# Project Description

```
ᜐᜒᜆ᜔.js
 1  ᜐᜒᜆᜒᜇᜓᜈ᜔ ᜊᜇᜓᜐᜓ() {
 2    ᜆ᜔ᜇᜒᜇ᜔ᜆᜓᜈᜓᜊ(400, 400);
 3  }
 4
 5  ᜐᜒᜆᜒᜇᜓᜈ᜔ ᜇᜇᜏᜓ() {
 6    ᜊᜆᜆ᜔ᜄᜇᜓᜈ᜔ᜇ(220);
 7
 8    ᜊ =("ᜆᜄᜏ);
 9
10    ᜊ= (ᜏᜆ);
11
12    ᜆᜊᜇᜆ.ᜆᜈ ={"ᜊᜇᜏᜇ᜔ᜇᜊᜆᜏ");
13
14    ᜐᜓᜇ(x=0;x<=9;x++){
15
16      ᜐᜊᜏᜓ();
17    }
18
19  }
```

My proposal is to create a p5.js library that allows users to program using the *baybayin* (ᜊᜌ᜔ᜊᜌᜒᜈ᜔) writing system and vocabulary. *Baybayin* is a pre-Hispanic Philippine script, or writing system. The word *baybayin* translates as "to spell" in English. As a written language system, the widespread use of baybayin ended with Spanish colonialism but the FLOSS nature of both Processing and p5.js present a chance to revive the relevance of the writing system in a different context. This project attempts to reframe the act and practice of writing code as not just participation in technological culture, but also the retention and reconfiguration of pre-colonial history and knowledge systems.

A few attempts at digitizing baybayin writing already exist online as virtual keyboards (https://www.lexilogos.com/keyboard/baybayin.htm), translators(http://baybayin.sourceforge.net/), and plugins that let users tweet and post to social to media. Baybayin characters have also been

encoded into Unicode.  I hope to build on and consolidate this earlier work to create a functioning and implementable p5.js library.

As an artist I build fragile, post-colonial technology using web programming and physical computing because I'm interested in manifestations of networked culture in the diaspora.  I have a background in the intersections of art and technology including working with Processing and p5.js. Much of this work is distilled through my own experience as Filipinx/American and aspires for intersectionality. I hope that this perspective will allow me to reach out and work with those who have already done extensive work in digitizing non-Western languages.

This project most directly addresses the Internationalization Priority Area.  As many have already pointed out, programming is inherently biased towards English speakers.  This project in many ways expands upon other work that creates coding accessibility focused on non-English languages such as Yorlang or p5.js for 50+.  These projects rightly directly addressed accessibility for users of a "living" language.  Baybayin is considered a "dead" language from colonized people, and it's "rebirth" as a p5.js library can demonstrate the potential for programming as a model for decolonization.

The planned end output of this project will be a p5.js library that allows users to write code using *baybayin* characters and vocabulary.   By incorporating the library in a p5 sketch, specific *baybayin* functions can be accessed.  The library can be integrated with a virtual keyboard to access the characters, but I thought it would also be interesting to fabricate a physical *baybayin* specific keyboard.  Documentation about using the library will also be produced as well as source code example sketches.

Will using a pre-colonial writing system extend to algorithmic processing in a different way? I'm interested to see if new programming constructs can arise from the use of non-English language programming.  Can a new kind of variable or logical operator emerge from use of baybayin specific syntax?  Or, what does a 'for' loop look like in *baybayin* and will it stay the same?

Programming is one of the fastest growing kinds of language acquisition in the developing world. Often, programming is seen as instrumental to participation in modern capitalist networked economies.  To participate, post-colonial people must embrace a more dominant culture.  Personal worth becomes conflated with the ability to communicate with technology (often targeted to end-users elsewhere).  Users of "dying" or "obsolete" language often see future generations losing language fluency in favor of more "productive" languages.  Projects like this one can make lost languages relevant to a networked culture.  Online communities (like the ones that already exist for Processing and p5.js) can form around this project and make the acquisition of "historically dead" language skills creative, fun, and experimental.