Did you feel learning to program with turtle graphics was useful? [closed]

Asked 16 years ago Modified 9 years, 10 months ago Viewed 5k times

11

As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, visit the help center for guidance.

Closed 12 years ago.

I'm preparing to teach someone to program. When I learned the course material, I used turtle graphics for the first few exercises. In reading introductory textbooks, I have not found one that uses the technique. Did others find this approach helpful? If not, what is a better way to learn to program?

turtle-graphics logo-lang

Share

Improve this question

Follow

edited Feb 5, 2015 at 21:37

Steven Doggart

43.7k ● 8 ● 70 ● 107

asked Dec 6, 2008 at 7:37

minty

22.4k ● 40 ● 91 ● 106

c.f. stackoverflow.com/questions/1003841/… – skaffman Nov 20, 2009 at 12:20

18 Answers

Sorted by: Highest score (default)

7

I think it depends on age of the target group.

If they are children (I would say up to 12-14 years), doing any easy graphics is a good way to motivate them; on the other hand, don't expect them to learn much about real programming or algorithms.

If they are teens (14-18), it's perhaps still good to use some algorithms that give pretty results (for example 3D or fractals), but since they are older and capable of more abstract thinking, I don't think 2D turtle graphics is interesting enough.

If they are older, doing any graphics is a distraction. At that age, they should have enough inner motivation to learn without anything fancy.

To sum up, I think that fancy graphics serves more motivational role (that you see what you did fast, and it's easy to show others what can you do with a computer) than learning role (that it would make learning real programming easier).

Share Improve this answer

Follow

answered Dec 6, 2008 at 9:19

J S

6

In the late 80s, before I was programming in C, I was programming in Applesoft BASIC and Logo. As a child I thought the turtle was great because it make programming simple. If I decide to teach my children Logo I will probably start here to get an actively developed Logo interpreter.

Share Improve this answer

Follow

answered Dec 6, 2008 at 9:01

Nathan

656 ● 4 ● 7

3

The key thing about LOGO is user-defined functions. It is very good at conveying that, as long as you emphasize it. Show interactively how to draw a square, then make a new word called square. Then show how you can draw patterns using square. Then make those patterns into words, and so on.

Share Improve this answer

Follow

answered Dec 6, 2008 at 9:22

Daniel Earwicker

117k ● 38 ● 208 ● 286

You could do worse in teaching programming than using a tool like Scratch. It's a drag and drop programming

3

interface and can be used to teach basic concepts of programming with some fun visual results (as can be seen from the gallery on their website).

Rob

Share Improve this answer

Follow

answered Dec 6, 2008 at 14:32

RobS

3,847 ● 29 ● 34

Scratch is great. It also has built-in turtle graphics as well (the pen commands) – tfinniga Dec 11, 2008 at 21:22

2

Logo gave me a very clear picture (no pun intended) on how recursive functions would work, and since I was doing assembly programming at the time, the need to return to the previous state when returning to a method became very clear with Logo.

Recursive implementations of things where also very easy to see the effect of.

Share Improve this answer

Follow

answered Mar 6, 2009 at 20:10

Lasse V. Karlsen

391k ● 106 ● 646 ● 844

1

I wrote script/code in a c-like dialect for a game called Doom2 before I knew what programming was, so when it came to seriously learning about concepts such as pointers, inheritance and polymorphism I found the basics



a breeze because I could construct a mental model to not only help me understand, but also appreciate how cool things like pointers and arrays are.

A friend of mine is a good programming student, but he gets frustrated when he can't visualize an algorithm working, when I was starting to help other students I found they had the same problem, if they can't see something working it's harder to appreciate as a fledgling programmer, the same friend eloquently suggested I "Show 'em some crazy pimp shit and *then* show them how it's done". He's right, even if someone really wants to learn something they'll be able to draw on more mental energy if they think what they're learning lets them do awesome things.

My best bit of advice is this: **AT THE START SPEND AS LITTLE TIME PROGRAMMING TO THE CONSOLE AS POSSIBLE**

It makes you feel constrained and your efforts appear futile, only after you appreciate it as a front end should it be used for learning to program. I wouldn't use logo myself because I don't think it can teach concepts such as the aforementioned polymorphism or inheritance nearly as well as other methods, I know a friend of mine is teaching a teenager how to program using XNA in a wrapper, I think anything that can let you blit an image to the screen is fine. That way you can see why you'd want an abstract base class called EnemyEntity with behavior that's inherited by zombie and dog etc. It's not that the

concepts are hard to understand, it's just that at first they're hard to appreciate.

I could go on but I think that puts across what I've learned by teaching others. I think using graphics in teaching programming allows students to gain the ability to build mental models of intangible concepts faster than any other.

XNA If you want to teach C# that's an amazing graphics library, just write a wrapper sprite class to hide as much complexity when first starting out and teaching concepts.

SDL A lower level library if you're going to start with c++

Share Improve this answer Follow

answered Dec 6, 2008 at 10:04

Tarks
4,237 ● 6 ● 39 ● 44

1

During one of my first-year computer science papers we used Java to create fractal patterns via a turtle object.

It was pretty fun to see visually whether or not we had correctly implemented the algorithm required to produce a certain pattern. However, so answer the main question, I wouldn't say that programming via a turtle is useful. I'd say the best way to teach someone to program is to get them to build their own app to do whatever they want it to do. This gives them creative control, plus if they get stuck they can learn how to resolve a problem.

Share Improve this answer

Follow

answered Dec 6, 2008 at 11:30

hazexp

1

I strongly suggest to start with a interpreted language like Logo (not compiled) because of the quality of the error messages. Reading error messages is very important in this process. Also, at the easy level, Logo allows you to run your instructions one by one in direct mode and carry them to your procedures when you get the expected results.

@ Alex: MicroWorlds is a commercial version of Logo and it does exist in English, Spanish, Portuguese, Italian, Russian, etc. it's a big plus if you are not a native English-speaking person.

Share Improve this answer

Follow

answered Mar 27, 2010 at 13:18

Alain

11 ● 1

1

LOGO is not only Turtle-Graphics. There are also other interesting concepts in it which come from LISP. 'Turtle' is just icing on the cake and the "imperative" side of Logo. :)

Share Improve this answer

Follow

answered Mar 28, 2010 at 11:34

Bert

11 ● 1

0

I learned to program in BASIC by writing simple programs drawing faces (I mean circles and squares) on the screen. Somehow the whole turtle programming was never my thing, although a few of my friends learned that way. Later on I moved to Pascal, then to Delphi, Java and C++/C#. In my opinion the trick is to "wow" your student and impress/empower with potential things that you can accomplish by writing your own programs. I would actually demonstrate some GUI programming or game programming. It's much easier to learn the basics by keeping the end goal in mind.

Recently I came across SmallBasic - a cool programming environment for kids designed to teach concepts. I would give that a try. It comes with a pretty complete paper describing how to use it.

Share Improve this answer

Follow

answered Dec 6, 2008 at 8:15

Filip Frącz

5,937 ● 11 ● 47 ● 67

0

When I got my first computer (VIC-20) and started programming it was very hard to explain to my parents what I was doing.

My mother tok a course in computing preparing for a project of computerizing the library she worked in. They had a couple of classes introducing them to programming. After learning LOGO she came home and said that she suddenly understood what I was into.

So LOGO with turtle graphics brought us closer together!

Share Improve this answer

Follow

answered Dec 6, 2008 at 9:36

Guge

4,599 ● 4 ● 37 ● 48

0

I did a "computing for kids" course in the late eighties, and there was an extensive section on turtle graphics using logo. In all honesty I was bored to tears, and learned virtually nothing from it.

I think "programming the turtle" might work better for someone who is artistically inclined, or hugely into geometry, but by and large, there are far more interesting problems to attack, even for kids.

Share Improve this answer

Follow

answered Dec 8, 2008 at 2:14

cygil

3,644 ● 1 ● 20 ● 10

0

Ah, the memories of good old Logo. I think I got more of a geometry lesson than a programming lesson out of it, e.g. figuring out how much to turn at various points to produce a particular shape, design or pattern. It may work if you plan on mixing geometry with the programming, but if the person doesn't have the basics of geometry, e.g. what is a square and how is it different from other 4-sided shapes, what is a triangle, etc.

Share Improve this answer

Follow

answered Dec 10, 2008 at 2:09

JB King

11.9k ● 4 ● 40 ● 49

0

I used logo and turtle at school too, a great introduction.

It looks like our kids will be getting a slightly updated interface with Microsoft Kodu. It looks very impressive. It's an icon based programming language made for creating games that runs on X-Box Live.

Share Improve this answer

Follow

answered Mar 6, 2009 at 20:07

Tristan Warner-Smith

9,771 ● 6 ● 49 ● 75

0

I'm currently learning python and using a little bit of turtle. In labs we haven't used it, but our homework does. It's nice to know it exists, and it's a good way to get certain commands and syntax in. Overall I don't feel it was completely necessary though.

Share Improve this answer

Follow

answered May 6, 2009 at 21:03

TIMBERings

165 ● 1 ● 2 ● 8

0

When I was young, I found it very interesting. It was one of the first programming languages that I've learned, even though I've used it for about two days. It started my interest in programming.

Nowadays, I think the syntax is a bit unclear because most statements are abbreviations. Nowadays, computers are far more powerful thus the language could profit from clearer statement. Another factor is the native language of the person who is learning to use it. If English is not your native language then Logo becomes a bit more complex to understand. So if you're teaching Logo to children, make sure they're familiar with English terms first. (Quite easy if you're a native English-speaking person. More complex if you're originally Dutch, German, French, Portuguese. Even more complex if you're Russian or Chinese because you'd have to adjust to a different character set too.)

Share Improve this answer

Follow

answered Aug 26, 2009 at 21:16

Wim ten Brink

26.6k ● 22 ● 91 ● 162

0

I have just begun teaching my 7-year-old how to program using Logo, and he is having a load of fun with it. The commands are easy enough for his limited reading ability and he just loves drawing cool pictures using the turtle graphics. I was amazed at how well he retained what he had learned using it, so I feel it was a good choice for his age.

For older kids (or adults) other languages might have more advantages as a beginner language though

Share Improve this answer

answered Nov 20, 2009 at 12:14

Follow

Serge Meunier
162 • 3 • 7

0

Personal experience, YMMV...

My first encounter with a computer was turtle graphics in my early teens. I loved and was immediately hooked. (Perhaps because for the first time someone [something] did exactly what I told it to do?)

The visual and instant feedback made me want to do more and more. I really wanted to figure out how to replicate the pictures I saw in the book I was using. Without me even classifying it as "work", it slowly built up my early programming skills and my confidence I could learn on my own.

I credit it with sending me in the path I'm in today, a happy software developer who can't believe I get paid to do this work (I know, I know - all corporate snickering aside, I like my work).

As I said, YMMV.

Share Improve this answer

Follow

answered Apr 4, 2012 at 0:36

Christian Garbin
2,522 • 1 • 23 • 31