

Flooding the desktop with learning tools

Martin Abente Lahaye

>> It looks good from here. So, we start in 5 minutes.

Mr. Martin Abente Lahaye: Yep. Perfect.

>> Okay. Perfect.

>> So, I think we can start as this is the last talk of today. So, we will continue with Martin Abente Lahaye. Flooding the desktop with the learning tools. Please.

Mr. Martin Abente Lahaye: All right. This is the last -- you will get this. I hope you can hear me well. Thanks for staying around. It must be late for some people. So, I appreciate that. So, before I begin with this talk, there is like one idea that I want to, you know, put you -- to put in your mind that kind of explains the motivation behind this project. And it's like how much value can we create by connecting existing dots? Right? That's like the whole motivation of this project.

And so, for those who don't know me, well, what you need to know is that for -- I was very lucky that for the last 10 years I have been able to work in these very small intersection between open source, education, and the best technologies. I'm still a member of Sugar Labs and a member of the GNOME Foundation as well.

So, what are these dots that I'm talking about? The first one is Sugar. In case you never heard of it. It's a desktop for Linux. But it was especially designed for kids to learn. It is also a huge set of learning tools that are also especially designed for the same goal. Both Sugar and the apps are based on completely different paradigms compared to, let's say, GNOME. Two examples of that is that kids do not access directly to the file system, but instead they use this abstraction code, the journal, which is basically a record of all the activities that they do. And they don't need to think about files and things like that. The other is collaboration that's deeply embedded in the user experience.

So, things like sharing files or doing activities collaboratively. It's very easy to do. It was originally released in 2006 as part of the one laptop per child project. But since 2008, it has its own community and it's been maintained since then.

And since this is a beginner's talk, I will be very -- I will oversimplify things a little bit. But, so, the other is Flatpak. And depending on who you ask. Right? So, for users, Flatpak is a better way to manage your app. Searching for apps, installing apps, running and updating your apps. And the nice thing is no matter what Linux distribution you are using, as long as it supports Flatpak, you can run the exact same app and they will behave exactly the same. And more importantly is that your apps can run more secure compared to other ways of doing this. And if you're a developer, it is a better way to package a list of your apps. And the nice thing is that you can only need to build this once and it will guarantee that it will run consistently in any other Linux distribution that supports Flatpak.

Then the very important thing to mention is that it's a lot easier compared to other ways to do this. Something that I like about Flatpak is that it's a distributed system. And it just means that many organizations can have their own backend and repository. And, you know, like a list of apps. So, we have like, for example, the endless one, Fedora. But there's a special one among specials called Flathub president for the user, it's a store to develop your apps. And for a developer, it's a place to distribute and maintain your applications over time. And for both users and developers, it's a very nice community to be part of.

So, how am I connecting these things? And here's like the premise. So, we have Sugar Labs that has hundreds of learning tools. But most of those only work inside Sugar. Because they are based on completely different power lines. And also, we have technologies like Flatpak. And Flathub that allows it to reach a much bigger audience. The premise is that Linux in general can benefit from the apps that Sugar has done over the last 14 years. And at the same time, Sugar can benefit from reaching a much wider -- wider audience. So, the way that I solve this problem is basically just created a small token a that actually allows Sugar applications to run in the desktop. And just gives you very small library and just gives you like a wrapper GTK application that acts as a middleman between the Sugar world and the desktop world. And it gives you some tools that you can use to are place functionalities that do not exist in the rest of the world, but they do in Sugar. And therefore, this application expects to have.

And we also generate almost automatically all of the metadata necessary for let's say packaging your application in Flatpak. So, how much value have we created with this project?

Here is basically a summary. So far -- and there's good reason why there's actually not more so far -- we have 20 apps that we're already -- very unique apps. And I'm going to talk briefly about every of these 20 apps now. But I want to share a little bit about the experience of how this project started and how, you know, it's going to move forward as I do the explanation. So, this basically started in October 2009. I had a lot of free time in my hands and in something that I really wanted to do since a few years now. And, you know, I just prototyped the idea of how this will work. And a few weeks later I was visiting Barcelona for the Linux apps Summit. And shared this with colleagues from endless and the community. And they loved it. Extra validation never does any harm.

Even while I was still there, they helped me with some very early reviews and helped me with some technical issues. And, you know, maybe a week later after I came back home, we already published the first two apps, right? One is Abacus. Which is literally what the name says. But it has many different presentations. It's interesting to expose kids to these. They can see fundamental sort of computational, how information is represented and how it works in the different presentations. The other is called Music Keyboard. And basically turns the kid's computer into a piano, right? You can select different instruments and you can record and share these, you know, your masterpieces with your friends.

So, after I, you know, got the first two applications published in the Flathub, I, you know, spent some time like writing a decent but complete documentation on how to reproduce this. And I assigned the Google Code-In with sugar Labs. And we did something kind of clever. We created a task for these kids to choose like any application they wanted. And they had to basically port it and port this and pub lush to Flathub, right?

But to do that, they will first need to port the app to Python 3 that -- it was required. And also port GTK2 apps to GTK3. And yes, there is still GTK2 apps in this world, believe it or not. And we have a lot of those. But it is amazing the amount of, you know, work that we did.

At the same time, you know, we fixed a lot of bugs because, you know, we wanted to make this app work as -- better as we could. And with things like many any bugs. And some of these applications did not have a release for years or many, many months. We make it work with newer versions of GTK and things like that. So, it was very -- very positive as well for the community.

Okay. Put your seat belt on. Because I'm going to go really fast here. I'm just gonna show you like all the applications that we actually got to publish in Flathub in this phase. Okay. This is Chart. It's basically an application that lets kids import data and visualize data. It's like very nice to just have like the very first steps about data visualization.

The next one. Color Deducto, is a game about recognizing patterns. But it is also about writing code that generates those patterns. So, it's very interesting. These codes -- this application was completely ported

from Python 2 to 3 to GTK 2 to 3. And also published in Flathub 100% by our students.

Dimensions, it's another like very nice app that is, you know, finding pairs of cards that matches some pattern. And it's, you know, just a very classic application for the game for, you know, training, exercising pattern matching.

Finance. It's, well, you know, like to learn how to do finance, you know, you have to do it right. So, this application -- what this application does is allows the kid to actually manage a real budget so they can basically manage their home -- their own budget at home and, you know, and they learn by doing like the real thing. Fully functional.

This is FotoToon. It was originally designed, you know, for kids to create comics. But we think about present day, it's more of a meme generator by today's standards. So, maybe, you know, some kids can create the next biggest meme with this.

So, FractionBounce, the name says it all. But it's an application where kids can understand or learn the concepts of fraction and percentages by basically bouncing this ball and it has to land in a specific, you know, one quarter or 50%. It's very simple. But it's nice.

Implode is just a Tetris-like game. These are kind of the family of apps that there is in Sugar that, you know, tries to exercise the problem, you know, the problem solving abilities. And the next one is basically the same, but the scenario is a maze. So, it's -- it can be very, very hard. Like I mean, this is -- and if you want to download this and go to the maximum complexity level, you will be getting nowhere.

Okay. And another app is Measure. Yeah. This is also one of my favorites. This basically takes the microphone as a sensor and it can help you visualize the sounds or voices, you know, in these waves. And it can also be used for tuning an instrument and thing like that. Something nice about it is that you can -- the kids can -- record this data and then, for example, like import it in the chart app and they can, you know, see the data that was, you know, stored. Yeah. Sensored.

Yeah. Memorize, yeah, at first it seems like just the classic, you know, memorize game where you have these cards and you have to find the pairs. But actually this is a tool for creating those games. So, kids can create their own games that they can define an org like card. Like in, you know, use images, sounds, and, you know, text-to-speech and they can create those games and then share to their friends.

Yeah. Physics is literally just a physics sandbox. So, kids can build the structures of machinery and then, you know, have the time of their lives just destroying everything. And it's really fun. Yeah.

And this is -- yeah, I have to say, this is my number one favorite. It's called Pippy. It's a very powerful Python editor. What's really nice about it is that it works very nice. It's very simple. And it comes with a lot of tutorials for making games and even GTK apps.

Another really good one, ReadETexts. It does a lot of things. To allows the kids to search for books from these like public domain libraries. And they can download the book. And this has this functionality where you have this robot read the book for you in many different languages. It's really, really good.

Yeah. Sliderule. It's another example of how Sugar -- there's a phrase that says for learning something, you have to learn it in more than one way. And, you know, Sugar has a lot of, you know, applications that teach you the same thing, but you -- with a different tool, right? To this purpose. This app lets you do calculations.

There's Speak. Basically a speech synthesizer with a face. But it has some very nice features. Like you can customize the face of the -- of the robot here. And also, anyway, you can actually use your face to do that. It comes with a chat bot called Alice. And it has a very funny sense of humor. So, yeah. I already tested this with -- with many kids and, you know, they love it.

Another one, it's called Story. This is about, you know, presenting kids with random -- with a random sequence of random images so they have to write a story that connects like all these images. So, it's funny because, you know, a lot of silly stories emerge from, you know, from the process. It's really nice. Another interesting one, Turtle in the Pond. If you look at it, and I have to admit that I have to improve the documentation. But it's a game about not letting the turtle escape from the pond.

But it actually it's about writing Go to make the turtle smarter. And how you do this? You can use it to write the code and import it here and test your algorithm to see how much smarter you made the turtle to be. And the last one, this is basically it's a dictionary. But it's also a -- like a very simple translator. And it can also be used to learn how to spell the different words in the different languages that supports.

So, yeah. Phrase two kind of finished around February. And, you know, at that point we brought in so many different apps that it like started to converge. So, we kind of know what was the, you know, all the different things that we needed. But the manifest, which is the recipe that you use to create the Flatpaks, there was still a little bit missing. So, I had good advice from Flathub admins. And so, I created a base app and I moved all the common pieces there. And, you know, like, I don't know if you can see -- you can actually read the -- what's in this image. But that's the full manifest for a general Sugar app. It's very clean and very simple to understand. So, it should be a lot easier now both to maintain and to port new applications. And in the process, also updating to the newer GNOME runtime.

And after that, there was a lot of releases of apps because of all the amazing work that the students did. So, I also updated all of that. And so, moving forward. Yeah. So, one thing, like I'm kind of imminent thing that you have to do is it's to update the tutorial because so much easier now to port this application. So, need to update that.

And now that I have data thanks to Flathub, I kind of know what are the most kind of most attractive or popular applications. Because there are hundreds of applications. And, of course, I don't have time to do it all by myself. So, I have to prioritize. But yeah. The one that you see in here, it's called Music Painter. And it's literally what it says, you paint music. And yeah, that's the next one that I'm thinking about.

So, conclusions. We have new -- 20 new unique learning tools. So far, over 6K downloads. Of course, that doesn't mean there's 6K new users. But, you know, some fit in between. I receive a lot of valuable feedback from Flathub maintainers, people from the community, users, you know, it needs to be very, very useful. And, you know, on a kind of personal note, for me -- for those who know me, they know how important Sugar is for me, right? And for me, this is -- like a personal attempt, you know, to preserve the legacy of what Sugar did in DOS, right? And there are many different -- many developers in the Sugar community did different things. For example, Sugar Icer, a complete rewrite of Sugar, but for the web. But I think that this is my way of, you know, preserving this legacy.

And, well, that was it. And thank you. And I don't know. Let's check the questions. If there are any.

>> Thank you, Martin, for a presentation. Yeah, you can find the questions in the other part.

Mr. Martin Abente Lahaye: Okay. Are there any academic research doing regarding actual usage of Sugar learning tools by pupils. How effective they are and what could be better -- what are the conclusions? Well, I'm not sure if there are direct specifically for individual applications. But because of the one app, there's a lot of evaluations like in different countries. And, you know, they have very different results. In the country, you have great countries like Uruguay where they use Sugar a lot. And other countries where for different reasons they could not really use it. I don't have a paper that I point to you at the moment, but I know people who do so. I can point you to them.

Yeah. The UI framework. I forgot to mention this. But all both the Sugar desktop and the applications are written in GTK and Python. So, yeah. It wasn't, you know, something completely alien to, you know, to port these things. How are games selected as candidates for porting for Flatpak? Well, so far, only now I do

have data, right? Just basic data from the feedback that I'm getting and also from the stats that I use from Flathub.

But at least for the free and initial phases I just make -- what is there in Sugar that doesn't exist not even remotely outside of Sugar, right? So, that was kind of my criteria. But now I have more criteria to move. Yeah, sure. Let's get in touch. Yeah, the -- Sugar apps for them. Yes. That is what I'm saying. Like big projects and very successful projects like -- in Uruguay have a lot of information.

All right. Oh, there's another question emerging. Newcomers, contributors? Well, basically, all the students from Google Go in were newcomers, contributions. Yeah. Of course, if someone is interested in, you know, helping me with this project, you know, you're more than welcome. And you know, trust me. I already did most of the heavy lifting for this. So, I can promise you, you know, to have more fun.

And okay. I think that's it. Thank you, everyone.

>> Thank you, Martin, for the just great presentation about the Sugar. And this concludes our today track with these talks. But in about 15 minutes we will have first social event for this GUADEC. It's a cocktail hour with -- it's a cocktail hour with Melissa Wu. So, stay here and prepare your drinks and see how good cocktails are made. And for the rest of the talks, please join us tomorrow where there are lots of good talks. And we start at 3 p.m. 15 universal time coordinate. So, see you tomorrow!

[End of day one captioning -- enjoy your cocktails!]

The cocktail will be on the same track 1, so you can easily stay here.