AOZ Studio Beta - Bug #113

Scancode function is returning the wrong scan codes. (Sometimes even NaN)

02/04/2020 04:25 AM - Brian Flanagan

Status:	Closed	**Start date:**	02/04/2020
Priority:	Normal	**Due date:**	
Assignee:	Francois Lionet	**% Done:**	0%
Category:		**Estimated time:**	0:00 hour
Target version:	0.9.5		
Affected version:	0.9.9		

Description

Example: Help_43 from AMOSPro_Examples disk.
The Scancode function results are completely screwed up.

Example results:

Keys: 1 2 3 4 5 6 7 8 9 0 - = (top row of keys [partial])
should return 1 2 3 4 5 6 7 8 9 10 11 12.
instead returns NaN NaN NaN NaN NaN NaN NaN 65 NaN NaN NaN NaN

Keys: q w e r t y u i o p (next row of keys [partial])
should return 32 33 34 35 36 37 38 39 40 41.
instead returns 0 0 0 NaN NaN NaN NaN NaN NaN NaN

Keys: a s d f g h j k l ; - (third row [partial])
should return 32 33 34 35 36 37 38 39 40 41
instead returns 64 0 95 0 0 79 76 78 77 NaN

etc.

Here's a good reference for the scan codes: https://wiki.amigaos.net/w/images/thumb/1/12/LibFig34-2.png/800px-LibFig34-2.png

History

#1 - 02/22/2020 09:55 AM - Brian Flanagan

- Affected version changed from 0.9.3.2 to 0.9.5

Correction to example results above:

Keys: q w e r t y u i o p
should return 16 17 18 19 20 21 22 23 24 25 26 27.
instead returns 0 0 0 NaN NaN NaN NaN NaN NaN NaN NaN 0

Also verified this is still broken in 0.9.5
Adjusted affected version from 0.9.3.2 to 0.9.5.

#2 - 02/22/2020 11:56 AM - Francois Lionet

- Status changed from New to Resolved

- Assignee set to Francois Lionet

- Target version set to 0.9.5

Fixed! And thank you for the table of scan code, I never could find one... (Y)

#3 - 02/27/2020 03:42 AM - Brian Flanagan

- Status changed from Resolved to Feedback

Thank you! ...but it's only *mostly* fixed. Some codes are still wrong:

```
\ to the right of the "querty" row should be 13 vs. 43.
Enter on the keypad should be 67 (different than regular Enter: 68).
```

Hopefully that's the last of them.

Also, (another issue), but Key State is all screwed up. Here's what Key State should return:

```
Bit 0:    1 Left Shift  instead returns 3 (combined left & right shift)
Bit 1:    2 Right Shift instead returns 3 (combined left & right shift)
Bit 2:    4 Caps Lock   returns NOTHING
Bit 3:    8 Ctrl        returns 8 (this is the only one working)
Bit 4:   16 Alt     returns 48 (16 + 32)
Bit 5:   32 ??? usused ???
Bit 6:   64 ??? usused ???
Bit 7:  128 ??? usused ???
```

Finally, I'm not sure about this, but I think some of the keys that return the states are *also* supposed to return scan codes. (I'll have to test this on a real Amiga to see.)

#4 - 02/28/2020 07:59 PM - Brian Flanagan

- Affected version changed from 0.9.5 to 0.9.5.1

#5 - 03/01/2020 05:09 PM - Brian Flanagan

- Affected version changed from 0.9.5.1 to 0.9.6

The problems I mentioned in 0.9.5.1 remain.

One more problem found. Both of the enter keys will not sustain the key down state.

They will reset while the keys are still held down.

(This is besides the fact that they should return different codes. Perhaps this is related?)

#6 - 03/13/2020 03:18 PM - Brian Flanagan

Regarding the key states: Here's the information I was missing on the other bits of key state. (The documentation was right inside Help_43!)

```
Bit 0:    1 Left Shift  instead returns 3 (combined left & right shift)
Bit 1:    2 Right Shift instead returns 3 (combined left & right shift)
Bit 2:    4 Caps Lock   returns NOTHING
Bit 3:    8 Ctrl        returns 8 (this is the only one working)
Bit 4:   16 Left Alt    returns 48 (16 + 32)
Bit 5:   32 Right Alt   returns 48 (16 + 32)
Bit 6:   64 Left Amiga  returns NOTHING (equivalent of Left Windows or Left Command (mac))
Bit 7:  128 Right Amiga returns NOTHING (equivalent of Right Windows or Right Command (mac))
```

Also, I found that we can return ALL of the errant keys and key states properly via JavaScript, as well as those keys missing from AOZ at this time.

Here's a script to show how:

```
<!DOCTYPE html>
<html>
<body>

Click in the input box and press any key.

<input type="text" onkeydown="CheckKey(event)">

<div style="font-weight: bold">(The location property is not supported in Safari or IE <= version 8.)</div>
<br />

<div id="results"></p>

<script>
function CheckKey(event)
{
  var k = event.key;
  var l = event.location;
  var w = event.which;
  var c = event.code;

//  var x = event.location;
  document.getElementById("results").innerHTML = "Key: " + event.key + " Loc: " + l + " Which: " + w + " Code:
 " + c;
}
</script>
```

```
</body>
</html>
```

The AOZ keyboard handling routines should be revamped to handle *all* of the keys properly using the event properties in the above example.

#7 - 03/13/2020 03:20 PM - Brian Flanagan

You can ignore / delete the commented line in the example above:
// var x = event.location;

#8 - 03/13/2020 03:22 PM - Brian Flanagan

- Affected version changed from 0.9.6 to 0.9.6.3

Oops! Forgot to up the "affected version". Problem still exists in 0.9.6.3

#9 - 05/08/2020 04:42 PM - Brian Flanagan

- Affected version changed from 0.9.6.3 to 0.9.9

Tested again in 0.9.9. It works better. Now some of the shift states work, but many keys are still messed up.

The shift states are about *half* fixed now.

```
Bit 0:    1 Left Shift   WORKING!
Bit 1:    2 Right Shift SHOULD return 2, instead returns 1.
Bit 2:    4 Caps Lock SHOULD return 4, instead returns NOTHING
Bit 3:    8 Ctrl            WORKING!
Bit 4:   16 Left Alt     WORKING!
Bit 5:   32 Right Alt   SHOULD return 32, instead returns 16
Bit 6:   64 Left Meta WORKING!
Bit 7:  128 Right Meta SHOULD return 128, instead returns "ContextMenu" in Inkey$  This will be a problem in B
OTH Amiga and PC manifests!
  When someone hits the "ContextMenu" key we don't need C o n t e x t M e n u in the input buffer!!!  This is
another modifier key and should return *nothing*!
```

I also still suggest adding:

bit 8: 256 to detect the Right Ctrl key. (Again, should return nothing in the input buffer, just a shift state.)

Also... right now, you can't enter a control key. Instead of having Ctrl modify the letter typed, it enters that letter into the input buffer / Inkey$

Ctrl-a returns the proper shift state, but instead of putting Chr$(1) into the buffer, it puts "a" in.
Ctrl-m again returns the proper shift state, but instead of putting Chr$(13) into the input buffer (just as Enter should), it puts "m" in.
As mentioned above, the Enter key, instead of putting Chr$(13) into the input buffer, puts "Enter" E n t e r into the input buffer. Definitely a problem no matter what manifest is in use!

(Think abut these keys with respect to say, writing a word processing program! MAJOR problem!)

What if someone wants to use the Right Meta key? Right now, instead via the input buffer & Inkey$, they'll get 11 keystrokes, "ContextMenu", instead of the appropriate shift state : 128! Again... a MAJOR problem no matter what the manifest.

#10 - 10/02/2020 09:17 AM - Brian Flanagan

- Status changed from Feedback to Resolved

I fixed these a while back. The modifier keys are all working now. There are slight differences betweeen browser implementations which have been accommodated. In the Amiga manifest, the proper scan codes are returned with the exception of 0. This is unavoidable due to the current implementation. It also better corresponds to the AOZ/PC version. In AOZ mode, for the most part, event.what is returned... which will be the JavaScript equivalent of scan codes.

NOTE: Perhaps in the future, we can add the option of returning *actual* PC scan codes for better compatibility with other PC applications.

#11 - 10/02/2020 09:17 AM - Brian Flanagan

- Status changed from Resolved to Closed