Spryntax - a programming racer

Team 5

John Polasek, Shrey Bosamia, Aidan Lusk, Kali Pulanco, Justice Peyton, Prathmesh Nitin Gite, Seojin Lee
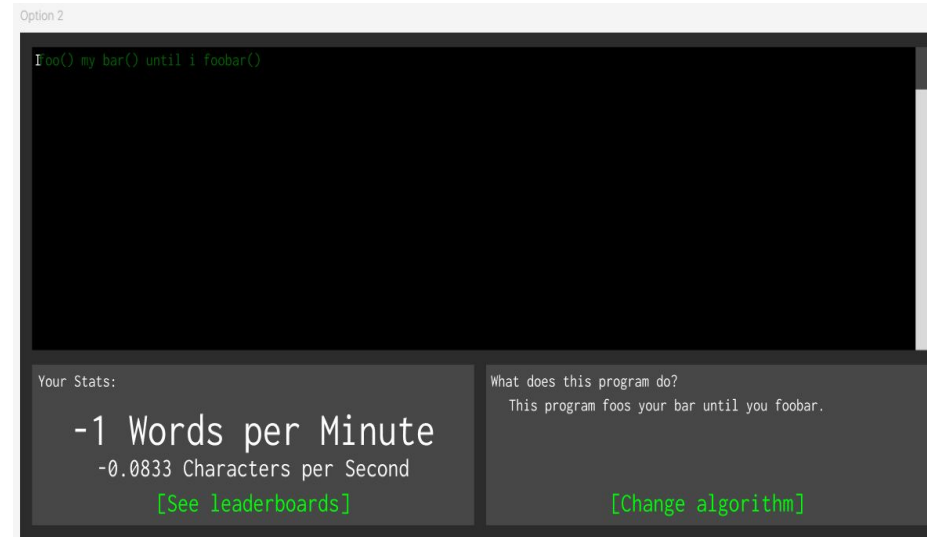
What is the project about?

- Traditional typing practice websites are useful for general typing but what about coding?

- What if you were shown algorithms in multiple programming languages that you can select and had to type it out? Not only will you practice typing code, but you'll learn more about the fundamentals of the particular programming language you selected.

- Existing practices have algorithms but our project will showcase what the program does.

Cool! What features are there going to be?

- Monkeytype-style of program that allows users to type characters needed to create a specific file, the file contains some useful-to-know algorithm
- Citations and extra resources provided for each assignment
- Explanation of the code that the user types (with comments)
- Leaderboard for fastest times
- Statistics for users
  - Time that a user took on a specific assignment
  - Words per minute (# of chars divided by 5)

Option 2

```
Foo() my bar() until i foobar()
```

Your Stats:

-1 Words per Minute
-0.0833 Characters per Second
[See leaderboards]

What does this program do?
   This program foos your bar until you foobar.

[Change algorithm]

(Mock-up of typing screen)

Goals and Timeline

The primary goal of Spryntax is to introduce beginners to the way modern programming languages are typed and to important concepts, as well inspiring competition in typing speed amongst programmers.

The stretch goals are having "Profile" for a particular user that other users can view, Support for multiple programming languages, and Support for users to compile/interpret code on the site

Timeline:
- Week 1: Discuss about the project idea
- Week 2: Functional requirements
- Week 3: Create database
- Week 4: Full website wireframes
- Week 5: Implementing html, css for the website design and node.js to connect to the database
- Week 6: Local functionality
- Week 7: Server functionality
- Week 8: Fine tune explanations
- Week 9: Polish javascript functions for non-static functionality
- Week 10: Add more languages and extend explanations.