AN OPTIMIZED WRITING EFFICIENCY TOOL WITH INTEGRATED GAMIFICATION AND WORK INTERVAL TIMER FOR ENHANCED ESSAY PRODUCTIVITY AND AI ADVICE

Summer Shen[1], Moddwyn Andaya[2]

[1]Saratoga High School, 20300 Herriman Ave, Saratoga, CA 95070
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This paper addresses the challenges students face in efficient essay writing, aiming to improve both productivity and writing quality. Students often struggle with overwhelming tasks, slow progress, and lack of motivation during the writing process. Our proposal combines the Pomodoro Technique, gamification, and an AI-powered writing quality checker to offer a holistic solution [8]. The Pomodoro Technique breaks down tasks, while gamification elements, including a playful chicken character, make writing enjoyable. An AI-based writing quality checker provides real-time feedback on grammar, sentence structure, and clarity [9]. Challenges, such as balancing interactivity and efficiency, were addressed through thoughtful design decisions.
Experimentation involved applying the application to various writing scenarios, showcasing its adaptability. Results demonstrated enhanced productivity, improved writing quality, and increased user satisfaction. This comprehensive approach addresses the shortcomings of existing methodologies and provides users with a valuable tool to navigate the challenges of essay writing efficiently and effectively.

KEYWORDS

Essay, Gamification, Unity, Productivity

1. INTRODUCTION

Writing is a crucial skill for students, but students often struggle with efficient essay writing. For many students, writing essays is a time consuming and laborious task. There are many reasons students struggle with essay writing. These reasons include having a hard time getting started and feeling overwhelmed by the task, feeling that the process of writing is slow and tedious, etc [5]. When students don't enjoy the writing process, they are not able to develop their writing skills at a productive pace. How to manage the overwhelming task of essay writing, and how to motivate students and bring enjoyment to the writing process are key problems to be solved.

In addition, students often feel that the paper never turns out the way they want even though substantial time and effort were spent [5]. This also demotivates them. The quality of the writing is caused by a lack of various writing skills, including grammar, sentence structure, spelling, communicating ideas clearly and concisely, constructing a reasoned, demonstrable argument, etc. [6] Providing necessary assistants during the writing process can help students improve writing

quality and skills. Completing an essay with high quality will definitely motivate the students in writing. A solution of writing assistance is needed.
Methodology A utilized existing tools like Grammarly and Hemingway Editor to address issues in spelling, grammar, and readability during the revision stages of writing. However, these tools didn't tackle challenges in formulating ideas or enhancing productivity during the writing process, limiting their scope.

Methodology B introduced Wordtune, catering to EFL writers by suggesting rewrites and improving sentence structure [10]. While beneficial for formulation, it lacked features to enhance overall writing productivity, leaving a gap in addressing efficiency during the writing process.

Methodology C explored 750.com, a platform promoting daily writing habits through gamification. It incentivized users but fell short in addressing writing quality and efficiency. The project aimed to fill these gaps by combining gamification elements with an AI-powered writing assistant, seeking to encourage consistent writing habits while also enhancing the quality and efficiency of the writing process. This approach aimed to offer a more comprehensive solution that addressed both formulation challenges and productivity shortcomings observed in the three methodologies.

To help students master efficient essay writing, our application uses the Pomodoro Technique and gamification to enhance productivity and make the essay writing process less tedious. The Pomodoro technique is a time management method, which uses a timer to break up work into timed intervals [7]. These intervals are usually 25 minutes and are separated by short breaks. This technique helps people focus on their work. When people have a big project to do, they may feel too overwhelmed to start the project. The Pomodoro Technique breaks down large tasks into smaller more manageable tasks [15]. Using a Pomodoro timer can make writing essays less intimidating for students. Setting time goals for and corresponding tasks for writing parts of an essay can also help students get their rough draft down quicker.

Gamification uses the elements we often see in games to motivate and engage users. These elements include things like point scoring, competition, and rewards. For our application, we implemented a chicken character that interacts with the user, showing different emotions based on the user’s typing speed. We also made a game-like interface for storing all the user’s essays. Elements like this can make writing an essay enjoyable. Both of these techniques can help users write essays more efficiently.

Another method to improve efficiency we have considered is providing the user with an outline for their essay. This could help the user break the essay down into small manageable tasks. However, different essays will have different formats that need different outlines. We have also considered letting the chicken character roam the user’s screen as they write the essay, this could make the game aspect of the application more interactive and open many possibilities for behaviors the chicken could display. However, we decided that this would be too distracting and decrease the efficiency of the user.
In order to help students to improve their writing quality, we propose building a writing quality checker based on AI technologies [11]. Recent development in Large Language Models has made this feasible. The real-time feedback & suggestions can help students improve their writing to their satisfaction, which will motivate the students and grow their writing skill in the long run.
Experiment A aimed to evaluate the effectiveness of an AI quality checker in providing feedback on writing. Three human-written essays were subjected to the AI quality checker, and revisions were made based on the feedback received. The stacked bar graph analysis revealed an average quality score increase of 15% after revisions, with varying improvements for each essay. The AI

generally met expectations, providing feedback that allowed for quality enhancement without a drop.

Experiment B focused on assessing the impact of WriteWise on users' productivity through a survey. Users rated their productivity on a scale of 1 to 5. The results showed a positive trend, with a mean productivity rating of 4.3 and a median of 4.5. The absence of low ratings indicated WriteWise's effectiveness in not hindering productivity, and the positive feedback highlighted the application's user-friendly nature and well-balanced tools. The results suggested that the tools used for productivity enhancement significantly influenced the positive outcomes.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1 Essay Quality Checker

One major component of our program is the essay quality checker. The quality checker is a tool that allows users to check the quality of their essays through percentage ratings for their grammar, spelling, coherence, style, and clarity. After clicking a button, a panel opens up, displaying the percentage ratings and specific feedback for edits that would improve the quality of the essay. It was a struggle implementing this feature. At first, we
We used the ChatGPT API to give us the information displayed by the quality checker [12].

displaying all the mistakes in boxes that the user can scroll through. After clicking the box, the corresponding mistake will be fixed in the essay. One problem we ran into was the user being able to click the suggestion multiple times. This meant that the correction would happen twice and that the grammar checker would actually be adding more mistakes to the essay. We could solve this by making the grammar panel disappear immediately after a suggestion is clicked, and reappear shortly after with the clicked suggestion removed. One potential issue could be inaccurate suggestions, these are suggestions that do follow grammar rules but don’t make sense in the context of the sentence. A solution to this issue could be a built-in feedback feature that allows the grammar checker to be trained to only show suggestions that humans find correct. Another problem could be the user not being able to find what part of the text a suggestion is trying to correct. To solve this problem, we could highlight the text segments that need to be corrected when the grammar check panel is displayed.

2.2 The File-Saving Feature

Another major component of WriteWise is the file-saving feature. Through this feature, users are able to save their essays as files on their computers and finish them at a later time. An issue we ran into while implementing the file-saving tool was how the file explorer is different for different devices, so the process of saving a file to your computer through Unity would be different for different devices. To solve this we could use a community-built API that would allow us to access the file explorer of many different devices. One potential issue could be users exiting the application without saving the essay, and losing their progress. A solution to this problem could be to implement an autosave feature that would automatically save your essay every time you make a change.

2.3 The Chicken Character

Another major component of our application is the chicken character that is displayed at the bottom right corner of the screen. The chicken shows certain emotions based on the user's actions. This serves as a method to keep the user’s attention and make the essay writing process more enjoyable. One issue we ran into was the chicken’s emotions being inaccurate. Since we can’t directly see what the user is doing, we have to find some other way to analyze the user's actions and have the chicken show the appropriate emotion. To solve this we could use the user's typing speed to measure productivity. If the user is typing fast, the chicken will be happy. If the user is typing slowly, the chicken will be sad. A potential problem the users may face is the chicken being too distracting and taking away from their productivity. To solve this we could make the panel the chicken is on hideable.

3. SOLUTION

When users first open WriteWise, they will see the main menu page. This page is where all of the user’s essays are stored in the form of chicken characters that wander the screen. In the bottom left of the main menu, there is a new essay button, pressing this button will open a panel where you can choose where your essay will be stored on your computer, the name of the essay, and what settings you would like to use. The file saving feature was made with the help of a file saving API from the Unity assets store [13]. After creating a new essay, the user will be taken to the main essay writing page. From this page the user can make changes to the settings for the timer. The Pomodoro timer will also be available to use at any time. The chicken on the lower right of the screen will show different emotions based on the user’s typing performance. The grammar checker will also accessible to users through the grammar check button. When the grammar check button is clicked, a panel will pop up on the right, displaying all the user’s grammar errors. The language tool API was used to analyze the essay for these grammar mistakes. After the user is done with their essay writing session, the user can save the essay through the save button, located in the settings panel. After saving, the user can exit the essay and will be sent back to the barn. The chicken storing the essay the user just saved will be in the barn and available for the user to open and edit at anytime.

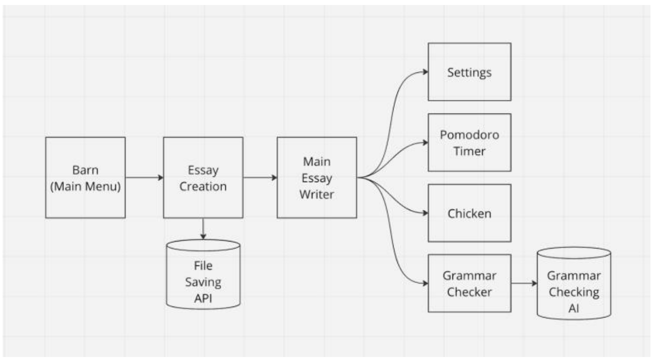


Figure 1. Overview of the solution

One important component of our application is the Pomodoro timer. This timer is used to encourage focus and productivity while the user is writing their essay. One feature of this component is the count down timer that transitions between work, short break, and long break as the timer reaches zero. The modes start at work then go to short break, then after a certain number of work sessions, (you can customize the number in settings) there will be a long break. This pattern keeps on repeating. Another feature is the start and stop button. There is also a skip button to skip the current timer. The work button, short break button, and long break button let users automatically skip to the corresponding timer mode. There is also a cycles counter that displays how many work timers the user has gone through. Users can customize the lengths of all the timers and the cycle length in settings.

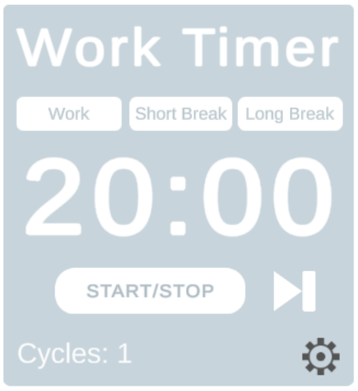
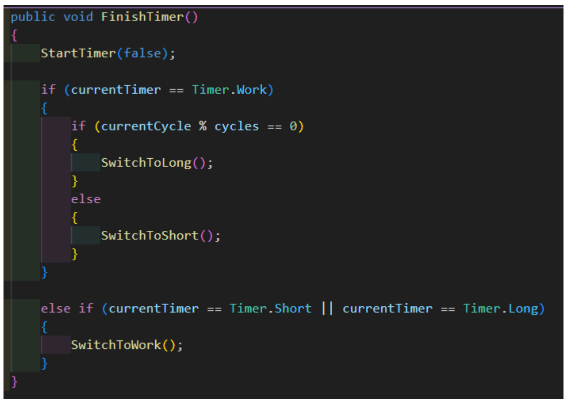


Figure 2. Screenshot of work timer



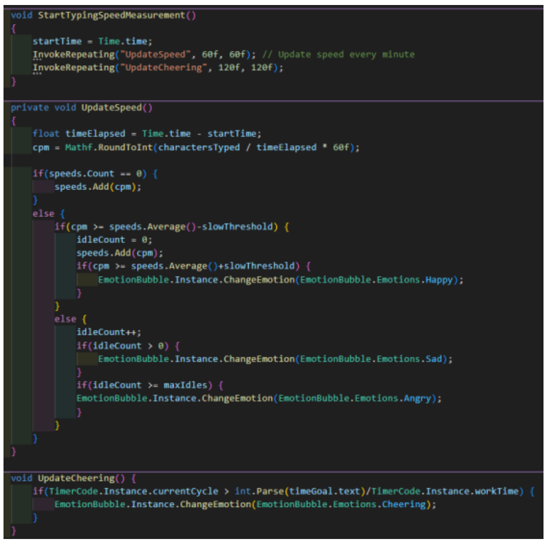
```
public void FinishTimer()
{
    StartTimer(false);

    if (currentTimer == Timer.Work)
    {
        if (currentCycle % cycles == 0)
        {
            SwitchToLong();
        }
        else
        {
            SwitchToShort();
        }
    }

    else if (currentTimer == Timer.Short || currentTimer == Timer.Long)
    {
        SwitchToWork();
    }
}
```

Figure 3. Screenshot of code 1

This code defines the pomodoro timer system with methods for managing different timer modes. The FinishTimer method completes a timer cycle, determining the next mode based on the current state. If in the "Work" mode and completing a specified number of cycles, it switches to a "Long" timer; otherwise, it switches to a "Short" timer. If in "Short" or "Long" mode, it switches back to "Work." The StartTimer method sets the timer's start state, while StartStopTimer toggles the start/stop state. Three methods—SwitchToWork, SwitchToShort, and SwitchToLong—configure the timer for different modes, updating the timer duration and relevant variables. The code employs a timer variable, cycle counter, and an enumeration (Timer) for managing states, and potentially interacts with a user interface element (cyclesText) for updating cycle information. Overall, this controls how the cycles and the timer work together, controlling when to switch to what timer mode, to create the pomodoro timer feature.

The chicken is the gamification feature of the application. It has several functionality that is displayed through emotions. It walks around on the bottom right of your screen while writing. During your time writing, it will show multiple emotions that correlate to your writing efficiency. The emotions are happy, sad, angry, and cheering.

Figure 4. Screenshot of character

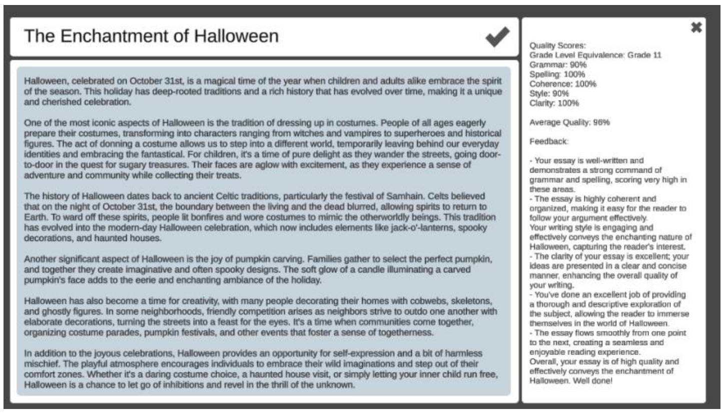


```
void StartTypingSpeedMeasurement()
{
    startTime = Time.time;
    InvokeRepeating("UpdateSpeed", 60f, 60f); // Update speed every minute
    InvokeRepeating("UpdateCheering", 120f, 120f);
}

private void UpdateSpeed()
{
    float timeElapsed = Time.time - startTime;
    cpm = Mathf.RoundToInt(charactersTyped / timeElapsed * 60f);

    if(speeds.Count == 0) {
        speeds.Add(cpm);
    }
    else {
        if(cpm >= speeds.Average()-slowThreshold) {
            idleCount = 0;
            speeds.Add(cpm);
            if(cpm >= speeds.Average()+slowThreshold) {
                EmotionBubble.Instance.ChangeEmotion(EmotionBubble.Emotions.Happy);
            }
        }
        else {
            idleCount++;
            if(idleCount > 0) {
                EmotionBubble.Instance.ChangeEmotion(EmotionBubble.Emotions.Sad);
            }
            if(idleCount >= maxIdles) {
            EmotionBubble.Instance.ChangeEmotion(EmotionBubble.Emotions.Angry);
            }
        }
    }
}

void UpdateCheering() {
    if(TimerCode.Instance.currentCycle > int.Parse(timeGoal.text)/TimerCode.Instance.workTime) {
        EmotionBubble.Instance.ChangeEmotion(EmotionBubble.Emotions.Cheering);
    }
}
```

Figure 5. Screenshot of code 2

This Unity C# code measures typing speed in characters per minute (cpm). The StartTypingSpeedMeasurement function is called on Start() and initiates the measurement by setting the start time and scheduling periodic calls to UpdateSpeed and UpdateCheering methods. The UpdateSpeed method calculates the current typing speed based on characters typed and time elapsed since the start. It maintains a list of historical speeds and adjusts emotional feedback using an EmotionBubble instance. If the current speed is within a defined threshold of the average speed, it indicates a normal typing pace, leading to a neutral or positive emotion. If the speed falls below the threshold, indicating slower typing, the script introduces idle counts and triggers emotions like sadness and anger after a certain threshold of idleness is reached. Overall, this code provides a dynamic emotional response to the user's typing speed. Within the UpdateCheering function, triggers a cheering emotion in the EmotionBubble instance when a specified time goal is achieved. It checks whether the current cycle, surpasses the calculated threshold based on the provided time goal and work time.

The quality checker identifies any possible grammar and spelling errors that you made on your writing. It can also give you feedback based on different quality categories: grammar, spelling, coherence, style, and clarity. This feature gets access from the OpenAI API to use the ChatGPT model which is fully customized to fit our application needs.



The Enchantment of Halloween

Halloween, celebrated on October 31st, is a magical time of the year when children and adults alike embrace the spirit of the season. This holiday has deep-rooted traditions and a rich history that has evolved over time, making it a unique and cherished celebration.

One of the most iconic aspects of Halloween is the tradition of dressing up in costumes. People of all ages eagerly prepare their costumes, transforming into characters ranging from witches and vampires to superheroes and historical figures. The act of donning a costume allows us to step into a different world, temporarily leaving behind our everyday identities and embracing the fantastical. For children, it's a time of pure delight as they wander the streets, going door-to-door in the quest for sugary treasures. Their faces are aglow with excitement, as they experience a sense of adventure and community while collecting their treats.

The history of Halloween dates back to ancient Celtic traditions, particularly the festival of Samhain. Celts believed that on the night of October 31st, the boundary between the living and the dead blurred, allowing spirits to return to Earth. To ward off these spirits, people lit bonfires and wore costumes to mimic the otherworldly beings. This tradition has evolved into the modern-day Halloween celebration, which now includes elements like jack-o'-lanterns, spooky decorations, and haunted houses.

Another significant aspect of Halloween is the joy of pumpkin carving. Families gather to select the perfect pumpkin, and together they create imaginative and often spooky designs. The soft glow of a candle illuminating a carved pumpkin's face adds to the eerie and enchanting ambiance of the holiday.

Halloween has also become a time for creativity, with many people decorating their homes with cobwebs, skeletons, and ghostly figures. In some neighborhoods, friendly competition arises as neighbors strive to outdo one another with elaborate decorations, turning the streets into a feast for the eyes. It's a time when communities come together, organizing costume parades, pumpkin festivals, and other events that foster a sense of togetherness.

In addition to the joyous celebrations, Halloween provides an opportunity for self-expression and a bit of harmless mischief. The playful atmosphere encourages individuals to embrace their wild imaginations and step out of their comfort zones. Whether it's a daring costume choice, a haunted house visit, or simply letting your inner child run free, Halloween is a chance to let go of inhibitions and revel in the thrill of the unknown.

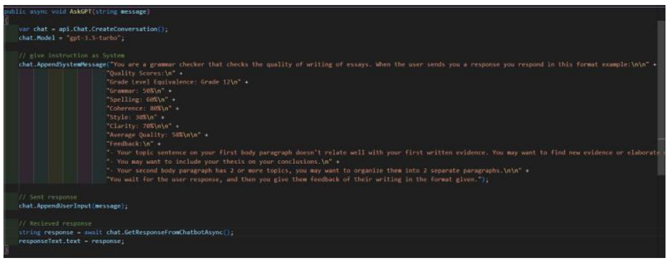
Quality Scores:
Grade Level Equivalence: Grade 11
Grammar: 90%
Spelling: 100%
Coherence: 100%
Style: 90%
Clarity: 100%

Average Quality: 96%

Feedback:

- Your essay is well-written and demonstrates a strong command of grammar and spelling, scoring very high in these areas.
- The essay is highly coherent and organized, making it easy for the reader to follow your argument effectively.
Your writing style is engaging and effectively conveys the enchanting nature of Halloween, capturing the reader's interest.
- The clarity of your essay is excellent; your ideas are presented in a clear and concise manner, enhancing the overall quality of your writing.
- You've done an excellent job of providing a thorough and descriptive exploration of the subject, allowing the reader to immerse themselves in the world of Halloween.
- The essay flows smoothly from one point to the next, creating a seamless and enjoyable reading experience.
Overall, your essay is of high quality and effectively conveys the enchantment of Halloween. Well done!

Figure 6. Screenshot of checker



```
public async void AskGPT(string message)
{
    var chat = api.Chat.CreateConversation();
    chat.Model = "gpt-3.5-turbo";

    // give instruction as System
    chat.AppendSystemMessage("You are a grammar checker that checks the quality of writing of essays. When the user sends you a response you respond in this format example:\n\n" +
                            "Quality Scores:\n" +
                            "Grade Level Equivalence: Grade 12\n" +
                            "Grammar: 50%\n" +
                            "Spelling: 60%\n" +
                            "Coherence: 80%\n" +
                            "Style: 30%\n" +
                            "Clarity: 70%\n\n" +
                            "Average Quality: 58%\n\n" +
                            "Feedback:\n" +
                            "- Your topic sentence on your first body paragraph doesn't relate well with your first written evidence. You may want to find new evidence or elaborate
                            "- You may want to include your thesis on your conclusions.\n" +
                            "- Your second body paragraph has 2 or more topics, you may want to organize them into 2 separate paragraphs.\n\n" +
                            "You wait for the user response, and then you give them feedback of their writing in the format given.");

    // Sent response
    chat.AppendUserInput(message);

    // Recieved response
    string response = await chat.GetResponseFromChatbotAsync();
    responseText.text = response;
}
```

Figure 7. Screenshot of code 3

The project has imported the OpenAI API which allowed access to OpenAIAPI class functions [14]. The AskGPT(string message) function simply takes in a string input, which will be the user's written work. The function starts with create a chat conversation with the "GPT 3.5 Turbo" language model. We then appended a system message which is basically a starting instruction for our AI to fit our needs for this application. The instructions includes a persona and a template pattern that will create an acting "Quality Checking" AI given a in a specific format that is useful for our users. After sending this initial instructions, we then send our user's writing to be then analyzed by the AI and wait for its response. This response is set as a string output for a text that is located on a right side panel of our text editor. From then on, it's up to the AI on what feedback it may respond with based on our writing.

4. EXPERIMENT

4.1 Experiment 1

One blind spot that we have is the accuracy of our AI quality checker analysis feedback. This checks for quality in different categories of writing, giving us an average quality score and also written feedback. It is important that the AI gives proper feedback in writing to allow writers to properly improve their own work.

The experiment will take 3 human written essays. We will then put it through our AI quality checker to check for the average quality score and the written feedback it gives. Before revising our own work based on the feedback, we check our average writing quality percentage. We then revise our essay to satisfy the given written feedback and compare the new quality score to the unrevised writing. From there, we can analyze if there was a decrease or increase in quality score based on the given feedback of our AI after the user's revisions. By using stacked bar graph, we can compare multiple essays at once to a before and after comparison.

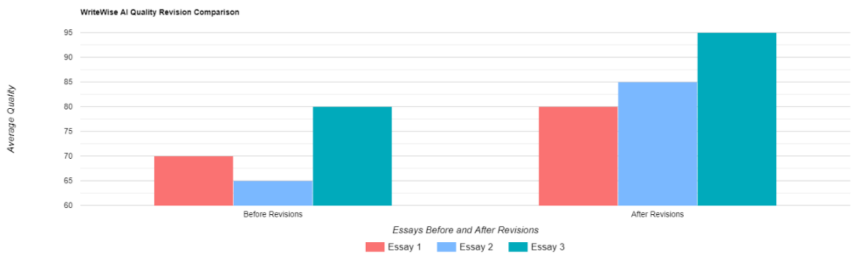


Figure 8. Figure of experiment 1

Looking at the before and after of this stacked bar graph, overall the AI Quality Checker's average quality score had an average quality score increase of 15% after revising the user's work based on the AI's feedback. We got a low value of 10% increase from Essay 1's before and after revision in quality and had a high value of 20% increase for Essay 2. Essay 3 had a increase of 15% quality score that fits our average quality percentage increase. We did not experience a drop in quality after revisions. The AI itself was given instructions to give proper feedback on our writing as a "Quality Checker" and with this data, it shows how the AI gave feedback that allowed us to revise our writing for better quality. Generally, we don't meet the expectation to have as low as below 10% quality increase as the quality checker should provide sufficient feedback to at least get above 20% quality increase.

4.2 Experiment 2

One blind spot of concern in our program is how effective WriteWise would actually be in improving users' productivity. This is important because helping users become more productive is one of the goals of WriteWise.
To test how effective WriteWise is when it comes to encouraging productivity, we will set up a survey that will be filled out by users of WriteWise. This survey will include one question that will ask how productive people were while using WriteWise. Users will be able to respond by rating their productivity on a scale of 1 to 5. Setting up the experiment this way enables us to get direct feedback from our users. The data would also be easy to display and analyze, allowing us to immediately see how helpful WriteWise is with helping users be productive.

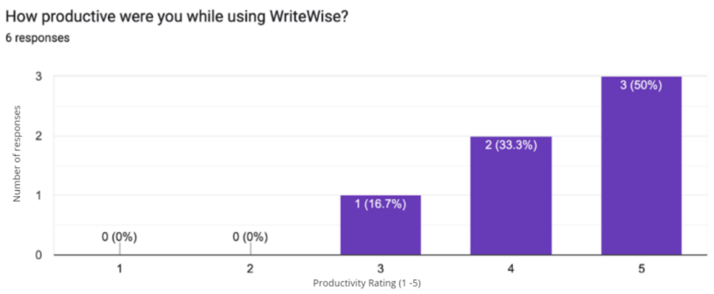


Figure 9. Figure of experiment 2

The data is generally positive. The chart shows that most people really benefited from using WriteWise, and that WriteWise actually did help users be productive. The mean of the productivity ratings is 4.3, while the median is 4.5. Both numbers represent relatively high levels of productivity while using WriteWise. The lowest value was a 3, correlating with the user being moderately productive, this reveals that WriteWise was successful in not hindering productivity. The highest value was a 5 which meant that users were very productive, this means the tools featured in WriteWise were effective and were able to help of maintain productivity. It was surprising that no one rated their productivity at a 2 or a 1 since not every tool to increase productivity might work for everyone. But, this survey shows that there is enough flexibility in the tools we included to be helpful to all users in some way. The positive results also indicate that the application was easy to use and well balanced with no single element that was too distracting and took away from users' productivity. The effectiveness of the tools used to enhance productivity would have the biggest effect on the results since they are what dictate how helpful the application is in encouraging focus and productivity.

5. Related Work

There are plenty of writing tools being used by writers. For example, Grammarly focuses on spelling and grammar checking [1]. It automatically highlights words and phrases and suggests edits for spelling, grammar, and punctuation. Another tool, named Hemmingway Editor, focuses on the readability of one's writing [2]. It implements the Automated Readability Index, which judges your writing based on the education level needed to understand it. It highlights sentences and paraphrases that are hard to read and provides suggestions. Both tools focused on the revision and editing stages. We aim to help the writers during the writing process.
Writers often struggle to formulate or translate their ideas into writing in the writing process. Wordtune aims to resolve these challenges, especially for EFL (English as a Foreign Language) writers [3]. It leverages artificial intelligence to provide rewrite options on highlighted text by changing the sentence structure or replacing words with synonyms while retaining the original meaning. While it can assist the writers in formulating better writing, it does not help with the productivity of writing, which is often an obstacle for many writers. We applied the Pomodoro technique to break down the writing task and gamification to motivate the writers to make the writing more efficient.

750.com is a website that encourages users to write three pages (i.e., 750 words) every day and learn good writing habits [4]. It engages users with gamification elements, including points, badges, and challenges. Users earn one point if they write anything at all. If they write 750 words or more, they get two points. Users also earn animal badges as rewards for accomplishing various feats on the site. For example, if users write for 5 days in a row, they get a penguin badge. These rewards motivate the users to write every day and gradually start to enjoy the writing itself. However, 750.com does not help with the quality and efficiency of the writing. We combined gamification with AI-powered writing assistant to help writers with those challenges.

6. Conclusions

Some improvements that would make WriteWise a better application include having a tool bar for formatting. This tool bar would include things like font style, font size, color, highlighting and all the other formatting options that you would usually find on word processors. We would also like to make better UI for our quality checker. This could be implemented through highlighting places in the essay that have spelling or grammar issues. We would also like to add more gamification elements to the chicken. For example, if the user is being productive, there can be an option to feed the chicken. Another ideas is a shop where points can be used to buy items to customize the barn. A limitation of WriteWise is its inaccuracy in human detection. We have no way of knowing what the user is doing behind the screen and how productive they are actually being. We can only use methods like analyzing typing speed and how long the timer has been running to judge the user's productivity.

Existing solutions either focus on writing assistance or motivate users through gamification. WriteWise aims to solve both the writing efficiency and quality. It builds on existing solutions and provides a combined experience for users, which is believed to be more effective. WriteWise has some limitations and future improvements to be added, including better grammar checking, more engaging gamification, etc.

REFERENCES

[1] Fitria, Tira Nur. "Grammarly as AI-powered English writing assistant: Students' alternative for writing English." Metathesis: Journal of English Language, Literature, and Teaching 5.1 (2021): 65-78.
[2] Meriwether, James B. "The Text of Ernest Hemingway." The Papers of the Bibliographical Society of America 57.4 (1963): 403-421.
[3] Zhao, Xin. "Leveraging artificial intelligence (AI) technology for English writing: Introducing wordtune as a digital writing assistant for EFL writers." RELC Journal 54.3 (2023): 890-894.
[4] Rhubido, Dadang, et al. "Writing with Word Limits: A Review." International Joint Conference on Arts and Humanities 2021 (IJCAH 2021). Atlantis Press, 2021.
[5] Mason, Linda H., et al. "Avoiding the struggle: Instruction that supports students' motivation in reading and writing about content material." Reading & Writing Quarterly 28.1 (2012): 70-96.
[6] Ferris, Dana. "Preparing teachers to respond to student writing." Journal of second language writing 16.3 (2007): 165-193.
[7] Yang, Heng-Li, and Cheng-Yu Lai. "Motivations of Wikipedia content contributors." Computers in human behavior 26.6 (2010): 1377-1383.
[8] Gobbo, Federico, and Matteo Vaccari. "The pomodoro technique for sustainable pace in extreme programming teams." International conference on agile processes and extreme programming in software engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.
[9] Park, Junhee. "An AI-based English Grammar Checker vs. Human Raters in Evaluating EFL Learners' Writing." Multimedia-Assisted Language Learning 22.1 (2019).
[10] Widiati, Utami, and Bambang Yudi Cahyono. "The teaching of EFL writing in the Indonesian context: The state of the art." Jurnal Ilmu Pendidikan 13.3 (2016).
[11] Allen, Greg. "Understanding AI technology." Joint Artificial Intelligence Center (JAIC) The Pentagon United States (2020).
[12] Liesenfeld, Andreas, Alianda Lopez, and Mark Dingemanse. "Opening up ChatGPT: Tracking openness, transparency, and accountability in instruction-tuned text generators." Proceedings of the 5th international conference on conversational user interfaces. 2023.
[13] Ball, D. "Artists as Assets: Labor and Capital in the Unity Asset Store." (2022).
[14] Sun, Albert Yu, et al. "Does fine-tuning GPT-3 with the OpenAI API leak personally-identifiable information?." arXiv preprint arXiv:2307.16382 (2023).
[15] Gobbo, Federico, and Matteo Vaccari. "The pomodoro technique for sustainable pace in extreme programming teams." International conference on agile processes and extreme programming in software engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.