**BuildTheHTMLPage Team*ID: PNT2022TMID44952***

<!DOCTYPEHTMLPUBLIC"-

//W3C//DTDHTML4.01//EN"

"<http://www.w3.org/TR/html4/str>ict.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"></script>

<![endif]-->

<styletype="text/css">

<!--

.example {margin:0.3cm;

margin-left:0.5cm;

}

.comment{

font-style:italic;

}

.term{

border-bottom:1pxdottedblack;

}

.cstr{

color:#007700;

}

-->

</style>

</head>

<body>

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

<divid="navigation">

<ul>

<li><ahref="<http://openkore.sourceforge.net/>">Mainwebsite</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 take awhile tounderstandit'sdesign.

<p>

All"intelligence"ishandledinsidethe

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

wehopetosplititinthefuture).

Asexplainedinthe<a>Mainloop&amp;initialization</a>page, the <code>AI()</code> function only runs less thanafractionofasecond.

<p>

Basically, the AI tells Kore to do certain things based onthe current situation. I'll try to explain it with someexamples.

<aname="ex1"></a>

<h2>Example1:Randomwalk</h2>

You'reprobablyfamiliarwithKore'srandomwalkfeature.

If there are no monsters and Kore isn't doing anything, itwill 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 random blockonthe map thatwe can walkon</span>

4 <b>do</b>{

1. $ai\_v{'temp'}{'randX'} = int(rand() \*($field{'width'}-1));
2. $ai\_v{'temp'}{'randY'} = int(rand() \*($field{'height'} -1));
3. } <b>while</b>($field{'field'}[$ai\_v{'temp'}{'randY'}\*$field{'width'}+

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

1. <spanclass="comment">#Movetothatblock</span>
2. message <span class="cstr">"Calculating randomroute to:

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

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

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

|  |  |
| --- | --- |
| 11 | ai\_route(\%{$ai\_v{'temp'}{'returnHash'}}, |
| 12 | $ai\_v{'temp'}{'randX'}, |
| 13 | $ai\_v{'temp'}{'randY'}, |
| 14 | $field{'name'}, |
| 15 | 0, |
| 16 | $config{'route\_randomWalk\_maxRouteTime'}, |
| 17 | 2, |
| 18 | undef, |
| 19 | undef, |
| 20 | 1); |

21}

</pre>

Wecallthisblockofcodean <emclass="term">AIcodeblock</em>.

In other words, an AI code block is <em>an entire block ofcodewhich dealswithacertainpartoftheAI</em>.

<h3>Situationcheck</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>

Whatisan<emclass="term">AIsequence</em>?Itisavaluewithinthe<code>@ai\_seq</code> array.

Thisarrayisa<em>commandqueue</em>.

<p>

AIcodeblocksprependvaluesintothisarraysotheycanknowwhen it'stheirturntodosomething.

WhenanAIcodeblockis donewith it'stask,it willremove thatvaluefrom thearray.

So,if<code>@ai\_seq</code>isempty,thenthat meansallAI code blocks have finished and Kore isn't doinganythingelse.

AndthisiswhentherandomwalkAIcodeblock jumpsin.

<p>

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

Ifavalueisprepended into<code>@ai\_seq</code>,thenavalue mustalsobeprependedinto

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

<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 have tounderstand whatthisblockdoes.)

<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.

Shortlyafterthis,theentire<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,therandomwalkAIcodeblockisneveractivated

(theexpression<code>'$ai\_seq[0]eq""'</code> isfalse).

<p>

TheAIcodeblockthathandlesroutingiselsewhereinthe

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

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

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

(TherouteAIcodeblock isverycomplexsoI'mnotgoingtoexplain whatitdoes, butyougettheidea.)

WhentherouteAIcode blockhasfinished,itwillremovethefirst 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"</span>

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

But as you may know, if you manually type "moveWhateEverMapName" in the console, Kore will move to thatmapwithoutattacking

monsters(yes,thisis intentionalbehavior).Whyisthat?

<p>

As seen in example 1, the <code>ai\_route()</code> functioninitializestherouteAIsequence.

Thatfunctionacceptsaparametercalled"attackOnRoute".

<code>$ai\_seq\_args[0]{attackOnRoute}</code>issettothesamevalueas thisparameter.

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

When you type "move" in the console, that parameter is setto 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 theautoattackAIcodeblock,Korecheckswhether

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

</pre>

<h2>Timeouts:Towaita whilebeforedoingsomething</h2>

Incertaincasesyoumaywant theprogram towaitawhilebefore doinganythingelse.

Forexample, you maywant tosenda "talktoNPC" packettotheserver,thensenda"chooseNPCmenuitem2"packet

2secondslater.

<p>

Thefirstthingyouwouldthinkof isprobably tousethe

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

However, that is a bad idea. <code>sleep()</code> blockstheentireprogram.Duringthesleep,nothingelsecanbeperformed.

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

youcanusethetimeOut()functionin anAIsequence.Thisexampleinitializesa

conversation with NPC 1337 (aKapra 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} = <span class="cstr">'Juststarted'</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

NPC 1337</span>sendTalk($net,1337);

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

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

<spanclass="comment">#Wewantto

waittwoseconds</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>

TheentireAIsubsystemiskepttogetherbythesetwovariables:

<ul>

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

Usually,AIcodeblocksarerunbasedonthevalueof 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>

Thedesignisprettysimple.Thisallowsthesystemtobeveryflexible:

youcandoprettymuchanythingyouwant.There aren'tmanyreallimitations

(butthat'sjustmyopinion).

<p>

The<code>AI()</code>functionrunsonlyveryshortly.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 value withinthe <code>@ai\_seq</code> queue (and an associated valueinsidethe<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="GetFirefox - Take Back the Web"><img width="104" height="32"src="<http://www.mozilla.org/products/firefox/buttons/getf>irefox\_small.png" alt="Get Firefox - Take Back theWeb"></a></li>

<li><a

href="<http://www.mozilla.org/products/firefox/>" title="Ifyouwere looking atthispageinany browser butMicrosoftInternetExplorer,itwouldlookandrunbetterand 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>