

# jAMAL - AMOS Animation Language reimplemented in Java!

Alpha release 0.52 – August 2<sup>nd</sup> 2011  
(c) 2011 Mequa Innovations

## Credits:

AMOS BASIC and its 'sublanguage' the AMOS Animation Language (AMAL) was originally created for the Commodore Amiga by François Lionet (now part of Clickteam - <http://www.clickteam.com> ), and originally sold by Europress Software.

Both Amiga products "AMOS The Creator" and "AMOS Professional" included AMAL, for both interpreted and compiled AMOS BASIC programs (the compilers for both products were sold separately). AMAL itself was always compiled to Motorola 68000 machine code.  
(This version of jAMAL is both interpreted AND supports direct translation to ASCII Java source.)

The original AMOS The Creator and its compiler are now available under a BSD-style licence - source available (Motorola 68000 assembly):  
<http://www.clickteam.com/eng/downloadcenter.php?i=58>

jAMAL uses Java. To install or update Java Runtime Environment (JRE), visit the Java website:  
<http://www.java.com>

jAMAL utilises the JGame engine by Boris van Schooten:  
<http://www.13thmonkey.org/~boris/jgame/>

Blue backdrop (c) Photos8.com:  
[http://www.photos8.com/view/blurry\\_blue\\_background-other.html](http://www.photos8.com/view/blurry_blue_background-other.html)  
Public domain background image (c) NASA.

For a complete guide to AMAL, see the [AMOS Professional manual](#).

## Notes:

This version runs from a single jAMAL.jar file.

This initial release is incomplete, and intended as a "proof of concept" that AMAL can indeed be reimplemented in Java.

The Lexer and Interpreter core are almost complete.

Sprite and Screen integration and editing are in early stages of development.

## Source release notes:

jAMAL\_src.zip is now importable into Eclipse IDE under "Existing Projects into Workspace".

It should now be buildable out of the box. (JGame libraries are in 'jgame' directory.)

It should also build out of the box with AmalPrograms.java replaced with a jAMAL-exported Java file.

## Editor limitations:

- You cannot yet copy or paste to or from external programs.

## Game engine limitations:

- Can't yet change the screen size or create more screens (stay tuned!)
- Smooth scrolling/VSync (and double buffering) is not yet implemented (requires JOGL/OpenGL).
- Collision detection is not yet implemented. Collision functions are parsed, however.
- Rainbow (emulated Copper) effects are not yet implemented using JGame.
- AMAL Bank movement patterns (PLay) are not yet implemented. PLay is correctly lexed, however.
- =VUmeter(..) to move sprites to sound, is not yet implemented. [=VU(..) is lexed but returns 0]

## Interpreter limitations:

- Currently very memory hungry.
- 'For' loops do not yet record the target value into a buffer. It doesn't work properly yet for "For R0=1 To Z(7)+3". Nested For loops are also not yet implemented.

## Lexer limitations:

- Currently very memory hungry.
- AMAL expressions currently may have only one operator. [Let A=Z(R0+5) is fine, but not Let R0=1+2+3].
- Using functions at the end of Move .....,.. is currently buggy [confused with ')', end of AUtotest]

## Proposed pedantic mode for forward compatibility:

For a later 'Fully AMOS Compatible' option in the Lexer, the following original limitations will be implemented:

- "Pause Let R0=1" currently works, this would be interpreted as "PLay R0=1" in the original AMAL without a separating semicolon.
- "Let A=\$8001B: ... " currently works, this would fail in the original AMAL without a separating semicolon.
- Collisions will be testable within an AUtotest when not using pedantic mode. (=C(..) currently returns 0 always)
- The game engine and editor will also optionally be able to respect the original limited number of channels, programs, sprites etc.
- Will eventually be able to import/export original AMOS .abk AMAL banks, loadable into AMOS. Will warn if not using pedantic mode.

## Changes in jAMAL alpha 0.52

### Changes from alpha demo 0.1 in alpha 0.1.1

- More examples added, and added more comments. Should be much easier for beginners now!
- Pausing improved. "Run" and "Pause" now both function as "Unpause".
- Will now officially translate AMAL to Java! (early alpha stage, not yet buildable)

### Changes from alpha demo 0.1.1 in alpha 0.2

- Fixed a major concurrency bug. All examples now load first time now.
- Improved the examples and added two "bouncy ball" demos.
- Can now save and load external files in ASCII format. (Uses OS-native file requester.)
- PLay is now lexed correctly (movement banks not yet implemented).
- You can now strip your code. Oh là là!

- Added the ability to view all your programs in standard or stripped mode.

### **Changes from alpha 0.2 in alpha 0.3**

- AMAL to Java machine-translated source code is now DIRECTLY BUILDABLE WITHOUT MODIFICATION!
- AMAL scripts can now run either interpreted, or translated directly into Java and built. (Requires full jAMAL source [with JGame] and Java SDK to build your own - former coming soon!)
- Can now export .java files which contain a complete buildable Java class (using OS-native endlines).
- A fully translated-to-Java and working non-interpreted AMAL example has been integrated into the editor.

### **Changes from alpha 0.3 in alpha 0.4**

- Now uses a higher resolution game window (640x512) and more sprite images.
- Sprites are now fully V-reversible as well as H-reversible.
- 90 degree rotation is possible for the pacman sprite (now also stored in rotated form).
- Some Amiga-style "boing ball" and "checkmark" sprites (and the jAMAL logo) were added to the examples.

### ***First source release on Sourceforge. BSD Licence.***

- jAMAL\_src.zip is now importable into Eclipse IDE under "Existing Projects into Workspace".
- It should now be buildable out of the box. (JGame libraries are in 'jgame' directory.)
- It should also build out of the box with AmalPrograms.java replaced with a jAMAL-exported Java file. Enjoy!

### **Changes from alpha 0.4 in alpha 0.41**

- Tidied up and refactored the Java source. It's now tidier and produces a smaller JAR file.
- Now launches JAMAL.java in jAMAL/src/jamal/ (instead of MyGame.java in jgame-sample/src/mygame/)

### **Changes from alpha 0.41 in alpha 0.42**

- Now supports AMAL functions with multiple arguments (=XScreen(1,2), =YScreen(1,2), =XHard(1,2), =YHard(1,2), =BobCol(1,2,3), =SpriteCol(1,2,3)). Collision detection and multiple screen support is not yet implemented.
- =Z function (for random numbers) no longer locks up when used without an argument (still needs more work).

### **Changes from alpha 0.42 in alpha 0.50**

- The AMAL Environment Generator has been reimplemented. This is based on a subset of AMOS BASIC itself.
- Can now load a background image from a file (or resource).
- Can now load sprite images from a folder or directory (disk or resource).
- The AMAL-to-Java Compiler has been greatly optimised (still needs more work though).
- A new background was added (from Photos8.com).
- Can now create a new project, and create a new AMAL channel and program.
- Added support for screen dragging and scrolling effects using Screen Display and Screen Offset in the Environment Generator. These can now be allocated to AMAL channels to allow AMAL scripts to control the screen directly. Some new examples were included.

### **Changes from alpha 0.50 in alpha 0.51**

- Bug fix: The playfield is now correctly reset after using Screen Offset (screen scrolling demo) on loading a new example or program.

### **Changes from alpha 0.51 in alpha 0.52**

- Added Bell command to Environment Generator (with AMOSPro bell sound)
- Enhanced the quality of the scrolling example background
- Added digital speech on startup, stripping and translating code
- Added a new game control example – Shoot.amal

## **LICENSE**

This software is distributed under the revised BSD license.

Copyright (c) 2011 Mequa Innovations.

Copyright (c) 2006 by Boris van Schooten.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the organizations nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.