

Today, I received an award for **Tactics Core** !!
I'm very excited, of course ^_^v !

FWA ONE AWARD
Favourite Website Awards



-V2C, aka Seed



Play **Tactics Core Demo** ! (Requires Flash 6+)

Thanks to everyone who sent me feedback.
It's been great help! =)
Special thanks go to **AnonT** for writing the guide below !

AnonT's Tactics Core Unit Summaries

Hunter is a relatively weak ranged combat unit. He has low health and defenses, and his arrows are less effective than the magic attack of the Talon Guard. His advantage is his speed, and he can attack before the enemies reach you. However, being unable to shoot over or around allies and not being able to attack at close range, the Hunter is very difficult to use during the main battle.



Ironclad Sentinel is like a tank: strong but slow. Unfortunately, he doesn't have the ability to shoot explosive shells across the battlefield, and his close range attack isn't quite strong enough to make up for that. The Sentinel is good for holding off the enemy's more powerful close range units, but he'll need backup if you expect him to actually kill the units he's fighting.

Shadowfist Fighter is one of your faster and more mobile close range combat units, but he sacrifices magic resistance for more health and a powerful attack. The Fighter is very useful for going after your opponent's Cleric and Sorceress, but doesn't last much longer in a close range fight with enemy swordsmen than the Assassin does.



Shadowblade Assassin is the fastest and most mobile unit in the game, but relatively weak otherwise. The Assassin is best used against physically weaker enemy units such as the Sorceress and Cleric, as she doesn't tend to last long in a close range fight with enemy swordsmen.

Templar Mage does fairly good damage at a decent range, and has the added bonus of being able to hit more than one unit at once. If you get a good change to hit two or three enemy units at once without too much risk to your Mage's life, take it. His biggest disadvantage is that his damage isn't strong enough to overcome the Sorceress' high magic resistance.



Red Wizard is the big guns on your team, so take my advice, don't let him die early on. He can do 1000+ damage to your enemy's melee units, and often hit more than one of them at once. Beware that he has low physical defense and speed, so getting him into the right position to deal out the massive damage will take planning.

Mystical Healer is, unfortunately, inferior in almost every way to your opponent's Cleric, but that doesn't mean she's worthless. Her biggest disadvantage is that she can only heal at close range, often putting her at risk. Be careful when and how you use her to heal if you want to keep her around long enough to save all your units.



Daemon is a mystery unit, bearing striking resemblance to the Healer. Her attack has great range, and will kill any unit at once. Daemon has no defenses, but that is hardly noticeable when the enemy can't even get close to attack her.

Templar Striker has a weak magical attack that can hit up to two of your units at once. While he has a shorter range and can hit fewer targets than your Mage, he has better speed and mobility.



Macabre Swordsman has good speed, decent movement, decent health, does fairly good damage, and is fairly resistant to physical attacks. His main weakness is magical attacks, and your Red Wizard can do 1300-1400 damage to him in one shot. While he is an all round melee attacker, he is far from the biggest risk to your party.

Talon Soldier is slightly slower than the Swordsman, but more than makes up for it with better health, equal or better damage, and slightly better magic resistance. Due to his high health and good defenses, you're best off ignoring the Soldier until the Cleric has been dealt with.



Talon Guard fires a long range magical attack that does better damage than your Hunter's arrows. The Guard shares one of the Hunter's weaknesses though: he cannot attack at close range. If you maneuver your units right, the Guard can be trapped and nullified.

Cleric has the ability to heal units from a few squares away and tends to return several hundred hitpoints more to your opponent's units than your Healer is capable of. The Cleric should at least be one of your first targets, if not the first enemy unit you target and destroy.



Mystical Sorceress deals good damage (often 800+) to more units at a time than your Red Wizard, and always seems to be able to hit more than one of your units without hitting any of her own. While she is nearly immune to magical attacks, she is quite vulnerable to close range attacks, and your Assassin and Fighter can dispatch her very quickly. Don't let her live too long.

February 2nd, 2003. 3:00am.

Take Flash MX, 3DStudio Max, Swift 3D, Cooledit Pro, plus one Seed ^^ , and the result is what I call **Tactics Core**.

So... what IS **Tactics Core**?? It's a 100% Flash based engine for developing TBS games in Flash. I've created a demo game to show what it can do.



You'll find it quite challenging at first, but once you figure out how all the units attack, it'll be much easier. Also, if you're lucky, the most powerful unit, *Daemon*, might appear on your team! There's a sneak peek of her in the screen shot below ;)

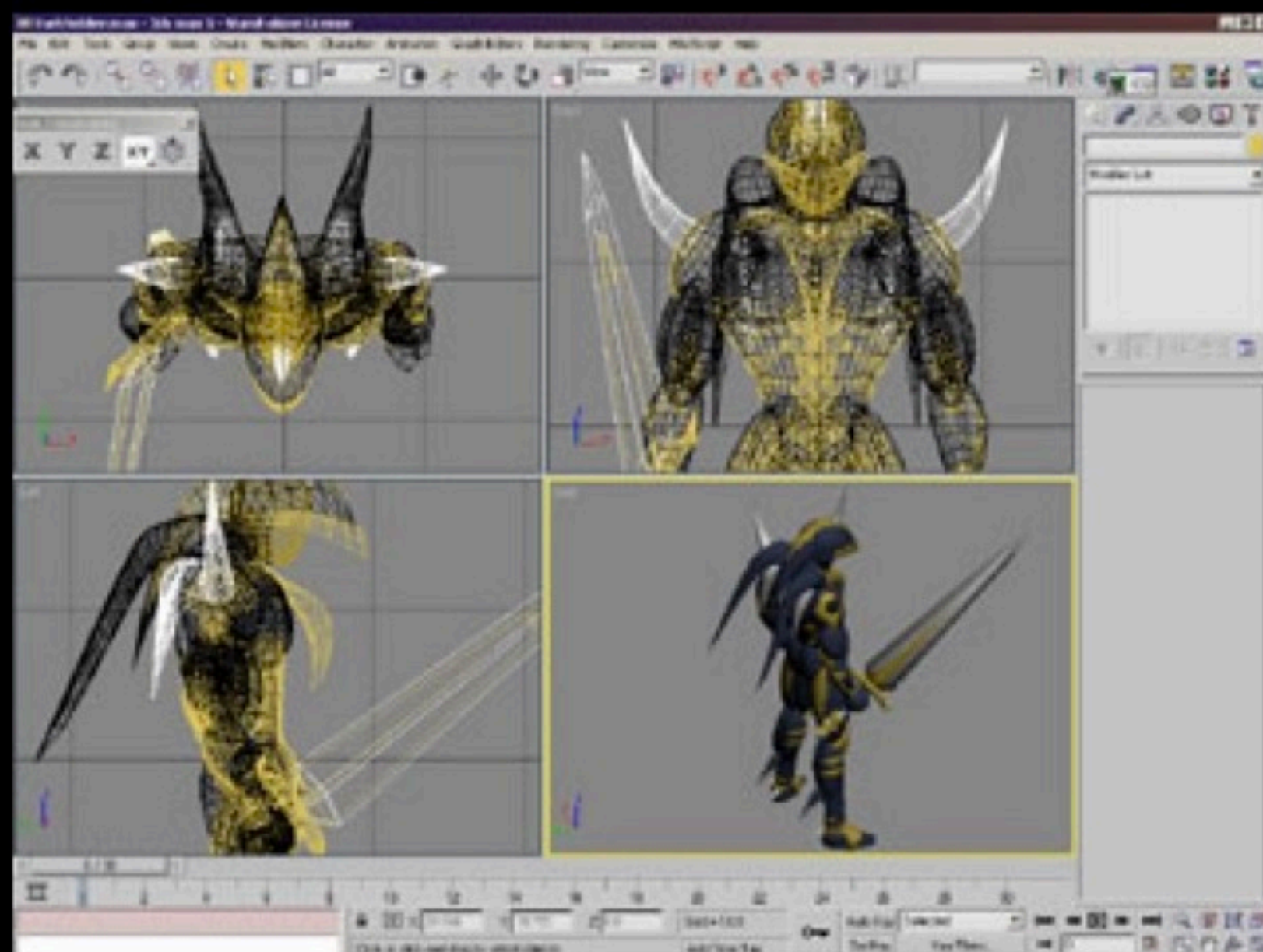
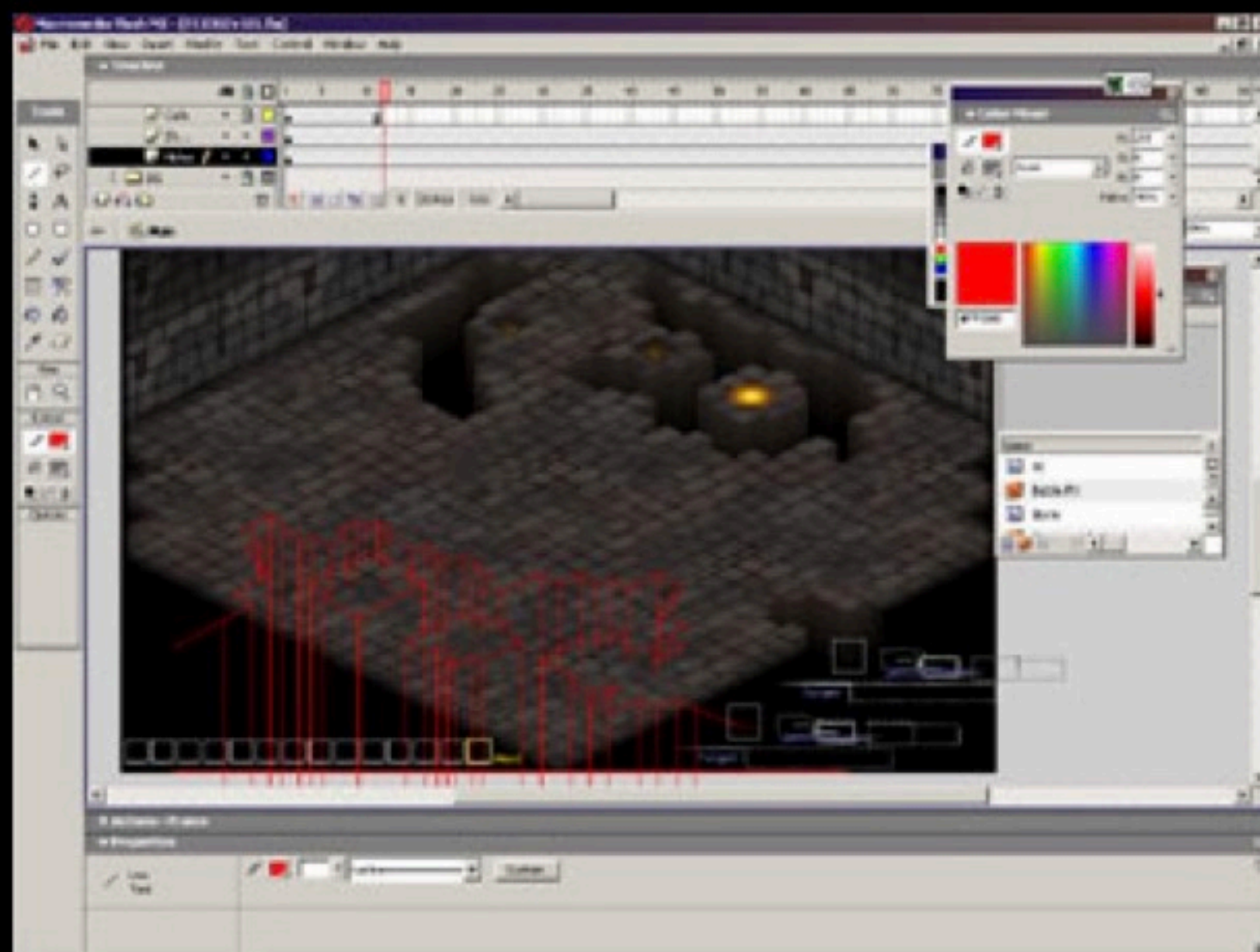
Depending on the feedback (IF there's any -_-), I may publish online documentation on how it was created, or develop this into a full length RPG. Of course I won't be able to do this alone...

Please email me at v2c@digisonline.com if you have any questions or comments (or feedback ^^"). Thanks!

-V2C, aka Seed



Page 2/7 [Prev] [Next]



Alright! I'd say **Tactics Core** was a success, all thanks to my friend FFuryFX who posted my link on **Flashkit** the day I found out that I WASN'T selected for **Flash Film Festival**, and of course I owe it all to **Newgrounds** for hosting me on their front page! And now, on to the documentation you asked for =)

Before you read this, please note that this documentation is NOT a tutorial in any way! It is only meant to give you an idea of my planning process and thoughts along the way. If you find something useful, be sure to search for more information about it with the links I've provided! Enjoy ^^

Where to start? Why **Flash**? **Flash** has an incredible graphics engine, not for its speed but for the way its graphics look. And almost everyone's got it installed already, what more can I ask for? If I took **Flash MX** and wrote a game that couldn't honestly be published in **Flash 4**, I knew it would attract some attention ;)

First, I had to do some research on this type of strategy game... tried to finish **Final Fantasy Tactics** overnight at my friend's place - _-" Another game I constantly referred to was **Dragon Knight 4**, but that must have been almost ten years ago.

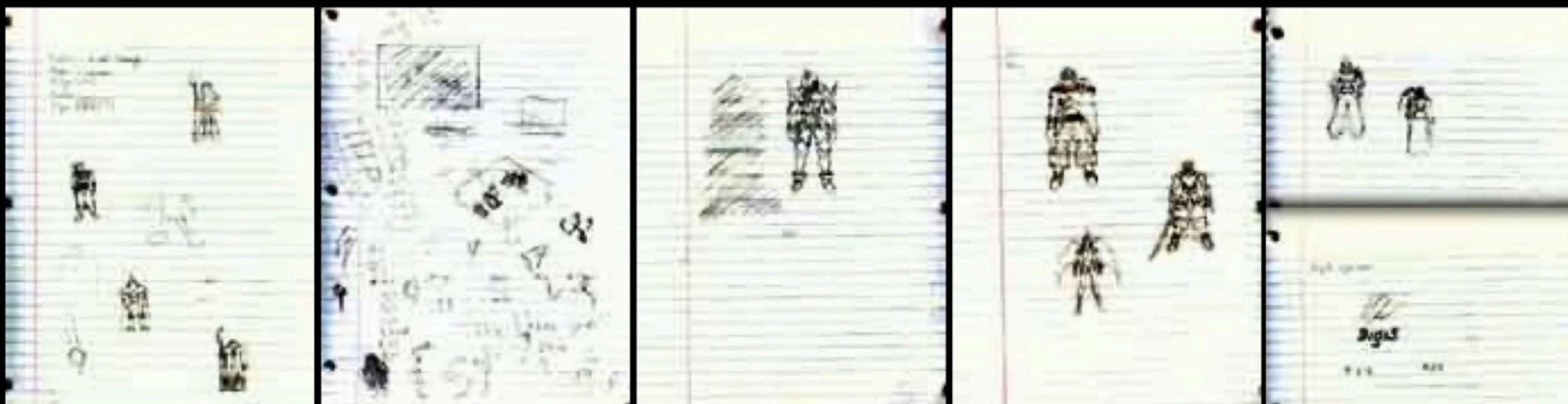
I liked the turn queueing system in **FFT**, so I decided to include that in **Tactics Core**, however I stuck with **DK4** on the unit/character design: each unit has one type of action at a time. This made it like a chess game requiring more strategy, and it's also easier for a new player to learn all the commands quickly. Plus, it's more feasible to program in **Flash** =D

I started making notes and doodling character design sketches. If the project's going to take a few months, it's best to keep some written notes. Try to complete everything you initially planned **BEFORE** adding or modifying features. If you can add in the features easily at the end, it's a sign of good coding design. For example, things like the *"Undo"* command were not added in until the whole game was completed.

OK. The design process is really up to you, not much more I can say there. You can see my notes here, they're **EXTREMELY** crude, that's why I made them small and blurry on purpose. ^^"

By: V2C, aka Seed

Email me at v2c@digisonline.com



By: V2C, aka Seed

Email me at v2c@digisonline.com

The secret is I CAN'T draw 2D graphics if I have to make them consistent in four directions! For me it was easier to learn 3D modelling instead, heh. I used **3D Studio Max 5** to model the characters, and **Swift 3D 2** to convert them to **Flash**.

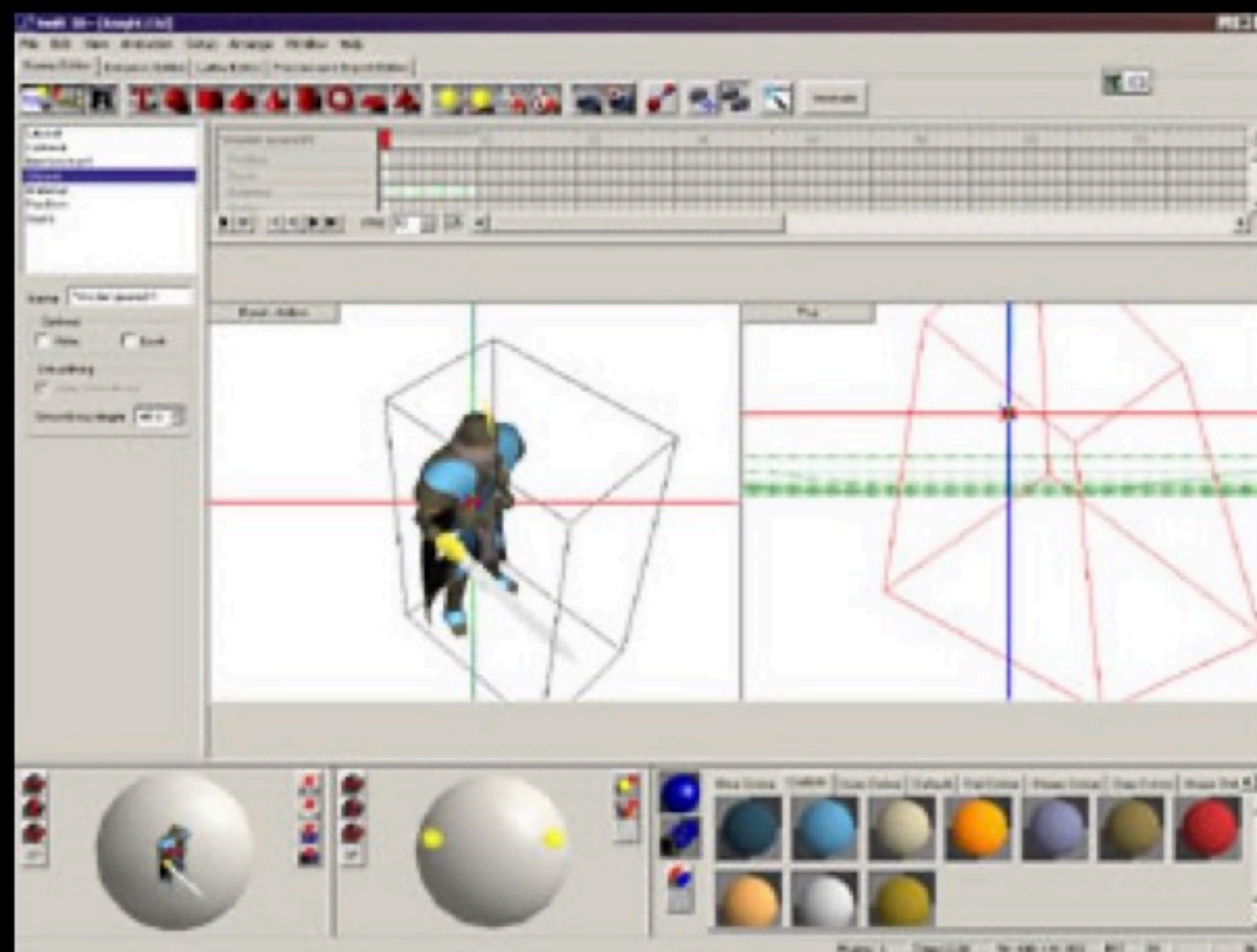
A great resource to get started for **3DS Max** is **3D Cafe**, look in the *Free Stuff* and *Tutorials*. Personally I only use one technique to model EVERYTHING, because it's the only technique I know: *Patch Grids*. I'm sure **3D Cafe** has tutorials on that.

I did not have to add texture or material to any of my models in **3DS Max**, but I had to give the objects colors since I exported as DXF (worked best for me when importing to **Swift 3D**). Note that since you are converting to **Flash** in the end, it is better to have many small simple shapes. If you tried to model a large smooth complicated shape, it will just turn out as a blob with a single gradient fill...

For each model, I made one standing pose, three walking poses (left foot forward, center, right foot forward), and two attacking poses. If the models were symmetrical, then I would only need two walking poses, but they would be less interesting. The left-walking pose is reused as the being-hit pose, and the center-walking pose is reused as the step-aside pose (to let other units walk pass).



I only used **Swift 3D** for converting the models to **Flash** format really. I reassigned all the colors/materials to the models inside **Swift 3D** first though. Color everything as flat/matte as possible because when **Swift 3D** tries to optimize your shapes, sometimes the specular shading does unexpected (unwanted) things.



I imported each pose separately, and rotated them around the vertical axis to the four directions. Then I rotated them down 45 degrees along the horizontal axis since that is the camera angle that I am trying to simulate. Remember to position the lights to the same position for every file, mine were basically just directly in front.

I never changed the lens length when exporting, so all my characters are rendered with perspective. That was a PAIN to animate because the stage is in an isometric/orthographic view! I would strongly recommend increasing the lens length to simulate an orthographic view!

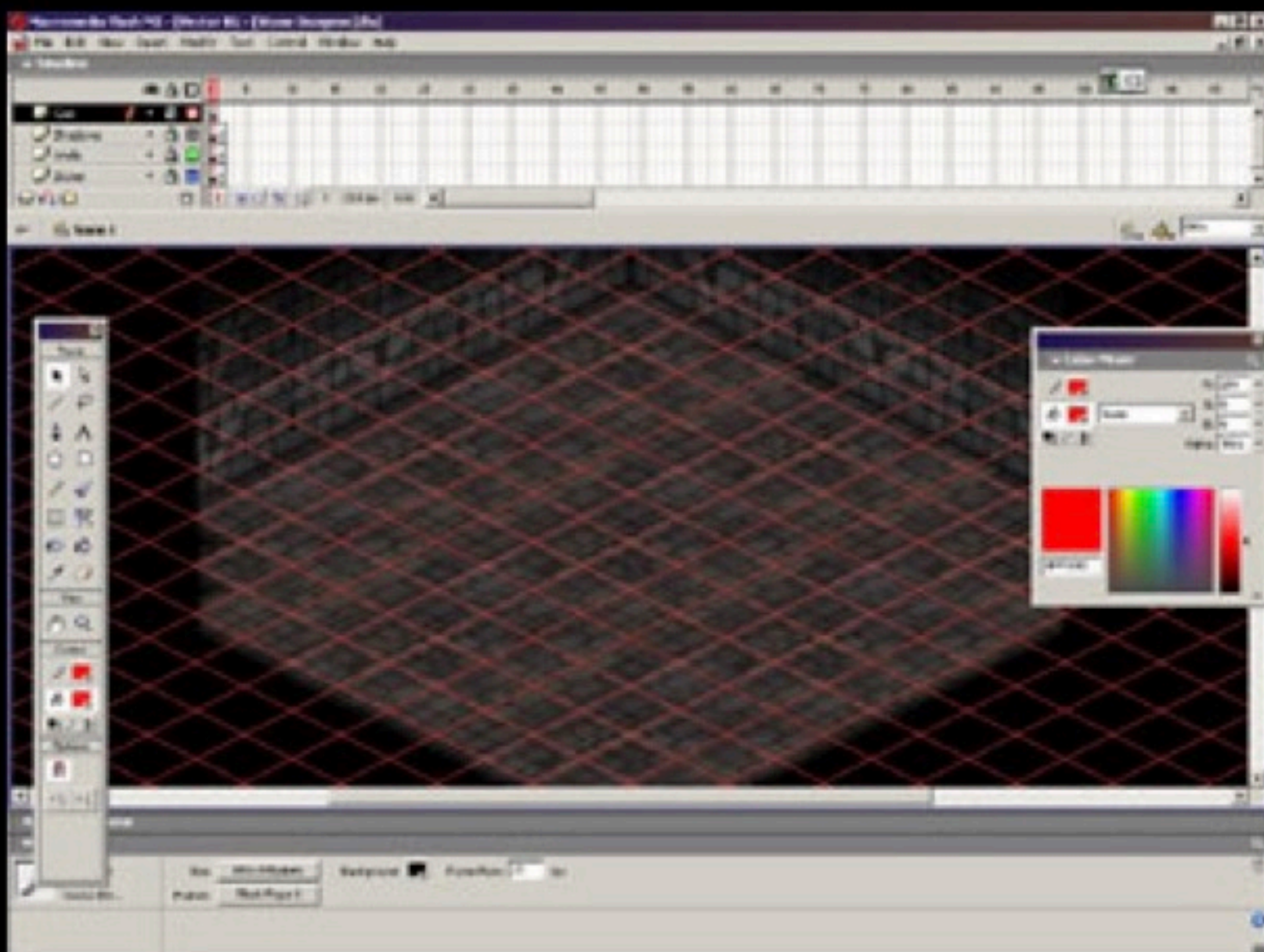


Everything in the **Tactics Core Demo** is vector graphics, or at least was at some point. Even the background!



As you can see the stones are composed of very simple vector graphics, however there were too many vectors when I tiled the whole ground plane, so I took a screenshot and imported it back in as a bitmap. The effect of raster images on the framerate is not dependent on the amount of detail, so at some point you will have to decide what is better for you.

To create the orthographic view with the camera at a 45 degree angle, I rotated the tiles by 45 degrees, and then scaled the height by half. When animating motion on this ground plane, moving 2 pixels in the X direction corresponds to moving 1 pixel in the Y direction.

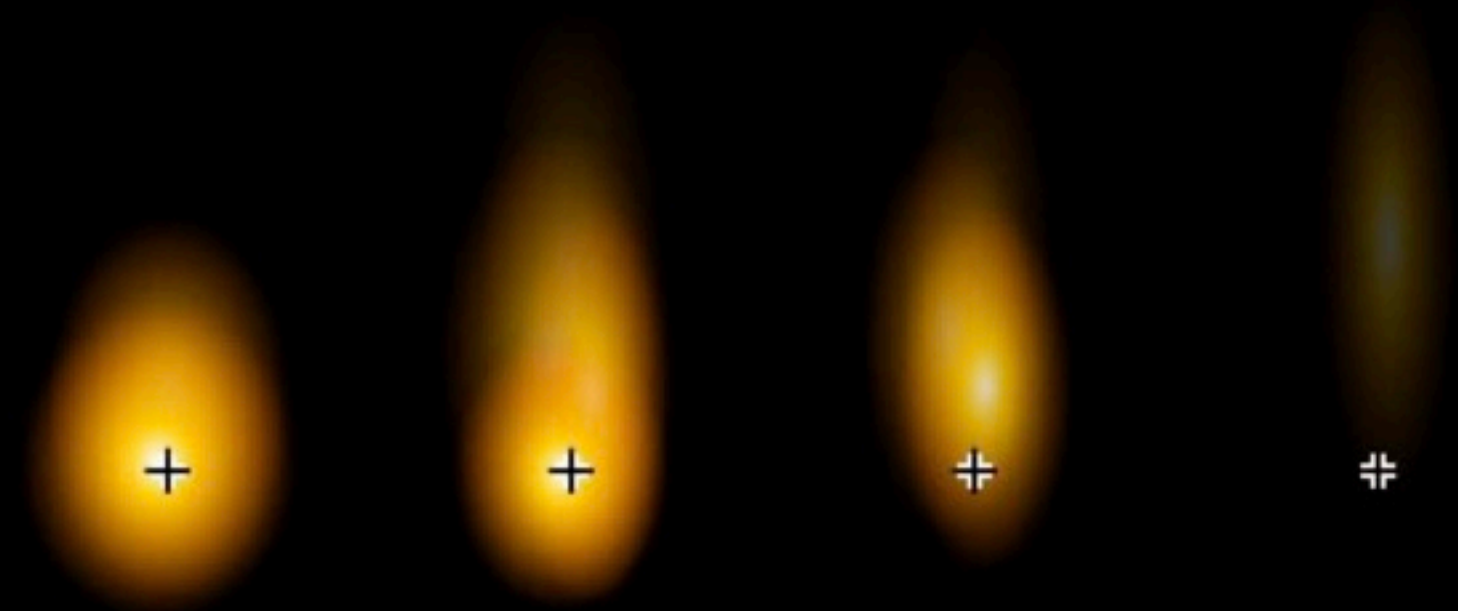


Before the imported 3D sprites could be animated, I had to scale them all to the same size. They had to be small enough so that with all the overlapping, you can still clearly tell which unit is standing in which space (by their feet).

To link the separate poses into a smooth animation, I drew a strip across the most prominent area of motion (such as the sword or staff), and shape tweened a gradient fill across it.



The great thing about **Flash** is how easy it is to make gradient fills with Alpha blending. I used that in almost every single one of my battle effects graphics. For example, just take a circle, gradient fill it so that the edges fade out to zero Alpha, and you can make fire!



I actually picked up this technique from **Flashkit** :D

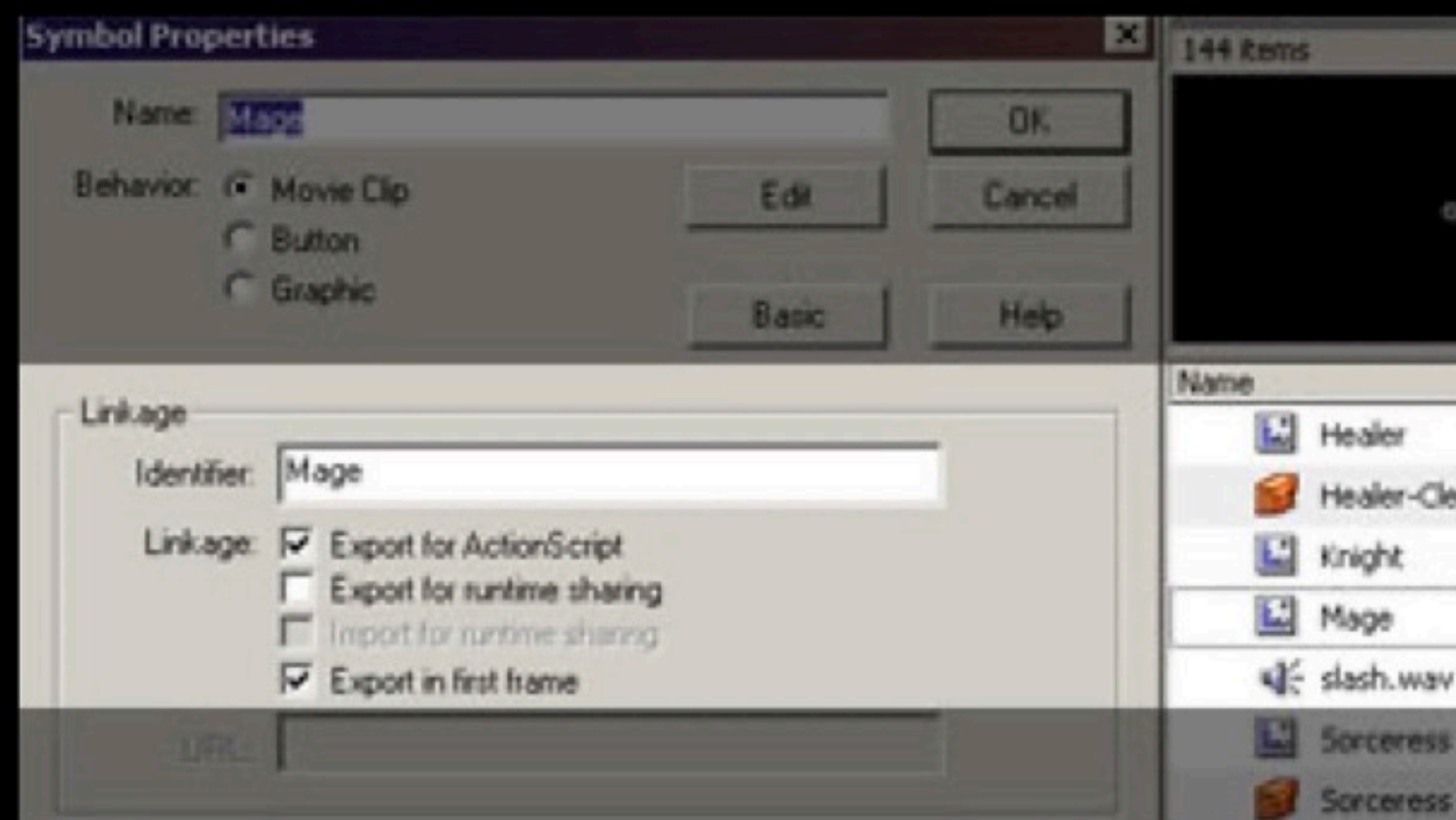
By: V2C, aka Seed
Email me at v2c@digisonline.com

By: V2C, aka Seed

Email me at v2c@digisonline.com

Tactics Core was written entirely in **Flash 6** Actionscript 1.0. To learn Actionscript, **Flashkit** is a great resource, and there are many experts in their Forums who answer questions very quickly.

Since **Tactics Core** is an engine meant for *developing* games, I wrote it to load almost everything dynamically using object oriented programming methods. Because of this, in the Demo game I had a huge library of movie clips, and a lot of them were linked for Actionscript, and set to export in the first frame. It takes a long time to load everything before even the first frame can be displayed, that's why a separate loading swf is used.



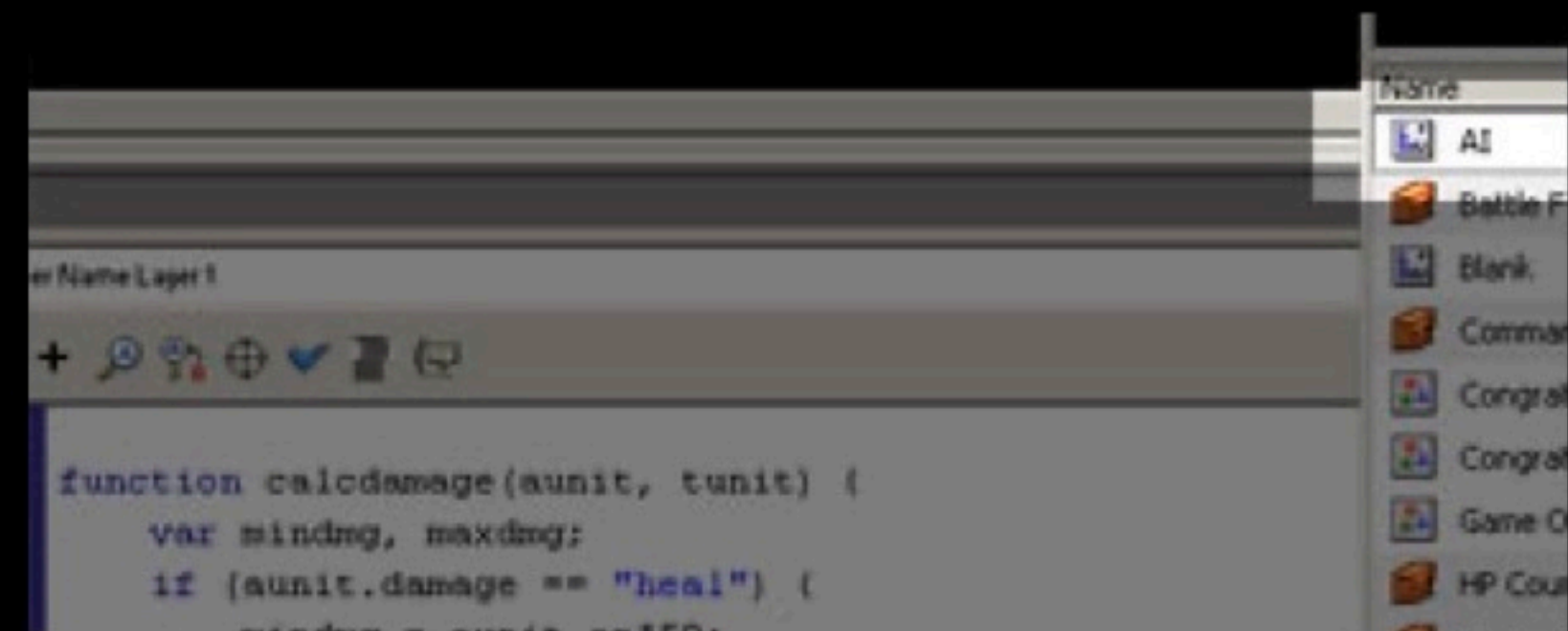
The grid is set with a two-dimensional array which determines whether each space is empty, occupied, blocked, a hole.. etc. This makes it VERY easy to program different stages within minutes! (Excluding the time to make the graphics =P)

Objects are loaded from the library using *attachMovie(...)*, and unloaded using *unloadMovie(...)* after they are done (like battle effects). I found this to be the most efficient method.

Maintaining the depth layering as objects move around and overlap each other is easily achieved with the *swapDepths(...)* function. Just assign each grid space it's own depth, and whenever an object moves onto that space, use *swapDepths(...)* to move that object front/back to the space's assigned depth level. I used a simple algorithm based on the Row and Column in the grid to assign depths. Just remember that you can't have more than one object occupying the same depth level! (The offset in my code below allows me to have two objects in one space.)

```
function setdepth(offset) {  
    if (offset == undefined) offset = 0;  
    swapDepths ( (row+col+1) *100+70-col*2+offset);  
}
```

The A.I. is built using a recursive function that calculates every possible action a unit can take during its turn, and picks the one that is immediately most beneficial. This A.I. algorithm is general to all units; no matter what kind of stats you give to any unit it will function accordingly ^_^v



HOWEVER, this A.I. only thinks for the immediate turn. As a result you may notice that there isn't much *teamwork*. If I add in multiple levels of recursion to think *ahead* a few turns, then the computer will learn to work as a team. Currently that kind of algorithm is too slow to run in **Flash**, so I decided against it ^^"

By: V2C, aka Seed

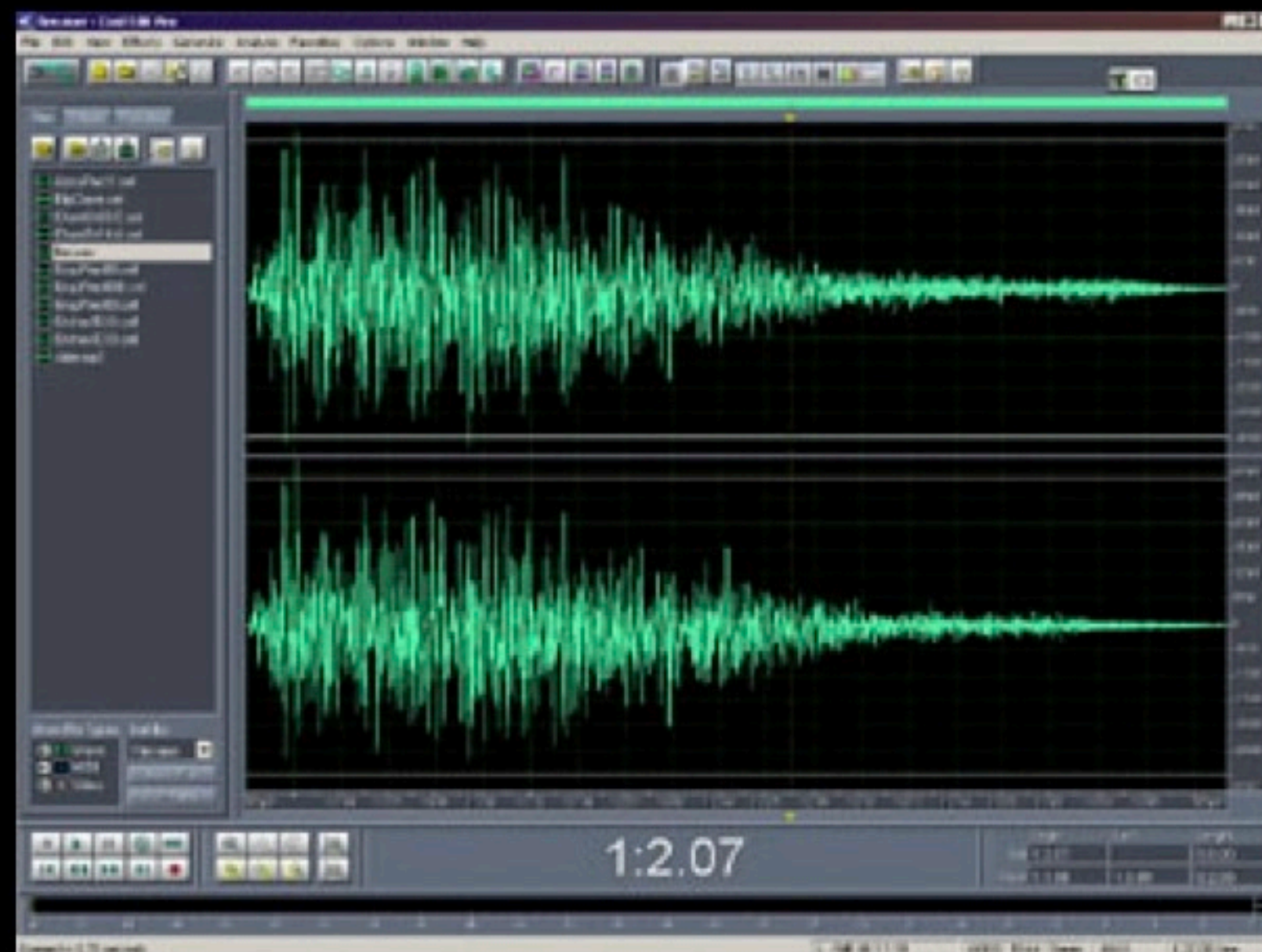
Email me at v2c@digisonline.com

Sound is definitely my weakest area, but I gave it a try anyway. For those of you who want free sounds, I know **Flashkit** has a lot available for download.

I used a program called **Cool Edit Pro 2** to create the BGM and sound effects. A great place for downloading sound clips to use in your **Cool Edit Pro** songs is **Loopology.com**. **Cool Edit Pro** allows you to edit and manipulate multiple tracks at the same time to create high quality song loops.



For the sound effects, I started off by going around recording various sounds I heard onto my MD. Then I imported these into **Cool Edit Pro** and used various filters to alter and clean them up, or even combine sounds.



Some of the sound effects were used in different combinations in the actual **Flash**, and this allowed more variation without increasing the file size ^^.

Well, I guess that's all I have to say now. Sound was the last thing I added, so that's everything I put into **Tactics Core**! Hope it's been interesting =)