BuildTheHTMLPage

TeamID: PNT2022TMID25963

```
<!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" ></script>
                                                <![ 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; initializationpage, the <code>AI()</code> function only runs less thanafractionofasecond.

<p>

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

<aname=" ex1" >

<h2>Example1:Randomwalk< /h2>

You'reprobablyfamiliarwithKore'srandomwalkfeature.

If there are no monsters and Kore isn't doing anything, it will walkto 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 ifit isn'tdoing anything:

<preclass=" example" >

1	*##### RANDOM WALK#####*
2	*if($config{'route_randomWalk'}&& $ai_seq[0]*

```
<b>eq</b>""&&@{$field{'field'}}>  1&  &
!$cities_  lut{$field{'name'}.'.rsw'}){

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

4                   <b>do</b>{

5                   $ai_v{'temp'}{'randX'}  =  int(rand()
*($field{'width'}-  1));

6                   $ai_v{'temp'}{'randY'}  =  int(rand()
*($field{'height'}  -  1));

7                   }
<b>while</b>($field{'field'}[  $ai_v{'temp'}{'randY'}*$field{'width'}+
$ai_v{'temp'}{'randX'}]);8

9                   <span  class=" comment" ># Move  to
                    thatblock</span>

10                  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>;

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>

We call this block of code an AIcodeblock.

In other words, an AI code block is an entire blockofcodewhichdealswithacertainpartof theAI.

<h3>Situation

check</h3>Inline1, itchecks:

<ol>

*
whethertheconfigurationoption <code>route_randomWalk</code>ison
*

*
whether there are currently no other active <emclass=" term" >AIsequences(seebelow)
*

*
whetherwe'recurrentlyNOTinacity
*

*
*

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

<p>

What is an AI sequence? It is avaluewithinthe<code>@ai_seq</code> array.

Thisarrayisacommandqueue.

<p>

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

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

So, if <code>@ai_seq</code> is empty, then that means allAIcode blocks have finished and Kore isn't doinganythingelse.

AndthisiswhentherandomwalkAIcodeblock jumpsin.

<p>

There is also the <code>@ai_seq_args</code> array, usedtostore 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 mapthatyoucanwalkon.

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

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

<p>

Theresultcoordinateisputintothesetwovariables:

<ul>

*
<code>$ai_v{temp}{randX}</code>
*

*
<code>$ai_v{temp}{randY}</code>
*

*
*

<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 notsynchronous.

<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" hey, nowit'smyturntodo something!" .

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

When the route AI code block has finished, it will removethefirstitem from<code>@ai_seq</code>.

If <code>@ai_seq</code> is empty, then the randomrouteAIcodeblock isactivatedagain.

<h2>Example 2: Attacking monsters while walking toarandom 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 oftheaboveAI 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, ifthatparameter 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");

if ($ai_v{'temp'}{'ai_route_index'} ne <spanclass=" cstr" >""){

$ai_v{'temp'}{'ai_route_attackOnRoute'}= $ai_seq_args[$ai_v{'temp'}{'ai_route_index'}]{'attackOnRoute'};

}

...

 # Somewhere else in the auto-attackAI code block,Korechecks whether

 # $ai_v{'temp'}{'ai_route_attackOnRoute'} is setto1.< /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 sequencesarenotrun, networkdataisnotreceived, etc.

<p>

The right thing to do is to use the <ahref="Utils.html#timeOut"><code>timeOut()</code>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 menuitem2" packet.

```
<preclass=" example" >

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

 # Store thecurrenttimeinavariable

$ai_seq_args[0]{waitTwoSecs}{time}=time;

 # We wanttowaittwoseconds

$ai_seq_args[0]{waitTwoSecs}{timeout}=2;

$ai_seq_args[0]{stage} = <spanclass=" cstr" >'Initializedconversation';

}elsif($ai_seq_args[0]{stage} eq 'Initializedconversation'

 # This 'if'statementisonlytrue iftwo seconds havepassed

#since $ai_seq_args[0]{waitTwoSecs}{time}isset

&& timeOut($ai_seq_args[0]{waitTwoSecs})

){

 # Twosecondshavenowpassed

sendTalkResponse($net, 1337, 2);

#We'redone; removethis AIsequence

shift@ai_seq;

shift@ai_seq_args;

}

}

...

}

</pre>

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

The entire AI subsystem is kept together by thesetwovariables:

<ul>

*
<code>@ai_seq</code> : a queue which contains AIsequencenames.*

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

*(though this doesn't have to be true; it depends on howtheAI codeblockisprogrammed).
*

*
<code>@ai_seq_args</code> : contains argumentsthat'sassociated withcurrentAIsequence.
*

*
*

The design is pretty simple. This allows the system tobeveryflexible:

you can do pretty much anything you want. Therearen'tmanyreallimitations

(butthat'sjustmyopinion).

<p>

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

<h3>Glossary</h3>

<ul>

*
An AI code block is an entireblock of code which deals with a certain part of theAI.
*

*
An AI sequence is a valuewithinthe<code>@ai_seq</code>queue(andanassociatedv alueinsidethe<code>@ai_seq_args</code>array).
*

*
*

<p><hr><p>

<divid=" footer" >

<ul>

*
<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" >
*

*
<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" >
*

*
<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" >
*

*
*

* *

* *

</body>

</html>