Taylor Smith

Written Content

Home Page

Quick overview of what I do:

I am Taylor Smith, a full-time college student at Texas A&M University majoring in Computer Science with a minor in Mathematics. This website is going to be used as a portfolio for projects I have worked on and accomplishments I have made.

About

Who I am, what I do, what I am passionate about. Image, information about me: school, experience, etc.

About Me:

Howdy, my name is Taylor Smith, I am currently a Junior CS Major at Texas A&M University who has a passion for coding; I'm seeking opportunities to broaden my knowledge through experience. My competitive nature causes me to strive for excellence; always seeking chances to challenge myself and my capabilities.

Currently I am working on various projects for the A&M Engineering Academies organization that I am an officer and ambassador in alongside personal coding projects that I find interesting. I am not very familiar with front end, as this website is my first time working with Html, CSS, etc.; however, the programing languages I primarily use are C++, Python, and Java. I was tasked with teaching myself JavaScript for my final project in my UT CS course in high school, but I'm more comfortable with the languages I listed.

How it started:

The first time I "programmed" was in elementary school, we had a computer lab where we messed around with website Scratch. I liked playing games growing up and realizing I had the chance to try and make my own was exciting for me!

Further down the line, in my junior year of high school, I finally took my first actual programming course; we learned Java and I caught on quite quickly. My teacher saw this and, by the end of the year, asked if I would be a part of the University Interscholastic League.

The following year I participated on our school's Computer Science team where we ranked first in our region and second in our state; individually I placed 13th in our state. Unfortunately, that was my senior year of high school so that was the only year I participated in academic UIL; however, it was an extremely fun and competitive challenge that made my passion for programming flourish.

College Life:

During my first year of college I took an Engineering Computing course which taught me python, this was one of the most interesting courses I have taken because the main focus was working together with a group of people in order to solve complex coding problems. The language itself came easy to me but learning how to work with others, especially on a project that we had to work on over the course of various weeks, was extremely valuable.

After my first year, I applied to be a computer science major at Texas A&M University, College of Engineering. Since I had a 4.0 GPA and my personal essays were strong enough, I was admitted to the department of Computer Science for a CS major! This was an extremely exciting part of my life because I know that only a very small percentage of people who apply to be a CS Major at A&M actually get in.

During my second year, I spent two semesters taking the most difficult computer science courses I have ever taken: Discrete Mathematics, Data Structures and Algorithms, and Computer Architecture. Before this, I was usually able to get by on just the information from lectures alone; I rarely had to sit down and study material. However, these courses were much more difficult than I anticipated; there was actually a time where I was at a failing grade in each of these classes midway through each semester. Luckily, I was able to pull back and manage high A's in my Data Structs and Computer Architecture courses, my Discrete Mathematics, however, was only a high B. This semester taught me a lot about what challenges I am going to face moving forward in CS, but more importantly, it taught me how to properly study and manage my time in general.

Extra-Curricular:
In high school, I was part of the National Honors Society for two years, volunteering over 60 hours for multiple organizations. During my time in UIL Computer Science, my mathematics teacher also asked me to be on the UIL calculator team. The team as a whole did not make it very far; however, individually I came 1st in our district and 5th place in our region. Unfortunately, only the top four qualified for state so I was not able to compete at the state level, but it was still a very fun experience for my first year doing it.

For my first year of college, I was admitted to the Texas A&M Chevron Engineering Academies. During this time, former students of the Academies became ambassadors of the program and would often talk to us current students about what a positive thing the academies were for them. We were able to ask them many questions about their experience in the program, how the main campus is, and what they're currently doing and hope to do themselves. They were great resources, as an incoming transition student, so I reached out to them to see if I could become an ambassador myself. I submitted my application and days later I actually received a call asking if I would be willing to work on more projects besides just being an ambassador. Due to my knowledge of computer science, they asked if I was willing to become, not only an ambassador, but an officer for the association. I would work on the communications technologies within the organization as well as working alongside the treasurer to create a fully automated verification system the org could use for new members. This seemed like an extremely daunting task and position to immediately jump to, but I decided to just go for it because I probably wouldn't get another opportunity as good as this one. If that project sounded interesting to you by the way, I have it listed in my portfolio if you would like to check it out.

Portfolio

Showcase of my best work:

- Maze Game – Spring 2022
  - This project was made in my senior year of high school as my final project for my University of Texas Computer Fluency course; I had never programmed or created anything visual using JavaScript before. The project required the use of sound, graphics, and some level of user interaction.
  - I was tasked with creating this project using the p5.js website: p5.js
- Zanzibar – Fall 2022
  - This project was made in my freshman year of college; it was my first time creating a visual interactive program using Python so, while the project does not look as visually appealing as it could be, I am still very proud of it.
  - It was also my first time working on a program in a group setting, where each member had their own specific tasks to complete which came into one whole project.
  - The project required us to:
    - Display the rules of the game and a set of instructions on how to use your program
    - Display a set of menu options for the various things your players can do
    - Make use of conditionals and loops effectively
    - Organize code by taking advantage of functions (including docstrings)
    - Use file I/O in a meaningful way
    - Use turtle graphics to animate something
    - Incorporate something to make the project stand out from others
- Datathon – Fall 2023
  - I had the privilege of participating in the 2023 TAMU Datathon, my first-ever datathon. This event brought together a diverse group of talented individuals, all with a shared passion for data science.
  - Our project is training an ai model using a dataset to determine the accuracy of a patient surviving given the specifics of their condition.
  - Our original plan:
    - The dataset given to us is in the form of a csv file, the first row having the titles of the columns, and each subsequent row will be containing information for a patient given by the hospital with one row representing one patient.
    - Provided with a dataset with inherent flaws, we had to adjust, scrape through, and clean up the data.
    - Afterwards, we used recursive feature elimination to find what features contributed most to survival.
    - This allowed us to train the Keras Sequential Model to predict data.
    - The predictions from this model were submitted to get a score on how accurate it was.

    - After countless hours of hard work, our team – consisting of Arul Dhar, Bao Nguyen, and me – had our project selected as one of the winners of the Datathon.
        - While our code did not provide the most accurate results, our writeup is what made us stand out the most (not to toot my own horn too much but that's what I did 🤠).
- y86-compiler for assembly code – Spring 2024
    - In the Spring of my sophomore year, my group – Logan Atkinson, Aidan Briggs, and I – were tasked with building a functional sequential y86-64 processor for assembly code: this was one of the most difficult projects I had worked on due to the expansiveness of it.
    - Over the course of a month, our group designed and implemented the
        - Fetch
        - Decode
        - Execute
        - Memory
        - Write Back

      stages of the processor using the *logism-evolution* software.
    - The project required us to:
        - Implement a fully functional processor that worked with a few assembly programs we wrote.
        - Write a detailed report explaining how each stage is structured and how they work.
        - Include timing diagrams of our processor.
        - Include transformation tables of the y86 instruction set we had to implement.
        - Orally answer questions about our processor and the design in general to ensure our understanding.
- Wordle – Spring 2024
    - This project was made in my sophomore year of college; it was my first time creating a visual interactive program utilizing JavaFX. The use of SceneBuilder made the entire process much easier and more fun to work on, I only had to worry about what the design would look like and easily implemented it using SceneBuilder.
    - I had been acquainted with Java for a few years by now so creating the logic for the game was quite straightforward, the bulk of my time was spent adding various features and designing a, somewhat, visually appealing experience.
    - The project required me to:
        - Create a working application of Wordle.
        - Have a random word selected through Java File IO.
        - Use a complex (more complex than a 2D array) data structure for storing letter data.
            - Also store this data for future game use using Java File IO.
        - Ensure the application was visually appealing and user friendly.

        - Record and calculate player statistics that could be seen anytime:
            - Wins/losses
            - Number of games played
            - Current streak/max streak
- Discord Verification Bot – Summer 2024
    - This is a project that the Texas Engineering Academies Student Organization tasked me with doing after I became an officer of the org.
    - While I had a decent amount of python knowledge, I had never created a discord bot before (or worked with an API in general for that matter), so I knew this would be an excellent opportunity to learn and apply my coding knowledge to an actual issue the organization was running into.
        - The organization has a one-time payment in order for students to gain access to all the important channels in the discord.
        - Members who pay the fee gain the role "Verified", this is what allows them to gain access to all the resources the org provides.
        - The bot aims to automatically change roles of members who pay the fee so that officers do not have to continue manually assigning roles.
    - The bot solves this issue by utilizing the Google spreadsheets API:
        - Members fill out a form in order to conclude their admission in the organization.
        - Information from that form is saved on a spreadsheet.
        - The bot utilizes Google's spreadsheet API to read through and confirm a user has gotten their information onto the spreadsheet, if the information matches then the user is verified.
            - This method works great because the API allows for real-time updates on the spreadsheet so anyone who fills out the form can immediately confirm their verification in the discord.
    - While this bot is primarily used for authentication of new organization members, I plan for it to do other tasks that the org's discord could benefit from. These changes will be documented as I do them.
    - I wanted to make the code as understandable as I could for future organization officers to be able to update the bot if any changes are needed.

Contact

Email, times of availability, LinkedIn, etc.

- smith.taylor@tamu.edu
- 7-10 pm weekdays
- 3-7 pm weekends
- in/smithtas

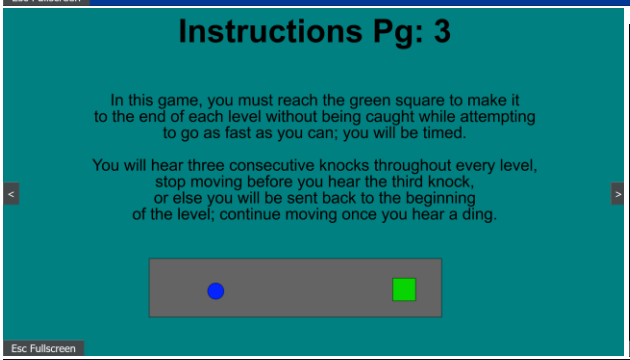
Images and Visual Content

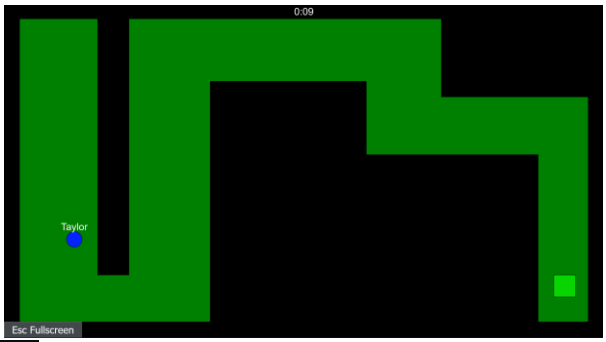
Pictures of Me



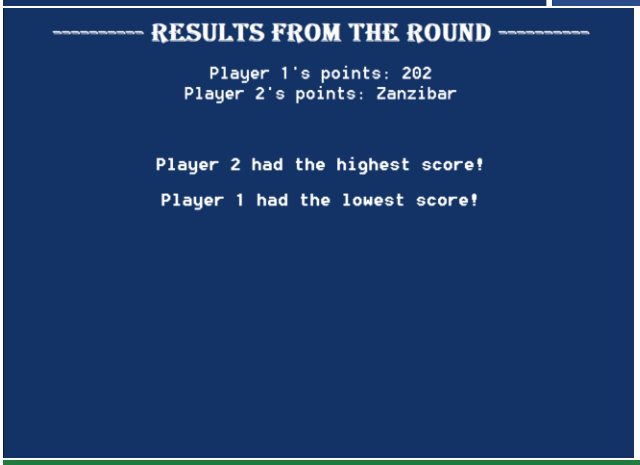


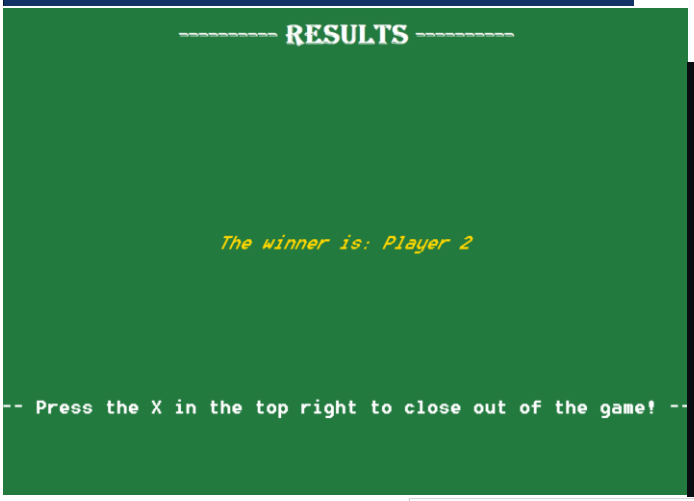
Pictures of my code/projects

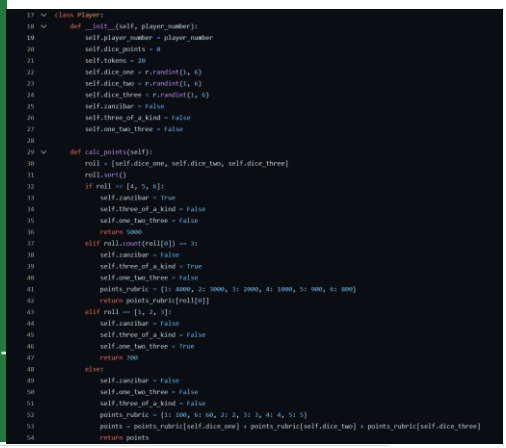




```
// this is the function for the fullscreen button
function pressedTheButton() {
  let fs = fullscreen();
  fullscreen(!fs);
  isFullScreen = !isFullScreen;
  if (isFullScreen) {
    resizeCanvas(screenWidth, screenHeight - 10);
    button.html("Esc Fullscreen");
    resizeFlag = false;
    buttonWidth = 250;
    button.size(250, 50);
  } else {
    resizeCanvas(720, 480);
    button.html("Fullscreen");
    buttonWidth = 180;
    button.size(180, 50);
  }
}
```

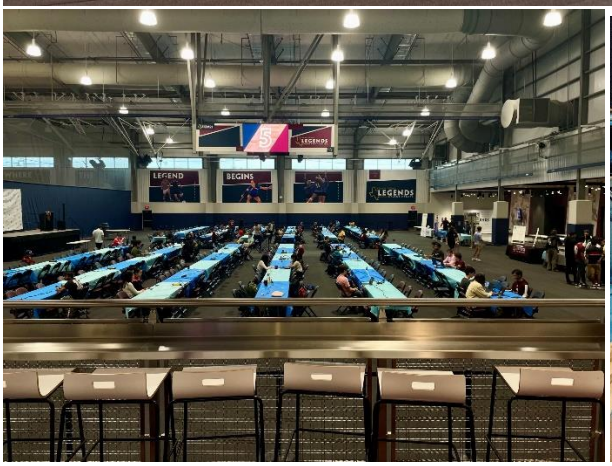


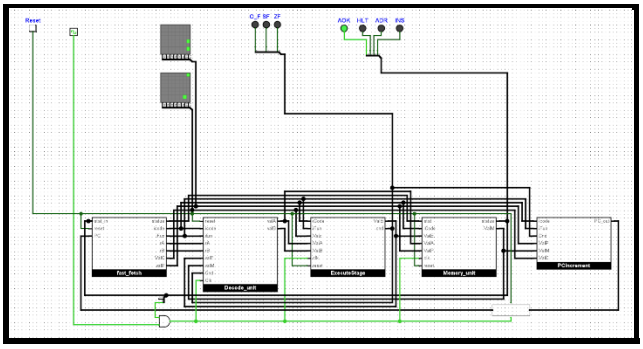




```
class Player:
    def __init__(self, player_number):
        self.player_number = player_number
        self.dice_points = 0
        self.tokens = 20
        self.dice_one = r.randint(1, 6)
        self.dice_two = r.randint(1, 6)
        self.dice_three = r.randint(1, 6)
        self.zanzibar = False
        self.three_of_a_kind = False
        self.one_two_three = False

    def calc_points(self):
        roll = [self.dice_one, self.dice_two, self.dice_three]
        roll.sort()
        if roll == [4, 5, 6]:
            self.zanzibar = True
            self.three_of_a_kind = False
            self.one_two_three = False
            return 5000
        elif roll.count(roll[0]) == 3:
            self.zanzibar = False
            self.three_of_a_kind = True
            self.one_two_three = False
            points_rubric = {1: 4000, 2: 3000, 3: 2000, 4: 1000, 5: 900, 6: 800}
            return points_rubric[roll[0]]
        elif roll == [1, 2, 3]:
            self.zanzibar = False
            self.three_of_a_kind = False
            self.one_two_three = True
            return 700
        else:
            self.zanzibar = False
            self.one_two_three = False
            self.three_of_a_kind = False
            points_rubric = {1: 100, 6: 60, 2: 2, 3: 3, 4: 4, 5: 5}
            points = points_rubric[self.dice_one] + points_rubric[self.dice_two] + points_rubric[self.dice_three]
            return points
```

```
def play_game():
    '''this is the main game loop that is played'''
    screen.bgcolor('#2B4B6b')
    t.color('black')
    p1 = players[0]
    roll_count = 0

    for i in range(3):
        t.clear()
        draw_round(p1)
        roll_count += 1
        p1.roll_dice()
        draw_die()
        p1.draw_outline()
        if p1.zanzibar:
            t.setx(0)
            t.sety(100)

            t.write('You got: (4, 5, 6) Zanzibar!', align='center', font=('Helvaeem', 30, 'normal'))

        elif p1.three_of_a_kind:
            t.setx(0)
            t.sety(100)

            t.write(f'You got: {(p1.dice_one, p1.dice_two, p1.dice_three)}', align='center',
                    font=('Helvaeem', 30, 'normal'))

        elif p1.one_two_three:
            t.setx(0)
            t.sety(100)

            t.write('You got: (1, 2, 3)', align='center', font=('Helvaeem', 30, 'normal'))

        else:
            t.setx(0)
            t.sety(100)
```



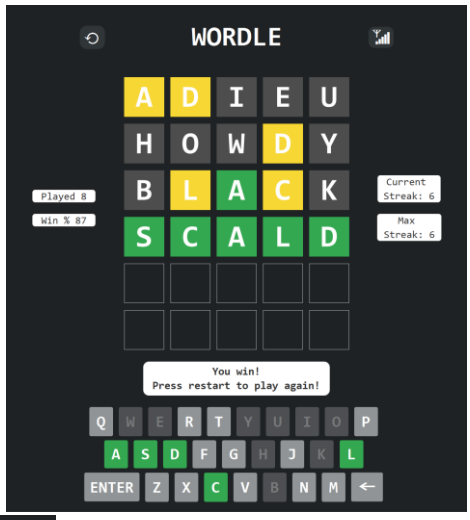


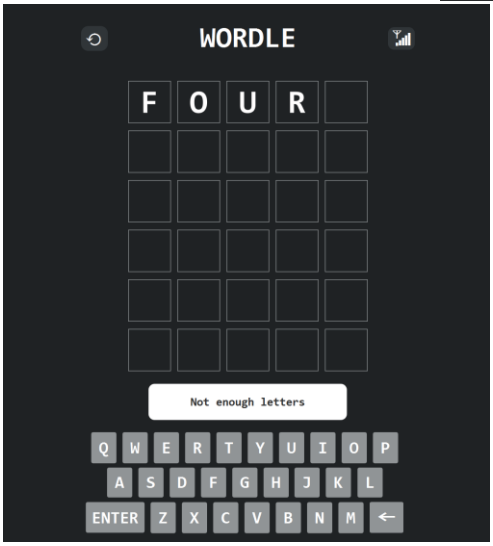
Stage	halt	nop	rmmovq **rA, rB**
Fetch	icode : ifun ← M_1[PC] valP ← PC + 1	icode : ifun ← M_1[PC] valP ← PC + 1	icode:ifun ← M_1[PC] rA:rB ← M_1[PC+1] ValP ← PC+2
Decode	_____	_____	ValA ← R[rA]
Execute	stat ← HLT	_____	valE ← 0 + valA
Memory	_____	_____	_____
Write Back	_____	_____	R[rB] ← ValE
PC Update	PC ← 0	PC ← valP	PC ← ValP

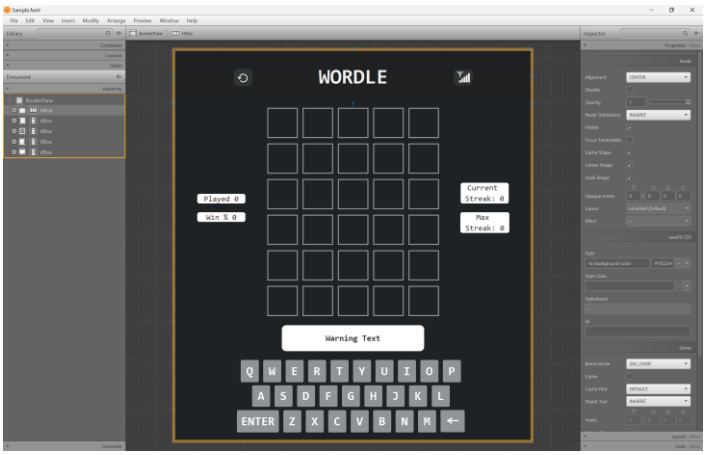
Stage	irmovq **V, rB**	rmmovq **rA, D(rB)**	mrmovq **D(rB), rA**
Fetch	icode : ifun ← M_1[PC] rA : rB ← M_1[PC + 1] valC ← M_8[PC + 2] valP ← PC + 10	icode : ifun ← M_1[PC] rA : rB ← M_1[PC + 1] valC ← M_8[PC + 2] valP ← PC + 10	icode : ifun ← M_1[PC] rA : rB ← M_1[PC + 1] valC ← M_8[PC + 2] valP ← PC + 10
Decode	_____	valA ← R[rA] valB ← R[rB]	valA ← R[rA] valB ← R[rB]
Execute	valE ← 0 + valC	valE ← valC + valB	valE ← valC + valB
Memory	_____	M_8[valE] ← valA	valM ← M_8[valE]
Write Back	R[rB] ← valE	_____	R[rA] ← valM
PC Update	PC ← valP	PC ← valP	PC ← valP

Stage	OPq **rA, rB**	jXX **Dest**	cmovXX **rA, rB**
Fetch	icode : ifun ← M_1[PC] rA : rB ← M_1[PC + 1] valP ← PC + 2	icode : ifun ← M_1[PC] rA : rB ← M_8[PC + 1] valP ← PC + 9	icode:ifun ← M_1[PC] rA:rB ← M_1[PC+1] ValP ← PC+2
Decode	valA ← R[rA] valB ← R[rB]	———	ValA ← R[rA] ValB ← R[rB]
Execute	valE ← valA OP valB Set cc	Cnd ← Cond(cc, ifun)	ValE ← CC[ifun] ? 0 : ValB
Memory	———	———	———
Write Back	R[rb] ← valE	———	R[rB] ← ValE
PC Update	PC ← valP	PC ← Cnd? valC : valP	PC ← ValP

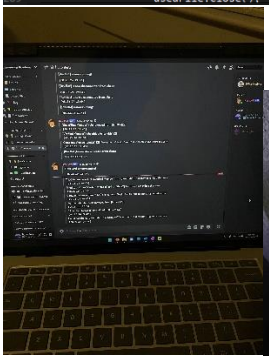
Stage	popq **rA**	push **rA**	ret	call **Dest**
Fetch	icode:ifun ← M_1[PC] rA:rB ← M_1[PC+1] ValP ← PC+2	icode:ifun ← M_1[PC] rA:rB ← M_1[PC+1] ValP ← PC+2	icode:ifun ← M_1[PC]	icode:ifun ← M_1[PC] ValC ← M_8[PC+1] ValP ← PC+9
Decode	ValA ← R[%rsp] ValB ← R[%rsp]	ValA ← R[rA] ValB ← R[%rsp]	ValA ← R[%rsp] ValB ← R[%rsp]	ValB ← R[%rsp]
Execute	ValE ← ValB+8	ValE ← ValB-8	ValE ← ValB+8	ValE ← ValB-8
Memory	ValM ← M_8[ValA]	M_8[ValE] ← ValA	ValM ← M_8[ValA]	M_8[ValE] ← ValP
Write Back	R[%rsp] ← ValE R[rA] ← ValM	R[%rsp] ← ValE	R[%rsp] ← ValE	R[%rsp] ← ValE
Pc Update	PC ← valP	PC ← valP	PC ← valM	PC ← ValC

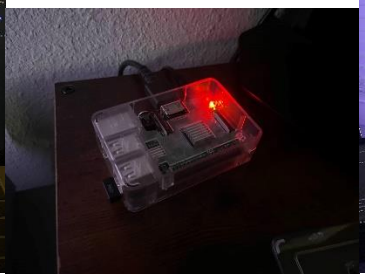


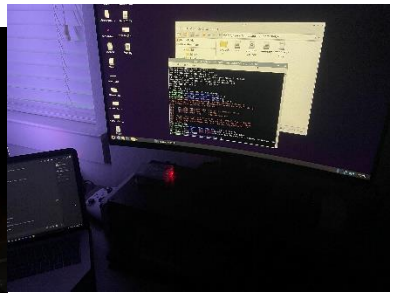




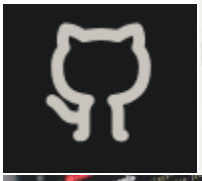
```java
    // randomly select a word that has not been used before
    private void wordInitialize() throws IOException {
        Scanner usedFile = new Scanner(new FileReader("UsedWords.txt"));
        Scanner statFile = new Scanner(new FileReader("statLog.txt"));
        // numUsed stores the number of times the user has played
        int numUsed;
        try {
            numUsed = statFile.nextInt();
        } catch (Exception e) {
            numUsed = 0;
        }

        Scanner infile = new Scanner(new FileReader("WordList.txt"));
        // get a random number
        Random random = new Random();
        int wordBuffer = random.nextInt(2310);

        if (numUsed < 2309) {
            while (true) {
                // go forward in the txt file the buffer amount
                // and select that word, essentially randomly accessing a word
                for (int i = 0; i < wordBuffer; ++i)
                    infile.next();

                boolean good = true;
                word = infile.next();
                // checks if the random word has been used already by checking it against the
                // UsedWords file
                while (usedFile.hasNext()) {
                    if (word.equals(usedFile.next().toLowerCase())) {
                        good = false;
                        break;
                    }
                }

                // if the word has been used, both files are closed and the process is repeated
                // until a new word has been selected
                if (!good) {
                    infile.close();
                    usedFile.close();
```







Filler pictures (code-related/a&m/some-icons/social-media)



TEXAS A&M UNIVERSITY
Engineering