been started have lasted for longer than a week.

However it appears that a recently launched project may well be an exception, the GMCG has its own designated section of the forum

and work has been in full swing. Just over a month after it came to life voting has been taking place and a decision has been made on the main plot line for the game.

Programming of the game looks set to come to life in the next few months.

GMTV and GMVision

Following a brief link up with GMKing Danny and the rest of the GMTV team made a decision to publish smaller shows rather than full feature episodes. After releasing five full-length episodes since 2007 the team hope that this switch to smaller features will lead to more frequent releases and keep community interest high – so far, it doesn't appear to be going too well. This week also saw the launch of a rival Game Maker video show. GMVision's first episode included news and interview and several game reviews.

Join GMTech Magazine

We have made quite a few changes to GMTech over the last few months and we are always looking to improve the magazine further and continue to impress our readers. In order to do this we are looking for writers, reviewers and issue checkers to join the team and help work on the issues. You can get full details about getting involved by visiting the GMTech forum.

TEAM GMTECH

Editors

*gamez93
Timoi*

Writers

*MnemonicPunk
Mike Alfus
David McClure
RhysAndrews*

Reviewers

*NAL
Xantheil (Alexander Earl)
Broxter*

Proofreader

Maverik94

Special Thanks

*Philip Gamble
Kelle Santiago
John Edwards
FredFredrickson (Martin Crownover)*

MORAL CHOICES

Good or Bad - Mike Alfus examines the choices developers offer gamers

Do you want to be a knight in shining armor, or a dark lord who rules over the land with fear? Do you want to be a Jedi or a Sith? Do you want to use your newfound powers to help the masses or to rise above them, making them slaves to your will?

Basically, do you want to be good or evil?

Over the last few years, moral choices in gaming have been on the rise. Some games, like Fable, even marketed themselves on the fact that you could be good or evil, and it was up to you to decide how the story played out. Are we approaching a day in gaming when there will no longer be one single, linear game play experience? Will every game have multiple paths and plot threads that players can spin themselves off on,



depending on what choices they make throughout the game? It's certainly possible, and seems to be the way the current trend in gaming is pointing.

Many games that incorporate a moral choice system do so in a way that leads the players down two distinct paths, good or evil. In games like Fable, you are able to choose different ways to complete quests, and also different encounters with NPCs. The choices are fairly clear as to which ones are good and evil, as opposed to being morally ambiguous. Your choices are usually reflected in your character's appearance, which either becomes more virtuous and noble as time goes by, or more sinister and vicious, even sprouting tiny devil horns if you're malicious enough. Many Bioware RPGs like, Mass Effect and Knights of the Old Republic have a similar system handled through dialogue as well, and the player can usually tell what answers will lead you down which path.

Moral choices can be so much more than that though. That's not to say that more linear moral paths are boring or wrong, because sometimes it's fun to be the bad guy, and it's good to know how to get there. However to make gamers face choices that are more than just good or evil, the choices need to be more ambiguous, with the outcomes not so clear cut. That way,

gamers have to really think about the choices they make.

One game that did this incredibly well was Grand Theft Auto IV. GTA has always been known for pushing the limits of gaming and coming up with new ways to immerse gamers in their massive city landscapes, so when they tackled the issue of moral choices, they didn't just go the good/evil route. GTA forces the player to really consider what they're going to do. The decisions aren't just black and white, good or bad. They're multiple shades of gray, with an equally gray outcome, just like life itself.

Isn't that the true goal of a moral choice, to have the gamer really struggle with the decisions? At certain points in the game, you're faced with choices that have no clear cut outcome. It comes to the point where one has to choose between killing two people that they had missions for. While it's true that pretty much every person you deal with in GTA is a criminal, it's still not an easy decision to make. Neither one had wronged you directly, so who are you to judge them? When a game can make me ask those kinds of questions, then it's moved beyond the traditional gaming experience.

And the kicker?

I drove around for awhile in game, finally made my decision, then "took care of it".

Continued: Moral Choices

When all was said and done, and Niko stood over the dead body holding a combat shotgun, a realization struck me. I made the wrong choice, there was no magic karma meter I could balance back the other way with a few good deeds. I had made my choice. The character was dead and there was no going back. I just hoped that I'd make the right decision the next time.

That's the level of moral weight that a game needs to strive for. It's one of the best ways to truly pull gamers into a universe and show different sides of a character. Gamers need to feel like they have a stake in the universe they're in, and that the decisions they make truly matter. When games can breach that fourth wall and directly ask the gamer, "So, what are you going to do?" with an unclear outcome, then gamers will become fully immersed in the gaming universe.

Otherwise, you'll just end up like Dr. Manhattan, asking yourself, "Why would I save a world I no longer have any stake in?"

On top of that, having the game environment change based on the decisions the player makes is another way games make you use your moral compass. In the game Infamous, which is set to be released this summer, you control a hero with newfound super powers, but that's just it, no one said you had to be a hero. Infamous has a Karma system that weaves itself into the game quite well.

Based on choices gamers make in different moments throughout the story and other random acts of kindness or destruction, your karma balance will shift towards good or evil. Like Fable before it, the further along you get down a certain moral path, the appearance of Cole, the main character, changes along with it. Not only that, but depending on how good or evil you are will influence what kind of super powers you get to use. For instance, there's an electricity grenade that Cole can throw, and the upgrades for the good and evil sides are drastically different. On the bright side of things, your grenades will explode sending out a shockwave that disables and stuns enemies around it. The evil version, on the other hand, has a massive initial explosion, followed by a cluster bomb sort of effect as many smaller grenades go shooting out to further carve out a path of destruction, obliterating anything they touch.



Not enough? Well it also turns out that the city itself hinges on Cole for its mood. If you do good deeds throughout the city, which was ravaged by destruction earlier in the game, you can see a positive change in the city as shops begin to open up and debris are cleared off the streets. If you decide that the city is yours for the taking, then there will be rioting, frightened people running through the streets, and general degenerative chaos. If you choose the evil path, then people on the street will also curse you and throw stones or other objects at your, cursing you for their misfortune.

Moral choices in games are a great way to make gamers feel more connected and involved with the experiences they have in game. Whereas before all the major conflicts in the game were handled in pre rendered, pre determined sequences, now they are slowly but surely falling into the player's hands, where they belong. With gamers expectations of interactivity rising, and publishers wanting to structure a certain amount of freedom in their games, moral choices and decisions affecting the game world are becoming more of an industry standard.

So, good or bad?

You decide.

It's your choice, after all.

ENHANCING GAME PLAY THROUGH CUT SCENES

MnemonicPunk

So your awesome game is making great progress, right? You've got the game play nailed, written an epic story, held a contest for the best graphics and sound artists you can find and now you're set to conquer the world, right? For some reason though, you can't help but feel that parts of your game aren't connected very well and that the dialog-based storytelling is just so 1990. You want to add cut scenes to your game to make it the best game ever? Here's how.

Cut scenes are short intermissions between two bits of game play, either pre-rendered or made from in-game assets to convey information. They move the story of your game forward or are just for entertainment. Those are all valid reasons for inserting a cut scene but remember this, they cut into your game flow, hence the name.

In recent games, cut scenes have become very important. You'll most likely remember some games solely or mostly for their great in-game cinematics (which is just another word for cut scene, really) like Metal Gear Solid 1-4 with its epic storytelling, Devil May Cry's Dante and his ubercool assortment of moves and hard-hitting punchlines, or the Final Fantasy franchise with pre-rendered videos that blow you away.

There's basically three types of cut scenes. Let's check them out first.

The pre-rendered cut scene

This type of cut scene allows game makers to use awesome graphics of any production value they want. They are pre-calculated and don't need to take into account the hardware they're running. Final Fantasy makes frequent use of this type of cut scene to showcase some of the sweetest computer generated movies to date. While this type of cut scene is easily the best looking, it has to be produced completely from scratch and takes the most time, effort, and monetary resources. This is the type of cut scene that'll make you sit back and relax your grip around the controller for the moment.



The in-game cut scene

In-game cut scenes use the running game engine to tell the story. They take control of characters, the camera and the audio aspects of your game to tell the story like a movie director. This type of cut scene covers almost any range of intermission, from simple dialog to full-blown in-game movies. Unlike pre-rendered cut scenes, this type can, and should reuse existing game assets. It doesn't break the game flow as much as pre-rendered scenes do. Also, some games even play with the "you'll never know when the game play continues" to achieve suspense in players.

The interactive cut scene

Nowadays, most interactive cut scenes resort to so-called "Quick Time Events" (QTE) to achieve interactivity in cut scenes but there are many more ways of making interesting, interactive cut scenes. Final Fantasy 8, for example, used pre-rendered background material along with a controllable player character to make the escape from a collapsing building truly nerve-racking. Half-Life 2 has cut scenes that don't even feel like such, you'll always stay in the body of your character, listen to others talking and see events unfolding while maintaining total control of what you do.

Continued: Enhancing Game Play Through Cut Scenes

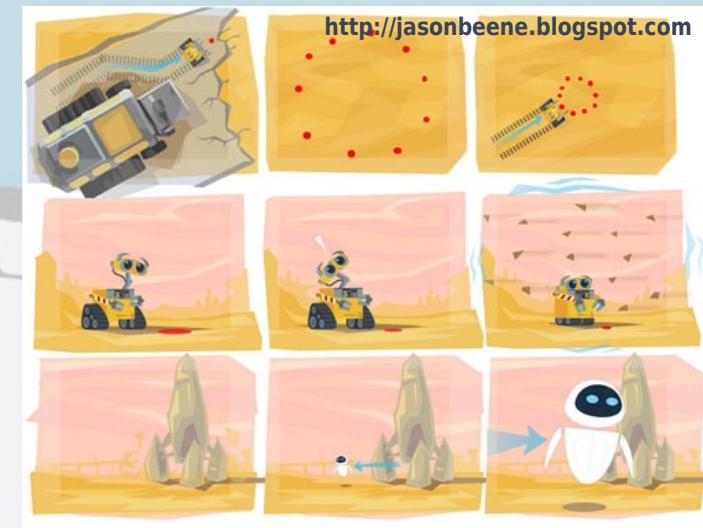
Any cut scene begins with two game states that have to be joined together. Once you have those two endpoints you want to connect, come up with a script for your cut scene. What is going to happen? Who will appear? What will the player, character, or party do? If you have multiple characters, you can also throw in some conflict ("I don't think our army can really invade that enemy country. Let's look for a different plan." - "Yes we can!") or some jokes, or anything that makes the cut scene worth watching. Players want to be entertained while not being able to play, they don't like having no control over the game for no obvious reason. Making a cut scene is really much like making a small movie, so give it some real effort.



Next is the camera. With your storyboard done, you can think about the behavior and movement of the camera. What is it going to focus at? Does it smoothly move from one perspective to another or jump there? Both can be good, it just depends on what you want to tell. In 3D games you'll also have to think about camera angles, close-ups and much more.

The choreography of your cut scene is important as well. Nobody wants motionless characters standing there that just rattle through their script for the sake of getting it over with. Let your characters express emotions while they talk, let them shake their fists in anger and sorrow, let them run around the screen in desperation, let them jump up for joy. The possibilities are endless, just make sure your scene tells the story as much by acting as by the actual dialog used.

While transitions to and from cut scenes are not really that difficult to make, make sure you don't get the basics messed up here. The famous 16:9 "watch out a cinematic begins" bars are common today. They make the screen look like a movie theatre for a reason and any of your players will know that a cut scene has just begun. You don't have to use those, maybe you don't even want to. That's fine, you still should give your player some kind of visual cue that a cut scene has just begun and, more importantly, ended. The latter is often missed even by big



gaming companies. Try to avoid that or your players will stand around and do nothing in expectation of the rest of the cut scene. This causes irritation and ultimately breaks your game flow. That's probably not what you want.

Voice overs are just enough of a topic by themselves to justify an article on their own so we'll just get down with some of the basics here. Use voice overs whenever you can get some decent material for them. Do NOT use voice overs if you have bad material. While you can often use resources of mediocre quality for other parts of your game and the players will gladly overlook them because your game is just so much fun, bad voice overs will cause your players to break down laughing, point at the screen and ridicule your character. This doesn't only break game flow but game ratings. If you don't have some real decent voice overs, don't use any. Also, give the game (at least optional) subtitles for players who can't or want to listen to what characters say.

Continued: Enhancing Game Play Through Cut Scenes

Did you ever have to replay a lengthy boss battle with 30 minutes of cut scenes before it? Yes? So you'll probably know this but I'll still spell it out for those who think their cut scenes are so awesome you could watch them every time. Make cut scenes skip-able!. It's okay for cut scenes with vital story elements to be exempt from this, this but as long as your player might have to repeat it, make sure he/she isn't forced to see the same scene twice. He already knows what's going on.

Review your own cut scenes and let others review them. You might make mistakes in story telling, the script or the actual implementation of your cut scene that you didn't notice first or will not ever notice on your own. This is especially true for logical mistakes in the script. Players who don't yet know your game are far more likely to spot things that simply don't make sense at that point in the story.

If you feel you're up to it, try your hands at some interactive cut scenes. QTE events are easy to do, they just stop or slow down an in game cinematic and prompt players to press the correct button in time. That still too easy for you? Half-Life 2 cranked up the difficulty in cut scene making by using only cut scenes from the ego-perspective and allowing the player to still move during them. This makes cut scene making a real pain because you

have to consider player actions that might alter the course of your script but if done right will really immerse the player.

There's plenty of good material about cut scenes out there, but one of the most important is learning from the pros. Some really good games to get inspiration from are Final Fantasy (almost any kind of cut scene is covered here, from the first instalment in the series to the most recent PS3 iteration), Devil May Cry (probably the best example for cut scenes with awesome fight sequences), Metal Gear Solid 1-4, Half-Life 2, the recent Suikoden Tierkreis for the DS (you won't always have cinema-scale hardware power and this game shows how to cope with that). If you want to keep it basic, turn to old Super Nintendo games like The Legend of Zelda, Chrono. Trigger and of course Final Fantasy. An overlooked but prime example of cut scenes in that time was Super Mario RPG, since it featured rich cut scenes without Mario ever saying a word and acting it out in motions.

Reading an article won't make you cut scene king. Make sure you get up so speed with the references here, do some experimenting on your own, and get feedback from your testers. Practice makes perfect and is actually a lot of fun.

GAME MAKER IN SCHOOLS

Philip Gamble

Over the past few years the game development bug has been spreading across schools around the world. More and more school pupils, college students, and summer school attendees are being given an introduction into game design and programming using Game Maker.

Whilst I was aware that Game Maker has long been taught around the world, I was inspired to write this article following the introduction of a Game Maker option for 11-12 year olds in an "Active Learning Week" at the school I used to attend. Whilst I was in my final year I spoke with a couple of my tutors about the software as they were aware that I ran a blog relating to it. Now I am reliably informed (by my younger brother) that Game Maker is one of the standard software items installed across the network.

Game Maker has long been used in education; it is a central part of the game design course taught at Utrecht University where Mark Overmars is a professor, and was used as the development tool in the 2005 Dutch Make-A-Game competition which attracted over 200 teams of high-school students.

In mid-April, Game Maker Community member Simon Donkers ran a Game Maker



class at the IMC Weekendschool at Tilburg University in the Netherlands. Forty eleven year old children with no previous programming experience created a variety of maze-based games.

Explaining his decision for deciding on Game Maker as the development environment for the class Simon Donkers wrote, "I've chosen Game Maker because it's easy to learn and allows students to create interesting games in a matter of hours. Also, because the environment is free to use and easy to install across all systems, no admin rights required to install Game Maker for instance."

Also, what I find great about Game Maker is that the only limit is your imagination (and

free time). Many other 'easy' programming worlds only allow one specific game type to be created while Game Maker allows a wide variety."

"I noticed in the last class for Weekendschool, as well as in previous years, that many of the 11 year old kids grasp the concept of Object Oriented Programming after only using Game Maker for a few hours. They quickly understood events and the separation with sprites, objects and rooms. In 2007, one girl's PC crashed halfway through the program and she created the game Dog House 1 in one hour. It's a maze game in which you must avoid the fire, catch the fish to lure the cat away, and grab the key to open the door to get to next of 3 levels. I don't know a single other tool in which an 11 year old kid can create a fun, good looking game like this with little guidance and hardly any training in just 1 hour time."

"Giving classes like this is very fulfilling when you see the wonderful games people create. It also shows that these kids don't start a remake of Tetris or Pacman, but that they have original ideas. For instance a game where you need to grab toothbrush and toothpaste and clean teeth from dirt whilst avoiding the evil bacteria in your mouth. But also a more political race game to see whether George Bush or Barrack Obama is

Continued: GM In Schools

the fastest in a race. Or a very challenging puzzle game where you control both yourself and the enemy with your arrow controls. This requires some careful planning to avoid a collision with an enemy. But also somebody who made a game around being a postman and collecting letters as fast as possible."

Blijbol who helped on the course said, "the children were very excited while playing each other's final games at a video projector" before adding that by the end of the course "they were more enthusiastic than they were anywhere in the programming process."

It isn't just pre-teens that are using Game Maker in education though; Simon has also given classes to 16 year olds as part of their computer science class. Describing the differences between this and the younger children Simon said, "It's a bit less creative and more organized because the goal is not to have fun but to get good grades, but I still see how these people learn lots of complex principles like object oriented programming, inheritance, code structuring, state machines, and many other principles."

Simon has also taught many other Game Maker classes for students, teachers, and for members of game design competitions.

When asked if he thought the use of Game Maker within education had been increasing

in recent years he said "within the Netherlands I haven't noticed a recent upsurge in popularity, Game Maker has been popular for quite some time already."



Writing in the Sec Ed Journal in 2003 John Ashford, head of ICT at North Walsham High School in the UK wrote of the many different benefits that come from teaching Game Maker. "Designing computer games requires students to practise many generic skills associated with Windows based software. Examples are file management, navigation of menu systems, and use of the mouse and keyboard. It is also possible to use a broad range of different software applications, for

example, when creating graphics, sound effects, and musical scores as part of the project. Alongside these there are abundant opportunities to promote students' literacy and numeracy as well as other important skills such as teamwork, planning, problem solving, decision making, and evaluation."

Reasons why Game Maker may be chosen above other game development tools include its combination of Drag & Drop actions alongside more advanced Game Maker Language scripts. Ashford explains, "Game Maker is a very flexible system, easy for even a novice programmer to get to grips with. On one level, games can be created by dragging and dropping items from a set of standard toolbars. However, it also includes a powerful built-in programming language, called GML for the more confident who wish to progress further with more complex tasks."

Ashford says that another reason for the popularity of Game Maker over other game design environments is the "extensive selection of resources" available online which makes lesson planning easy. Free online resources such as a Moodle (free open source e-learning package) course created by Darren Smith and tutorials and challenges available at <http://www.mindtools.tased.edu.au> are just one slice of the offering as traditional Game Maker tutorials such as the official ones at

Continued: GM In Schools

YoYoGames.com can also be used in schools.

For more than five years Michael Fenton from the Nexus Research Group has demonstrated how Game Maker can be used to create educational games, simulators, and eLearning resources such as interactive whiteboard applications for teachers in New Zealand. Michael has also developed educational simulators based around the subjects of Math and Science such as a game in which you must manage a nuclear reactor developed for year 12 physics students.

The final word from Simon Donkers, "A great quote a teacher once said to me after a class is that 'the biggest problem is that students spend too much time on their games and forget other classes'. While using Game Maker you learn a great many things, from simple things like English to more advanced things like creating design documents and writing neural networks. Game Maker lets people learn without them recognising it as homework but rather recognising it as a fun challenge. That's the real power of Game Maker and that's why many schools have been using it and will be using it in the future."

Thanks to Simon Donkers and Jeroen van der Gun who provided me with an insight into their work at the IMC Weekendschool.



The games created by children at the 2007 and 2009 IMC Weekendschool can be found at <http://weekendschool.simondonkers.nl/>

IDENTITY AND THE PLAYER / AVATAR RELATIONSHIP

David McClure

Nearly all games involve a relationship of sorts between a player and their avatar, the entity they control within the game world. Good design decisions related to this relationship are critical to expanding the range and popularity of games as an art form. Bad design decisions will annoy the player and diminish their immersion in the game world. Unfortunately, most coverage of this issue focuses on MMORPG players who have an intense bond with their chosen character; as this area has been covered amply it will be ignored in order to cover what is a sadly under discussed facet of the player/avatar relationship.



Nintendo

WWW.GAMEMAKERTECH.INFO

The immediate personas that players take on in games fall into three categories:

1. A predefined character, as exemplified by Nico Bellic in the game Grand Theft Auto IV
2. A player defined character, which the player has created and shaped, as in Fallout 3
3. Themselves in another guise or occupation, as in managerial simulations like Football Manager or strategy games like Spring.

Different games place a different level of emphasis on who the actor the player controls is and on any inherent limits to their actions. Gee theorised that during play there are actually three different identities which are at work at any given time; the virtual identity, that is, the character a player controls and any constraints which are inherent to playing as this character; the player's real world identity, the person playing the game and any mistakes they make in terms of input or strategic choice; and the projective identity, which Gee describes as both the player projecting their "values and desires onto the virtual character" and as "seeing the virtual character as [their] own project in the making". [1] Clearly, most characters do fall under a broad version of this definition, but it is important to note that, except in rare cases, players of most board games and

many simple games do not project themselves onto the pieces or assume any elements of their identity. It could even be argued that in some respects these types of games, which tend to rely on skill, strategy, and timing and tend to eschew plots, character development, and even characters, are actually much purer games, stripped of aesthetic distractions and boiled down to sheer game play. For the purposes of this essay, this is neither here nor there and we will be focusing on the first two categories of immediate personas, those of the character and the player created character. As we will see, although they can easily cause problems for designers, these immediate personas are arguably one of computer gaming's biggest strengths and are one of the factors which give games the potential to become truly fantastic pieces of art and communication.

In some respects, the underlying mechanism of playing a video game can be described as "understanding a system then attempting to apply efforts so as to create a desired result". Take Pac Man. The ghosts kill us so we avoid them. We eat the pellets because we get points. If fruit appears it may be worth risking death for bonus points. When we eat the power pills the ghosts are vulnerable to us. For a limited time, we may eat the ghosts for points or avoid them to concentrate on the pellets, depending on our

Continued: Identity and the Player / Avatar Relationship

goal. We make our decisions and decide on strategies based on the maze, the pellets, the power pills, the ghosts, the bonus fruits which appear, the score, and Pac Man; these elements are what make up the system. We then apply our efforts, by way of using the joystick to control Pac Man in order to try to achieve our goal, be it survival or a high score. This understanding is drawn from what Gee calls The Probe, Hypothesize, Re-probe, Rethink Cycle;

1. The player must probe the virtual world (which involves looking around the current environment, clicking on something, or engaging in a certain action).
2. Based on reflection while probing and afterward, the player must form a hypothesis about what something (a text, object, artifact, event, or action) might mean in a usefully situated way.
3. The player re-probes the world with that hypothesis in mind, seeing what effect he or

she gets.

4. The player treats this effect as feedback from the world and accepts or rethinks his or her original hypothesis.

This occurs at an incredibly rapid pace and is a naturalistic way of learning. It is also aided by transfer of knowledge from other games that a player has experienced. There is a pleasure in learning how a system works and being able to use it for one's advantage. This phenomenon is often at its most notable in the case of emergent game play and speed running, where players consciously analyse the best ways to exploit a system in order to create a desired result. This can easily be seen in message board discussions regarding Resident Evil 5's sub-game, The Mercenaries, the game play of which seems to almost be designed for this kind of analysis and strategising. Most of our daily actions are governed by some kind of understanding of various systems in the world around us, such as preparing food, following social conventions, and obeying laws.

The problem which was alluded to earlier tends to arise when the player's experience of real life systems encounters a version of these systems within the game and finds that these systems have either been simplified or lack a vital component in a manner which can be quite jarring. To give an example, this commonly occurs when a



character needs a small amount of money or when it is possible to bet. Although gambling beyond your means is quite irresponsible in real life it is a simple matter to bet and consistently make small amounts of money, provided you decide your wagers using mathematics and seek to exploit any advantages which the betting industry offers to you. To generalise, the options available in this field to a character in a game are somewhat limited and generally the results of these choices do not chime with reality. This may seem a moot point, in that, with regard to Gee's theories, it is reasonable to expect a player's character to have limitations. However, if the player is the character and thus limited in some activities why should no such limitations be imposed when the player has to decide how to solve a difficult puzzle, what tactics to use, or what equipment to buy? It seems unreasonable to apply such standards arbitrarily. Surely

Continued: Identity and the Player / Avatar Relationship

veteran fighter Nico Bellic should be a much better shot than he seems to be when I am in control of him? Shouldn't a character which I have created to be an expert physician express displeasure or disbelief if I purchase inappropriate tools? Further, limiting players' pool of sensible decisions from which to draw from when they approach an obstacle is bad game design which harks back to the archaic formula of blue keys which open blue doors and such. One solution would be to remove a player's decisions from the equation, but this would reduce games to a series of quick time events. It seems the best route to take would be that of endeavouring to include sensible, realistic systems where possible but without spending too much valuable development time on them. If used properly, this could even add an extra layer of depth and pleasure to games. Take the example of repairing a vehicle in Far Cry 2; Imagine that if instead of simply pressing the appropriate button when the wrench icon was displayed, the player had to evaluate what had caused the problems and attempt to fix them with the resources at hand, perhaps just having to keep the vehicle moving for long enough to reach a mechanic who could restore it more thoroughly.

There is a rather brilliant upside to this problem though. Nearly all cultures have an

idiom which corresponds to the old adage "Before you judge someone, walk a mile in their shoes." Over the centuries, various types of art have attempted to conjure up emotions in people and to express ideas and mental states. With modern video games we have

an art form which could almost have been tailor made for this kind of project. It is unlikely that a player will ever really forget who they are for any extended period of time, but it is fairly plausible that if they are emotionally invested in a storyline or caught up in a section of the game, they might well empathise strongly with the character's plight. The Sapir-Whorf hypothesis states that the language we think in has some effect on how we think. Neuroscience tells us that the brain is plastic and that different specialised areas develop more if they are used more. It is therefore sensible to assume



that although most people will generally see many aspects of the world in the same way, there will be minor differences in the manner in which we comprehend it, which may well be exacerbated by education, social background, and chosen culture. If we are careful and subtle in our decisions we can express such concepts via games. This will both allow for great, communicative pieces of art and for more rounded, differentiated characters.

SURFACES EXPLAINED

MnemonicPunk

So you have heard a lot of good things about those... surfaces and that they can be used for some really awesome stuff, but somehow you don't understand how they work. Read on then, and discover the secret art of using surfaces to make your game even more awesome than it already is.

There are some really good applications for surfaces that can be used to add split screen modes to your game, draw things off-screen, create sprites in RAM for later use, and add some really cool effects that wouldn't be possible otherwise.

But first of all, What is this surface thing I'm talking about? A surface is basically a canvas that you can draw on. Actually, you're using at least one every day, your screen! It has a height, a width, one color per pixel, and you can draw stuff on it in whichever way you want.

To use a surface in Game Maker, you first have to create it. You can't just use something that isn't there yet, right? Let's create a surface and assign it to a variable, for example.

```
mycoolsurface = surface_create(640,  
480);
```

This line of GML will create a surface which is 640x480 pixels in size. Right now it's like an empty sheet of paper. Let's draw something on it, shall we? As mentioned before though, the screen is a surface itself, so whenever you draw something in your game, be it via GML or drag&drop commands, you draw it on the screen surface. We want to draw on our cool new surface though, so let's tell Game Maker that.

```
surface_set_target(mycoolsurface);
```

This tells Game Maker to put aside the "screen" sheet of paper and replace it with our new one. Anything you draw from this point on will be drawn on the new surface.

```
draw_sprite(sprUbercoolSpaceship,  
image_index, x, y);
```

This will draw our totally awesome spaceship sprite on our new surface, just as if you had drawn it to the screen. There's a hook to this though. Only one surface is ever displayed automatically, and that's the screen. More on that in a second, just keep that in mind. Only the screen surface will be visible. So your spaceship isn't, yet.

Alright, we now know how to draw things on surfaces. That's all well and good, but not have much use if we can't display it., so let's

change that. Tell Game Maker to select the screen surface again so we can draw on it. Because the screen is a special surface, it also has a special command for this.

```
surface_reset_target();
```

This line does the same as the above line that selects our surface, except it now takes the screen again. Any sprite you draw from now on will be drawn to the screen once more. If you forget this line, you will still be drawing on your surface later on so make sure it's there. Well, cool, so we drew a sprite to a surface which is not shown at all. Let's display it.

As said before, you can draw anything to surfaces, even other surfaces! Not only that, it's as easy as drawing any other sprite. Just the thing we need, eh? Let's draw our cool new surface to the screen so we can see it.

```
draw_surface(mycoolsurface, x, y);
```

Now we can see the surface on the screen and the spaceship on it. It doesn't look any different from a normal sprite, does it?

Why the hell would you want to draw your sprites on a surface then? Because you can! This opens up some major new possibilities. You can...

Continued: Surfaces Explained

...draw a set of sprites once on a surface and reuse them wherever you need them! This is much faster than redrawing all of the sprites every step.

...apply post-processing effects to your game. This includes all kinds of fullscreen blur, fake HDR, morph effects, crazy color transformations, rotations and other effects. Just draw your sprites on a surface, then use that surface like a sprite.

...create your own custom lighting system! Paint a surface black, then cut out the parts you want to be lit by filling them white. With the correct blend modes this will allow for any kind of lighting you can come up with.

...create your own custom view system! If the built-in view system of Game Maker is too limited for you, just make your own. Draw what every player sees on a surface and then draw those to screen at their respective position. Bingo.

...create transitions! Game Maker expects an initial image and a final image for every transition you want to create. Since those are surfaces too, you can manipulate them like any other surface.

...draw complex menus in a breeze! Windows' Aero effects use surfaces to draw any window before it appears on the screen. This is what makes the cool animations seen in Windows Vista and Windows 7 possible

and now you can use this technique to make your ingame menus look just as cool!

When working with surfaces you will notice that they don't clear automatically. This means that – unlike the screen which is emptied at the beginning of every step – a surface will not lose what you painted on it unless you tell it to. Even if we draw a second spaceship, our first one is still there because we didn't empty the surface first. This will cause ugly smear effects if you move sprites on surfaces without deleting them regularly. A good point to empty a surface is right after choosing it as target, just before you start drawing on it:

```
draw_clear(cl_black);
```

There is one final thing you need to do with the surfaces after you don't need them anymore, though: You need to destroy them. Unused surfaces still occupy RAM on your computer. They'll pile up and make your program hog memory if you don't delete them. Here's how to destroy a surface that you no longer need:

```
surface_free(mycoolsurface);
```

Now that your surface is freed, the RAM it used can be used by other programs again.

There are some rules to keep in mind when

working with surfaces, though:
Never change the drawing target in the draw event of an object. Game Maker can't handle that and will do unspeakable things to you, including killing your kitten and not flushing your toilet after peeing. Really, never do this, it will mess up your graphics.

Do not use surfaces in 3D mode. Same as the one before, Game Maker can't handle this and it will seriously screw your stuff up. Surfaces might be automatically cleared when you switch the resolution so if you do, don't expect the things you drew on them to still be there. Redraw them to be sure. Like all other custom resources, surfaces are not saved along with save games.

Alright, this marks your initiation in dabbling with the forbidden art of surfaces. Always remember, with great power comes great responsibility.

There are many more commands available that deal with surfaces, make sure you read about them in the manual. You'll really love those nasty little buggers once you've gotten used to them.

USING DATA STRUCTURES (PART TWO)

RhysAndrews

Following up on the first part of "Using Data Structures", where I introduced you to some of the basics of data structures, this is a case study just to give you an idea of some of their practical elements.

Case Study

You might understand all of this, but sometimes it can be difficult to place yourself into the mindset of using it logically. You need to play with it a few times and see for yourself how a data structure works and how it can benefit your own projects. Sometimes a case study helps, so I've grabbed a block of code from my Parabox Console, which relies heavily on data structures. This particular block handles errors that occur in the game, if the error hasn't occurred before, it stores the error message in a data list and sets a repeat value as '1', meaning the error has occurred once. If it occurs again, instead of adding another copy of the error message (potentially huge data structures!), it simply adds to the repeat value. This allows the console to display error messages sophisticatedly, showing how many times each error has occurred instead of constantly popping up the error again like game maker's inbuilt handling.

```
//Detect Errors
if error_occurred = true && global.pbx_showerrors
{
    var ref,i2;
    i2=false;
    for(i=0;i<ds_list_size(global.pbx_errorlist);i+=1)
    {
        if error_last=ds_list_find_value(global.pbx_errorlist,i)
        {
            ref=ds_list_find_index(global.pbx_outputlist,"External Error no."+string(i)+" (rep
"+string(ds_list_find_value(global.pbx_erronumlist,i))+"): Use command 'errorinfo
"+string(i)+"' for more information on this error.");
            ds_list_replace(global.pbx_erronumlist,i,ds_list_find_value(global.pbx_erronumlist,i)
+1);
            ds_list_replace(global.pbx_outputlist,ref,"External Error no."+string(i)+" (rep
"+string(ds_list_find_value(global.pbx_erronumlist,i))+"): Use command 'errorinfo
"+string(i)+"' for more information on this error.");
            i2=true;
            break;
        }
    }

    if i2=false
    {
        global.pbx_visible=1;
        ds_list_add(global.pbx_errorlist,error_last);
        ds_list_add(global.pbx_erronumlist,1);
        ds_list_insert(global.pbx_outputlist_col,0,global.pbx_txtcol_error)
        ds_list_insert(global.pbx_outputlist,0,"External Error
no."+string(ds_list_size(global.pbx_errorlist)-1)+" (rep 1): Use command 'errorinfo
"+string(ds_list_size(global.pbx_errorlist)-1)+"' for more information on this error.");
    }
}

error_occurred = false
```



gamemakerblog

NEWS - COMMENT - OPINION - DISCUSSION

500 + POSTS

GAMEMAKERBLOG.COM
TWITTER.COM/GAMEMAKERBLOG

ANOTHER DAY IN THE LIFE OF BOB

Developer : PolystyreneMan

Reviewer : NAL

Type : Complete

[Download](#)

A while ago, Polystyrene Man released his first game, a fairly small life simulator by the name of "A Day In The Life Of Bob". Although it was very good for a first game, it was hampered by the creator's own limitations. Now, the sequel is out, which Polystyrene Man claims will be his LAST game. This one, which has had a lot of love poured into its creation (and it shows), is a huge improvement over the first. It's bigger, better, and importantly more fun.

It's is ot necessarily a bad thing, when you load the first game you're standing (well, floating) in the middle of a big city, and you get a sense of "what the hell do I do first?" The best move is probably to explore the area - no need for a job yet. You'll initially find it difficult to remember which building is which, as apart from the hospital none have any

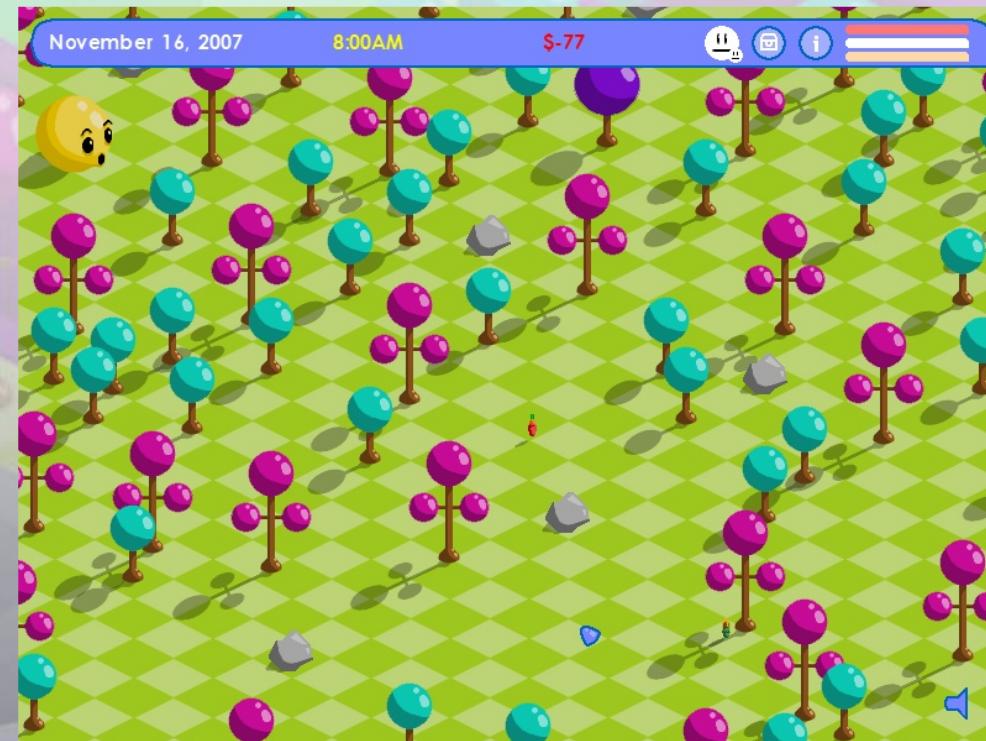
indication, but after getting into the game a bit, you'll start to remember the locations of the essential buildings (where you work, where you buy food, etc) and eventually also learn where other places are.

On the subject of places within the game, there's enough variety to please anyone. As well as your apartment, there are corner shops, gyms, restaurants, concerts (you can't watch them, but get your musical skill high enough and you can play in them), herb shops, a prison, a crack house, and even a place that'll kill you for the reasonable

price of \$1000.

So what is there to do within the world? I can't really describe it all without GMTech getting angry at my word count, so here's a small selection. You can work in a corner shop for some cash and spend it on cactus seeds, plant them in your garden, then when grown, sell them to some bloke off the street at an elevated price. You can wander the forests finding such characters as a giant carrot and Pacman. You can hunt for the many hidden eggs which will complete an objective. You can help a slug in need by taking him from the piles of salt that surround him. If you're bored of all that, you can buy and take drugs, which have a "slight" impact on how everything looks.

The huge amount of variety and potential exploring makes the game fun to play for a fairly long time. Although to be successful you have to go through the tedious process of learning, you'll want to so you CAN be successful. As you progress,



Continued: Another Day In The Life Of Bob

more and more things to do are unlocked, keeping the game fresh for a while. I can imagine quite a few people dismissing the game after five minutes for not being action-y enough though, which I think is stupid. You have

to let the game grow on you.

Graphically, although the game is very good looking, it suffers from all being a little inanimate. There are no vehicles and pedestrians do not walk about -

they just stay in the same spot. There's also no walking animation, or even direction changing. All the characters lack a face, arm, and legs. People bob up and down a little - that's where the small amount of animation does exist. There are some snazzy effects for when you get drunk or stoned, though, and generally everything within the game is well sprited.

In terms of sound, they are fitting. There are decent sound effects in the game such as door opening and the music loop, which grates on the ears a little after a while, but can thankfully be turned off.

Controls are chosen wisely. For a while I was playing with the

directional buttons (which provide the isometric "up = up and right a bit, right = right and down a bit etc" standard), wishing Polystyrene Man had chosen the simpler "up = up" mechanism. Then I found WASD, which DOES give me that.

Everything else about the game is very pleasant. There are some welcome additions such as a solid inventory system, a well laid out HUD, clear instructions in buildings, and in general, a real air of professionalism. The one or two glitches are forgivable.

For his last game outing, Polystyrene Man has taken a solid foundation and turned it into a treat. It's addictive, it's packed with features, and it looks good. It's a winning formula and a highly recommended download.



Gameplay:	8.5/10
Graphics:	8.5/10
Sound:	4/10
Originality:	8/10

DUO WARFARE

Developer : Toadfrogs
 Reviewer : Broxter
 Type : Completed

[Download](#)

If you've ever played one of the classic Worms games, Duo Warfare will feel familiar at once. This strategic turn based platformer gives the player(s) a choice between playing against another human (not online) or against what has to be said is quite reasonable AI.

Duo Warfare is moderately simple to play, even though the game play takes a number of variables into account, such as wind speed. As far as content goes, there isn't much, but fortunately the replay value is increased by the fact that every play is unique due to the nature of the game.

The game's most innovative element is the ability to create custom maps using an external 'Map Maker'. This allows relatively easy creation of environments for the game to take place in, and increases the replay-ability by a great deal. It gives the player the opportunity

to try out different kinds of environments and see the effects of them in battle. One snag I have with this though is that there are no preset maps, meaning the player must create one to play the game. As a result, anybody who lacks creativity or any decent artistic talents will struggle to make anything that will make a good game. For this reason, I hope Toadfrogs decides to release a map pack of some kind; it would

be a good idea that would decrease the chance of players being scared off by the thought of creating their own content. As for game play, there is a good range of weapons which means tactics will come into play throughout battle. Also, as I've previously mentioned, the decent AI reacts rather well to events in the game, nevertheless there is unquestionably room for improvement.

The visual aspect of Duo Warfare could do with a little renovation.

Although the interface is easy to navigate, it doesn't look that great and isn't very attractive. Generally, the look of the game is fairly boring, this is emphasised by the dull choice of fonts and unimaginative methods of presenting information in-



game. However, on the other hand, there are several good effects, such as destructible terrain, but these still don't improve the look very much. Overall, the look could really do with a polishing, as everything does somewhat rash.

Duo Warfare's audio is fairly limited. Although the game has a flowing atmospheric background soundtrack, I feel like the game would benefit from a track with a more dramatic tempo to give the player the impression they're in a highly intense battle. Also, the sound effects are much too infrequent, even if the ones which are used are of good relevance and quality.

Gameplay:	7/10
Graphics:	5.5/10
Sound:	5/10
Originality:	6/10

ASTATINE

Developer : **Cesque**

Reviewer : **Xantheil**

Type : **Completed**

[Download](#)

Astatine by Cesque is a game made up of simple yet nice graphics, featuring some good sound effects, plenty of music, a thrilling storyline and of course awesome game play. You start as an amnesiac who just remembers his name (I believe), guided by a spirit, and as you proceed you start to remember (the spirit doesn't tell you anything) your quest to kill the (of course) Evil one. Thus you move from place to place, walking, running and even driving a train to reach East, the place where the bad guy had retreated to.

The graphics are retro-styled and look nice, the HUD, the text boxes, the player graphics, the enemies, and the various animations reflect the effort put it. All details were paid attention to, as can be seen from the animations of weapons and the flies hanging about the

corpses (that really made me laugh). All the cities/places had distinct styles and color schemes. I have never been a fan of retro-styled graphics and thus, but still I found the graphics appreciable.

The music differed with the different places/cities, which promise abundant musical variety (though I didn't really like all the tracks). The sound effects are plenty too, from the sound of the revolving chainsaw to the death cry of enemies Overall, the musical elements were well-done which made this game even more enjoyable.



The game play was a major high. You could switch between various weapons and pick up ammunition from time to time for the guns, but what I really loved was the chainsaw. When you use a chainsaw, you not only damage the enemy but also prevent him from moving forward, which was very realistic (after all, how many of us can walk straight through a chain saw?). All this made the chainsaw an ultimate weapon since it doesn't require any ammo. However to balance it is the unique feature where chainsaw loses energy every time you use it, which it regained as soon as you stopped using it. Cool, eh ?

The rest of the game was like any other platformer RPG,

collecting items , killing enemies, moving onto new cities, fighting bosses, nearing your destination, and finally winning. Also like most other platformers, you really miss replay-ability. Sure, you can go on and try to do the game again, but what's the fun in that. A sand-box mode would have really enhanced the replay-ability and made this game better. But nonetheless, this game is surely worth trying out. It may not stand out from the rest, but hey, do you care as long as you're getting the fun?

Gameplay:	8/10
Graphics:	7/10
Sound:	8/10
Originality:	5/10

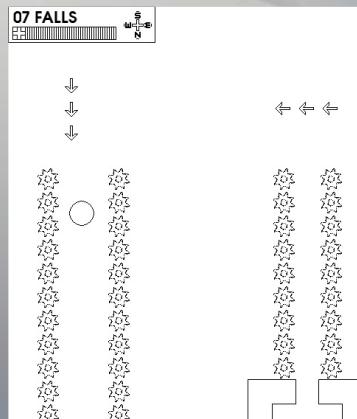
HIDDEN GEMS

Broxter sheds light on some buried treasure...

Many readers in the past have said that GMTech should take a look at some of the games that people may not know much about; games that (let's be fair) haven't received the recognition they deserve. We at GMTech now hope to please all of those who sought after this, bringing you this new feature, 'Hidden Gems'. Every issue we'll present a selection of games that, in our opinion, have not been justly acknowledged.

Innoquous 2 by NAL

Innoquous 2 is a follow-up to the hugely popular Innoquous, although interestingly the game has received nowhere near the interest that the original



acquired in a relatively short period of time. It takes the idea of gravity switching further, with a total of eighteen levels and overall, much more content.

The core feature of the game is gravity switching. This is what makes up the majority of levels and really allows for some interesting level design. This sequel also includes new features such as new obstacles that the first did not to give more varied game play. NAL, the creator, felt that the original did not deserve the credit it got and that this puts the mistakes of the original's right. It should also be noted that there is a secret rabbit hidden somewhere in the game that can earn you the editable version of the game – a cute little idea.

Probably the most interesting and original element of the game is the graphics. The effect that the circular character creates with the platform it is standing on sets it apart from other games in the genre. Overall, the visual look of the game is very basic with only two colours in the

whole game, relating somewhat to the simplicity of the game play.

The game's music is rather basic and allows the game to flow quite nicely. Unlike the original, Innoquous 2 features a few sound effects for actions such as jumping and switching gravity. I'm not sure if I prefer the game with the constant sound effects, as they add an unneeded addition of complexity that can disrupt the flow, but the addition evidently shows that more care and thought has been put into this sequel.

[Download](#)

Social Experiment by AndrewAP

Social Experiment is a co-operative platformer works similarly to many of the games submitted for the YoYo Games Co-operation competition. The player can choose to play alongside AI or with another human to journey through the puzzling adventure that lies ahead. The game sports a bit of a retro style since it was created for the Retro competition at Zerosoft Games.

The player must guide two

humorous characters, Malky and Wilkinson. These funny chaps must work together to rescue Eve, Malky's girlfriend, from the Wicked Magic Man. The levels contain many puzzle elements that will keep you playing for a while, as the special abilities that each character has allows for some good level design. The AI in the game is better than average, but playing as two players is much easier. Quite annoyingly, the game progress cannot be saved, so you'll have to face those tricky levels all in one go.



The graphics remind me of the sensation that is cactus. They do slightly resemble those of Psychosomnium, but I'm still not entirely sure why I think "...cactus?" The graphics are not bad, but they're not great either. It could be argued that the game is meant to be retro though, so we'll let the game off with that.

The audio is fairly interesting and pretty nice to listen to. The level music is not really retro, but this

Continued: Hidden Gems

isn't too much of a problem. The retro styled sound effects break up the background music nicely and they go well with the retro theme.

[Download](#)
**Jumper Tree by
CRYSTALL_m**

This environmental puzzle game was entered into the fourth YoYo Games competition, being one of many that didn't get the notice it truly deserved. Working similar to the popular game, Coredase, the player has to steer a ball around a map, planting trees as it steps over each map tile. The most fundamental rule of the game is that once the player has stepped on a tile he or she cannot return to that same tile. This rule gives the game a lot of opportunity for challenging level design, which there is. The



Arcade mode features a total of 96 demanding levels with enough variety to keep you entertained.

This 3D puzzler is thoroughly enjoyable to play and the whole game is very well organised. There is a tutorial that teaches the basics so that the game can be played effectively without confusion. The third dimension in Jumper Tree really expands the idea of the game and it's executed very well. It seems like everything is covered in this game, from detailed user settings to a well thought out level editor.

The graphics in this game are the only major problem. They test Game Maker's 3D capabilities but unfortunately, they are not up to scratch. As so often with 3D games, they look too plain and blocky. Also, some of the textures of the objects clash with others and it really makes the game unpleasant to look at. The menu visuals are okay, but they're nothing special either.

The music is nice and enjoyable. The main loop in the game is

long enough not to be repetitive and the rhythm goes nicely with the bounciness of the ball. Sound effects are few and far between, but they aren't too overwhelming and they fit the game well.

[Download](#)
**Miscellaneous_Quest by
\$pecter**

This is another game that was submitted into the Zerosoft Games Retro competition and this is a very good platformer that has been largely overlooked.

The player must guide Zerry, the main character, through the colourful worlds to take down the evil spirit, with a friend Jasmine joining him on the quest. Passing many interesting enemies and through good enjoying level design, the adventure would be smooth and enjoyable if it wasn't for the terribly slow dialogue cut scenes. Another nice addition is the array of tricky bosses that meet the player throughout the quest, breaking up the adventure into manageable chunks.

The graphics are particularly well done in this game. Even though the game is retro in appearance,

the graphics are very well made and attention to detail is certainly shown. The nice environment lets the game flow in a charming manner and is a large part of the reason why this game is so good. The platformer also includes a number of different environments, so that the look doesn't become boring, a really good idea. More work on the main menu would have been good, but this is small negative in a very visually pleasing game.

The audio was particularly enjoyable, with a wide range of tracks making the game really flow. The music built up tension at appropriate times and was not too cyclical. No sound effects are present, but I feel this would have disturbed the brilliant background music anyway.

[Download](#)


INTERVIEW WITH: THATGAMECOMPANY

The team behind 'Flower' - the video game version of a poem

thatgamecompany

'thatgamecompany' was founded in the Spring of 2006 by Jenova Chen and Kellee Santiago, two graduates of the University Southern California Interactive Media MFA program. They are the team behind the multi-award winning student game, "Cloud," they aim to make video games that communicate different emotional experiences the current video game market is not offering. We spoke with President Kellee Santiago and engineer John Edwards.

[1] Tell us a bit about how thatgamecompany came to life, the reason behind the name, and the sort of things that you were thinking about when putting together the company.

[Kellee Santiago] Jenova Chen and I started thatgamecompany just as we were graduating from the MFA Interactive Media program at the University of Southern California. It's a three year program, so we had known each other for a while. While working together on the student project "Cloud" we realized that we had very similar interests and goals for what we wanted to do in the video game industry and that we were very good at working together for long periods of time, a very important trait when you are thinking about starting a business

with someone. Simultaneously, digital distribution was just taking off. XBLA had launched, PSN and Wii-Ware were around the corner, and Steam was really exploding. Publishers were starting to look for unique content to distribute in these new channels, and that's how we came to talk to Sony.

We were asked to pitch a version of "Cloud" for the PS3, but we thought, "when might we ever get the opportunity to talk to these people again?" So our presentation opened with about 25 minutes of us talking about how we wanted to change video game design, and how we wanted to make games that would push the communicative boundaries of the medium. I can't even imagine what those Sony executives were thinking., but that pitch led to a three-game deal for the PlayStation Network!

As for the name, "Cloud" we created the url <http://www.thatcloudgame.com>, so it became a bit of a running joke. When Jenova and I sat down to think of company names, my brother started texting me to see what I was up to. When I told him we were trying to think of names for the company, he responded with, "oh, thatgamecompany?" And it just stuck!

[2] Your latest game 'Flower' was just released. You describe it as a "video game



version of a poem"- what exactly does that mean, what exactly is this game?

[Kellee Santiago] We think of "Flower" as a video game version of a poem because, unlike games that spell out the story through cinematics, the player in "Flower" is presented with images of a stressful urban environment, which is contrasted by the actual game space, in which you play as petals on the wind, flying through serene, imaginary natural landscapes. As you move through the game, the contrasting images become more tightly woven, and we invite the player to interpret the meaning for his or her self. "Flower" has also been described as "a dance". It changes slightly depending on the player and what they bring to the game.

The idea for "Flower" grew organically, no pun intended, from a number of inspirations.

One was to try and capture the feeling of being in a large flower field; to capture both

Continued: Interview With thatgamecompany

a sense of beauty when you see them all, but also the visceral feeling of being up close to an individual flower. If you do a Google image search for "flower" you discover photographs from people all over the world, all fascinated with this aspect of nature.

Technically, it was an exciting challenge. What would happen if we took this aspect of video games that is normally an afterthought on the edge of the world, the bushes and grass, and put it right in front, and make the entire game about it?

What I really like about "Flower," though, is that it feels like a very personal expression of Jenova's. The story and the themes that are expressed in the game really evolved from what I see as his current interpretation of the world around him, and the game invites the player to participate in that space. It has an



intimate feel to it.

[3] Your first game was only available on PC, then we saw a two year gap before you released 'fIOW', then another two year gap before you released 'Flower' - how do you go about planning your projects in order to reach a set deadline?

[John Edwards] Planning is something we haven't always been especially good at, but to be fair, we spent most of that time between Cloud and fIOW finishing school! In fact, Jenova and Nick built the Flash version of fIOW in a couple months while they were still in school, and then the team spent seven and a half months building the PS3 version immediately after graduating. fIOW was the first commercial game any of us had ever worked on from beginning to end, and consequently the first time we'd ever had to plan out production. The plan was actually pretty good, assuming Nick was three people and we were on Pluto, where days last 154 hours. On Earth, fIOW took twice as long as we originally intended and had half the features, and left the team delirious and ulcerated with the insane hours and constant stream of worried Sony producers passing through our offices.

After taking some time and Pepto Bismol to recover, we decided planning would be one of the main areas we focused on for Flower. As a result, we were able to hit every

deadline and never once "crunch" (meaning to involuntarily work longer than a normal work week). It turns out, for us there wasn't any magic to it. Quite the opposite in fact. It was just a matter of periodically going through the tedious process of listing every single task left to make the game. The "details of the task" lists were often wildly inaccurate, but the picture they gave of the overall scope of all our features was invaluable for helping us decide what was truly important for the feeling of the game and we could cut out.

[4] You work with Sony Computer Entertainment to create games for the PS3 Network, what lead you to make this leap from PC to making PS3 games?

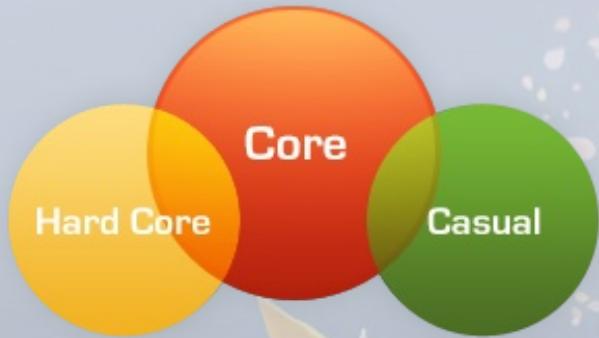
[Kellee Santiago] Part of it was that SCEA made us an offer we couldn't refuse! Not only was a three-game deal a great offer for students just coming out of school, but we also clicked with the people at Sony Santa Monica immediately from our first meeting.

We had pitched to and spoken with a lot of publishers before them, and I remember after our meeting in Santa Monica Jenova and I hugged each other in the parking lot.

Finding a publisher is similar to finding a business partner, because they are very much collaborators in the development process, so you want to work with people who really get what you're about. And they got it.

Continued: Interview With thatgamecompany

Technologically, the benefit to working on consoles is that you know exactly what system your players will be using, which is very different than in the PC market, where you have to worry about connection speeds, video cards, processors, and all the possible combinations of parts. While there were still plenty of technological hurdles to overcome on the PS3, in some ways it did allow us to focus more on the creative side than the trenches of debugging a PC game.



[5] You create 'core games', a mix between hardcore and casual, would you say it is harder to go about catering for two different sets of gamers, or easier as you have no set guidelines for a specific audience?

[John Edwards] We try to make our games emotionally engaging and accessible. While many hardcore games focus on different goals than ours, that does not mean hardcore gamers are incapable of

appreciating emotions or simplicity. In fact, everyone on our team could have been considered a hardcore gamer at one point in their lives, and most of us still are. We all grew up playing NES (or Sega Master System, in my case). We understand what makes Halo, Madden and Metal Gear so good. But as we've gotten older, it's gotten harder and harder to devote the time necessary to truly appreciate traditional games.

Over their 40-year existence, video games have built an extremely complex vocabulary. And not just things like 1-up, d-pad and deathmatch, but also: enemies die when you jump on their heads, red barrels explode when you shoot them, and your health regenerates completely after 5 seconds. Learning and retaining the language of games takes hours a week. Most people have never learned game language or have lost touch and forgotten it, and so all the good material in modern games is completely inaccessible to them. If you can't read English, Shakespeare is no more fulfilling than the back of a cereal box. We're trying to take what's so great about games today and translate that to a format that doesn't require extensive study and practice to enjoy.

So, our job is not made harder by having to cater to two different audiences since we don't cater to two different audiences. We



never specifically say, "Okay, throw this feature in for the hardcore gamers, and that feature in for the casual gamers." The hard part is trying to build games without relying on hardcore gamer conventions. Partially because, being hardcore gamers ourselves, it's so easy to slip one in without even realizing it, but mainly because abandoning the existing gamer language requires that we invent a new one. The task is so hard, in fact, we still get this wrong all the time.

Flower is littered with hardcore vocabulary. Luckily, we aren't the only ones working on this new language for games. The indie scene is full of games exploring all sorts of directions, many focusing on the same kind of accessibility that we do, Knytt by Nifflas being a prime example on the Game Maker Community.

[6] After spending months working on a game and you are set for release, what is going through your mind, what are your hopes and what are your fears?

Continued: Interview With thatgamecompany

[John Edwards] I think what you feel right after a game comes out is very personal, and it's different for everybody. So, while I love almost everything about making games, release is one of the toughest times for me. You spend the last two months of a project in "beta," or "code complete," meaning you can no longer make any changes unless you are directly addressing a bug reported by the Quality Assurance team. This puts you in a strange place psychologically, because to hunt down bugs you're forced to play through the game over and over again, constantly reminded of every poorly conceived design or missing feature, but completely unable to do anything about it. This process rapidly erodes any hope that the game does, in fact, have some redeeming value, to the point where all you can think if you read initial positive reactions is, "No! Don't you understand how bad this is? Can't you see that so much is missing?" There's also an element of Stockholm syndrome, where throughout the project's development, there's been a lot of blood, sweat and tears poured into it and you really should be relieved to be done with it, but you're just not ready to let go, and you feel listless and lost for a couple of weeks. With any luck, something will go terribly wrong right after launch and you'll have to jump back into the code to patch it up.

fIow and Flower were both released about a

month before the Game Developer's Conference, and both times that's been where everything sunk in and I felt ready to move on to the exciting planning phases of the next project. For Flower, specifically, it was also when I began to really appreciate the positive player and press responses, because Flower represents a type of game that I want to play, and the warm reception it got will hopefully encourage more developers to make tight, short, emotionally-focused games.

[Kellee Santiago] Yeah, my feeling when the game is released is that I've thought everything there is to think about the game, good and bad, so it's just a matter of what aspects will other people notice. Something that pleasantly surprised me about the release of "Flower" is that now that we have both "fIow" and "Flower" out as a studio, people are really getting a sense of who thatgamecompany is. That feels really great; to be at a point where we've been successful enough to build a small library of games under our banner.

[7] You started the company after going through the University Southern California Interactive Media MFA program. Do you feel that the extra years you spent studying helped shape where you are today? Do you think you



still would have been successful if you didn't go through the university route?

[Kellee Santiago] Absolutely. The USC Interactive Media program is housed in the School of Cinematic Arts, and during our three years we had to study film production, fundamentals of animation, screenwriting, and critical studies, in addition to our classes on virtual reality, mobile media, interactive installation, and of course, game design. Having this rich background in media feeds directly into the games we make at every level, from inspiration to execution. I think now we are seeing an explosion in critical study when it comes to games, but at that time, even studying it in school seemed too fluffy. However, going through school allowed us to mature as designers much more rapidly than if we had been working in the bowels of some large studio for that same amount of time. We also made great contacts and friends in the industry, who we keep in contact with and who we still use as

Continued: Interview With thatgamecompany

sounding boards for our ideas.

[8] What are your tools for developing? What are some of the previous ones that you have used and dismissed as you don't like them and found their price tags ridiculous considering they didn't work well?

[John Edwards] Our most important tool has a ridiculous price tag, but it's so useful it's worth it: Perforce Source Control. Source control keeps track of every version of every file in your project, so you never have to worry about a temporary experiment or a misguided optimization ruining everything.

If something goes wrong, you can just revert back to the last known good version. Source control also makes it easier to share files between team members, by either preventing multiple people from working on a single file at the same time, or, in the case of text files, allowing two people to work

simultaneously, and then intelligently merging the two different versions.

Depending on the project, we use a variety of different editors and compilers, but we always use Perforce. It's also free to use on networks with 2 or fewer simultaneous connections, so I use it at home for my own projects. If you have a large team and can't afford the \$750 licenses, CVS and Subversion are good free alternatives. It seems like very few indie studios use source control, which is a shame, since it's such a productivity boost. It takes a week or two to get really comfortable with, but after that you can't imagine working without it.

For production, we use all the standard game developer tools: Photoshop, Maya, Visual Studio, and so on. On Flower, we also built some of our own tools, such as our level editor, Gardener. For prototyping, the best tool is free, and it's sitting right between

your ears. A large part of Flower's game play was prototyped in Microsoft's XNA. The idea is to always use the simplest tool that can get the job done, but no simpler. Of course, this is something we almost never get exactly right. We learned the hard way during one of our game jams that PHP and mySQL are not good tools for making a real-time multiplayer game. We also learned that big fancy scheduling tools like Microsoft Project are cumbersome, and you're generally better off with a simple Excel file or even stickies on a wall.

When we begin working on a game, we start with a single simple emotion or feeling. For Flower, the feeling was standing in a wide open field of Flowers. Development is all about exploring that feeling, and figuring out how to most effectively convey it through a game. We try to pick feelings that are relatable, but if some people don't like the particular feeling, that's okay. Not everyone likes standing in a field of flowers, but maybe they'll like the feeling of our next game.
John Edwards



**thatgamecompany
Website**

Spiderweb
SOFTWARE



FANTASY ADVENTURES FOR WINDOWS AND MACINTOSH

GM CLASSICS

HOVER TANK 3D

[Download](#)

In Game Maker, games that are 3D have always been limited by the power of the GM runner and the artistic skills of the developer. Compared to Crysis and Unreal Tournament, GM's 3D capabilities are extremely poor, and the majority of GM developers are sadly lacking when it comes to true artistic talent.

Being mainly young teenagers, many current GM developers have been inspired by these high-poly, processor intensive games, and are understandably disappointed when it turns out that GM is nowhere near as powerful. However, Martin Crownover (aka FredFredrickson) got inspiration for HT3D not from these mega budget games, but from the older more basic ones. "Back around the time that I had discovered perspective drawing, 3D graphics in more mainstream games were also starting to appear. The original SNES StarFox, and an older PC game, which I believe was called JetFighter II, inspired me a lot".

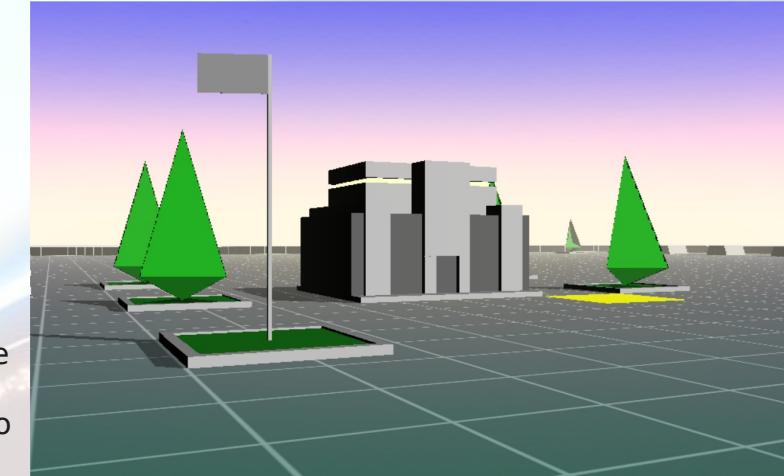
Even today, 3D games made using Game Maker rarely manage to be as slick and polished as HT3D, but how does it compare to most 3D games? "I think HT3D would still receive a fair reception, though the expectations for better graphics and sound, and more complex game play have

increased since it was completed. I definitely think that the multiplayer would have to be re-written, and maybe a bit more polishing applied to the overall game, but generally I think it's still a decent addition to the Game Maker collective."

So did the final game match his original vision? "The trees in the game and the look of the buildings are almost exactly how I pictured them many years before it was even possible for me to do such things on the computer."



For years, Martin had dreamed about making a similar game, and with the release of GM6 and its 3D capabilities he had his chance. "It was quite incredible to me to be able to make something I'd dreamed about years before, and I was excited to finally be able to put some of my skill in 3D modeling



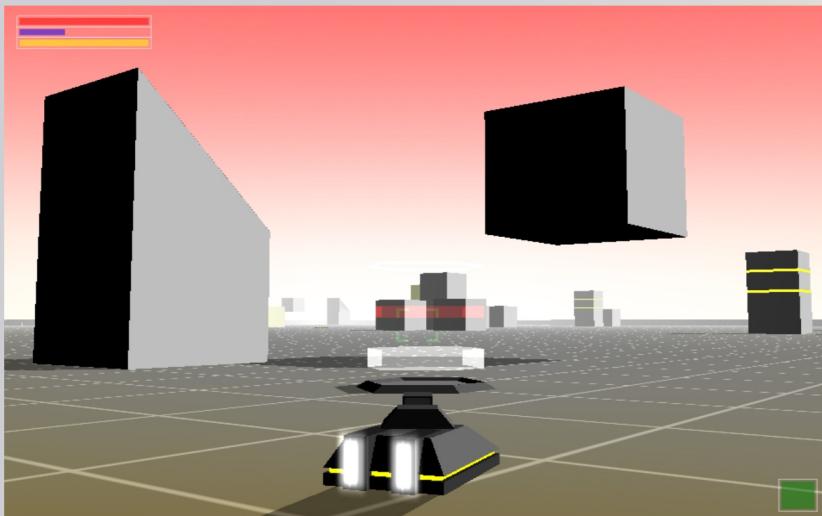
and graphics to a good use".

As if creating one of the first (and best) 3D games with Game Maker wasn't challenging enough, he decided to also add fully functional online modes, and this wasn't without its problems. "The multiplayer portion of the game used an older DLL (I think it was called WinSoc, or something like that) which used another GMC member's server to direct communication between players' games. Unfortunately, the proprietor of this system became disgruntled by all the complaints GMC members had, so he turned off the system, and as you would expect, anything that used it was instantly broken. HT3D's multiplayer was included in the casualties, and at the time, the thought of re-writing the entire thing to work with a different system that I didn't know was daunting". But Fred didn't let this put a stop to his making the game he had dreamed about, "After a break of a few weeks, I decided that I shouldn't give up, and began looking for an alternative multiplayer solution". Using a GM helper service he met the user Fox-NL (Wilco

Continued: Hover Tank 3D

Schoneveld), who taught Martin how to use 39DLL as a replacement for WinSoc. Together, they got the multiplayer portion of HT3D sorted as well as then going on to create the Reflect Games registration system that has been used by over 8,000 people. "So I guess you can say that if WinSoc wouldn't have been turned off, I would have never became friends with Wilco and probably would have never created Reflect Games".

It was one of the first large scale 3D games in Game Maker, did Martin feel like he was breaking new ground while developing it? "It did on a personal level, because I was finally working on something that other people seemed to be interested in, and something which I found very interesting as I learned new techniques and added new things to the game.



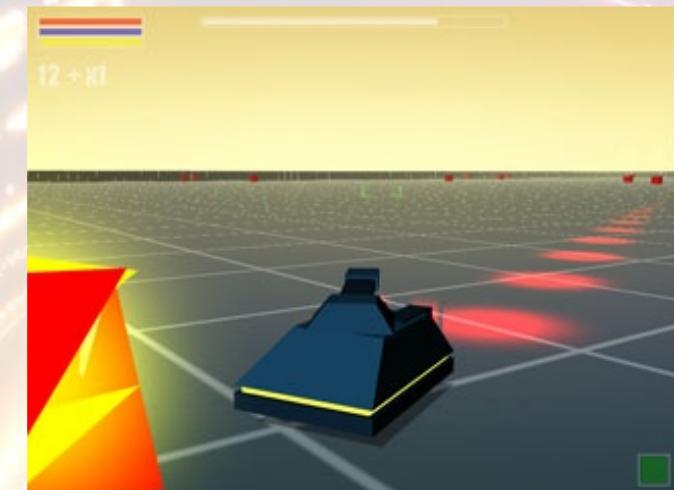
On a broader scale however, I never felt like I was really breaking new ground with Game Maker, because I wasn't seeking out other 3D games while I worked on it. I didn't want to be tempted to borrow ideas from other people, and I felt that I had enough ideas on my own. A lot of HT3D was developed within the vacuum of my own process, and from the feedback I got from the WIP topic I had for the game; I was mostly unaware of what others were doing with Game Maker's 3D at the time."

Martin had to learn many new things for the development of HT3D as he was finding out about GM's 3D functions for the very first time, but this helped motivate him further. "In some ways, I liked the challenge, it was fun to see ideas go from some drawings on paper to something I could actually play on the computer. I like to try to encourage people to make the game that they want to make, and then worry about whether or not people will like it, but I can't deny that finding people who support you, and who give good advice is a big help.

HOVER TANK 3D 2?

In a previous issue of GMTech we featured a preview of Hover Tank 3D 2, since then there hasn't been any news on it, so what has happened to it? Fred explained; "I do have plans for it, but progress is slow. I've been working on assets that I might use for HT3D2 for some time now, but I haven't come up with a firm idea of what exactly I want to do with a sequel yet. It's easy

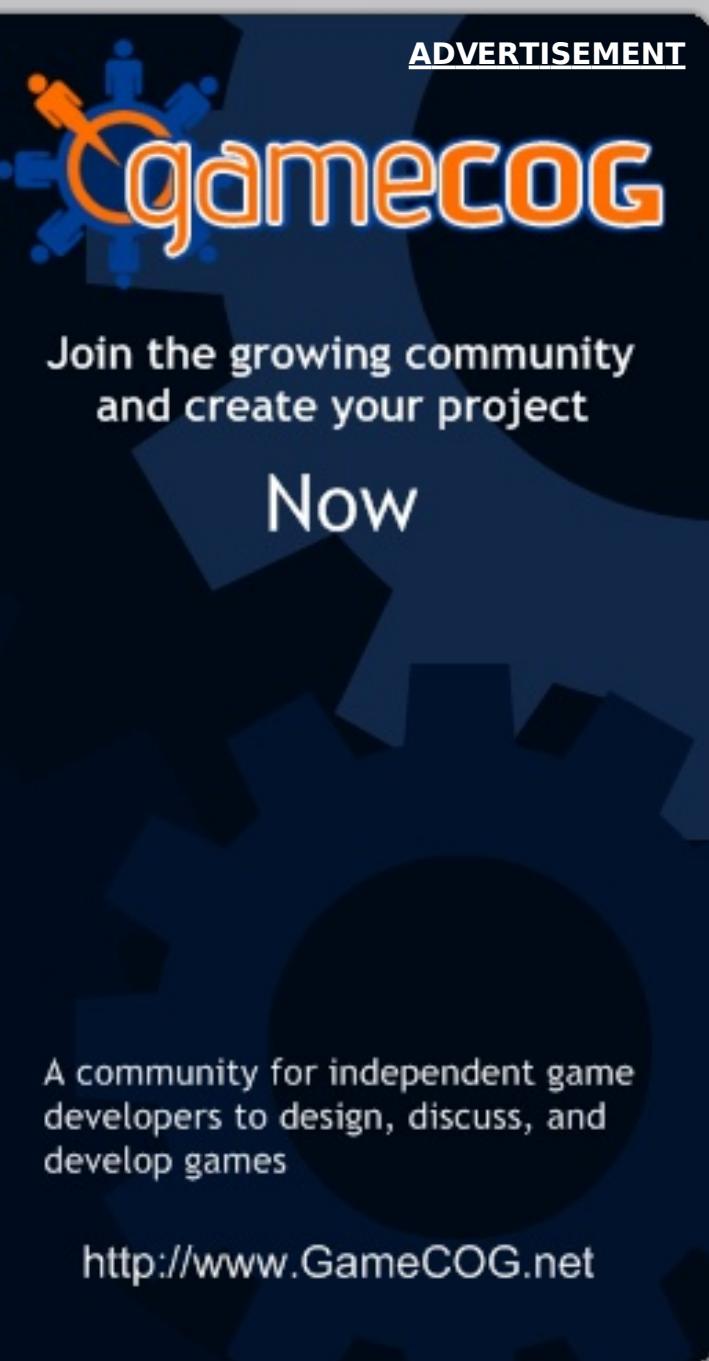
to just throw some new graphics into the old engine and add a different story, but I want to do something better than that. Unfortunately, I haven't had as much time to work on things as I did with the first game, so I haven't been able to get as much done yet. I haven't abandoned the idea though, so hopefully I'll have more to show sometime in the future!"



What are your classics?

If you have a particular game you'd like us to feature here (must have been first released at least over 2 years ago), you can get in touch with us in the usual ways (email/forums/PMs).

ADVERTISEMENT



The advertisement features a dark blue background with a stylized gear and cogwheel pattern. In the top left corner, there's a small icon of two people working together. The word "gameCOG" is written in a large, orange, sans-serif font where the 'g' has a gear-like shape. Below it, the text "Join the growing community and create your project Now" is displayed in white. At the bottom, a circular badge contains the text "A community for independent game developers to design, discuss, and develop games". A URL "http://www.GameCOG.net" is at the very bottom.

WORTH A LOOK

Jub Jub talks about his GM Highscore website

Game Maker Highscores (GMH) is a free tool that enables anyone to add online highscore capabilities to games designed with Game Maker 7.0 Pro. Unlike other online highscores systems, GMH was designed from the very start to be easy to set up and use.

Game Maker Highscores allows games to access scores without opening a browser window, so that gameplay is not interrupted. Netread.dll is used to access the highscores. This is implemented as simply as possible, so that it would be easier for users who are new to Game Maker.

GMH also uses a verification system for submitting highscores. This helps protect against users hacking into the highscore system. Various security features have been implemented recently, including blocking highscores submitted by using a browser, instead of netread.dll.

This program has its start six months ago in August 2008 by the creators of the site as something for their own use. It was then located at gilsoftware.net and some users found that it was either hard or impossible to submit their highscores.

Game Maker Highscores began to take shape in December 2008, when the developers set off to design something so that others could create online highscores quickly and for free.



Welcome to the GMH

About a month later, the website was ready, and was announced online.

Within the next month, the Game Maker Community posted more than 30 comments on the GMH topic, and over 50 games were registered with GMH. In March 2009, the site was moved to gmhighscores.com and was rebuilt so that highscores were easier and more secure. The site is still at gmhighscores.com today.

As of May 2009, GMH serves over 160 games, 120 users, and 2500 highscores. The GMC topic has almost 100 comments, and the GMH helpdesk has been used extensively. Check it out at gmhighscores.com, and stay tuned for the many updates soon to come!

Visit Website

Get In Touch

Have you made a new Game Maker related website, forum, or even a useful application, and want to write about it in this column? Then get in touch with us by posting on our forum.

BE PART OF GMT

Join The Team

Becoming a part of the GMTech team is easy. You just need to be a good writer and competent with GM. Simply make a topic in the Join The Team forum and we'll set you a little a test. Once accepted, you can be proud of the fact that you're part of the longest running Game Maker magazine.

Join The Team

Freelance

You don't have to be a staff member to get your writings into GMTech. You just need to post that you have written an article for us on the GMT forums, then we'll get in touch with you. As long as it's of a high quality we'll put it into the next issue. This is a great way to contribute without having to commit to the team or if you have less time to spend writing.

Submit An Article

Website

You can download and read all of our issues from our website, as well as find out more information about GMTech. News about GMTech is always posted there first.

www.gamemakertech.info

LEGAL STUFF

Corrections

GMTech Magazine is happy to correct any published errors. If you spot any mistakes and would like to alert the editors to them, send a message to gmtech.magazine@gmail.com outlining the error, including the correct information.

We will endeavour to correct any published errors as quickly as possible.

Thanks

CLOSING

If you're reading this, I'm presuming you haven't actually read the entire magazine, but I hope that you've enjoyed what you've read so far. The interview with ThatGamesCompany is probably the longest article we've ever had!

Looking to the future, a new version of the DTP (DeskTop Publishing) software we use is coming out soon which will allow us to keep on improving the look and feel of the magazine.

All that remains is for me to say thank you to all the writers in the magazine and to everyone that have (or will) encourage us. A special mention should go to some non staff people: Elmernite, ToadFrogs and Giorgio.

Tim (aka Timoi)
Assistant Editor

Copyright

The Game Maker Technology Magazine is licensed under Creative Commons license. This means you are free to copy, distribute and transmit the work. This work is for Non-Commercial use only. You are NOT allowed to alter, modify or build upon the work contained in the magazine.



ISSUE 17
Coming soon...