Roby Provost-Blanchard
Application for the Processing Foundation Fellowship
*Using Processing and game design to teach high-school maths*
December 2016

# Project Description

In order to introduce programming and facilitate math learning, my project is to create a serie of 10 exercises that combine high-school math and game design. Being a Fab Lab technician in four different high schools, I've tried to find educational way to better integrate and facilitate the learning of math through 3D modeling, technical drawing, etc. Processing being free and open-source, it efficiently lower the entry bar (ie: having access to a Fab-Lab) while providing an interesting and very fluid approach to understanding math and basic physics. These exercises would we be freely available in french and english (hopefully from a processing-affiliated website) for anyone to download, in a printable PDF form.

While it is possible to find similar tutorials online, this project objective is to ease the learning of high-school math (vectors, fractions, trigonometry, cartesian coordinate system) while introducing programming concepts (if/then logic, functions, procedural programming, variables, I/O, etc.) in a cohesive way. From experience, a considerable amount of teacher can be intimidated by new technology; the proposed assignments should not scare the teacher but convince them that programming is an easy and interesting way to teach their class. This project would also provide file-management tips and grading guide for the teacher. Each exercise would end with a summary of the concept learned and 2-3 quiz questions in order to validate the student's learning.

Not only does Processing provides a visual and reactive feedback, it also enable youngster to be creative with mathematics. Instead of answering generic problems, they're now drawing on their screen, creating a game, etc. This simple fact gives them much more motivation to learn the theory in order to reach their goal.

In order to to this, I will work closely with some high school teacher with whom I've already developed a few Fab Lab projects. Some of them already expressed that they would love to include programming to their curriculum, given that they would have adequate resources available to them. Also, it is highly likely that I could assist them in teaching the class, giving me very good insight on what seem to work or not. Working with different teacher will also provide me very valuable data as how much time should be spent on this programming concept, or how to translate the mathematical theory into something usable on processing. I plan on making a general draft of all the exercise, then show them to the interested teacher before they commit to the project. From that draft, we will work on different scenario and appropriate teaching methods. Ideally, I'd like to test these exercises with 3 teacher, in order to broaden the experience and make sure that the time allocated for each lesson is adequate.

Since 2014, I've been a Fab Lab technician in most of the labs in Montreal. Over those past 3 year, I've been teaching people from very different age and technical background on subject ranging from basic electronics to how to use a laser cutter. Since I joined Fab Lab CSMB, I have given many workshop to teachers and I have been able to sense how both Fab Lab and Processing could fit and be taught in their respective curriculum.

As for result, I'd like to get to be able to get  the student complete projects, along with somewhat detailed teacher feedback on their experience teaching Processing. Not only will I be able to fine tune the exercises using their feedback, I'd wish to get more people interested in using Processing as a teaching tool. Naturally, I expect the whole package to be done by the project's 100 hours timeline.

I feel like this project as the potential to ease the learning of mathematics on a high-school level, while developing a very useful programming skillset. For Processing, this mean more useful resources for teaching and wider audience. Learning Processing at such a young age also make them prone to use it as a sketchpad for all their future programming experiments.

# Development document

The proposed package could look like this :

0 - Teacher manual (Intro to processing, file-management, grading guide, etc)

1 - Processing basics (void setup, void update, size)
2 - Drawing shapes (rect, ellipse, lines, points)
3 - Color! (fill, stroke, explaining RGB, background)
**Summary 1 (keywords, ordering, shape position, commenting, what's next?) + Test quiz**
4 - Using variable (int, float, names, random)
5 - Moving things! (incrementation, mouse position, variable)
6 - Computer I/O (mouse, keyboard)
7 - Condition (if / then, boolean)
**Summary 2 (why use variables, how to use conditions, how to make a game?) + Test quiz**
**Game prototyping (how is your game gonna work? what do you need to do?) + Plan**
8 - Game jam #1 (starting the game, basic behaviors)
9 - Game jam #2  (continuing game, using text, some trigonometry basics)
10 - Game jam #3 (game testing with friends, improving the game visual, fixing bugs)
**Written assignment (What sort of math did you use? how did you use it? was it a fun experience?)**

As for the final game, it is important not to limit the student to what they're going to do, but to give them clear indications on what is expected (ie: something that moves using the keyboard or mouse, something that moves by itself at a variable speed, translation, rotation, conditional, etc.). The student will have a game checklist where he/she will need to describe each of these element with their general behavior.

Ideally, I'd spend 2 hour maximum on each exercise draft. These drafts would include the general concept and a step by step approach. I'd then be able to team up with a teacher, go over the first few assignments to see how they feel about it. After adjusting them, I'd assist the teacher during the the Processing workshop, mostly to see where they're could be misunderstandings, or where extra programming explanation could be necessary.  After some fine-tuning, I'd try the second revision with another teacher for some further analysis on what works or not.  I plan to spend around 7 hours developing each exercise, from the first draft, to fine tuning after each class, getting screenshots and code snippets, and formatting the content to a pre-defined layout.

The infographics and page layout will be made by Maude Fafard, a talented industrial and graphic designer that I've worked with in different Fab Lab. She estimated 10 hours to create a standard page template, along with graphical elements to make the whole package cohesive and enjoyable.

These documents should be clear and downloadable in PDF form. The layout will be color coded and should clearly indicated code snippets, newly added coded from a previous step and so on. I'd make sure that everything they are kept safe, up-to-date and accessible using GitHub.

I would work on the project over the span of 3 months, starting in March 2017. Right now I've planned to use around 70 hours on the course material, 10-15 on english translation (being a native french speaker) and 10 hours for Maude Fafard to create the visual document, giving me 5-10 extra hours for further editing, packaging, reviewing etc.

As for final presentation, I would love to present the project at PERTE DE SIGNAL, a new media artist-run center in Montreal that would be an adequate fit. The presentation would consist in a general overview of the project and exercises, a couple of stations where people could try the simple games, hopefully with the young programmer present. It would also be possible to present the finalized package to the school board that I work with.

I didn't get her answer in time, but I also thought of Sofian Audry as a mentor. She was my programming teacher at Concordia, and had a great experience with her.

# Previous project

*Presqu'un Sega Dreamcast* is a game made in 2015 for a creative computing class taught at Concordia University, Montreal. It was made with processing for the class, then later was rebuild using the Blender Game Engine.

Project page -> http://robyprovost.com/PSD
Github -> https://github.com/lerabot/Presqu-un-Sega-Dreamcast

*Pour te retrouver* is a durational, generative installation piece made for Art Matter : Pied-du-courant. Also made with Processing.

Project page -> http://robyprovost.com/pourteretrouver

Finally, I also made *un genre de totem*, a web based travelling notebook, partially made with p5.js

Project page -> http://robyprovost.com/totem