



P-055-05 GaP-BG Book One

Gambas Programming Beginner's Guide

Book One: The First Steps

Contents

- * Where to start
- * Where Gambas stands
- * What would be "The first step"
- * The Second Step
- * Communication: Third Step
- * A wonderful tool: Gambas Classes
- * About the GaP-BG Project
- * About this book *
- * Instead of "Good bye"
- * Getting to me *
- * Contents * EOF *

[F 27-03-2020 h 10:00]

[F 27-03-2020 h 16:20]

—

Where to start

Is it a question or an answer?

[F 27-03-2020 h 10:10]

Hello!

It is likely that this is what you are thinking now, when reading this chapter's title:

"Is this guy going to answer the question, or is he asking the question and expects me to know the answer? Is it a joke, or what?"

Neither one is my answer.

It is an introduction to something that I find deeper than the implied question "**Where to start programming in Gambas?**"

It's all about learning

Yes! This is all about! Learning as much as you can about yourself, about who and what you are, about your skills and what you can do with those skills, because **this is LIFE Herself!**

The more skills you have, the better prepared for life you are!

So, since you now my approach, you know my paradigm, let's move to something more obvious related to programming: LANGUAGE.

The languages

[F 27-03-2020 h 10:20]

Actually, **the real problem is called : COMMUNICATION.**

"**Language**" is in fact **THE SOLUTION.**

We all need to communicate, we are taught to do so since the first steps in our life. In most countries, school means learning the primary language and a second one, up to somewhere in the 8th grade, then some people choose to learn the 3rd language, and some others, go on with the second to a higher level of practice and understanding.

In fact, there are much more languages then we are aware of: Geography, Maths, Physics, Chemistry, Music, Computers, (another branching!!) and so on.

The essential idea here is the following:

While in school, we learn about any language *the main sections it has:*

1. **Vocabulary.** That is: the words we use to build sentences and phrases.
2. **Grammar.** The rules we follow when building a sentence (proposition) and then a phrase.
3. **Semantics.** The meaning of each word we use, and then the meaning of a sentence, of a a phrase.
4. **Syntax.** While grammar rules apply more often to the small entities called "words", Syntax applies to larger entities, such as sentences and phrases.

I am sure you now understand where am I leading you to:

Gambas, IS A LANGUAGE. One of the many languages designed to fill in the gap between a computer that knows only a two-letters alphabet and a very small set of words, a tiny grammar, a tiny syntax.

For me, for you, those are useless. Instead of learning to speak directly like "**0011001010100011111010110010111010?**"

"01111011101110111010101!" I bet you wold do whatever it takes to do something else, like... go shopping, have a nice trip to Bahamas or whatever

you really please. Note that the question and exclamation marks, are HUMAN language. For the computer, they are supposed to be another series of "01011100" or whatever other sequence of the cryptogram follows.

Programming languages

Like in real life, where different countries and different cultures have different languages so do the computers. There are many programming languages. Gambas, is only one of the many others: C, C++, C#, PHP, Perl, Python, HTML, XML, CSS, Fortran, Pascal.

Far more than that, there are FAMILIES of computer programming languages.

In the real world, the families of languages are usually called "dialects" and so is in the computer programming, we have **DIALECTS**.

One of the oldest families of the computer programming languages, is **BASIC**.

That is the acronym for "**B**eginner's **A**ll-purpose **S**et of **I**nstructions **C**ode". So, as you see, we already have a communication rule: this is what **BASIC** means. A neat twist to help us remember what is this about.

Where Gambas stands

[F 27-03-2020 h 11:00]

While the BASIC family grew steadily over the years, it worths mentioning that the original implementation of BASIC, was somekind primitive, allowing someone to do very simple tasks, like introducing some input from a keyboard in a certain format and then getting somekind of an output on a TV screen. Some graphics were available too, but comparing that with what we now understand through "graphics", well... there is a long way behind, and a huge hiatus to cover in a very large book, that we might call "**The Computer Graphics Hystory**".

I'll skip the details and move back to **GAMBAS**.

While the first BASIC dialects were bound by the existing technology at that time, such as CPU low power, extremely expensive memory modules (RAM), now things got way above those limitations.

Programming itself, evolved from what is usually called "stepped execution" or "**top-down**" meaning that a computer could execute a program consisting of only a predefined sequence of code lines (instructions), **line by line, from the beginning to the end.**

Since the industry developed extremely fast and now the computing power available for the average people exceeds by far the real skills in using a computer, almost all "traditional" languages, including the BASIC family, moved to a different paradigm: OOP. That is: **Object Oriented Programming.**

Simply explained, this kind of programming, does its best to mimic the Real World, where we have objects and many other entities that can be defined through properties and actions. While at the beginning we might have a list of different entities very unique in their own way, the Real Life teaches us that there actually are groups, or families of entities, "objects", that share a large common base of "properties" and "actions", and they are individualised only through a small set of particular properties and actions.

This concept of "families" in programming translates into CLASSES.

The concept of "Object" in programming applies to many different "things".

GAMBAS, as we can read in its presentation, is an OBJECT ORIENTED BASIC DIALECT.

"GAMBAS is a free implementation of a graphical development environment based on a BASIC interpreter and a full development platform. It is very inspired by Visual Basic and Java."

Where Gambas stands

The civilised world

[F 27-03-2020 h 11:30]

In my view, "*a civilised world*" means way too many things and the list of all those, might take me lots of pondering, lots of time and lots of pages to list them all.

It is easier to shrink all the countless words **in three fundamental principles:**

- 1. RECOGNITION (GRATITUDE).** I'll skip the long story and tell a short version of it: **(1) Cognition.** Learning something until Knowing that something. **(2) RE-Cognition.** Going deeper into knowledge of something that you know at some level, but want to go deeper into the nuances of that knowledge. It is THAT SIMPLE! **(3) Public RE-Cognition = GRATITUDE.** This is the most advanced stage of cognition, when we publicly acknowledge that we have learned something from someone (a book, a teacher, a mentor, a situation), be it directly, through a close contact or indirectly. This last form of Cognition is actually a combination of the three fundamental principles of what I call "**Civilised world**": **Sharing (distribution)** with the others the information about what you've learned and where from. It is also a **contribution** so, the circle closes this way. You can find **this pattern** everywhere in the Nature, IF YOU really OPEN YOUR HEART and YOUR EYES. All three of them! Including the "Third Eye"!
- 2. CONTRIBUTION.** Contribution is what YOU do. Your actions your results and what exactly you have to offer to the community, and what you actually OFFER.
- 3. DISTRIBUTION (SHARING).** This is simple to understand: When you give away something you "crafted" — so to speak — **something that has some part of YOU** (Time, sweat, skills, knowledge, LOVE, passion).

Even passing along a free ebook, is an act of DISTRIBUTION.

Gambas and the civilised world

[F 27-03-2020 h 11:30]

Since I presented above my view on what is a civilised world, let's see if we can find those PATTERNS in Gambas:

- 1. Recognition.** Gambas is a collective piece of work. As you start the IDE, you can see a long list of contributors. **Benoît Minisini** is somekind of a director, like in a movie, or a ballet, or an orchestra: to be able to keep the tempo, you need to have someone in front, that is capable of seeing "the big picture" and remind you the "tact", the "tempo" "the heart beat" that sincronizes all the individual contributions. **I acknowledge here all the individual contributions that made Gambas what it is today.**
- 2. Contribution.** **Look at the IDE.** There's enough there to tell you about "**Contribution**". If you are interested in *my contribution* since February 2020 when I started coding in Gambas, I brought some contribution, beyond this book I'm working at now. Click here: "**MoSeS-ZENLA**". I guess it's a good start. :)
- 3. Distribution.** While Gambas itself is under the GNU-GPL Licence, I guess that tells enough about what "**distribution**" means in the particular case of Gambas.

So... Where does Gambas Stand?

[F 27-03-2020 h 12:20]

I presented above the three principles of what I call "**Civilised World**"

I also applied those principles to ponder Gambas.

CONCLUSION:

I believe that Gambas is a part of a CIVILISED WORLD that is emerging from the old world.

While the new world is still in its infancy, *I am very much aware of that*, we already have the essential marks, milestones and tools to continue to develop this NEW WORLD.

It is up to me and even if considering only this book, ***my choice is obvious.***

But it is also up to you if you join this world that opens in front of you.

It is all a matter of **CHOICE.**

What would be "The first step"

Learning the Language

[F 27-03-2020 h 12:30]

I believe I explained the idea of COMMUNICATION, and the MEANS to do that, in chapter "**Where to start**".

I also believe that the same you learned in school your primary language, applies as well here.

You need to repeat the same kind of tasks like you did when you learned your primary's language vocabulary, grammar, syntax and all the other.

Unfortunately, Gamas has a major issue: the documentation.

It is scarce and spread all over the internet. It is a work in progress and unfortunately, the manpower needed to get to a minimum level of documentation, is way below what might be called "close to the needs".

This is why I'm writing this book.

While for me was an extremely painful experience to find useful code and clearly written documentation, I believe that this can come to an end.

Maybe my contribution will be like drop of water in an ocean but I guess what matters here, is **THE CHOICE**.

My favorite choice for the first steps

[F 27-03-2020 h 12:40]

While there are countless approaches to learn something, *my favorite choice is:*

Programming Gambas From Zip

by Gerard "Gerry" Buzolic

Download: <http://wordpress.gambas.one/a-book-by-gerry-buzolic/>

While I have some programming background, especially RapidQ programming, actually I am a self-taught programmer and nothing more than an enthusiast. Nothing like a professional programmer or anything like that.

So, being somekind of an "Average Joe", I guess that my choice will be obvious if you are looking for something to "warm you up".

The book is wonderfully written and I guess that I am more than "qualified" to make this statement, since I write for well, over 50 years, I published more than 30 books so far, regardless the books written for the Apps uploaded on Gambas Farm.

If programming is something very new you are approaching now, this book is what you really need!

Assuming the other possible scenarios:

[You have enough experience to make your way through](#), only with a few "bumps". Well, Gerry has that covered too:

There are 30 Projects, discussed aslmost line by line so, if you want to move stright to coding, there you are!

Unpack them, load them into the IDE and press F5! That's it! You're in business! :)

The Second Step

[F 27-03-2020 h 12:30]

While in the chapter "What would be "The first step"" I outlined what I believe would be the most beneficial "**First Step**", now let's see what might be the "**Second Step**": **John W. Rittinghouse**.

I know that reading a book is somehow uncomfortable, mostly in programming, where the fun comes from writing code, testing it and improving it, the first stage though, is **LEARNING what others accomplished**.

Why is that so?

Well, maybe reinventing the wheel suits some people, but I seriously doubt that this is the way of **being proficient**. That is, **productively efficient**. See Alvin Toffler's books on that.

That is WHY it is better to know what others did before.

A book, is the shortest way to efficiency since it squeezes the time spent by the author to get to the point where he is: you read the book in... say a couple of days, or maybe a week, but the book required many months for acquiring the experience needed for writing it.

So, the "time compression", might well go beyond a 10:1 factor. Meaning that what you can achieve in one month, it took the author 10 months to achieve. Going further, what you achieve in one year, took 10 years to achieve for the author.

The book is 15 years old, I know that, many things have changed since, still, the code explained in this book, will help you figure out how to get your project(s) done, at least some portions of them.

A beginner's Guide to Gambas

John W. Rittinghouse

Download:

<http://distro.ibiblio.org/vectorlinux/Uelsk8s/GAMBAS/gambas-beginner-guide.pdf>

While this download might be an old version, it's the only working link I could find. Enjoy it!

Communication: Third Step

[F 27-03-2020 h 13:30]

While coding might seem to be very much like a monk's life, living hidden behind the walls, in front of a computer, we all soon discover the true nature of human beings: **WE ARE SOCIAL ANIMALS.**

So, what about the social part of Gambas?

It took me many frustrating hours to gather all the tiny pieces of information about "**Where, What, How**".

Does it have to be the same for you?

As you can see I have a different opinion on that matter so, here is a list of the resources you need in order to increase both efficiency and enter the community, by exchanging information, asking and answering questions.

Best Resource:

Forum Gambas One:

<https://forum.gambas.one/>

Users Mailing List:

<https://lists.gambas-basic.org/listinfo/user>

Search the lists:

<https://lists.gambas-basic.org/cgi-bin/search.cgi>

Gambas One, Gambas Farm:

<https://gambas.one/gambasfarm/>

Blogs

Blog Gambas One!:

<https://wordpress.gambas.one/>

John Rittinghouse blog:

<https://beginnersguidetogambasdotcom.wordpress.com/>

German and some English blog with Search:

<https://gambas-buch.de/dwen/doku.php>

Peter Bauer:

<https://www.bitkistl.com/>

Gambas Documentation

<http://gambaswiki.org/wiki>

<https://en.wikibooks.org/wiki/Gambas>

<http://gambas.sourceforge.net/en/main.html#>

<https://gambas.one/playground/>

A wonderful tool: Gambas Classes

[F 27-03-2020 h 14:30]

While there are many tools that might help you along the way on "[Gambas Farm](#)", I have a "**choice of the heart**":

Gambas Classes

Gambas Classes, beyond being a tool, is a use-case for some controls and how to deal with the online documentation, in a very nice organised manner.

The author however, [Charlie Ogier](#), is a constant contributor and the Apps he uploaded, can be a very good start for developing complex applications. Clicking on his name, will lead you to the whole list of available Apps he offered so far.

However, as I said above, my "**heart's choice**" IS "[GAMBAS CLASSES](#)". Clicking the link, gets you directly to the Download page. Pick it up and give it some minutes. I'm sure you'll love it!

It provides you the available documentation on all the classes, with a search function that will speed up the process of studying and making the right choices for the projects you plan to develop.

Sadly, I found the App a little too late, when I was already exhausted by jumping from a page to another, back and forth endless hours trying to figure out how to do something...

Since I'm sort of a tough nut to crack, I give up only when I see a clear end of the road.

Since nothing like this seems to be darkening the horizon, I go on.

See the next chapter! :)

About the GaP-BG Project

Obviously, the short form of this project looks like a joke and it really is. "**GAP**" means from a different approach, "something that is missing".

And it sure is. A proper documentation.

However, there are enough people in this community that use Gambas, and therefore, **WE can fill in this GAP.**

Now, going beyond the obvious side of this joke, the **GaP-BG** Project, has behind a similar work I used to put into another BASIC dialect: **RapidQ.**

RapidQ, is something very similar in many ways to Gambas, except that it was meant for Windows **(a)** and **(b)**, development stopped somewhere in 2003, when the lead developer, William Yu, got hired by a company and RapidQ became a proprietary project. That is, "Closed Source" or, commercial Project.

The free version is still available, but as I said, only for Windows.

Between the autumn of 2000 and the fall of 2005, I worked a lot with RapidQ, I developed a number of Apps but mostly, I was the Lead developer of the "**RQDP Project**", meaning **The RapidQ Documentation Project.**

After 2005, the lead maintainer and developer became a friend of mine and co-worker at the project, **Andrew Shelokovenko.**

His site is still up and running so you can check it out, although most of the available parts are in Russian.

So, as I said above, I gathered some experience in developing such a Project.

This book, is the first book in what is my intention to do from now on: a series of (assuming it is possible...), **short books**, on various themes regarding Gambas Programming.

While "**short books**" looks nice and sounds nice to me now, even doable, only TIME can tell what will get out of my idea. For now, here it is, the

first book!

There are many aspects to cover and I am aware that this is a very long run call. More than that, it requires a large team for making things happen.

Maybe life will point me towards other directions, or who-knows-what might happen one hour or less from now.

Still, I declared my intention and supported it with the chain of actions needed for the project to come into Reality.

A simple question pops up in my mind:

Will you join me in this Journey?

About this book

Writing the book

[F 27-03-2020 h 16:20]

I have to say that: writing the book was the easiest part of the job. This is true, for at least two reasons:

1. **Writing practice.** I started writing 51 years ago. That is some practice! I learned how to be productive, how to write fast, even how to write on-the-fly, meaning without a "TOC" or any outline. When you do something for many years, some internal mechanisms get better and better and you do the job somewhere deep inside you, sometimes even without knowing that at a conscious level. However, the outline is somehow done there, inside, the chapters are already sketched and all you need to do is act for making them happen.
2. **Writing in English.** That was also easy enough, given the fact that I constantly read in English many articles on various themes, mostly Personal Development, which is my choice for as long as I can remember. Personal growth, is in my views the only thing that really matters for anyone, whether they are aware of that or simply live from day to day... I use English almost on a daily basis, for almost 50 years. 49, to be specific...

What was awfully hard

[F 27-03-2020 h 16:30]

While writing the book took me somewhere up to 10 hours including the graphics, *gathering the knowledge and information needed to write the book*, took me *over 270 hours*. That is a total of 280 hours, meaning 35 working days, 8 hours/day, every day. That leads to 7 workweeks. Close to two months of my life.

And this is only about *the theory*. It took me another 150 hours of programming, to be able to understand what am I dealing with. That is, about 430 hours of work.

While I am used to work about 12 hours daily — *and there still is room for better, Nikola Tesla worked 22 hours/day* — the fact still remains: average people, barely work 6 hours/day and feel exhausted every weekend. At a rate of 12 hours/day, it took me 36 days to get where I am now with Gambas. Less than 6 weeks.

How much do you think will it take you to read this book?

Grossly overrated, **about 3 hours**.

Now, given the fact that inside the book there are clear references to Gambas resources that you'll want to study, that will take more time.

Anyway, instead of repeating my experience in crawling and digging for the basic information, ***you have it at your fingertips!***

That is what I meant in the chapter "**The Second Step**", when I said that I know that reading a book is somehow unpleasant, but ***it worths doing that***.

Otherwise, you would have spent the same amount of time looking for breadcrumbs and then trying to put them together into something meaningful to you.

Instead of "Good bye"

[F 27-03-2020 h 15:10]

Long time ago, I remember I read this in a book:

"Every Good Bye, means dying a little".

It so happens that I felt that being true.

So, instead of saying "Good Bye!", I say:

See You soon!

What will come next?

Well, for now, it's a little above my horizon. I have many things to do besides programming in Gambas or writing Gambas books so TIME, is of essence.

The idea for now, is to gather some snippets of code from my logs (over 15 for now!) and put together some guide for using the controls I already used successfully.

How exactly am I supposed to do that, it's something I still need to figure out.

I suppose that the best approach is to start from explaining "the nuts and bolts" of [DirLister](#), which is already available for download (*clicking the link gets you there! :)*).

There are many things there that might be used for branching the development of many different applications.

The only problem is that it already is somekind of a large application (over 1800 lines of code...) and I still have to figure out the best approach.

The best idea I have now, is to write down the chain of events and then

step into each link in the chain, explain the code that does this and that, at every moment of the chain of events.

Beyond that, I still have to figure out a lot of things.

Anyway, I truly believe that studying an application that has 1800 lines of code, offers enough examples of "**How to do**" this and that.

Since **DirLister** uses some of the most frequent used controls, the way I used them is a showcase. Comparing the code with my view on the program flow, I believe will give you a pretty good idea on how to use a snippet, a procedure or whatever might suite your application's requirements.

For now, I stopped the development because I need to work with DirLister and get the perspective of the user.

There are many things that might pop up when using an app, such as some performance improvements, some interface layout that "does better", some feature that at development stage was impossible to foresee but usage points to it and so on.

Getting to me

Exchanging information, requires at least two people so we need means to communicate. I said that at the beginning I say it again, now.

Nowadays, when the Internet became somehow "granted goods", available at reasonable prices and mostly everywhere, there we are!

Below are the means you can contact me.

Ignore the pages in Romanian, as you can see, it's OK if you write in English.

Maybe I'll start a group on Facebook, but I'm still pondering that.

The Gambas Farm, Vendor=MoSeS-ZENLA:

<https://gambas.one/gambasfarm/>

User's List:

<https://lists.gambas-basic.org/listinfo>

Twitter

<https://twitter.com/sserbanro>

YouTube

<https://www.youtube.com/user/serbansvideo>

Facebook

<https://www.facebook.com/serbanstanesuro/>

<https://www.facebook.com/sserbanro>

LinkedIn

<https://www.linkedin.com/in/serbanstanescu/>

This is it!

See you soon!

Waiting for your feed-back!