

Reder's A.I. WARS (The Insect Mind)

© 1996-1997 John A. Reder, all rights reserved

What is A.I. Wars?

A.I. Wars allows you to design the **Artificial Intelligence** of an insect like robotic unit (Cybug) and send it into battle to test it's wits against other Cybugs. This game is about programming. There I've said it, yes programming. Now don't get discouraged this is a game, not a computer science course. This game's A.I. Command language is easy to learn, and simple to apply. Some of the most fearsome Cybugs can be created with a just few lines of well thought out code.

A.I.Wars goal is to be fun. This is not an action game that involves hand eye coordination but rather a game that is designed to give you the pleasure every programmer gets when they write a successful program and the excitement of watching your creation pound its competition.

A.I. Wars will give you the basic tools to design your bug and experiment with it in varied battle situations. This game has multiple options to define the layout of each battle simulation. It is designed to encourage friendly competition between A.I. designers with options like encrypted ASCII A.I. files for file trading and a Tournament mode to hold tournaments and contests.

Remember, "Smart bugs never Die!" and "May your A.I. Bugs be without A.I. Bugs!"

How To Play A.I. Wars

Creating an A.I. unit

Some example units are included with A.I. Wars, these units are basic and are designed to let you see how units may be programmed. The scenarios that are included will utilize these example units and give you the basic opponents to pit your home-grown units against. Once your home-grown units are developed to the point where the example units present little challenge, then you are ready to send them to your friends via E-Mail, BBS's, FTP and Web sites to participate in contests and tournaments. (see Tournament Mode.)

1. Click on the Program Cybug button.
2. Enter an A.I. file name in the A.I. file to edit field.
3. Develop the Units AI using the Commands and Syntax A.I. Command language to design the A.I. . (Save your work periodically using the save button.) Note: there is a 1000 line limit on A.I. File length.

Example A.I. Code: This is the “survivor.ai” file’s contents. Notice how simple the code is, one mistake many Cybug developers make is to get lost in sophistication and forget about survival and offense. A good basic strategy and compact code are usually the best ingredients for success. As you can tell by it’s name it’s primary job is to defend and evade.

```
name survivor
iff code x111
author John A. Reder

raise shield
phase1:
long range scan
if scan found enemy then
    gosub killit
end if
scan perimeter
if scan found enemy then discharge energy
if bump barrier then move backward
turn right
if damage is > 1 then attempt repairs
goto phase1

killit:
lower shield
if missile ready then launch missile
raise shield
turn left
move forward
return
```

4. Once you are done select the Save button.

Note: to edit the file after you have left the edit screen you may choose the file to edit by clicking on it’s name in the file list and select the Program Unit A.I. button. This will autoload the A.I. file.

A.I. File Security:

If an A.I. file uses the password command the file will not allow you to edit it without entering the Security Password in the Security Password field.

Preparing for Battle

A battle requires the following options:

- One or more A.I. units in the **Units in Next Battle** List. (The shareware version only allows 5 players in a battle and the registered version allows up to 10.) (also see scenarios.)
- A Map Selected in the **Battle Map** field.
- The **Starting Ammo** option is allows you to set the basic ammo load of the units when the battle starts. Units can gain more ammo after the game starts when they gather flags.
- The **Maximum Damage** field. (an A.I. unit dies when it's damage reaches the maximum damage number. 10 is the maximum for the Shareware version and 99 is the maximum for the registered version.)
- The **Starting Fuel** Field. (A unit burns fuel for movement. Damage to a unit causes the Fuel to burn faster as the fuel cells are damaged. The fuel cells cannot be repaired during battle.) Note: a small amount of fuel burns every so often due to the A.I.Bots need to cool his onboard computer.
- Note: These fields can be populated automatically for you if you select a scenario from the Scenario list. You can save your entries by choosing the **Create Scenario** button. This creates a .SCN file. (see scenarios.)
- Once all Fields have the desired values select the **Start Battle Simulation** button to start the battle.

If the **Show Unit Names** check box is checked the units will have their names next to them. They will have a default name of AI# and their assigned number unless the A.I. file contains the name command. Following their name you will see their ammo, damage and their remaining fuel.

Example: #1 Crusher (15,3,1700)

Sounds

If your PC has the proper hardware and drivers you have the option of having digitized sound effects and or CD background music (using a CD of your choice). Select the **Sound Effects On** and **CD Audio** check boxes before battle begins.

During the Battle

The battle can be stopped by closing it using the **Cancel Battle** button. You are also given the chance to toggle sound effects and unit names on and off using the buttons displayed above the battle view window.

You can also toggle Names, Sounds and Music on and off using the toggle buttons at the top of the display.

End of the battle

The battle ends when all A.I. units run out of fuel or when all but one unit is dead or when the battle is canceled by the user.

Debugging:

You will be able to debug your A.I. unit by clicking on the A.I. file name in the battle list before starting the battle (This will place a red dot inside of a yellow square on the selected unit so you can follow it as it moves through the battle field.. When the unit fires or launches it's weapons, enables shields or scans you will see the actions represented in the battles graphics. If you selected the **Debug Mode for Unit** check box the debug data will be displayed in battle including the current line of code being processed by the selected unit. When Debug is on, the battle can be slowed down using the scrollbar at the top of the display so you can have time to analyze the debug information. A special map has been created called **debug.map** to help you see the data better. This map will not allow units and graphics to enter the debug display area. Note: You may use any map but the debug map is recommended.

Registered Users will have the additional debugging commands and options:

You may place the commands **debug on** and **debug off** in your A.I. code. When activated this will save any code that the unit sees into an area called the **Debug Watch buffer**. The buffer will display any commands that the unit sees and its reactions to the commands including error and warning messages about possible A.I. Command language errors. You can view the buffer contents by viewing the **debug watch** window from button in the battle summary screen.

Note: The bug must be tagged with the red dot by clicking on the Cybugs name in the battle list found in the battle setup screen.

Example Debug Watch Buffer Information for HULKV1.AI:

```
level1:
begin:
math v3 = 10 - 1
assign v3 9
if value 0 = 9 then self destruct
if facing south then turn right
>>> condition met executing command: turn right
turn right
if facing east then turn left
if facing west then turn right
>>> condition met executing command: turn right
turn right
if x coordinate is < 17 then turn right
if x coordinate is > 17 then turn left
>>> condition met executing command: turn left
turn left
start:
if bump barrier then generate random
if bump barrier then if random is 1 then turn right
if bump barrier then if random is 2 then turn left
if bump barrier then if random is 3 then turn right
if bump barrier then if random is 3 then move forward
if bump barrier then end if
if bump barrier then if random is 4 then turn left
if bump barrier then if random is 4 then move forward
if bump barrier then end if
if bump barrier then move forward
end if
if x coordinate is < 19 then math v0 = 19 - 21
if x coordinate is > 19 then math v0 = 21 - 19
>>> condition met executing command: math v0 = 21 - 19
math v0 = 21 - 19
assign v0 2
long range scan
>>> scan identified: enemy
if scan found flag then if damage = 0 then goto avoid
if scan found mine then goto avoid
assign v1 0
moveloop:
move forward
if bump barrier then goto start
>>> condition met executing command: goto start
goto start
start:

if bump barrier then generate random
>>> condition met executing command: generate random
generate random
```

```

if bump barrier then if random is 1 then turn right
>>> condition met executing command: if random is 1 then turn right
if random is 1 then turn right
if bump barrier then if random is 2 then turn left
>>> condition met executing command: if random is 2 then turn left
if random is 2 then turn left
if bump barrier then if random is 3 then turn right
>>> condition met executing command: if random is 3 then turn right
if random is 3 then turn right
if bump barrier then if random is 3 then move forward
>>> condition met executing command: if random is 3 then move forward
if random is 3 then move forward
if bump barrier then end if
>>> condition met executing command: end if
end if

if bump barrier then if random is 4 then turn left
>>> condition met executing command: if random is 4 then turn left
if random is 4 then turn left
>>> condition met executing command: turn left
turn left
if bump barrier then if random is 4 then move forward
>>> condition met executing command: if random is 4 then move forward
if random is 4 then move forward
>>> condition met executing command: move forward
move forward
if bump barrier then end if
if bump barrier then move forward
end if
if x coordinate is < 20 then math v0 = 20 - 21
if x coordinate is > 20 then math v0 = 21 - 20
>>> condition met executing command: math v0 = 21 - 20
math v0 = 21 - 20
assign v0 1
long range scan
>>> scan identified: barrier
if scan found flag then if damage = 0 then goto avoid
if scan found mine then goto avoid
assign v1 0
moveloop:
move forward
if bump barrier then goto start
math v1 = 0 + 1
assign v1 1
if value 1 <> 1 then goto moveloop
skipit:
if y coordinate is = 14 then if x coordinate is > 20 then if facing west turn left
if y coordinate is = 14 then if x coordinate is > 20 then if facing north then turn right
if y coordinate is = 14 then if x coordinate is > 20 then if facing south then turn left
if y coordinate is = 14 then end if
if y coordinate is = 14 then if x coordinate is < 20 then if facing east turn left
if y coordinate is = 14 then if x coordinate is < 20 then if facing north then turn left
if y coordinate is = 14 then if x coordinate is < 20 then if facing south then turn right
if y coordinate is = 14 then end if
if y coordinate is = 14 then goto start3
end if
if x coordinate is = 20 then goto start2
end if
if y coordinate is = 14 then goto start2
end if

goto begin
begin:
math v3 = 10 - 1
assign v3 9
if value 9 = 9 then self destruct
>>> condition met executing command: self destruct
self destruct

```

After the Battle

The Battle will display the battle statistics. They give a description of the battle settings, a play by play description of the of the battle and the unit standings and scores.

You can **Print** this or **Save** this to a file from this screen.

You can view this screen later using the **Show Battle Summary** button from the main battle setup screen.

Example: Battle Summary

Battle number: 1
Starting Ammo: 30
Maximum damage before death: 10
Starting Fuel Value: 2500
Battle Map: OPEN.MAP

Battle Results:

```
-----  
1st Place   #2 slammer  Damage: 1  Fuel: 2046  Score: 5099  John A. Reder  
2nd Place   #5 hulkv1   Damage: 3  Fuel: 1862  Score: 3996  John A. Reder  
3rd Place   #1 drone5   Damage: 12 Fuel: 0     Score: -13   John A. Reder  
4th Place   #3 tracker  Damage: 18 Fuel: 0     Score: -3419 John A. Reder  
5th Place   #4 mlauncher Damage: 18 Fuel: 0     Score: -4019 John A. Reder
```

Battle play by play:

```
-----  
Click: 29 #3 tracker is crushed by #1 drone5's missile direct hit doing 9 damage!  
Click: 38 #4 mlauncher is crushed by #2 slammer's missile direct hit doing 9 damage!  
Click: 39 #1 drone5 protected by shield is shaken by #3 tracker's missile direct hit doing 6  
damage!  
Click: 59 #1 drone5 protected by shield is shaken by #3 tracker's missile direct hit doing 6  
damage!  
Click: 60 #1 drone5 is killed! ...Memorial flag planted!  
Click: 65 #3 tracker is crushed by #1 drone5's missile direct hit doing 9 damage!  
Click: 65 #3 tracker is killed! ...Memorial flag planted!  
Click: 218 #4 mlauncher is crushed by #5 hulkv1's missile direct hit doing 9 damage!  
Click: 219 #4 mlauncher is killed! ...Memorial flag planted!  
Click: 466 #2 slammer discharged energy doing 1 point of damage to itself!  
Click: 466 #2 slammer zaps #5 hulkv1 with energy discharge inflicting 3 points of damage!  
Click: 1086 Battle Terminated by User!
```

```
-----  
End of Battle sequence  
-----
```

Scoring:

Units get points for inflicting damage on other units and keeping their damage low.

Scenarios

Scenarios allow you to select from a list of pre-selected battle configurations. These configurations include a group of Cybugs, a battle map and preset ammo, damage and fuel settings. You may also have a pre selected MIDI music title. A.I.Wars comes with some default scenarios to give you some basic battle settings to test your Cybugs in.

You may save any battle configuration by choosing the **Create Scenario** button.

Music Options

You are given 3 Music options from the battle setup screen.

4. MIDI (this plays MIDI files located in the A.I.Wars home directory.)
5. CD Audio (This plays any CD you have in your CD Rom Drive.)
6. No Music

Select which ever option you desire. The select Music will play during the battle simulation mode.

The Map Editor

You can Edit Maps to be used by the Battle Simulator by selecting the **Map editor** button.

The editor has a pull down window that allows you to select the graphic that you want to add to the map. Use the mouse to place the graphics.

Note: the unit start graphic depicts where the units will start when the map is used in battle. Only the first 10 will be used. If starting positions are not created then the battle simulator will use default starting locations.

From the pull down window you can **save**, **clear the map** or **exit the editor**.

Note: Only the **Registered version** will allow you to save a map created by the editor.

Tournament Mode

Registered version only. This mode allows you to play all A.I. unit files in the games directory (up to 500 A.I. files) against each other. The final results will be displayed in the battle statistics screen at the end and the entire statistics and pay by play readout will be saved to a file named **contest.txt** (you have the option to rename this file in the tournament screen). You could print or E-Mail this file to the authors of the A.I. files.

Note: You are given the option to disable IFF Codes during tournaments. This disables the Cybugs ability to recognize friendly units.

Feel free to create internet and BBS sites that have A.I. Wars file trading and tournament areas! If you hold a tournament and you want me to add a link to it from my web page just E-Mail me the URL and I will post it on my Official A.I. Wars page.

Note: An A.I. file doesn't have to be prepared in a registered version of A.I. Wars to participate in tournaments.

Example Tournament Mode Output:

Note: Actual play by plays are stored into a text file defined in the tournament setup screen.

Battle Results:

```
-----  
1st Place      #1 drone1   Damage: 0 Fuel: 0      Score: 5271   John A. Reder  
2nd Place      #4 slammer  Damage: 1 Fuel: 0      Score: 5079   John A. Reder  
3rd Place      #5 drone3   Damage: 5 Fuel: 0      Score: 2814   John A. Reder  
4th Place      #2 drone2   Damage: 5 Fuel: 0      Score: 2594   John A. Reder  
5th Place      #6 scanner  Damage: 5 Fuel: 0      Score: 2529   John A. Reder  
6th Place      #9 swarm4   Damage: 10 Fuel: 0     Score: 117    John A. Reder  
6th Place      #10 swarm1  Damage: 10 Fuel: 0     Score: 117    John A. Reder  
7th Place      #7 swarm2   Damage: 10 Fuel: 0     Score: 41     John A. Reder  
8th Place      #8 swarm3   Damage: 12 Fuel: 0     Score: -933   John A. Reder  
9th Place      #3 destruct Damage: 12 Fuel: 0     Score: -1013  John A. Reder
```

Battle Results:

```
-----  
1st Place      #1 drone4   Damage: 0 Fuel: 0      Score: 6599   John A. Reder  
2nd Place      #2 drone5   Damage: 0 Fuel: 0      Score: 4999   John A. Reder  
3rd Place      #7 hulkv1   Damage: 10 Fuel: 0     Score: 490    John A. Reder  
4th Place      AI #8       Damage: 10 Fuel: 0     Score: -1     Author Unknown  
5th Place      #4 lrscanne Damage: 12 Fuel: 0     Score: -513   John A. Reder  
5th Place      #9 drone6   Damage: 12 Fuel: 0     Score: -513   John A. Reder  
6th Place      #10 TheFrog Damage: 13 Fuel: 0     Score: -614   MikeSlaunwhite  
7th Place      #5 nester   Damage: 12 Fuel: 0     Score: -1013  John A. Reder  
8th Place      #3 mlaunche Damage: 12 Fuel: 0     Score: -1022  John A. Reder  
9th Place      #6 tracker  Damage: 18 Fuel: 0     Score: -3719  John A. Reder
```

Battle Results:

```
-----  
1st Place      #2 drone1   Damage: 3 Fuel: 279    Score: 3889   John A. Reder  
2nd Place      #1 chaser   Damage: 10 Fuel: 0     Score: 9      Colin McConnell  
3rd Place      #3 drone4   Damage: 11 Fuel: 0     Score: -411   John A. Reder
```

Note: In this example Drone 1 took First Place, Chaser took Second Place and Drone 4 took Third Place.

Play By E-Mail , File Copy or Transfer

The tournament mode is a great way to hold contests for all of your friends A.I. files. You can have them give you their A.I. files on disk or have them E-Mail them to you. The A.I. files are encoded but they are pure Ascii text so you can just E-Mail the text and cut and paste it into a file with the .AI extension.

Note: each line after the word start can not contain more than 15 encoded numbers. and must be copied and pasted exactly as they appear in the A.I. file.

You may use the **View Encoded Version** button in the A.I. editor develop mode screen to view the A.I. file in its encoded form. You may cut and paste this into your E-Mail editor for distribution if you wish.

Example ASCII encoded AI file: (tracker.ai)

You can place any comments you like here before the word start. This text is striped from the file if it is edited.

```
<start>
178
114 101 113 105 36 120 118 101 103 111 105 118 17 14 101
121 120 108 115 118 36 78 115 108 114 36 69 50 36 86
105 104 105 118 17 14 17 14 102 105 107 109 114 62 17
14 109 106 36 106 101 103 109 114 107 36 119 115 121 120
108 36 120 108 105 114 36 120 121 118 114 36 118 109 107
108 120 17 14 109 106 36 106 101 103 109 114 107 36 105
101 119 120 36 120 108 105 114 36 120 121 118 114 36 112
105 106 120 17 14 109 106 36 106 101 103 109 114 107 36
123 105 119 120 36 120 108 105 114 36 120 121 118 114 36
118 109 107 108 120 17 14 109 106 36 124 36 103 115 115
118 104 109 114 101 120 105 36 109 119 36 64 36 39 105
114 105 113 125 99 124 36 120 108 105 114 36 120 121 118
114 36 118 109 107 108 120 17 14 109 106 36 124 36 103
115 115 118 104 109 114 101 120 105 36 109 119 36 66 36
39 105 114 105 113 125 99 124 36 120 108 105 114 36 120
121 118 114 36 112 105 106 120 17 14 17 14 119 120 101
118 120 62 17 14 113 115 122 105 36 106 115 118 123 101
118 104 17 14 109 106 36 102 121 113 116 36 102 101 118
118 109 105 118 36 120 108 105 114 36 120 121 118 114 36
118 109 107 108 120 17 14 109 106 36 124 36 103 115 115
118 104 109 114 101 120 105 36 109 119 36 65 36 39 105
114 105 113 125 99 124 36 120 108 105 114 36 107 115 120
115 36 119 120 101 118 120 54 17 14 107 115 120 115 36
102 105 107 109 114 17 14 17 14 119 120 101 118 120 54
62 17 14 109 106 36 125 36 103 115 115 118 104 109 114
101 120 105 36 109 119 36 64 36 39 105 114 105 113 125
99 125 36 120 108 105 114 36 17 14 13 109 106 36 106
101 103 109 114 107 36 123 105 119 120 36 120 108 105 114
36 120 121 118 114 36 112 105 106 120 17 14 13 109 106
36 106 101 103 109 114 107 36 105 101 119 120 36 120 108
105 114 36 120 121 118 114 36 118 109 107 108 120 17 14
105 114 104 36 109 106 17 14 109 106 36 125 36 103 115
115 118 104 109 114 101 120 105 36 109 119 36 66 36 39
105 114 105 113 125 99 125 36 120 108 105 114 36 17 14
13 109 106 36 106 101 103 109 114 107 36 123 105 119 120
36 120 108 105 114 36 120 121 118 114 36 118 109 107 108
120 17 14 13 109 106 36 106 101 103 109 114 107 36 105
```

```
101 119 120 36 120 108 105 114 36 120 121 118 114 36 112
105 106 120 17 14 105 114 104 36 109 106 17 14 17 14
119 120 101 118 120 55 62 17 14 112 115 114 107 36 118
101 114 107 105 36 119 103 101 114 17 14 109 106 36 119
103 101 114 36 106 115 121 114 104 36 105 114 105 113 125
36 120 108 105 114 36 112 101 121 114 103 108 36 113 109
119 119 109 112 105 17 14 107 115 120 115 36 102 105 107
109 114 17 14 17 14 17 14 17 14 17 14 17 14 17
14 17 14 17 14 17 14 17 14 17 14 17 14 17 14
17 14 17 14 17 14 17 14
```

Check my web site for additional AI files that you can cut and paste for use in this game. I'm encouraging players to develop swap areas to E-mail and distribute A.I. files freely. Any A.I file that is sent to my E-mail address may be posted on my web page for other players to try. See the **How to register screen** for my WEB page address. My E-mail address will be posted on this web site along with any updates to A.I. Wars.

Note: the map files, scenario and A.I. files are all in ASCII text formats so you can transfer them the same way as the A.I. files.

Credits

This entire program was written and developed by me: **John A. Reder**.

Please send any comments, bugs reports and suggestions to my E-mail address posted on my web page.

Registration

To learn how to register click on the How to Register button.

Happy Hunting!