



MATES Computer Science

## Senior Capstone Project Bi-Weekly Progress Report

Project Title	Bit by Bit
Team Members	Anna Pitera
Dates Covered by Report	02/12/24 to 02/22/24
Link to Github	<a href="https://github.com/anna-pitera/Bit-by-Bit">https://github.com/anna-pitera/Bit-by-Bit</a>

### 1. Summary of Project

#### Overview:

Bit by Bit is a gamified to-do list and habit tracker website that encourages users to complete their daily tasks and maintain good habits by rewarding them with in-game currency, which can be used to purchase customization options and in-game benefits. The website is created with HTML, JavaScript, CSS, and Phaser, a JavaScript library for creating web games. The Django web framework is used along with Pythonanywhere, which allows programmers to host web apps online for free with restrictions that can be removed by purchasing a subscription. To use Bit by Bit, players must create an account with a username and password which stores their progress in a MySQL database, allowing them to close and reopen their progress at any point from any computer. Bit by Bit “resets” every day at midnight so users can check off their habits each day, and users are rewarded for “checking in” each day.

#### Website Design & Layout:

The design of the website is in a simple, easily reproducible pixel art style, and the main page of the website consists of five key features: the user’s character profile box, a to-do list layout, a store where users can spend their in-game currency on customization options and character accessories, a habit tracker display, and a few different number displays. The user’s character profile box displays the user’s customizable avatar/character and is made using Phaser sprite functionality. The to-do list is a list where users can enter and complete their daily tasks, and task completion rewards the user with currency. The store is a tiled display showing all obtainable items that can be purchased with currency, including character accessories and UI color themes. The habit tracker displays the user’s entered habits that they choose to maintain, along with a check box where users can check off whether or not they maintained their daily habits each day. The user’s habit “streak,”

---

(the number of days they have maintained their desired habits in a row), currency amount, and total number of tasks on their to-do list are all shown in various number displays. Simple quality-of-life features are also included, such as the ability to undo the accidental completion of a task.

## 2. Summary of Progress this Period

The beginning of this progress period was spent practicing HTML, JavaScript, CSS, and Phaser on Codecademy. After getting far enough in the Codecademy courses to feel more comfortable with these topics, a few days were spent experimenting, testing, and messing around in VS Code to get more hands-on experience. While this sounds vague and unimportant, I consider it necessary for getting more comfortable with coding completely independently in new languages. It included things like finding the right VS Code extensions for live previewing websites, going back and forth between using certain JavaScript packages, and finding programming resources (videos, articles, Stack Overflow posts) for later use. The rest of the progress period was spent researching and learning about concepts essential to the development of Bit by Bit, such as Django, Pythonanywhere, and MySQL. Overall, a lot was learned in the past couple of weeks, and I feel that I understand the flow of project development much better now.

## 3. Detailed Progress this Period, separated by Team Member

The first thing I did was create the project repository. Then, after practicing HTML, JavaScript, CSS, and Phaser on Codecademy for a couple of days and installing the necessary software, I began experimenting with making the Bit by Bit website using the Live Preview VS Code extension.

I made the mistake of trying to rush into frontend development for the first few days of the progress period, so I spent time figuring out how to import custom fonts, how to resize Phaser environments, how to add pixel borders to text boxes, etc. Admittedly, I probably should've spent that time focusing more on the backend, but I think it taught me a valuable lesson of time management and setting more realistic expectations for progress periods.

Here's a very basic example of what the custom fonts and text box borders look like:

What is your name?



And when you type in the text box, the font persists:

---



Here's the style.css file for that (cut off because there's like 90 lines of just the pixel border, which thankfully I didn't have to write myself because I found a pixel art border generator online):

```
@font-face {
  font-family: "m5x7";
  src: url("../fonts/m5x7.ttf");
}

body {
  font-family: "m5x7";
  font-size: 40px;
}

input[type="text"] {
  font-family: "m5x7";
  font-size: 40pt;
  text-align: center;
}

.pixel-corners,
.pixel-corners--wrapper {
  clip-path: polygon(0px calc(100% - 12px),
    6px calc(100% - 12px),
    6px calc(100% - 6px),
    12px calc(100% - 6px),
    12px 100%,
    calc(100% - 12px) 100%,
    calc(100% - 12px) calc(100% - 6px),
    calc(100% - 6px) calc(100% - 6px),
    calc(100% - 6px) calc(100% - 12px),
    100% calc(100% - 12px),
    100% 12px,
    calc(100% - 6px) 12px,
    calc(100% - 6px) 6px,
    calc(100% - 12px) 6px,
    calc(100% - 12px) 0px,
    12px 0px,
    12px 6px,
    6px 6px);
}
```

And here's the HTML file for it (the JavaScript file is commented out for now because it's not needed right now but all it does is create the Phaser character box environment):

---

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Bit by Bit</title>
  <link rel="stylesheet" href="/bitbybit/bitbybit/static/style.css">
</head>

<body>
  <script type="text/javascript" src="/phaser.min.js"></script>
  <!-- <script type="text/javascript" src="/characterbox.js"></script> -->



  <h4>What is your name?</h4>
  <div>
    <input type="text" class="pixel-corners pixel-corners--wrapper" size="20">
  </div>
</body>

</html>
```

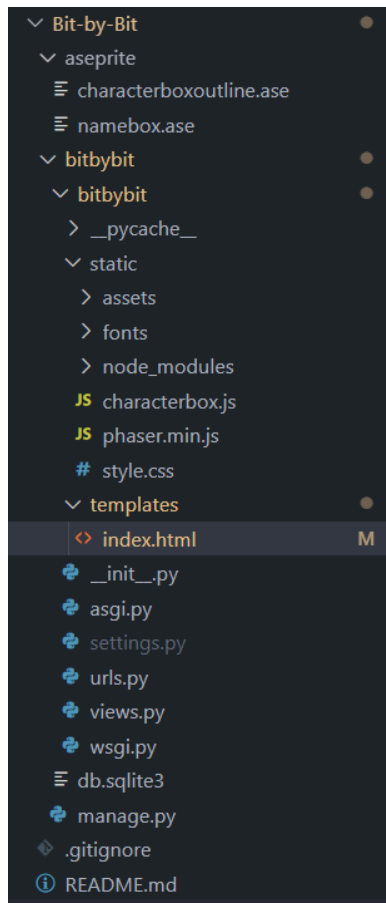
I also created my PythonAnywhere account and cloned into the repository, so this is what my dashboard looks like now:

The screenshot shows the PythonAnywhere dashboard for user 'annapitera'. At the top, there's a navigation bar with links: Dashboard, Consoles, Files, Web, Tasks, and Databases. The main header includes the PythonAnywhere logo and the text 'Welcome, annapitera'. Below this, a status bar shows 'CPU Usage: 2% used - 2.12s of 100s. Resets in 19 hours, 12 minutes' and 'File storage: 1% full - 4.3 MB of your 512.0 MB quota', both with 'More Info' links. An 'Upgrade Account' button is on the right. The dashboard is divided into four panels: 'Recent Consoles' showing a 'Bash console 32480178' with a 'View all' button; 'Recent Files' listing files like '/var/www/annapitera\_pythonanywhere...', '/home/annapitera/Bit-by-Bit/index.html', and '/home/annapitera/.bashrc'; 'Recent Notebooks' with a message that the account does not support Jupyter Notebooks and an 'Upgrade your account' link; and 'All Web apps' showing 'annapitera.pythonanywhere.com' with an 'Open Web tab' button. At the bottom, there's a 'New console:' section with buttons for '\$ Bash', '>>> Python', and 'More...'. At the very bottom, there are buttons for '+ Open another file' and 'Browse files'.

Finally, after spending time learning more about Django, I adjusted the file structure to cooperate with Django's project structure. So this is what the file structure looked like before:

 <b>anna-pitera</b>	Created settings.py for Pythonanywhere deployment	77832d4 · 2 days ago	 8 Commits
aseprite	Found a package called "NES.css" that builds in pixel art text...	last week	
assets/ui	Found a package called "NES.css" that builds in pixel art text...	last week	
fonts	Created ui folder within assets folder as well as style.css. Set ...	last week	
.gitignore	Created settings.py for Pythonanywhere deployment	2 days ago	
README.md	Initial commit	3 weeks ago	
characterbox.js	Centered scene for character box outline. Also added Asespr...	last week	
index.html	Decided against using NES.css because doing it manually se...	2 days ago	
phaser.min.js	Added phaser.min.js and tested using Phaser with world.js file	2 weeks ago	
style.css	Decided against using NES.css because doing it manually se...	2 days ago	

And this is what it looks like now with all the Django settings files and everything:



#### 4. Difficulties Encountered this Progress Period

- Node.js & Phaser
  - Because I had forgotten to install Node.js when we were doing our software installs, it took me about a day or two to figure out that I needed to install it and to figure out what I needed to do with it to install Phaser. This was resolved by installing Node.js and reading a few installation guides online.
- Django & Pythonanywhere
  - At the beginning of this progress period, I didn't understand Django, or even Pythonanywhere, nearly as well as I needed to. In fact, I hadn't even set up the file structure of my repository to cooperate with the file structure of Django projects, so all I had was a few simple folders and a couple of unorganized HTML, CSS, and JavaScript files. To resolve my confusion, I went through some of the Codecademy course on building interactive websites with Django and Pythonanywhere, which really helped me understand everything a bit better. After that, I reformatted the project's file structure using Django as mentioned earlier. I obviously still have a lot to learn with Django, though.
- Planning and setting expectations for progress
  - I think I had slightly too high expectations for the amount of progress I would be able to complete this progress period, which actually made it harder to make significant progress. I better understand now that I need to focus on one thing at a time and keep reasonable expectations.
  - I wasn't really using the Trello board as much as I wanted to, so I'm going to try to utilize it to better plan out my tasks and goals this upcoming progress period.
- MySQL
  - Just like with Django, I don't fully understand how to integrate a MySQL database with a website. In the next progress period, I'm going to continue a few Codecademy courses and work through a couple of guides/tutorials that I have saved on the Trello board, so I'm hoping I'll understand it better.

#### 5. Updated Trello Board and Discussion

Link: <https://trello.com/b/Lzd1bk2c/bit-by-bit>

##### Changes:

- Added a few extra cards in the "Resources" list for me to store resources for other topics (Node.js and Django specifically)
- Added a couple of reminders for when I have to write the final README.md file

#### 6. Tasks to Be Worked on in Next Progress Period

Connecting MySQL database to Django project

- Continue Codecademy course
  - Watch/read tutorials and guides (already have some saved in Trello board and bookmarked in browser)
  - Practice by making a test form where user can enter tasks and they get saved in the database
  - Update Pythonanywhere website with new site
  - Possibly work on username and password login system with MySQL if time permits? (just a very basic login system)
-

- Possibly work on connecting Phaser input to MySQL database? (again, just a very basic example for proof of concept)

## 7. Additional Information

N/A