***BuildTheHTMLPage TeamID: PNT2022TMID29172***

<!DOCTYPEHTMLPUBLIC"-

//W 3C//DTDHTML4.01//EN"

"<http://www.w3.org/TR/html4/strict.dtd>" >

<html>

<head>

<meta http- equiv=" Content-

Type" content=" text/html; charset=UTF- 8" >

<title>Artificial intelligence : OpenKore sourcecodedocumentation</title>

<link rel=" stylesheet"

type=" text/css" href=" openkore.css" >

<!- - FixbrokenPNGtransparencyforIE/Win5- 6+-

- >

<!- - [ ifgteIE5.5000] >

<script

type=" text/javascript" src=" pngfix.js" ></scri pt>

<![ endif]- - >

<styletype=" text/css" >

<!- -

.example { margin:

0.3cm; margin- left:0.5cm;

}

.comment{font-

style:italic;

}

.term { border- bottom: 1px

dottedblack;

}

.cstr{color:

#007700;

}

- - >

</style>

</head>

<body>

<divid=" title" >OpenKoresourcecodedocumentation</div>

<divid=" navigation" >

<ul>

<li><ahref="<http://openkore.sourceforge.net/>">Mainwe bsite</a></li>

<li><a href=" index.html" >Table ofcontents</a></li>

<li><b>Artificialintelligence</b></li>

</ul>

</div>

<divid=" main" >

<h1> HowtheAIsubsystemisdesigned</h1>

The AI subsystem isn't really complex, but it could takeawhileto understandit'sdesign.

<p>

All" intelligence" ishandledinsidethe

<code>AI()</code> function (right now it's one bigfunctionbutwehopetosplititinthefuture).

As explained in the <a>Main loop & amp; initialization</a>page, the <code>AI()</code> function only runs less thanafractionofasecond.

<p>

Basically, theAItellsKoretodocertainthingsbasedon the current situation. I'll try to explain it withsomeexamples.

<aname=" ex1" ></a>

<h2>Example1:Randomwalk< /h2>

You'reprobablyfamiliarwithKore'srandomwalkfeature.

If there are no monsters and Kore isn't doing anything, it will walk to a random spot on the map, and attack

anymonstersitencounters.

The following piece of code (within the

<code>AI()</code>function makes Kore walk to a random spot if it isn'tdoing anything:

<preclass=" example" >

1. *<span class=" comment" >##### RANDOM WALK#####</span>*
2. *<b>if</b>($config{'route\_randomWalk'}&&*

$ai\_seq[ 0]

<b>eq</b>""&&@{$field{'field'}}> 1& &

!$cities\_ lut{$field{'name'}.'.rsw'}){

1. *<span class=" comment" ># Find a randomblock on the map that we can walkon</span>*

4 <b>do</b>{

1. *$ai\_v{'temp'}{'randX'} = int(rand()*

\*($field{'width'}- 1));

1. *$ai\_v{'temp'}{'randY'} = int(rand()*

\*($field{'height'} - 1));

7 }

<b>while</b>($field{'field'}[ $ai\_v{'temp'}{'randY'}\*$field{'width'}+

$ai\_v{'temp'}{'randX'}]);8

1. *<span class=" comment" ># Move to thatblock</span>*
2. *message <span*

class=" cstr" >" Calculatingrandom routeto:

$maps\_lut{$field{'name'}.'.rsw'}($field{'name'}):

$ai\_v{'temp'}{'randX'}, $ai\_v{'temp'}{'randY'}\n" </span>,

<spanclass=" cstr" >" route" </span>;

1. *ai\_ route(\% {$ai\_v{'temp'}{'returnHash'}},*
2. *$ai\_v{'temp'}{'randX'},*
3. *$ai\_v{'temp'}{'randY'},*
4. *$field{'name'},*

15 0,

16 $config{'route\_randomWalk\_maxRouteTime'},

17 2,

1. *undef,*
2. *undef,*

20 1);

21 }

</pre>

We call this block of code an <em class=" term" >AI codeblock</em>.

In other words, an AI code block is <em>an entire blockofcodewhichdealswithacertainpartof theAI</em>.

<h3>Situation

check</h3>Inline1, it checks:

<ol>

<li>whethertheconfigurationoption

<code>route\_randomWalk</code>ison</li>

<li>whether there are currently no other active

<emclass=" term" >AIsequences</em>(seebelow)</li>

<li>whetherwe'recurrentlyNOTinacity</li>

</ol>

If all of the above is true, then Kore will run the codeinside thebrackets.

<p>

What is an <em class=" term" >AI sequence</em>? It is avaluewithinthe<code>@ai\_seq</code> array.

Thisarrayisa<em>commandqueue</em>.

<p>

AI code blocks prepend values into this array so they canknowwhen it'stheirturntodosomething.

When an AI code block is done with it's task, it willremove thatvaluefrom thearray.

So, if <code>@ai\_seq</code> is empty, then that means allAI code blocks have finished and Kore isn't doinganythingelse.

AndthisiswhentherandomwalkAIcodeblock jumpsin.

<p>

There is also the <code>@ai\_seq\_args</code> array, usedto store temporary variables used by the current AI codeblock.

If a value is prepended into <code>@ai\_seq</code>, then avalue mustalsobeprependedinto

<code>@ai\_seq\_args</code>.Mo reonthislater.

<h3>Findingarandompositiontowalkto</h3>

Line 4- 7 tries to find a random position in the map thatyoucanwalkon.

(<code>$field{field}</code> is a reference to an arraywhichcontainsinformationaboutwhichblocksyoucanandcan't walkon.

But that's not important in this example. You just havetounderstand what thisblockdoes.)

<p>

Theresultcoordinateisputintothesetwovariables:

<ul>

<li><code>$ai\_v{temp}{randX}</code></li>

<li><code>$ai\_v{temp}{randY}</code></li>

</ul>

<small>(In case you didn't know,

<code>$foo{bar}</code>isthesameas<code>$foo{'bar'}</code

>.)</small>

<h3>Moving</h3>

Line11- 20isthecodewhichtellsKoretomovetotherandom position.

Ittells<code>ai\_route()</code>whereitwants togoto.

<code>ai\_route()</code> prepends a <code>" route" </code>AI sequence in <code>@ai\_seq</code>, and arguments in ahash

(which is then prepended into

<code>@ai\_seq\_args</code>andimmediatelyreturns.

Shortly after this, the entire <code>AI()</code> functionreturns. The point is, <code>ai\_route()</code> is

<em>notsynchronous</em>.

<p>

Inlessthanafractionofasecond, the

<code>AI()</code>functioniscalledagain.

Because the <code>@ai\_seq</code> variable is not

emptyanymore, therandomwalkAIcodeblockisnever activated (the expression <code>'$ai\_seq[ 0] eq ""'</code> isfalse).

<p>

The AI code block that handles routing is elsewhere inthe<code>AI()</code>function.

Itseesthatthefirstvaluein<code>@ai\_seq</code>is

<code>" route" </code>, andthinks<em>" hey, nowit'smyturntodo something!" </em>.

(The route AI code block is very complex so I'm not goingtoexplain whatitdoes, butyougettheidea.)

When the route AI code block has finished, it will removethefirst item from<code>@ai\_seq</code>.

If <code>@ai\_seq</code> is empty, then the random routeAIcodeblock isactivatedagain.

<h2>Example 2: Attacking monsters while walking to arandom spot</h2>

YoumightwanttowonderhowKoreisabletodeterminewhetherto attack monsterswhenit'swalking.

Let'stakealook atasmallpiece ofit'ssourcecode:

<preclass=" example" >

<spanclass=" comment" >#####AUTO- ATTACK#####</span>

<b>if</b> (($ai\_seq[ 0] <b>eq</b>

<spanclass=" cstr" >"" </span> || $ai\_seq[ 0] <b>eq</b>

<spanclass=" cstr" >" route" </span> || $ai\_seq[ 0] <b>eq</b>

<spanclass=" cstr" >" route\_getRoute" </span>||$ai\_seq[ 0]

<b>eq</b><spanclass=" cstr" >" route\_getMapRoute" </span>

|| $ai\_seq[ 0] <b>eq</b>

<spanclass=" cstr" >" follow" </spa n>

|| $ai\_seq[ 0] <b>eq</b>

<spanclass=" cstr" >" sitAuto" </span>||$ai\_seq[ 0] <b>eq</b>

<spanclass=" cstr" >" take" </span>||$ai\_seq[ 0] <b>eq</b>

<spanclass=" cstr" >" items\_gather" </span>||$ai\_seq[ 0]

<b>eq</b><spanclass=" cstr" >" items\_take" </span>)

...

</pre>

As you can see here, the auto- attack AI code block is runifany of theaboveAI sequencesareactive.

So when Kore is walking (<code>$ai\_seq\_args[ 0] </code> is" route"), Korecontinuestocheckformonsterstoattack.

<p>

Butasyoumayknow, ifyoumanuallytype" moveWhateEverMapNam e" intheconsole, Korewillmovetothatmapwithoutattacking

monsters (yes, this is intentional behavior). Why isthat?

<p>

As seen in example 1, the

<code>ai\_route()</code>functioninitializesthe routeAIsequence.

Thatfunctionacceptsaparametercalled" attackOnRoute".

<code>$ai\_seq\_args[ 0]{attackOnRoute}</code> is set to thesamevalueas thisparameter.

Kore will only attack monsters while moving, if thatparameter issetto1.

Whenyoutype" move" intheconsole, thatparameterissetto 0. The random walk AI code block however sets thatparameter to1.

<p>

Inside the auto- attack AI code block, Kore checks whetherthe

argument hash that's associated with the " route" AIsequencehasa 'attackOnRoute'key, andwhetherthevalueis1.

<preclass=" example" >

...

$ai\_v{'temp'}{'ai\_route\_index'}=binFind(\@ai\_seq,

<spanclass=" cstr" >" route" </span>);

<b>if</b> ($ai\_v{'temp'}{'ai\_route\_index'} ne

<spanclass=" cstr" >"" </span>){

$ai\_v{'temp'}{'ai\_route\_attackOnRoute'}=

$ai\_seq\_args[ $ai\_v{'temp'}{'ai\_route\_index'}]{'attackOnRoute'};

}

...

<span class=" comment" ># Somewhere else in the auto- attackAI code block, Korechecks whether

# $ai\_v{'temp'}{'ai\_route\_attackOnRoute'} is set to1.< /span>

</pre>

<h2>Timeouts:Towaita whilebeforedoingsomething</h2>

In certain cases you may want the program to wait a whilebefore doinganythingelse.

Forexample, youmaywanttosenda" talktoNPC" packettotheserver, th ensenda" chooseNPCmenuitem2" packet

2secondslater.

<p>

Thefirstthingyouwouldthinkof isprobably tousethe

<code>sleep()</code>function.

However, that is a bad idea. <code>sleep()</code> blocksthe

entire program. During the sleep, nothing else can beperformed.

User command input will not work, other AI sequences arenotrun, networkdataisnotreceived, etc.

<p>

The right thing to do is to use the

<ahref="Utils.html#timeOut"><code>timeOut()</code></a>function.

The API documentation entry for that function has

twoexamples.Here'sanotherexample, demonstratinghow

you can use the timeOut() function in an AI sequence.ThisexampleinitializesaconversationwithNPC1337(aKa pra NPC).

Then two seconds later, it sends a " choose NPC menu item2" packet.

<preclass=" example" >

<span class=" comment" ># The AI() function is run in themainloop</span>

<b>sub</b>AI{

...

<b>if</b>($somethingHappened){

<b>my</b>% args;

$args{stage}=<spanclass=" cstr" >'Just

started'</span>;

<b>unshift</b> @ai\_seq,

<spanclass=" cstr" >" NpcExample" </span>;

<b>unshift</b>@ai\_seq\_args,\% args;

$somethingHappened=0;

}

<b>if</b> ($ai\_seq[ 0] <b>eq</b>

<spanclass=" cstr" >" NpcExample" </span>){

<b>if</b>($ai\_seq\_args[ 0]{stage}

<b>eq</b><spanclass=" cstr" >'Juststarted'</span>){

<spanclass=" comment" >#ThisAI

sequencejuststarted

#Initializeaconversationwith

NPC1337< /span>

sendTalk($net, 1337);

<span class=" comment" ># Store thecurrenttimeinavariable</span>

$ai\_seq\_args[ 0]{waitTwoSecs}{time}=<b>time</b>;

<span class=" comment" ># We wanttowaittwoseconds</span>

$ai\_seq\_args[ 0]{waitTwoSecs}{timeout}=2;

$ai\_seq\_args[ 0]{stage} =

<spanclass=" cstr" >'Initializedconversation'</span>;

}<b>elsif</b>($ai\_seq\_args[ 0]{stage}

<b>eq</b> <span

class="cstr">'Initializedconversation'</span>

<span class=" comment" ># This 'if'statementisonlytrue iftwo seconds havepassed

#since

$ai\_seq\_args[ 0]{waitTwoSecs}{time}isset</span>

&& timeOut(

$ai\_seq\_args[ 0]{waitTwoSecs})

){

<span class=" comment" ># Twosecondshavenowpassed</span>

sendTalkResponse($net, 1337, 2);

<span class=" comment" ># We'redone; removethis AIsequence</span>

<b>shift</b>@ai\_seq;

<b>shift</b>@ai\_seq\_args;

}

}

...

}

</pre>

<h2>Conclusion& amp; summary</h2>

The entire AI subsystem is kept together by these twovariables:

<ul>

<li><code>@ai\_seq</code> : a queue which contains AIsequencenames.

Usually, AI code blocks are run based on the value of thefirst itemin thequeue

(though this doesn't have to be true; it depends on howtheAI codeblockisprogrammed).</li>

<li><code>@ai\_seq\_args</code> : contains arguments that'sassociated withcurrentAIsequence.</li>

</ul>

The design is pretty simple. This allows the system to beveryflexible:

you can do pretty much anything you want. There aren'tmanyreallimitations

(butthat'sjustmyopinion).

<p>

The <code>AI()</code> function runs only very shortly. SoAI code blocks shouldn't do anything that can block

thefunctionforalongtime.

<h3>Glossary</h3>

<ul>

<li>An <em class=" term" >AI code block</em> is an entireblock of code which deals with a certain part of theAI.</li>

<li>An <em class=" term" >AI sequence</em> is a valuewithinthe<code>@ai\_seq</code>queue(andanassociatedv alueinsidethe<code>@ai\_seq\_args</code>array).</li>

</ul>

<p><hr><p>

<divid=" footer" >

<ul>

<li><ahref=" <http://validator.w3.org/check?uri=referer>" title=" Valid HTML

4.01!" ><imgsrc=" <http://www.w3.org/Icons/valid->html401" alt=" ValidHTML4.01!" height=" 31" width=" 88" ></a></li>

<li><ahref=" <http://www.mozilla.org/products/firefox/>" title

=" Get Firefox - Take Back the Web" ><img

width=" 104" height=" 32" src="<http://www.mozilla.org/products/firef> ox/buttons/getfirefox\_small.png" alt=" Get Firefox - Take Back

theWeb" ></a></li>

<li><ahref=" <http://www.mozilla.org/products/firefox/>" title=

" If

you were looking at thispagein any browser butMicrosoft Internet Explorer, it would look and run betterand faster" ><img width=" 45" height=" 45" src=" <http://linuxart.com/img/noIE->small.png" alt=" If

youwere looking at this page in any browser but MicrosoftInternet Explorer, it would look and run better andfaster" ></a></li>

</ul>

</div>

</div>

</body>

</html>