

## AOZ Studio Beta - Bug #280

### For loop misses first iteration in some cases.

03/01/2020 02:40 AM - Brian Flanagan

|  |                                  |
|--|----------------------------------|
| <b>Status:</b> Closed  | <b>Start date:</b> 03/01/2020    |
| <b>Priority:</b> Normal  | <b>Due date:</b>                 |
| <b>Assignee:</b> Francois Lionet   | <b>% Done:</b> 0%                |
| <b>Category:</b>   | <b>Estimated time:</b> 0:00 hour |
| <b>Target version:</b> 0.9.6.1   |                                  |
| <b>Affected version:</b> 0.9.6   |                                  |
| <b>Description</b>   |                                  |
| <p>For loops should ALWAYS complete the first iteration (even if the Step is in the wrong direction), but in AOZ they don't. (This is probably related to bug <a href="#">#259</a>. I've included a similar example here.)<br/>I'm guessing this is because the comparison is being done at the For instead of at the Next, as it should be.</p> <p>In the 3 test cases shown in the following example, AOZ fails the first (and only) iteration:</p> <pre>#manifest:"pc" #splashScreen:false  Print "2 To 1" For x=2 To 1   print x Next x  Print "1 To 2 Step -1" For x=1 To 2 Step -1   Print x Next x  Print "1 To 1" For x=1 To 1   Print x Next x  Print "Done"</pre> <p><b>Results (FAIL):</b></p> <pre>2 To 1 1 To 2 Step -1 1 To 1 Done</pre> <p><b>Result SHOULD be:</b></p> <pre>2 To 1   2 1 To 2 Step -1   1 10 To 10   10 Done</pre> |                                  |
| <b>Related issues:</b>   |                                  |
| Related to AOZ Studio Beta - Bug #259: For...Next from 1 to 1 doesn't enter t...   | <b>Closed</b> <b>02/27/2020</b>  |

### History

#### #1 - 03/01/2020 07:48 AM - Baptiste Pillot

- Related to Bug #259: For...Next from 1 to 1 doesn't enter the loop added

**#2 - 03/01/2020 11:35 PM - Brian Flanagan**

- *Affected version changed from 0.9.5.1 to 0.9.6*

1 to 1 example corrected. Other two still fail in 0.9.6.

**#3 - 03/07/2020 04:51 PM - Francois Lionet**

- *Status changed from New to Resolved*

- *Assignee set to Francois Lionet*

- *Target version set to 0.9.6.1*

OK, tough one.

In modern languages, when a loop- has no reason to run, it is not.  
In Basic, in the Amiga days, the loop was always run once.

So, the only solution here is differentiating between Amiga and PC, so in Amiga mode, the loop is always ran once, and in PC mode, if the initial values are out of range, it does not enter the loop.

**#4 - 03/27/2020 05:11 AM - Brian Flanagan**

- *Status changed from Resolved to Closed*

Tested in the current version 0.9.7 test 2, and all cases work fine!

I would still recommend warning messages in the transpiler if the loop direction is wrong.