MAGNETO

INTRODUCTION

Magneto is a prototype game that I have developed as a proof of concept for animated games running under Windows 3.1. Being a prototype, I am distributing it as shareware. Here are the ground rules:

1.   If you distribute MAGNETO, you must include this README.WRI file.

2.   If you would like updates of MAGNETO, please send your name, address and a small contribution (suggested $10) to:
        Marc-Thomas Clifton
        The Path Maker
        P.O. Box 703
        Cardiff CA 92007
        Windows Developer's BBS: 619-689-2248
        CompuServe ID: 71754, 1204

3.   If you would like to make a suggestion to the game, send your name, address and your suggestion. If I implement the suggestion, I will put your name in the credits and you'll get a free update for your suggestion. If you feel you have a really great suggestion and want a piece of the action, then drop me a note and we'll talk about.

INSTALLATION

Copy the two files, MAGNETO.EXE and README.WRI, to any directory of your making. From Windows, run the MAGNETO.EXE file. Since there are a variety of desktops that people now use (for example, I use NORTON desktop), I am not going to go into details about how to do this!

WHAT IS MAGNETO?

Magneto is a non-violent arcade style game in which you are a magnetic pole on a 2D frictionless playing surface. Also on this surface are other magnetic poles, fixed or moving. You move your pole by controlling its polarity. At the center of the playing field is a transporter that advances you to the next level. The purpose of the game is to rack up as many points as possible by advancing to the next level before the timer expires.

TARGET MACHINES

I developed Magneto on a TI TravelMate 4000 WinSX (for those of you that are curious). This is a 486SX-25 machine. I've noticed that with more than four "sprites" moving around, the game play slows down "slightly". I'd suggest that a 386-40 or a 486-25 are the minimum clock speeds for their respective CPU's. I've spent some time optimizing the code already, but I haven't written anything directly in assembler (it's all in Borland C++ 3.1). At this point, the screen BITBLT routines take the most time, so a graphics accelerator card should help out a lot. Also, the force vector calculations take some time (but less than the screen BITBLT), so a 486DX or a 386 with a math coprocessor will improve game play.

MULTITASKING

This game in interrupt driven at a 25ms interval. Some of the higher levels are just fun to watch, and you can certainly perform other tasks with Magneto running in the background. However, Magneto will bog down whenever Window's ungraciously takes over the system, which appears to be quite often (see Disk Caching).

SCREEN SAVERS

Screen savers are very annoying, and given the newer technology used in color monitors (employed now for several years) they are actually not necessary anymore. I've noticed that Magneto and some screen savers don't interact very well together. For example, some screen savers don't pay attention to mouse clicks. Since Magneto only utilizes the mouse buttons (and not the ball), some screen savers will step on top of the program. Besides, screen savers eat up CPU clocks, and Magneto needs them!

SCREEN FORMAT

Magneto will start up in a maximized window. Everything scales nicely to your window, however, screen performance is severely degraded at resolutions above 640x480, unless you have an accelerator card. If you are running Windows under 800x600, 1024x768, or higher, you can easily improve performance just by going back down to a 640x480 resolution.

DISK CACHING

Both SMARTDRV and LIGHTWIN will cause the game to "jerk" when the driver flushes its write-back cache. This "jerking" persists until the write cache is completely flushed. With both programs, you can tell them to perform their disk writes immediately instead of delaying them. Otherwise, another tactic is to hang out in the "About" box until the driver completes its task.

MAGNETO

The Screen

When you run the game, it opens up a full screen window in which is located the arena. Above the arena is a level counter (left) and point counter(right). Between these two counters are two bars. The top bar displays the available energy in yellow. The bottom bar displays the remaining time in green.

The Arena

The arena contains the objects. At the center of the arena is the transporter pad. At random places around the arena are stationary and moving poles. The first level starts with just one stationary pole. Additional poles are added for each level, in different combinations.

The Poles

You are the BLUE colored pole. The other poles are either GREEN or RED colored. A moving pole will attracted and/or repelled from all the poles, including your own if you are polarized. Once a moving poll collides with another pole, the polarity of the moving pole is set to the same polarity as the pole with which it collided, thus forcing the moving pole to repel rapidly. If a moving pole hits the edge of the arena, it reverses polarity. Your pole is completely in your control and simply bounces off other poles and the edge of the arena.

The Transporter

The transport occupies the center of the screen, and is represented by the blue diamond with the yellow circle in the center. In order to move to the next level, you must direct your pole onto the yellow circle at the center of the transporter diamond. Your pole must be entirely inside the yellow circle for "transport" to occur, however, you can be moving through the transporter at any speed. Originally, I had a speed limit which prevented the transporter from "locking on". I found that this reduced playability, because I didn't employ some of the fancy tricks that you can use to get you moving in the right direction, albeit at a high velocity!

Controlling Your Pole

*** IMPORTANT *** In order to have control over your pole, the mouse pointer must be below the menu bar. If you click on the buttons and nothing happens, check the pointer's position on the screen, and then your energy.

*** ALSO IMPORTANT *** Only the mouse buttons are used in this game, not the mouse "ball". Almost everyone that has played this game assumed that the mouse pointer needed to be moved somewhere! This is not the case. Once you have the mouse pointer positioned below the menu bar, leave it there.

You control the direction your pole travels (and its velocity) by polarizing it either RED or GREEN. Your pole is attracted to other pole's of opposite color, and repelled from pole's of the same color. When your pole is blue, it is neutral. Here are the button combinations and their colors:

Both buttons up	- neutral	BLUE
Left button down	- polarized	GREEN
Right button down	- polarized	RED
Both buttons down	- Brake and Recharge	YELLOW

When your pole is polarized either GREEN or RED, it takes up energy which can be observed by the decreasing yellow energy bar above the arena. Once you run out of energy, your pole remains BLUE until you recharge.

Recharging

You can recharge your pole by converting the magnetic fields of surrounding poles back into energy. Beware though! If you are not sufficiently near another pole, there is no energy gain, and you may be marooned. This can happen only on levels in which there are no moving poles. On levels with moving poles, you just have to wait long enough for a moving pole to come close enough.

Recharging also has the nefarious affect of slowing you down. You can use this to your advantage if you need to slow down, but if you are out of energy and you hit the recharge mode, your forward motion will stop very quickly and you may find that you are not close enough to another pole to recharge!

Orbiting

You will notice that if you attach yourself to a stationary pole, you will be able to orbit it. There is an amusing story behind this. I had wanted this to be a feature anyways, however, as a result of my math optimizations, sufficient error was introduced in the force vector calculations that this affect happened automatically without writing any code!

Orbiting is a nice feature because, as long as you are attracted to the pole, you can circle it. Once you neutralize your pole by releasing all buttons, you will start to move away tangentially from the stationary pole. This is a great way of controlling your direction.

Level Advancement

There are 13 levels in the game. After the thirteen level, the game stays at that level. To advance to the next level, you must control your pole so that it moves into the yellow circle at the center of the blue transporter diamond. This automatically and unceremoniously plops you into the next level. Any time remaining (the green bar below the yellow energy bar) is added to your points. The first level gives you 20 seconds. Each additional level gives you an additional 5

seconds.

BUGS

I have that there are some bugs with the collision detection (objects sometimes pass through each other or into each other halfway before bouncing off). I also have noticed that my collision vector calculations lead to some strange new vectors. Also, the velocity of the colliding poles is not really factored in to the collision, so you will notice that a fast moving poll banging into your slow moving pole does not impart any of its velocity onto you. I actually like this feature, but would like it to be an option!

WHAT'S MISSING

Here are some things that I feel are missing from the game:
  Help
  High scores
  Better collision
  Pole animation
  Transporter animation
  Sound
  Additional "zones"
  Automatic disk cache flushing

CONCLUSION

Well! After reading all that you should be dying to play the game! Hopefully you will enjoy it. I dream of two things (actually, more than two, but these are pretty important): making a game as popular as Tetris (or NYET), and catching the attention of Microsoft or Borland!

"I'll be back!"

Marc-Thomas Clifton