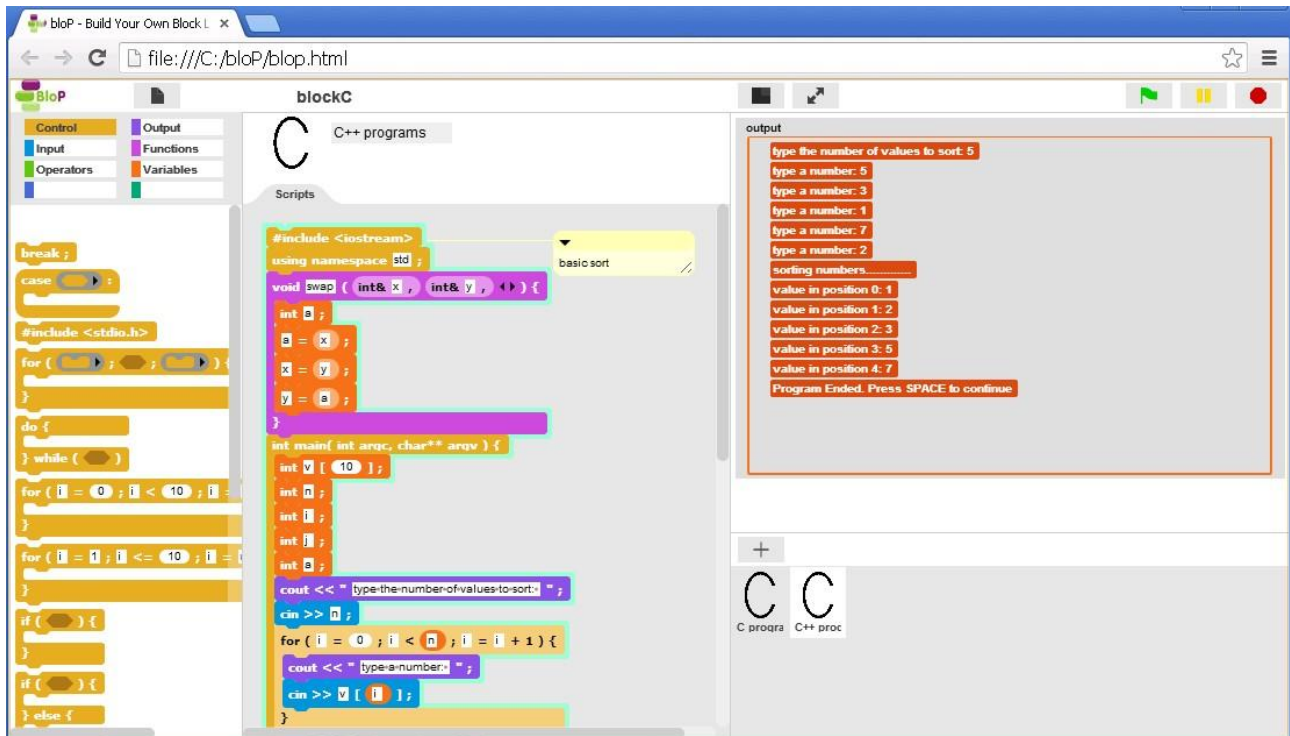


bloP – Build Your Own Block Language

by Stefano Federici
www.blockprogramming.org

bloP is a development environment for block languages. The environment is a mod of Snap¹. bloP stands for "BLOck Programming".

You can find more information about block languages at www.blockprogramming.org.



bloP running blockC, a block implementation of a subset of the C/C++ programming language

bloP is delivered with a bundled implementation of the "block-C" language, a subset of C/C++ programming language. You can download bloP and blockC at <http://goo.gl/auglHp>.

WARNING: while bloP runs fine either in Chrome, Firefox or IE, the implementation of a heavy block language could crash or freeze your browser. The only browser that is known to work with blockC on my very old Intel Core2 Duo T7250 @ 2.00GHz is at the moment Chrome.

How to Startup a block language

To startup the included blockC language follow these steps:

1. open bloP.html in Chrome
2. in File > Open select the blockC.xml file (if while loading blockC.xml Chrome tells you that there is a script that is not responding... just wait a little bit)

¹ Many thanks to Jens Moenig and Brian Harvey for developing the wonderful Snap environment (<http://snap.berkeley.edu>). bloP is based on Snap version 24/7/2013.

3. you can run the sample programs that you find in "C programs" or "C++ programs" or you can create your own blockC programs by adding or duplicating a program (to do so use the "+" button or rightclick the program icon in the corral) or by using File > New in order to create a completely new blockC program. Just snap together the C/C++ blocks you will find in the Input, Output, Control, Functions, Operators, and Variables categories.

4. save your programs by using File > Save As... The programs will be saved in a new XML file (NOTE: in this version of bloP the XML file includes both the programs and the whole definition of the block language)

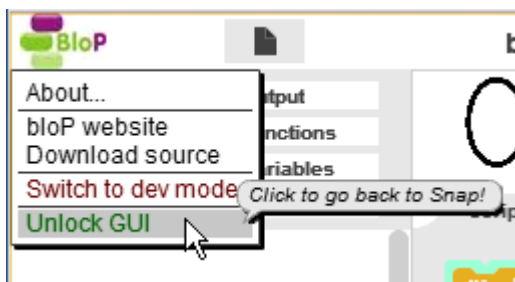
In order to run a program, just click on its script. By clicking the green flag, the *topmost* script of the selected sprite will be run. When you click on a different script, the program that is running (if any) will be automatically stopped.

WARNING: if you don't load a block language or a program (that is an xml file including both the definition of a block language and several programs) you *won't be able to run scripts* in BloP either by clicking them or by clicking the green flag button. If you want to run a script without loading a block language in advance, you need to *switch to Snap* (see following section).

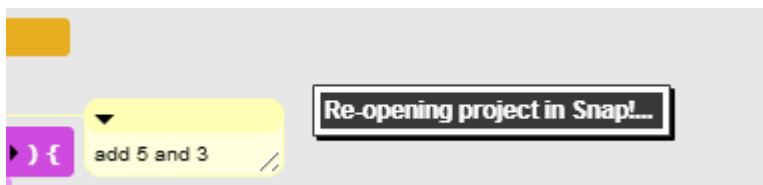
How to Modify or Create a block language

To create a new block language, just follow these steps (if you want to modify the blockC language, follow steps 1 and 2 above in order to load blockC in advance):

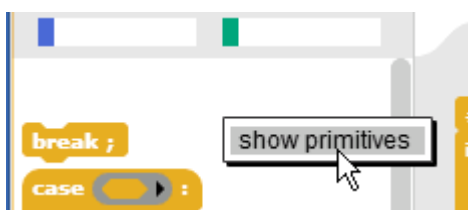
1. shiftclick the bloP logo and select "Unlock GUI" in the menu



The project will be re-opened in Snap



2. show all the primitives you need by rightclicking the palette and selecting "show primitives" in the menu

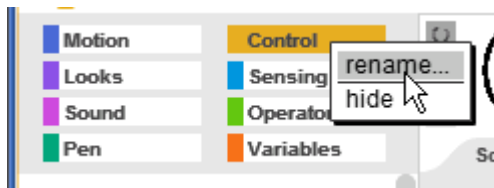


3. make all icons visible in the corral by rightclicking the corral and selecting "add removed objects back to corral"

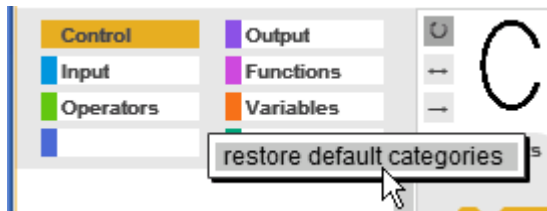


4. modify the environment as you like by:

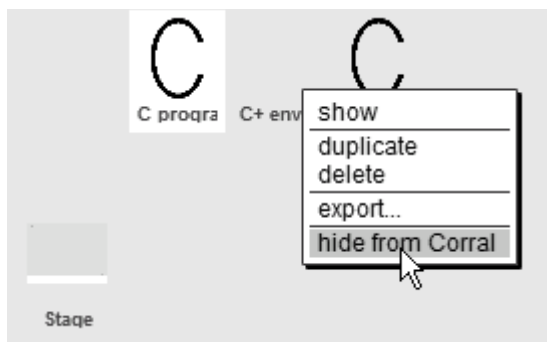
- modifying custom blocks or adding new custom blocks as you normally do in Snap
- renaming categories by rightclicking them



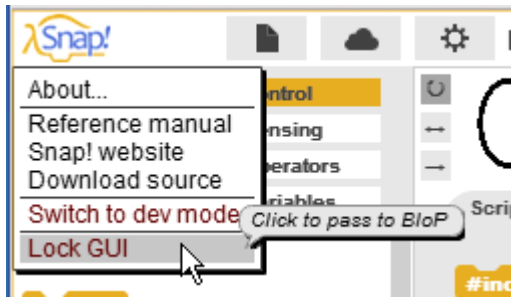
- restoring default names and visibility for all categories by rightclicking the category frame



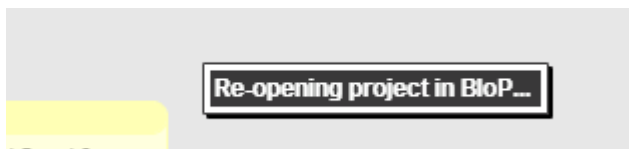
- hiding sprite icons or the Stage icon in the corral by rightclicking them



5. When you are done, go back to the bloP environment by shiftclicking the Snap logo and selecting "Lock GUI"



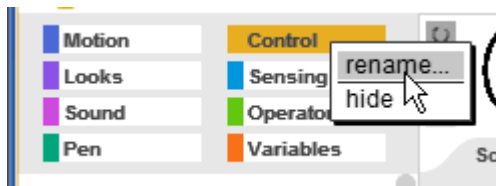
The project will be re-opened in bloP:



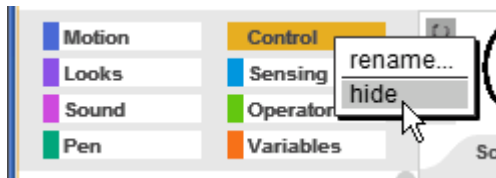
bloP Philosophy

The main objective of bloP is to allow you to create a safe environment in which you can develop and run your favorite block language programs without the risk of impairing the environment. In order to get this goal Snap has been modified, trying not to change the way it works (you should be able to load and run Snap XML files without any modification). Here you are the list of modifications to snap:

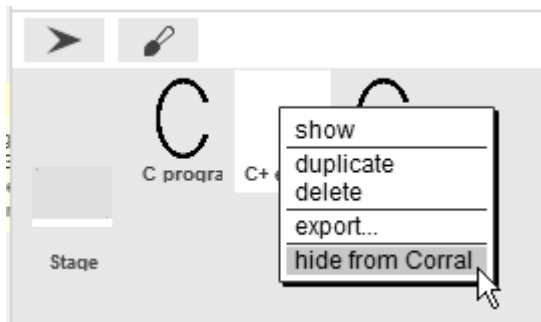
- you can rename categories



- you can hide categories (when you hide a category, all primitives of that category will be hidden by default but you can still make them visible by rightclicking the palette and selecting "show primitives")



- you can hide the Stage and the sprites icons in the corral (but you need at least one visible sprite icon in the corral)



- you have 3 new Control blocks:
 - "when loading" hat: the script under this hat is run when a program is loaded in Snap/bloP



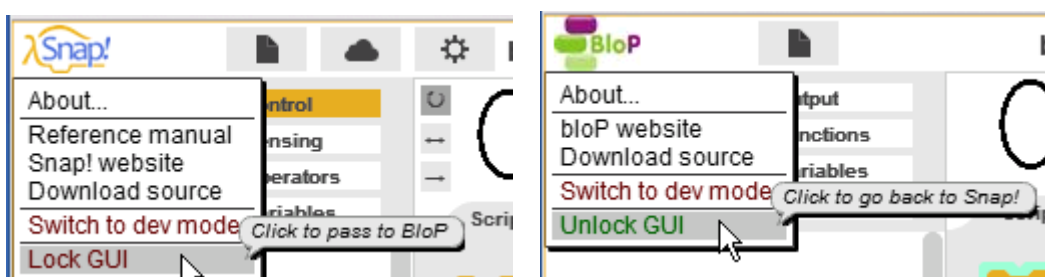
- "when running a script" hat: the script under this hat is run by bloP before another script is run (that is when you click on a script or you click the green flag)



- "run other scripts" block: this block, used at the end of a script that starts with the "when running" block, runs the script you intended to run when you clicked on it or clicked the green flag



- you can switch from Snap to bloP (and viceversa) by shiftclicking the logo in the topleft corner and selecting "lock/unlock GUI".



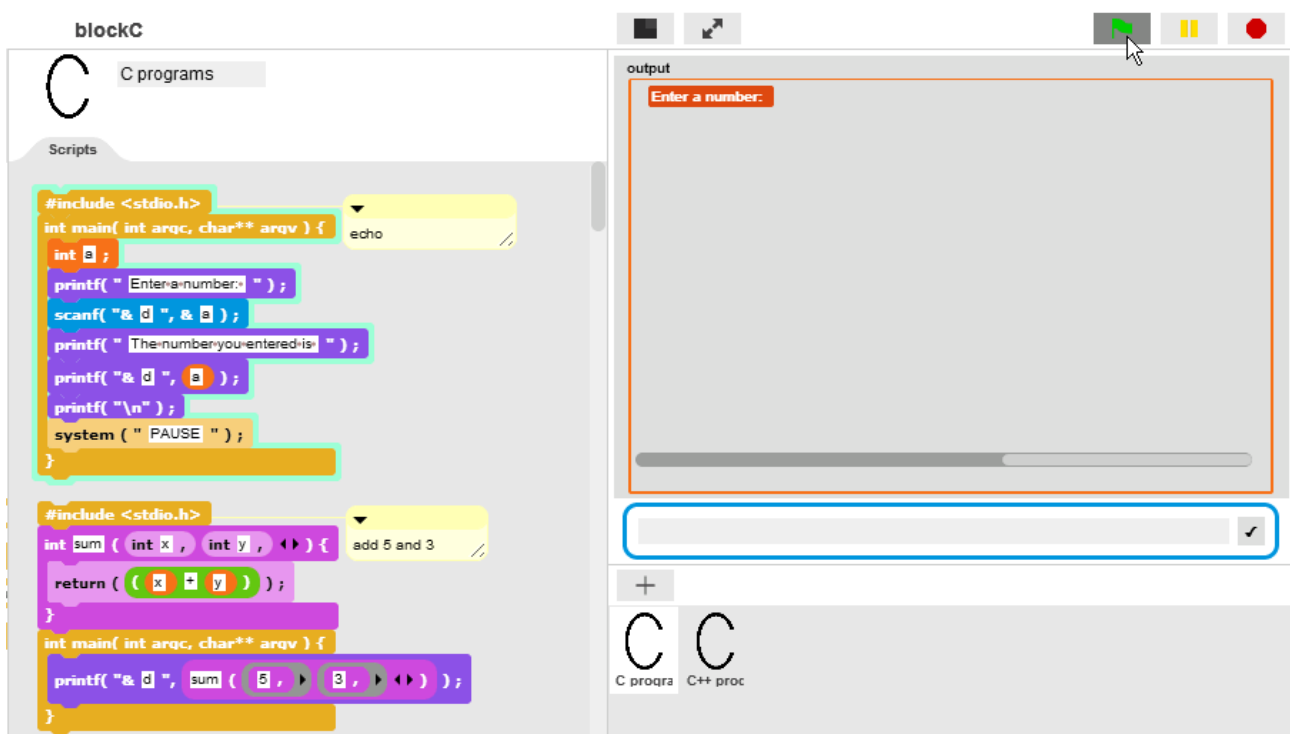
When you lock the Snap GUI and switch to bloP, you will have a different environment in which:

- sprites and watchers on the Stage are *non draggable*
- variable and list watchers *cannot be modified*:



The list watcher in this pictures is missing all the usual active areas

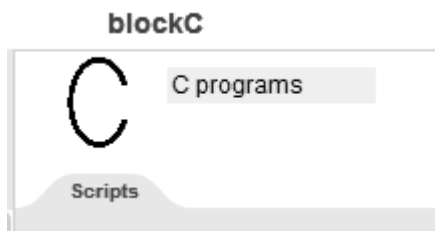
- by clicking the green flag you run the topmost script of the selected sprite:



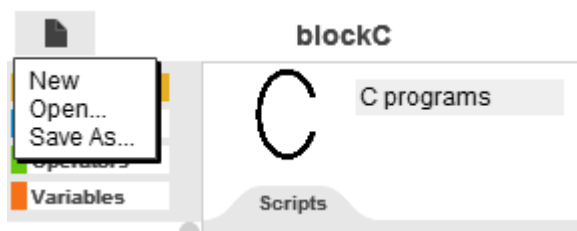
- you can't have programs (sprites) with the same name

Several features of the Snap gui will then be locked, that is

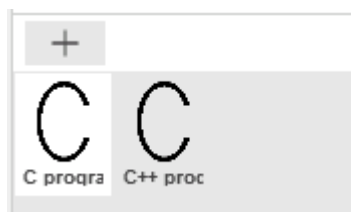
- sprite infos (such as rotation style and druggability) and the costumes and sounds tabs are hidden:



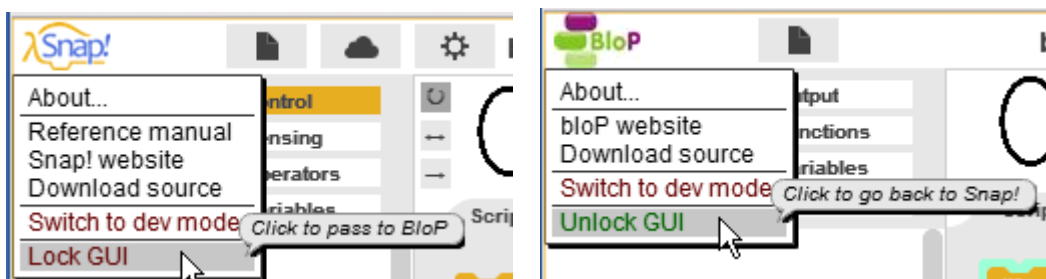
- you can't hide/show primitives
- you can't hide/rename categories
- "sprites" are renamed "programs"
- you can't define or remove custom blocks
- you can't change settings or use the cloud storage



- the button to create new sprites by drawing them is hidden:



By shiftclicking the Snap logo you will be able to lock the GUI again so that your language users won't be able to impair the environment by doing something wrong. But they will still be able to modify the environment, if they want, by shiftclicking the bloP logo and knowingly unlocking the environment.

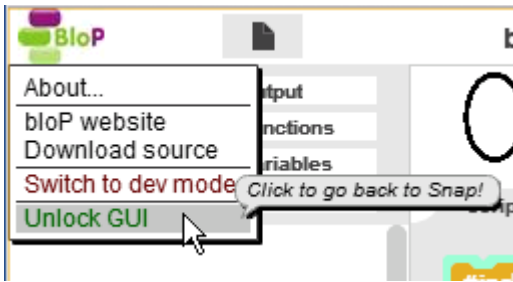


WARNING: if you don't load a block language or a program (that is an xml file including both the definition of a block language and several programs) you *won't be able to run scripts* in BloP by clicking them or by clicking the green flag button. To do so, you need at least a script containing the "when running another script" and "run other scrips" blocks (see following section).

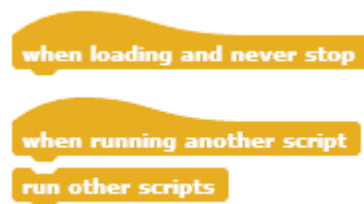
Create your own block language

These features should allow you to develop your own programming language built by blocks.

By shiftclicking the bloP logo you will be able to unlock the GUI so that you will have the full power of Snap available to develop your own language blocks.



Use the "when loading" and "when running" blocks to define scripts that you need to run at setup time in order to prepare your environment ("when loading" hat) or to cleanup your environment each time a new program runs ("when running" hat):



Remember: your users will be able run a new program even before the running program has halted (the running program will be stopped).

WARNING: if you don't load a block language or a program (that is an xml file including both the definition of a block language and several programs) you *won't be able to run scripts* in bloP by clicking them or by clicking the green flag button. The reason is that a script, in order to be able to run, needs another script starting with the "when running another script" block and ending with the "run other scripts" block, that is at least the following minimal script:



This script, hopefully containing other blocks inside, is usually stored into a sprite whose icon has been hidden in the corral so that the user s won't be able to remove it and impair the environment.

bloP always runs in "locked" mode at startup so, when you are done, all you need to do is saving your language: there is no need to switch back to bloP before saving (remember: in this version of bloP the XML file includes both the programs and the whole definition of the block language). Furthermore, when your users will load your language or their programs in bloP they won't be able to impair the environment by doing something wrong.

Comments are Welcome!

The present design of bloP is mainly oriented to the implementation of text languages (such as C, Perl, Python, etc) but new features are already in progress. If you have any suggestion about what you would need in order to implement your own block language (textual or visual), or you think that you had a good idea on how to improve bloP or you just found a bug, drop me a line at stefano.federici@tiscali.it.

Future Work

I'm working on importing in bloP several block languages I had started to develop in BYOB, namely ASSL (a visual language to design searching/sorting algorithms) and a subset of Berkely Logo. If you want to collaborate to create an extended library of block languages, please, do not hesitate to contact me.

Known Problems:

- you can't use single quotes in block names (see "A" block in Operators category. It should be 'A' instead...)



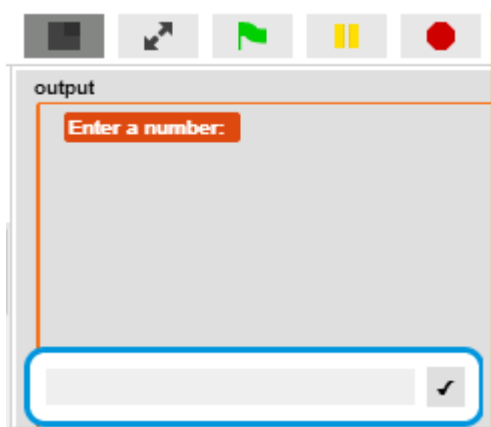
- the comma after the last arguments of a function shouldn't be there... (you can notice that I started working on this in the dialog of the properties of input arguments for custom blocks)



- rings can be really confusing to people that don't know what they are (this can happen to users of your own language that don't know of how Snap works). So I'm going to remove them when bloP is in locked mode.



- switching between Snap and bloP is a bit slow (now the whole program is saved and then reloaded). I'm planning to speed up this operation together with the operations of loading/saving bloP projects.
- watchers on the Stage don't grow or shrink in small Stage or presentation mode:



- having menu inputs for custom blocks would help a lot to create block languages that are much simpler and clear (you can notice that I started working on this in the dialog of the properties of input arguments for custom blocks)
- lists do not scroll to show the last added element