MarblePunk
- A fictional programming language to learn coding

Jan Niklas Bingemann (University of Paderborn, jnb@uni-paderborn.de)

I designed a programming language, in which programs can be build with marble runs. Looping track pieces are loops, branches are if-statements and so on. But instead of defining a visual programming language for this, I instead defined the programming language on a fictional universe, in which the laws of the world define the language.

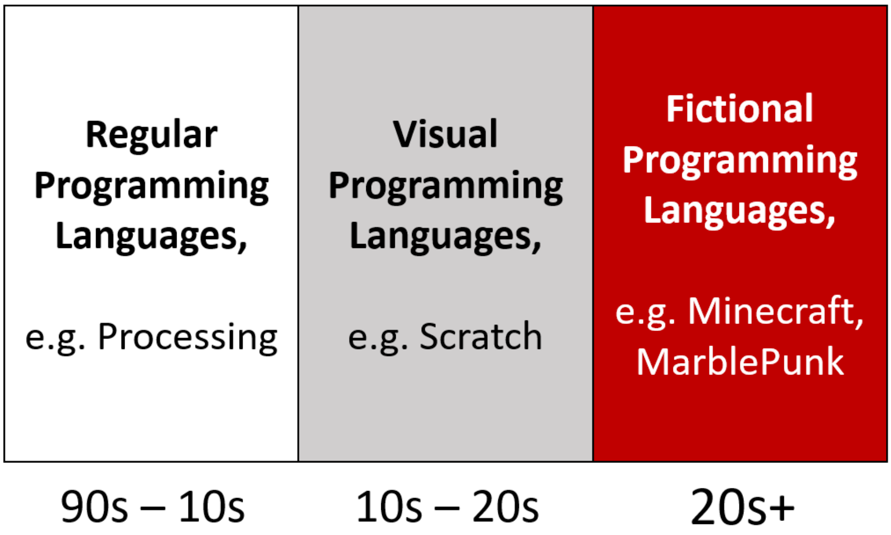
The language could be adapted as a visual programming language but could also be adapted in the context of computer game like Redstone in Minecraft. This makes the language very flexible and dynamically applicable to different forms of media and I hope it could be used to teach programming through different forms of media in the class room.

Reasons for MarblePunk, or: why Minecraft's Redstone holds the future of computer science education

There is a trend in computer science education. In recent decades, educators have moved from teaching actual "hard" programming language to more softer versions like Scratch, in which programming comes with nice visual components and is integrated in game-like or creativity-inducing contexts.

A more recent development is to integrate programming into a narrative and by that make it more exciting. An example for this is Coding with Minecraft, in which people use visual programming to solve simple programming problems in Minecraft. However, the programming there often feels like an afterthought, because programming is done visually and it is not part of the game world. For crafting items we have a crafting table but the visual programming components are not explained in the context of the world. They don't make sense and therefore, don't feel "real" in the game.

Instead, redstone, which already has elements of programming, is a part of the world and feels much more natural to players. It is only a natural development step here to think about ways to integrate programming more consistently in the game world. And I think, regular and visual programming both come to an end here and I think we need a new kind of programming language.



Regular Programming Languages, e.g. Processing	**Visual Programming Languages,** e.g. Scratch	**Fictional Programming Languages,** e.g. Minecraft, MarblePunk
90s – 10s	10s – 20s	20s+

What is a Fictional Programming Language?

In contrast to regular programming languages, the meta-model of fictional programming languages is a fictional universe by which natural laws the programming language is defined. In the context of the fictional universe, the programming language should be able to be naturally derived from the natural laws of the world.

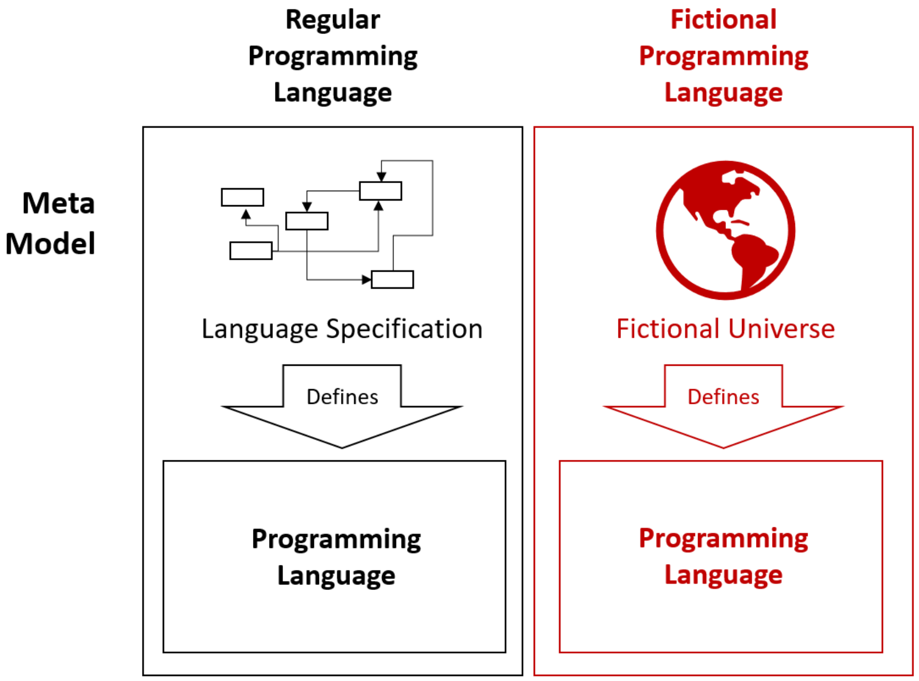
That also means that by exploring the fictional universe, the programming language would come natural to one. This could probably make learning programming much easier, if it is just a natural part of a fictional world, which is great.

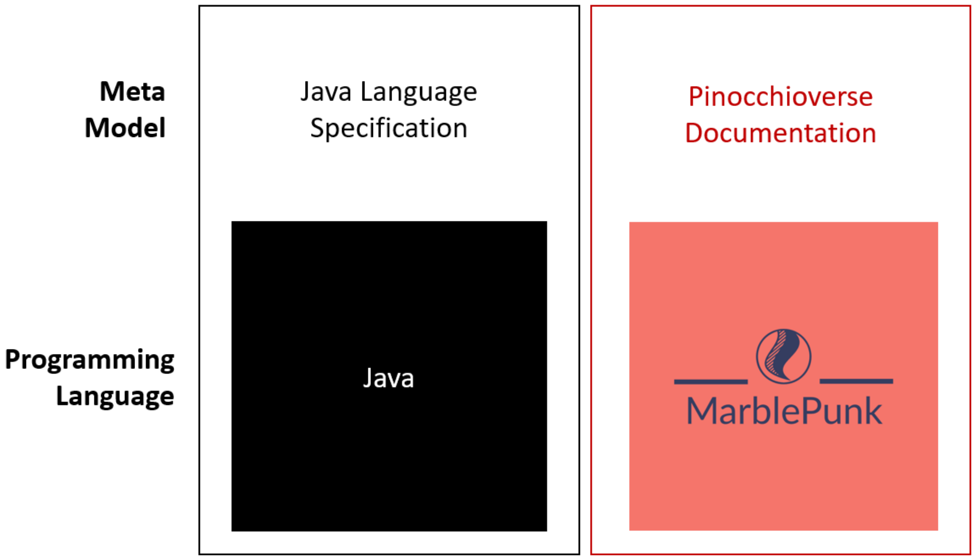
That also means that the programming language only really exists within the world of the universe, which, however, is not much of a problem, because it also accounts for any other programming language. But instead of regular programming language, this world can be traveled to through stories and by this, make the programming language directly expieriencable instead of being something abstract.

These advantages could be used for teaching programming in schools and in this topic, the school material, which would be used in these contexts would have the advantage, that it could be any medium really. It could be a book, but also a computer game or a VR learning environment. With this, the integration of different forms of media in the classroom could be much more flexible and dynamical and by this, more convenient and likely to apply in practice.

Other examples for fictional programming languages are Redstone in Minecraft, more or less the magic system in Harry Potter and many more examples in video games.

I think fictional programming languages could be perfectly suited for teaching programming. What they mainly have to offer, is immersion, which has been an aspect to much neglected in the design of programming languages and could move the whole area of computer science education forward.





The Pinocchioverse - The Universe that defines MarblePunk

The universe, which defines the MarblePunk serves also as the place to run it. Its like the Java Meta Model and the JVM in one.

The idea of the Pinocchioverse is the following: In this version of fictional history, Pinocchio did not come to life through magic - instead, Geppetto found a magic ore in the nearby forest, with which he could automate his machines and even do simple calculations. With the help of the marble ore, he build his new son: Pinocchio and together, they shared their knowledge with the local village folks to make all their lifes better.

And the marble ore turned out to be so powerful, that the digital revolution could already be started in this medieval setting. In a way, by listening to the story, everybody can participate in the unfolding of the digital revolution in a more easy to understand environment and hopefully, they can by this understand better what is going on around them. But let's already see how it actually works.

Marbles

The workings of marble tracks are at the heart of marble punk.

They work based on touch and writing symbols on them. In general, marbles have access to an infinite amount of energy inside of them, which can be activated if one marble is in contact with another marble. If this is the case, the marble can do stuff based on the symbol, which is written on it. This can for example be the rotation symbol. Then, the marble would start to rotate.

Another way of activating the power of the marble is using symbols that are not the rotation-symbol. If two marbles share the same symbol, they are activated, too. This also works if a marble rols over a symbol. All mechanics of further sections are based on this.

Marble Runs

Marble runs make use of the properties of the marbles described in the last section to automate devices. They can be build with the basic track pieces like loops, branches, linear pieces and curves. Seals can be pasted on these track pieces. If the marble runs over an action seal, the action is triggered.

A seal could for example trigger the feet movements of Pinocchio. A marble run could now be constructed to help Pinocchio through a hay maze.

Additionally there are certain mechanical items based on marbles. Mechanical items like buttons and gearing wheels can be easily build with marbles and can be used to build simple machines. In combination with the marble tracks from last section, they can be used to construct more complex devices like a remote-controlled car.

But you can also do simple calculations with the marbles and by this do a little programming. The following marble run is a sort algorithm (if run multiple times in a row).

Further explained here (also with illustrations): https://pinocchioverse.org/documentation/

Exercisebook

To better talk about the idea, I described the idea in a fictional universe, which I called the Pinocchioverse and only later realized that defines a programming language in itself.

As a first adaption of the universe, I created a free, open-source exercise book, which can be found here: https://pinocchioverse.org/teach/

In it, one can read about how Geppetto created Pinocchio with marble runs and follow Pinocchio on great adventures while learning basic concepts of computer science and the digital.

The Marble Revolution and how to experience it first-hand

In the exercise book, we came to now the age in which Geppetto and Pinocchio discover the power of marbles, how does it go from there? After all, the idea behind marblepunk was to witness the unfolding of the marble/digital revolution (or at least something similar to it) in the class room.

For that, Marblepunk (like Steampunk) comes with different ages, starting with the early marble age, which is set in a somewhat medieval setting, in which Geppetto discovers marbles, over to a mechanical marble age, to a digital and beyond. It could be thought of integrating this into group works for the classroom or a separate exercisebook. However, I haven't fully worked out this part yet.

Conclusion

I created MarblePunk and more generally speaking, the Pinocchioverse as a educational environment, which could possibly be adapted for every kind of media: books, computer games, VR environments, etc. to provide a new approach for teaching computer science. Instead of having to learn concepts in theory, in Marblepunk, they can be actively experienced. Programming for example, isn't an abstract concept but comes as natural as a practical art or craft.

My hope is that it could provide a new way to teach children about the digital world and empower them to change it, too and ultimatively, to become mature actors in it rather than mere consumers.

With that, MarblePunk is not specifically aimed at children, but for people in general. If the idea would grow, maybe people could more easily get an understanding of the digital age we are living in and learn that they can actively participate in its development.